NEURAL NETWORK APPROACH TO THE CLASSIFICATION OF URBAN IMAGES

by

Hywel F. J. Evans, MBE BA

Thesis submitted to the University of Nottingham in fulfilment of the requirements of the degree of

Doctor of Philosophy

December 1996

CONTAINS DISKETTE

UNABLE TO COPY

CONTACT UNIVERSITY

IF YOU WISH TO SEE

THIS MATERIAL

Contents

List of Figures

List of Tables

Abstract

Over the past few years considerable research effort has been devoted to the study of pattern recognition methods applied to the classification of remotely-sensed images. Neural network methods have been widely explored, and been shown to be generally superior to conventional statistical methods. However, the classification of objects shown on greylevel high resolution images in urban areas presents significant difficulties. This thesis presents the results of work aimed at reducing some of these difficulties. High resolution greylevel aerial images are used as the raw material, and methods of processing using neural networks are presented. If a per-pixel approach were used there would be only one input neuron, the pixel greylevel, which would not provide a sufficient basis for successful object identification. The use of spatial neighbourhoods providing an $m \times m$ input vector centred on each pixel is investigated; this method takes into account the texture of the pixel's neighbourhood.

The pixel's neighbourhood could be considered to contain more that textural information. Second order methods using mean greylevel, Laplacian and variance values derived from the pixel neighbourhood are developed to provide the neural network with a three neuron input vector. This method provides the neural network with additional information, improving the strength of the relationship between the input and output neurons, and therefore reducing the training time and improving the classification accuracy. A third method using

a hierarchical set of two or more neural networks is proposed as a method of identifying the high level objects in the images.

The methods were applied to representative data sets and the results were compared with manually classified images to quantify the results. Classification accuracy varied from 69% with a window of raw pixel values and 84% with a three neuron input vector of second order values.

ACKNOWLEDGEMENTS

The author has many to thank for their assistance and encouragement during the writing of this thesis. Most particular, I wish to thank my supervisor Professor Paul Mather who has provided his unfailing guidance and wisdom. He has provided continued faith in my work and moral support during some difficult times. I would also like to thank Dr. Chris Higgins for his advice and expertise in computer science during the early part of my research.

I would like to thank the W.H. Revis Bequest Fund for its generous financial support and I would like to acknowledge the help and advice provided by the Ordnance Survey.

Finally, I must thank my family and friends for their forbearance and understanding over the past few years.

CHAPTER 1

1. INTRODUCTION

We are the Pilgrims, master; we shall go

Always a little farther; it may be

Beyond that last blue mountain barred with snow

Across that angry or that glimmering sea.

The words of James Elroy Flecher's *The Golden road to Samarkand,* have been adopted by the Special Air Service to reflect the demands placed on them to push to the limits both their endurance and knowledge. It can also be seen to reflect mankind's continual need to go that little farther in a quest to discover the limits of himself and his environment. In the quest to discover the world around him he soon realised that in order to record his newly acquired knowledge a map was the tool that would become the foundation of all his records. Once he moved away from the local surroundings, maybe across that *blue mountain*, knowledge of position suddenly became paramount.

The period in which this research was carried out includes the Gulf war and the war in the Former Yugoslavia. I spent those few short days of the Gulf war and the long years of the conflict in Bosnia, with a never ending need to know my position and furthermore what was in my path. We had Global Positioning Systems (GPS), but what use is position without a map to tell you what is there? So, as with those first explorers, the map is the essential basic tool of all those wishing to know about the world around them. My bachelor degree is in mathematics in which I had a special interest in differential geometry which often forms the core mathematics for mapping.

In warfare, time is of the essence, and the identification and position of targets must be updated in as near real-time as possible. This has formed the driving force of this research, the aim of which is to develop automated computer based identification techniques in aerial reconnaissance. Image classification covers a broad area of research and applications and therefore to narrow the area down to a more manageable size, the classification of aerial images in one particular environment was chosen. Urban images presented the most relevant challenge to my military background where it is man-made structures that are of interest. More importantly, techniques used to classify urban images can be used for peaceful as well as military purposes.

Because of the period of time spanned by this research there have been many advances in computer technology and classification techniques, therefore ideas

and methods have had to be continually revised to keep pace. This might be seen as a disadvantage but the contrary is true since there are both hardware and software solutions available now that could only be dreamt of at the start of this research. The most influential of these advances has been the rapid development of fast desktop computers and the rebirth of neural networks as a pattern recognition tool.

1.1 Background

General Sir Peter De La Billiere in the Gulf and town planners at home both require the same type of information from remotely sensed images. They both require the classification and position of buildings and other similar structures. Reconnaissance in military circles and aerial survey in civilian circles produce the relevant results required. In reconnaissance Photographic Interpreters (PIs) analyse images and attempt to classify objects, while in civilian surveying mapping and Geographical Information Systems (GIS) are maintained by the analysis of images. That the military and civilian worlds often depend on each other for mapping shown very clearly by the formation of the Ordnance Survey Board, which is often thought of as the birth place of modern mapping methods.

The cost of mapping the United Kingdom is today in the region of 3.5 million pounds. Therefore, there is a strong motivation to automate as many functions

of map production as possible. Barnsley *et al.* (1991) produced several new tools for a GIS specifically to monitor urban areas and only limited success in automating the identification of buildings and as a result concentrated on a purely interactive system Precision farming, using satellite sensor images and GPS, is also expanding as higher standards with minimum cost are demanded by the consumer. Precision farming aims to manage crops precisely to their needs and reduce costs to the farmer as fertiliser and insecticides are only applied to the required areas. Aerial survey photography is not used as much as satellite images but could be of great use in areas where satellite image data would be considered too expensive or in areas where persistent cloud cover makes its use impossible. The continued development of methods using greylevel aerial photographic image data are required for land use classification in military reconnaissance, civilian mapping, land management using GIS and precision farming.

1.2 Aim

The overall aim of the research work reported in this thesis is to develop computer-based techniques for the recognition of objects in grey-level aerial photography of urban areas. The specific aims are the study of neural networks as a method of identifying objects and the use of derived (second-order) descriptions of tone, texture and edge strength.

The lack of success of conventional computer based pattern recognition methods, outlined by Muller *et al.* (1991), and the development of classification techniques using neural networks determined the possible areas of research that could be used to achieve the thesis aim. Neural networks are used for image classification and considerable research has been carried out since the early 1990s. The main thrust of the research has been to develop methods for classifying land cover, for example in terms of crop type, with smaller projects aimed at water run-off and farming related topics. The question posed by this research is: will they stand up to the difficulties presented by urban images?

Pattern recognition using a form of neural network was developed by Aleksander *et al.* (1984). His single layer networks were designed for use in industrial applications where the automated recognition of sub-assemblies was used on production lines. The technique was also adapted for a security application as a face recognition device. It was the neural network's ability to interpolate between objects it had been trained on that led to the belief that there was a future for them in conventional areas of image recognition where the possible variations of an image were to great for convention processing methods. It seemed in those early days that any recognition task that humans performed well would be suitable for a neural network system. This seems to have proved the case especially in land cover classification techniques.

The majority of neural network methods used for land cover classification use multi-layer networks and multi-spectral data mainly from the Landsat and SPOT sensors. Military reconnaissance and local aerial mapping surveys have traditionally used mono-spectral data and since this was the main area of interest of this research, any technique that was developed would need to be viable with mono-spectral data such as greylevel photography. Alexander's methods are optimised for binary image data which was considered a limiting factor on their use. The later more commonly used neural network, the backpropagation network, proved to be more adaptable to greylevel image data.

The stronger the relationship between the input and output vector of a neural network the better it is at classification. Multi-spectral data provides a natural selection for structure of the input vector to the neural network, with each input being simply the value of each pixel in one of the spectral bands of the data set. The seven Landsat bands thus provide seven input neuron values to the network. However, mono-spectral image do not have an immediate parallel in terms of input vector. Therefore groups of pixel values were tested and as a result a more novel approach using second-order values of groups was developed. This second method is presented as the culmination of the present research in this thesis.

For the future there are several new avenues to follow which will improve the results of the current research. One possibility is the use of more than one

network, using the results of the first to form the input vector of the second. This hierarchical method of classification would use a first network to determine object primitives, such as shape greylevel and texture, while the second would determine to which class of object these primitive belonged. A second possibility would be to use larger groups of pixels, in the order of the size of the objects. The development of faster computer hardware is the only limiting factor on the possible future neural network techniques. This is a area of research where size (memory) and speed are critical if the methods are to be considered practical.

1.3 Outline

The following is a guided tour of the thesis. To give an overall view of the research Figure 1.1 shows an outline of the thesis. Along the road there are milestones which as with real milestones identify particularly important or crucial parts of the research. The destination is the development of a new method of image classification based on a neural network approach. The first chapter defines the aim of the research: the development of techniques for classifying urban images and an introduction to this area of research. Chapter two discuses previous research directly related to the aim and also, in order to understand the methods used and possible problems that are posed in trying to achieve this aim, the discussion is widened to cover related areas of research. The areas of research that are covered include human vision, photogrammetry,

image processing methods, land cover classification techniques and neural networks.

Human vision provides a suitable starting point for any research involving pattern recognition. The work of Julesz (1960) and Marr and Poggio (1978) are reviewed to highlight the human solution to the problem of pattern recognition. Aleksander (1985, 1987) links the discussion of human vision and possible methods of machine vision. As photogrammetry is at the heart of cartography and surveying, both conventional and digital photogrammetry are reviewed in order to cover areas such as image refinement, measurement and height calculation from stereo images. These methods are used in classification methods using image processing techniques and neural networks. A method of classification based on image processing techniques relies on several basic image processing methods, hence these are discussed in some detail prior to a review of five methods which employ conventional image processing and / or stereo image correlation. The methods reviewed are: digital terrain models (Muller *et.al.* 1988), broken segment matching (Henderson, Miller and Grosch 1979), motion stereo (Lacina and Nicholson 1979), shadows (Huertas and Nevatia 1988, Liow and Pavlidis 1990, Irvin and McKeown 1989, and Hsieh, McKeown and Perlant 1990) and stereo correlation (Gruen 1985, Pridmore, Mayhew and Frisby 1990, and McKeown 1990). Land cover classification, until the recent employment of neural networks, has used statistical classifiers. Three of the more common classifiers are reviewed: centroid, parallelepiped

and maximum likelihood. These classifiers give and insight into classification using multiple image attributes such as multispectral data provided by SPOT and Landsat sensors. The use of multispectral data is relevant to the later discussion on the use of neural networks for land cover classification. Therefore, a detailed review of Aleksander's single layer neural network 'WISARD' and other multi-layer neural networks is included. Although WISARD is not strictly in current terminology a neural network it is discussed in detail as at the start of this research it was the only practical network method that would run on a PC in a reasonable time. WISARD showed great promise as it was a method that did not require programming, with all of the restrictions of having to define rules, but was taught by example, as is the case with a human and neural networks. Although not developed here some of the concepts developed by Aleksander may be applicable to multi-layer networks. It is possible that WISARD techniques could be adapted efficiently to greylevel images considering the increased processing speed of modern serial computers. This chapter concludes with a detailed study of multi-layer neural networks in order to provide the necessary background for the neural network approach presented in chapter four.

Chapters three and four detail two approaches to urban image classification. The first approach uses conventional image processing methods while the second is a new approach using neural networks. Chapter three presents work carried out in the early stages of this research which attempted to approach the

problem using conventional image processing techniques that were within the capabilities of the PCs of the time (the Intel 8086 at 16Mhz). These techniques, as with most of the early attempts at pattern recognition, processed the image into various primitive forms: edges, lines and areas. Attributes such as lines or shadow are identified so that higher-level reasoning can be used to classify the objects within the image. Images were processed to calculate the average greylevel and angle of the strongest gradient, for a fixed window size around each pixel, and use this data to determine whether the pixel was part of a line segment. Line segments were considered to have the same average greylevel and angle within a defined range. Building shadow is also considered and an variation on density slicing is used to reveal the detail of buildings that would normally be hidden by deep shadow. At this point these methods could have been developed to include AI methods such as a rule-based filtering process, in line with McKeown *et al.* (1985), to classify the lines that are extracted from the original image. It was felt that rule-based methods lacked flexibility and could become overwhelmed by the complexity of urban images. At this time great strides were being made in the use of neural networks for land cover classification by Benediktsson, Swain and Ersoy (1990), Bischof, Schneider and Pinz (1992), Civco (1991), and Dreyer (1993), and so this method was chosen as a more promising direction than the rule-based method. The neural network's non-parametric and interpolative features presented a possible new methodology.

The basics of neural networks are introduced in Chapter two, Chapter four presents the application of neural networks, in a selection of conventional and novel methods, to the classification of urban images. The methods use a combination of the knowledge gained in the previous chapters so that a degree of pre-processing is used before the image data are presented to the network. Successful training of a neural network depends primarily on a strong relationship between the input and output vectors and therefore the methods described in this thesis concentrated on achieving this requirement. Jensen (1981) included a measure of texture in the input data for land cover classification and found it to be an important factor in the training data for urban image classification. The neural network suitable for this type of application had to have supervised training, as a distinct class set was required and the code or suitable ready-made software had to be available. Backpropagation is one of the most commonly used supervised training algorithms and is the chosen model for this research. The images used are panchromatic and were scanned at several resolutions and 256 greylevels. This chapter describes the software tools that were used for capturing the data. It identifies the type of input data required and investigates the possible output classes using an unsupervised image processing technique. Nearly all of the previous research into land cover classification using neural networks uses multi-spectral input data, therefore panchromatic images required the development of a new method. Three methods of classification are considered: the first using raw pixel data (first order), the second using a novel technique

which pre-processes the image to provide the network with 'second order' data and a proposed third method that uses more than one network. The initial work in this chapter was restricted by the available computing power and the training set were minimal or synthetic in order to achieve reasonable training times (less than 12 hours). Fortunately, with the advent of higher speed PCs the research became less restricted and reasonable training times using more realistic training sets were feasible.

Chapter five discusses future developments, possible applications and a review of the thesis.

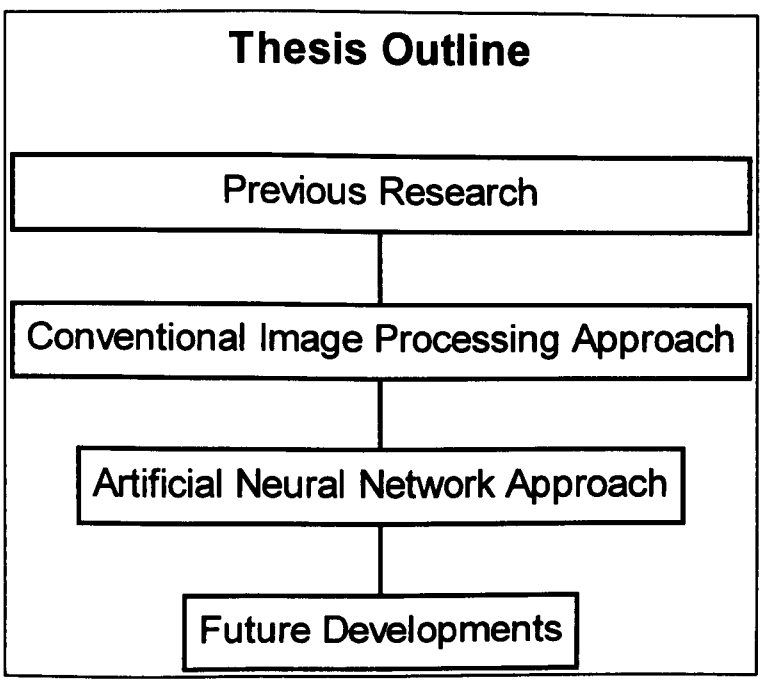

Figure 1.1 Thesis Outline.

CHAPTER 2

2. PREVIOUS RESEARCH

2.1 Introduction

In order to understand the nature of the problem of classifying urban images it is necessary to consider its history. Previous research into the automated recognition of buildings in digital images has used methods based on well-established mainstream image processing techniques. These methods invariably segment the image using edge and line extraction methods and then use this simplified image as the basis for pattern recognition. The pattern recognition techniques are designed around image attributes such as shape or shadows. Basic image processing has been fairly successful on test images but has had limited success when used on realistic complex urban images. Interestingly, as much of the work has been carried out in America, the test images have almost always included flat-roofed buildings and therefore this work may not be applicable to the pitched roof buildings of Europe. Building shadow has proved to be a powerful but simple attribute in building recognition (Liow and Pavlidis, 1990; Irvin and McKeown, 1989). Huertas and Nevatia (1988) also use shape descriptors to assist the shadow analysis.

Correlation of stereo pairs of images has been used very successfully in the production of Digital Terrain Models (DTMs) and these techniques have also been used on urban images. The correlation method that is normally used is a direct extension of the methods used in the production of DTMs (Pridmore *et al.*, 1990; Muller *et al.*, 1990). These and similar methods based on the least squares calculation fail when used for buildings because they assume that height change over space is gradual and the surface is continuous and single-valued; whereas the vertical wall of a building represents a multi-valued point. The underlying problem with the use of stereo matching in urban areas is the complexity of an environment which has buildings, shadows and occlusions. The lack of success of automatic stereo matching for urban areas is generally confirmed by results reported by groups such as McKeown *et al.* (1989), Lowe (1987), Horeud and Skordas (1989), Mohan *et al.* (1989). McKeown *et al.* (1989), Aviad *et al.* (1991) and Hsieh (1990) have achieved significant results using basic building shapes. Attempts to improve the method have been made by modifying the basic stereo matching process by using other features. Textures and edges were used by Pridmore *et al.* (1990), shadow by Irvin and McKeown (1989) and by Liow and Pavlidis (1990), and high-level domain reasoning by Huang *et al.* (1986) and Hsieh *et al.* (1990).

The lack of success of these more conventional methods, specifically outlined by Barnsley *et al.* (1991), and the apparent exhaustion of these techniques has led to a search for a new path. Because of the large variations in building

shape and size a method that could interpolate between known objects to identify objects belonging to the same set was required. This is a human skill that we often take for granted. It is important therefore to start any survey of past work with a brief review of the theories of the working of the human visual system. Neural networks mimic some of the abilities of the human visual system and they have been successfully applied to recognition of objects in industrial and security applications, and one of the objectives of this research is to investigate whether such methods can be applied to the problem of building recognition. This review starts by looking at the human vision system, then moves on to pattern recognition and finally considers the topic of neural networks.

2.2 Human Vision

Marr and Poggio (1978) proposed a theoretical framework within which psychophysical and neurophysiological data on stereopsis available at the time could be organised. This work was carried out during the same period that Aleksander *et al.* (1973) were rediscovering neural networks. The route taken by Marr and Poggio (1978) was technologically driven while that of Aleksander *et al.* (1977) was partly theoretical and could only be partly supported by the then-available computer power. Since that time, computing power has increased considerably, and it has been possible to examine theories

that can only be realised with the almost certain knowledge that technology will eventually catch up with the research.

Marr and Poggio (1978) studied the very complex problem of stereo vision and theorised on the human mechanism of vision by breaking down the task of visual processing into more manageable steps. This approach was very much in keeping with the existing computer programming techniques. This theory made the assumption that the brain also carried out the process of vision by a sequence of steps. Present theories suggest that the brain is a vastly parallel processing system and is very unlike a serial computer.

Julesz (1960) found that it was possible to interpret random dot stereograms, which are stereo pairs of images that consist of random dots when viewed monocularly but fuse when viewed stereoscopically yielding patterns separated in depth. This ability was not understood by Marr and Poggio (1978) as they could see a problem in that false targets would exist if any correspondence between the dots was calculated using existing methods. Figure 2.1 shows that of 16 possible targets only four are correct.

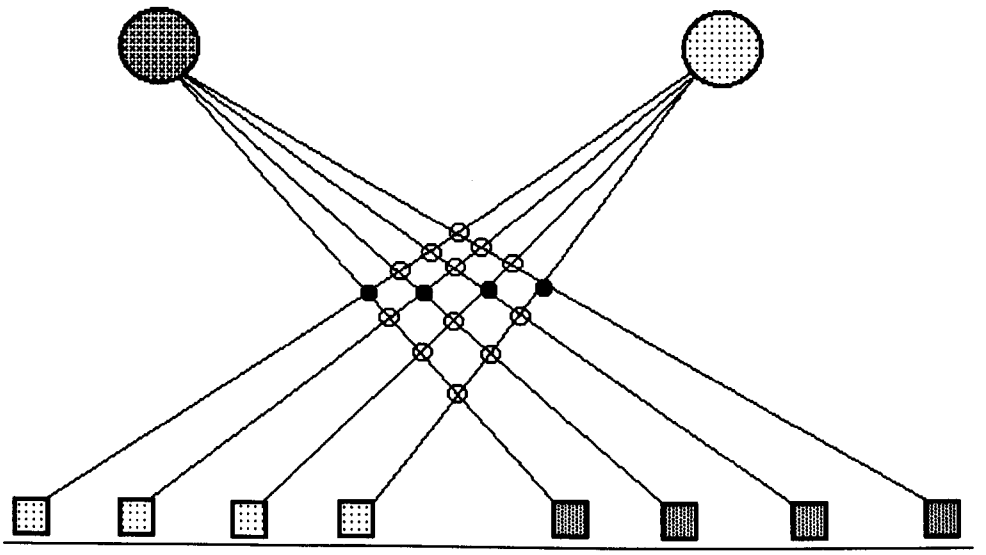

Figure 2.1. If the human visual system interprets random dot stereograms by simple correlation then 12 false targets, the open circles, would exist. Redrawn from Julez (1971).

In order to solve the false target problem Marr and Poggio (1978) suggested two constraints. First, that a given point on a physical surface has a unique position in space at any one time. Second, that matter is cohesive, it is separated into objects, and the surfaces of objects are generally smooth and the size of any surface pattern is small compared with the distance to the viewer.

Marr and Poggio (1978) were not to know that just seven years later Aleksander (1985) would demonstrate a single-layer neural network that could fuse random dot stereograms without any of the above or other limitations. More importantly the operation was carried out in a single pass calculation.

Matching a stereo pair of images can be made a much simpler task if the images are processed to place corresponding pixels in a stereo pair on corresponding scan lines. This is referred to as epipolar alignment. With this type of alignment the search process for correlating pixels is limited to single scan lines. In order to do this it is necessary to recover the camera geometry. One method is to match a number of prominent features that exist in both images. Prazdny (1982) called this a chicken and egg problem since in order to match points in the images the camera geometry must be known. However, in order to recover the camera geometry it is necessary to match points in the two images. The human visual system does not seem to have this problem as shown by the ability to fuse a random dot stereogram.

The recovery of depth information by the human visual system from random dot stereograms seem to indicate that shape and texture are only part of this process. The brain must be able to resolve the large number of ambiguities created by the possibility of one dot matching many others, if indeed point matching is used in the process. Aleksander (1985, 1987) gives evidence to support the view that point matching is not required by the human visual system but only by systems using an algorithmic approach. Most available evidence lends weight to the theory that the brain does not function algorithmically either. Hence, a computer based system that is required to recover depth information from a stereo pair of images perhaps should also follow a non-algorithmic approach. Following this path removes the need for a

system designed to recover depth information to overcome the problems normally associated with stereo image correlation.

2.3 Photogrammetry

Photogrammetry, the measurement of features in a photographic image, is a natural extension of photographic interpretation. It can be subdivided into metric and interpretative photogrammetry. Metric photogrammetry is concerned with the measurement of individual feature parameters such as volume, elevation, distance and area. Interpretative photogrammetry relates to the recognition and analysis of features. Both types of photogrammetry are included in remote sensing. An understanding of these techniques provides a baseline from which computer based digital photogrammetry can be developed.

Topographic mapping is probably the single largest application of photogrammetry. Remotely sensed data and computer based methods are used to produce digital terrain models (DTMs) for the identification of the contours for topographic maps. Urban areas present a problem to existing DTM methods as the algorithms assume the terrain to be a continuous surface which can be modelled by a bivariate polynomial. Urban areas present a surface that cannot be modelled this way because most buildings have vertical sides. Measurement, refinement, stereoscopic parallax measurement and

orthophotography are covered in this chapter. The techniques discussed are related to photographic images rather than images derived from current remote sensing sources because for high resolution work, where individual building boundaries must be defined, low altitude aerial photography is the most widely used image source. However, in the near future fine spatial resolution remotely sensed images will be available.

2.3.1 Photogrammetric Measurement

Measurement includes co-ordinates, length, angle and colour density, the latter being an essential first stage in digital photogrammetry. The co-ordinate system is invariably a rectangular grid with the origin at the centre of collimation. In order to establish a grid mapping cameras mark the photographs with side and / or corner marks, these are the called fiducial points (Figure 2.2). Metal and glass scales are used to measure positions relative to the fiducial origin and short distances between points. For more accurate measurement comparator instruments are used. These instruments are designed to compare photographic positions with respect to scales fitted to the instrument. If the photograph is to be used for mapping then accurate measurement is essential.

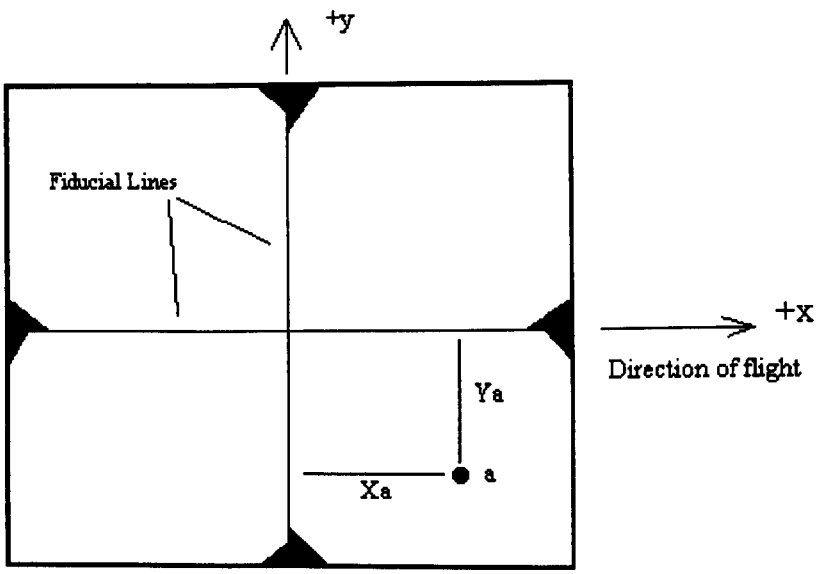

Figure 2.2 Side fiducials and rectangular co-ordinate system for photographic co-ordinate measurement.

2.3.2 Photogrammetric Refinement

Refinement of the measured image co-ordinates is necessary to correct distortions such as shrinkage, expansion, lens, atmospheric effects and Earth curvature. In addition, corrections are usually required to position the point of collimation at the principal point. The latter correction is only required for very high precision work as the error is normally in the order of micrometers. Shrinkage and expansion of the negative can be corrected by comparing the measured distances between fiducial points and calibrated distances. Another measurement method is to use a camera which has a reseau grid. The photograph then has this fine grid superimposed on the image. The grid provides calibration data for many more points on the photograph. If the distortion is non-uniform then a reseau grid becomes essential if corrections are

to be made successfully. The principal lens distortion is radial. It is assumed to be symmetric and corrections are applied with reference to a calibration curve. This curve can be approximated by a polynomial if a numeric method is required. Other lens distortions, asymmetric and tangential, are applied in high precision work.

2.3.3 Stereo Photogrammetry

Parallax differences between one point and another are caused by the different elevations of the two points, (Figure 2.3). Detection of abrupt elevation changes caused by buildings could be used to assist in extracting buildings from an image. In addition, if the images are to be mapped to ortho-images then the elevation of every point in the image is required. An urban area does not have a continuous height surface with a gradual change in elevation, the buildings and other features with abrupt elevation changes must be identified first before they can be mapped to their correct geographical co-ordinates. A mathematical mapping can not be applied as would be possible with an image of bare terrain.

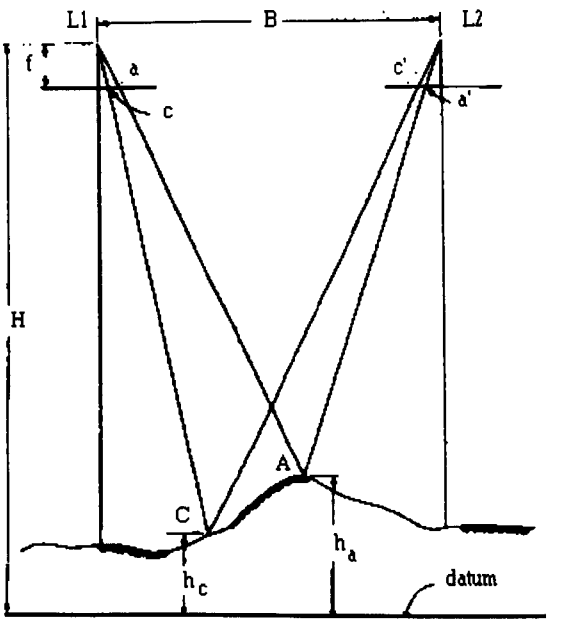

Figure 2.3 Parallax differences.

The parallax differences can be calculated from:

$$p_c = \frac{fB}{H - h_c}$$

$$p_a = \frac{fB}{H - h_a}$$

if point *c* is taken as the datum then

$$p_a - p_c = \frac{B(h_a - h_c)}{H}$$

2.4 Digital Photogrammetry

The need for accurate mapping has existed since we wandered beyond the immediate horizon. In digital photogrammetry accuracy is not dependent on the measuring instrument but on the scanning resolution. Today GIS extend the use of a map way beyond basic navigation. The ultimate dream of GIS design is the provision of timely and accurate information on nature's and human's influence on the Earth. The amount of information retrieval required for a GIS can only be achieved with the assistance of computers. An ultimate goal would be to process raw data and produce input data for a GIS without user intervention.

2.4.1 Digital Photogrammetric Refinement

If the image distortion is non-uniform the traditional method would require the original image to have a reseau grid and a polynomial method of least squares would be applied. It may also be possible to use a neural network model which can learn to approximate any continuous mapping, minimising the error in the same sense as least mean square methods. The idea behind this method was first put forward by Kolmogrov (1957) and later both Hopfield (1985) and Kohonen (1984) published work on self-organising neural network models that could perform error minimisation. An example of the self-organisation of an initial definition of 2-D space is shown in Figure 2.4. As the number of points

in an image requiring adjustment, e.g. for shrinkage or expansion of the original, increases the method of least squares becomes less viable even on a very fast computer. A self-organising network could theoretically cope with a very large number of points once the initial mapping had been completed.

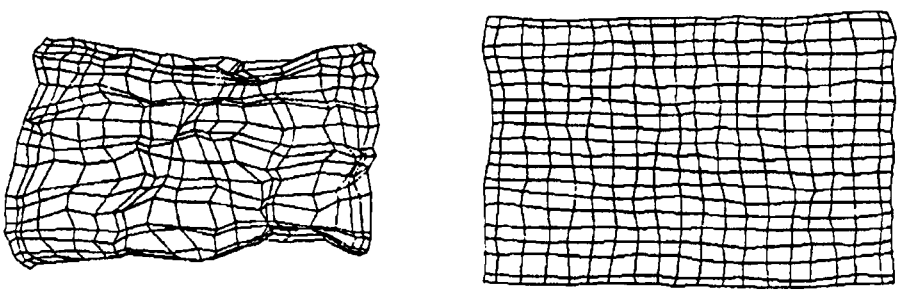

Figure 2.4 Minimising error using a self-organising network.

2.5 Digital Image Processing Methods

Although it is not the intention of this research to use primitive image processing techniques it is important to describe these basic building blocks as they are used to pre-process the images prior to use in neural network models. Neural networks require known pattern associations in order to predict unknown associations. For example SPOT three band images can be used in neural networks for land cover classification in a similar way to clustering techniques. The three bands are used as an input vector and the output vector represents the class membership. A 256 greylevel image must be processed to produce a set of unique patterns such as gradient or standard deviation to

provide an input vector for the network that produces a strong relationship between input and output vectors.

2.5.1 Contrast stretching

Contrast stretching can be used to process images before image matching. It operates by increasing the contrast of the image and making use of all of the available greylevel values, in this case 256. The effect of contrast stretching on edges will be to increase the edge gradient by the same amount as the increase in contrast. Therefore the ratio of the gradients of any edges will be maintained. However the difference in gradient will be increased. Hence any image processing method that uses differences will be enhanced by the use of contrast stretching. Gradient function results will therefore be improved by contrast stretching the image first. Other functions could be used to stretch the contrast non-linearly. If a non-linear function is used then some prior knowledge of the brightness distribution would be required in order to decide which specific function would be the most effective. Histogram equalisation is a common method of automatic contrast stretch. Figure 2.5 shows the effect on an image that has very low initial contrast and very little detail can be seen in the original image. The contrast stretch spaces out the high histogram frequencies and combines the low frequencies.

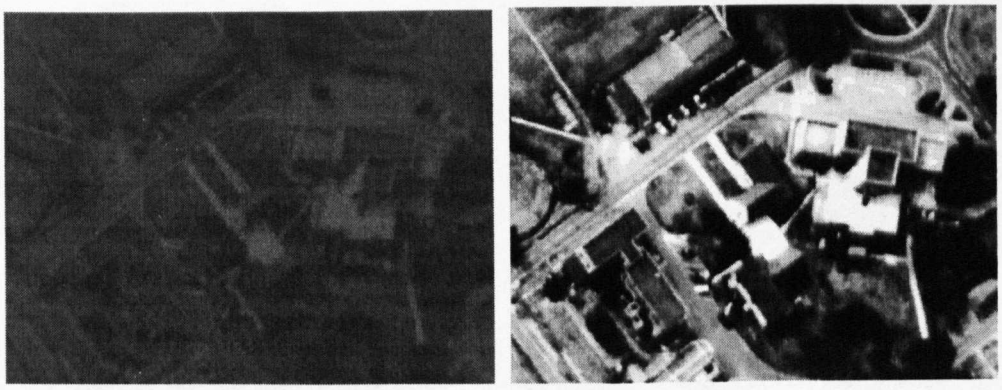

Figure 2.5 Histogram equalisation of an image.

2.5.2 Density slicing

Density slicing is a method of mapping a range of greylevels to a new single grey level or colour (Figure 2.6). It is often used to visualise areas of a single-band image which are significant. The improvement in the interpretation of the image is balanced by an accompanying loss of detail. The most common method defines a number of slices and then divides the image into an equal number of slices. This form of density slicing gives a linear distribution of greylevels. Non linear forms of density slicing include adjusting the greylevel distribution in relation to the histogram frequency. Density slicing may also be used as a pre-processing technique prior to using the greylevels as input data to another image processing method or to a neural network.

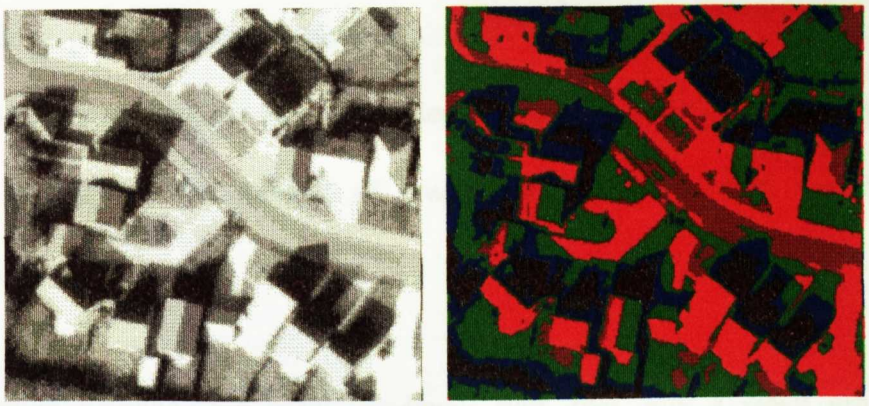

Figure 2.6 Density sliced image where the greylevel slices are represented by colours.

2.5.3 Gradient functions

One method of edge detection is the use of a measurement of the greylevel gradient. Most gradient techniques use some form of difference function. It may be the difference of a pixel from the local mean or the difference between neighbouring pixels. These difference measurements are the discrete equivalent of differentiation of the image. Most methods use first differences but some also use second and higher order differences. If the images were one dimensional then the problem of choosing the most suitable method of measuring gradient would be straightforward. The introduction of the second dimension means that the gradient also possesses an orientation. Possible types of gradient profile are: step, ramp and curve. If pixel position is plotted against greylevel then step, ramp and curve describe the shape of this graph. By choosing the correct gradient type for an application the noise and spurious edges detected can be reduced. Figure 2.7 shows a section of a building wall as

a graph of greylevel against pixel position. It is interesting to note that the gradient is not a discrete step but a steep ramp. At this high resolution, 10 pixels per metre, the method of gradient measurement must be chosen carefully.

Comparing neighbouring pixel greylevels will give only a small gradient, whereas comparing greylevels 10 pixels apart will produce a tenfold increase in the result, in this example. Figure 2.8 shows an example of measuring gradient across three pixels and Figure 2.9 across nine pixels. This type of gradient measurement is in fact a measure of the gradient type, the value of a step gradient will be greater than that of a ramp. The value calculated is a measure of the sum of the gradient across the number of pixels chosen. This method produces less noise, but thicker edges as the number of pixels chosen increases.

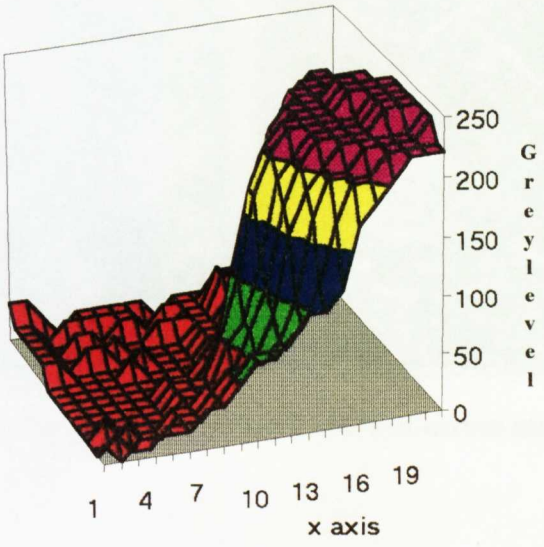

Figure 2.7 Graph of greylevels of a section of building wall.

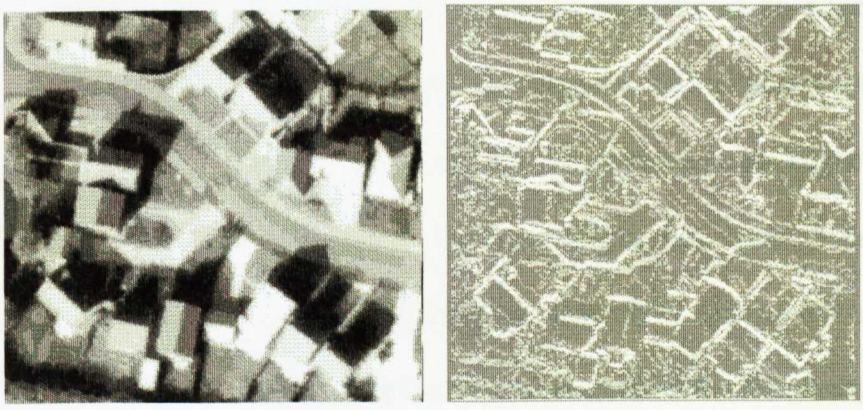

Figure 2.8 Measurement of greylevel gradient across three pixels.

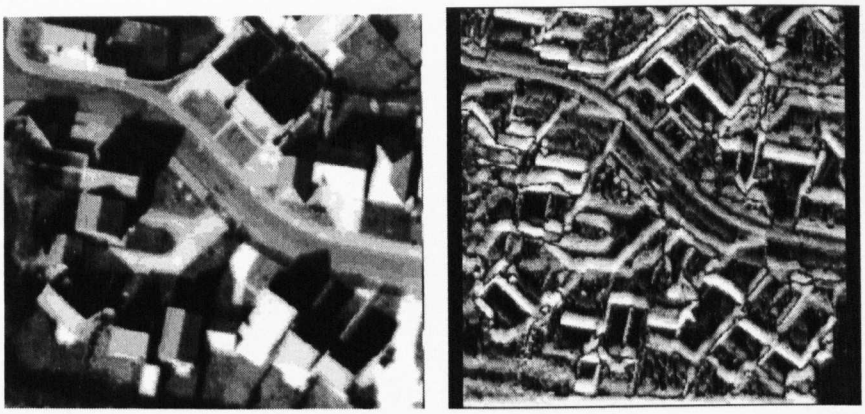

Figure 2.9 Measurement of greylevel gradient across nine pixels.

2.5.3.1 First Difference

The first difference, (Figure 2.10), will give the gradient and sign of the difference between pixel greylevels. The first difference can be used as an edge detector as the gradient will be highest where there is an edge. The resultant image will give gradient values wherever the greylevel changes, therefore in order to extract stronger gradient levels and hence the edges within the original image a threshold value for the gradient will have to be used. However, in an unsupervised system it may be difficult to determine a suitable threshold. The gradient can be considered as a measure of the frequency of the variation in greylevel; the first difference therefore is a direct measure of frequency.

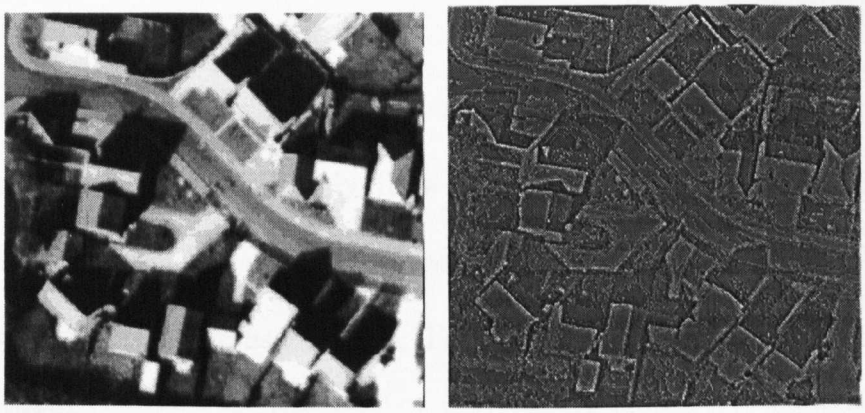

Figure 2.10 First difference image.

2.5.3.2 Second Difference.

The second difference is the change in gradient. This is the basis of the Laplacian operator (Figure 2.11). The second difference is useful mainly for finding lines since the zero crossing will denote the change in gradient from positive to negative across the line. It is not therefore ideal for detecting step edges where the gradient changes abruptly but may not change sign.

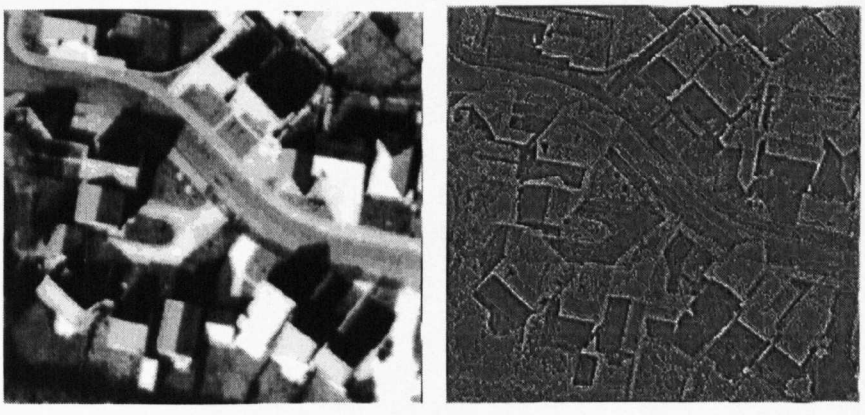

Figure 2.11 The result of applying the Laplace function to an image.

2.5.3.3 Robert's cross-gradient

Robert's cross-gradient measures the differences across the diagonals. This type of function comes directly from the analogue method of calculating gradient. The gradient is scaled to remain within the range 0..255 so that the results can be visualised as an image, (Figure 2.12). Robert's cross gradient is found from:

$$g = \sqrt{\left((a-d)^2 + (b-c)^2\right)}$$

where:
g = gradient
pixels arranged: a b / c d
this expression is sometimes simplified to:

$$g = \text{abs}(a-d) + \text{abs}(b-c)$$

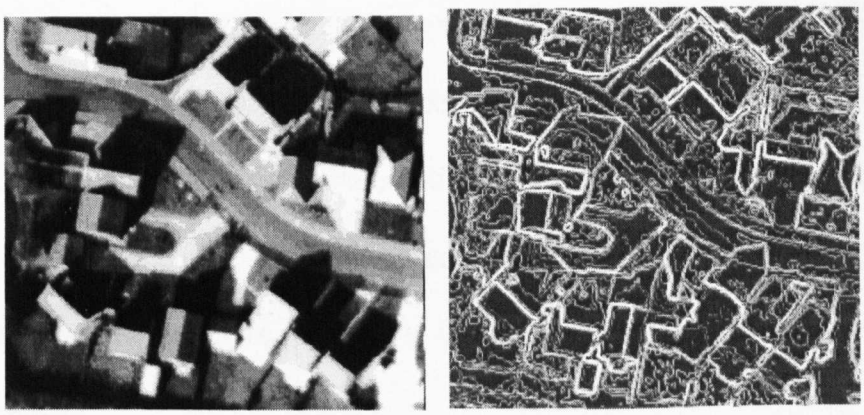

Figure 2.12 The result of applying Robert's cross gradient to an image.

2.5.3.4 Gradient functions and stereo image processing

Only the edge gradient values that are from edges across the direction of flight in a stereo pair of images are required in stereo image correlation as the parallax differences exist along the flight axis. Therefore, the gradient function applied to the images could be restricted to one dimension.

2.5.4 Filters

If the greylevels in an image are plotted against position a wave form will be produced which may contain a mixture of low and high frequencies. A filtering that separates the low frequency components is called a low-pass filter and a filter that only leaves the high frequency components is called a high-pass filter. A low-pass filter can be implemented by replacing the greylevel of an individual pixel with the average greylevel of the surrounding pixels. High-pass filtering could be achieved by subtracting a low-pass filtered image

from the original image thus leaving only the high frequency components of the image. The average greylevel can be found by calculating the mean or the median (Figure 2.13). The median, unlike the mean, filter maintains the edge detail whilst smoothing the grey levels and reducing high frequency noise. The median filter is therefore preferable if the next process is edge or line extraction.

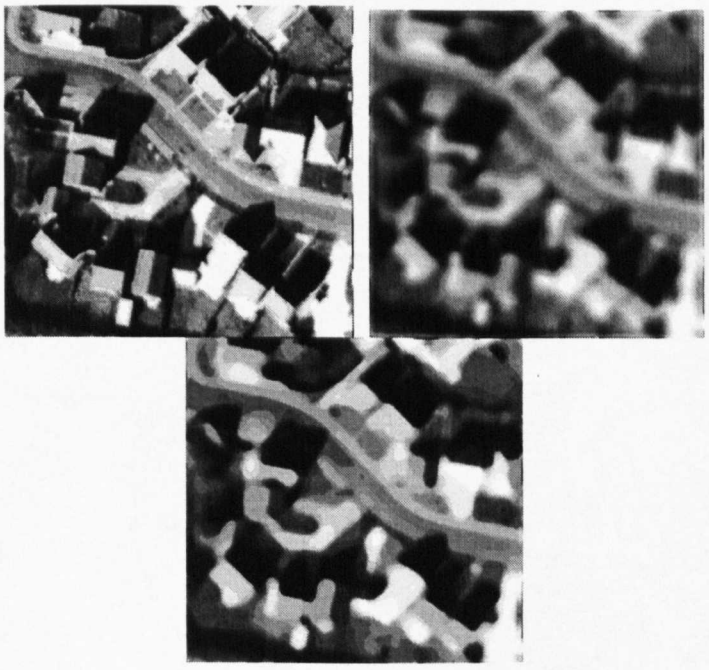

Figure 2.13 Original image with mean and median filtered images respectively using a 9x9 mask.

2.6 Methods Based on Image Processing

Without some or all of the previous basic image processing techniques more complex manipulation and analysis of images would not have been developed.

This section includes image processing methods that automate the production of terrain height contours, use the correlation of stereo images to attempt to identify building edges, predict the position of a designated point in a sequence of images and use shadows to identify building edges. Some of these methods are the work of military research teams for use in autonomous weapon aiming and terminal guidance systems. The remainder are related to civilian GIS research projects.

2.6.1 Digital Terrain Models

The derivation of digital terrain models (DTMs) of bare terrain are based in general on two techniques: feature-based and area-based. The derivation of DTMs is discussed in chapter 3. Feature-based techniques include those developed by Pollard, Mayhew and Frisby (1985) and Barnard and Thompson (1980). Area-based techniques include the most cited algorithm by Gruen (1985) and a variation on this by Otto and Chau (1988). All of these methods make the assumption that the terrain is a continuous surface with no abrupt changes or occlusions. Although the above techniques were not originally designed to operate on urban images, some of the underlying principles involved are relevant and are therefore discussed here.

There are also two significant stereo techniques that set out from the start with prior knowledge of the difficulties outlined above. The first, due to Henderson,

Miller and Grosch (1979), uses edges within the image to guide the matching process and is called broken segment matching. The second, introduced by Lacina and Nicholson (1979), extends the idea of stereo to multiple images using the known motion of the sensor platform, termed motion stereo, to predict the position of matching points in successive images. Both of these techniques were developed in response to the need for a terminal guidance system for cruise missiles and as a result they are adaptable to low oblique and vertical images.

Strangely, the later two methods are not referred to by any of the other research groups working in this area. Muller *et al.* (1988), a very important research group in this area, make no mention of the work of Henderson or Lacina. This seems unusual as their work makes a significant break away from all other research in this area. It can only be assumed that the military nature, cruise missile guidance, has restricted the access for some considerable time.

2.6.2 Broken Segment Matching

Henderson, Miller and Grosch (1979) developed a technique for automatic stereo reconstruction of cultural targets such as buildings. The need for this research was prompted by a military requirement for automatic terminal guidance of certain weapon systems. This method, termed broken segment

matching, is based on stereo image correlation but is specially adapted to cope with the problems that occur where the change in height across the image is not continuous, as with images containing buildings. This method does not use statistical correlation techniques but directly relates edges that occur along the same scan line in each image of the stereo pair. As this method is not described in standard texts or in the literature it is covered here in some detail.

Broken segment matching is carried out in a series of steps, starting with epipolar alignment of the stereo pair of images. The camera alignment must be known to be able to epipolar align the images. The reason for using images that have been processed in this way is to reduce the stereo matching process from 2-D to 1-D. Epipolar aligned images have all points common to both images along the same scan line as shown in Figure 2.14. The process of epipolar alignment is dependent only on the camera orientation and not the scene geometry, which is a vital property since the content of the images is assumed to be unknown. Moving the matching from 2-D to 1-D is a very important step because it simplifies the search pattern for matching points required to deal with occlusions caused by the buildings in the images.

The method proceeds by matching each scan line at a time, the matching positions along the scan line noted separately as shown in Figure 2.15 for line 6 of Figure 2.14. The points where there are changes in gradient in are referred to as breakpoints and these breakpoints indicate the position of vertices. The

positions of matching segments in a line can be used to assist in matching the positions of points on the following line as the displacement of the points on a line will only vary slightly, and unless there is a occlusion then the gradient will be 100% until another matching point is found.

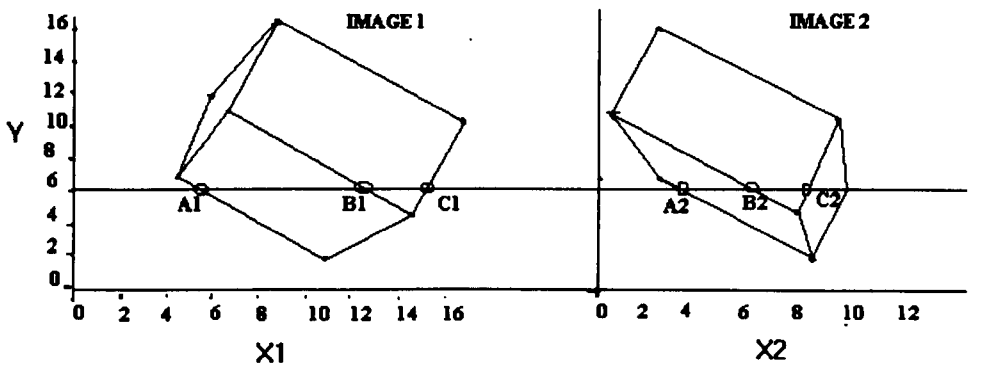

Figure 2.14 Epipolar aligned images ready for broken segment matching.

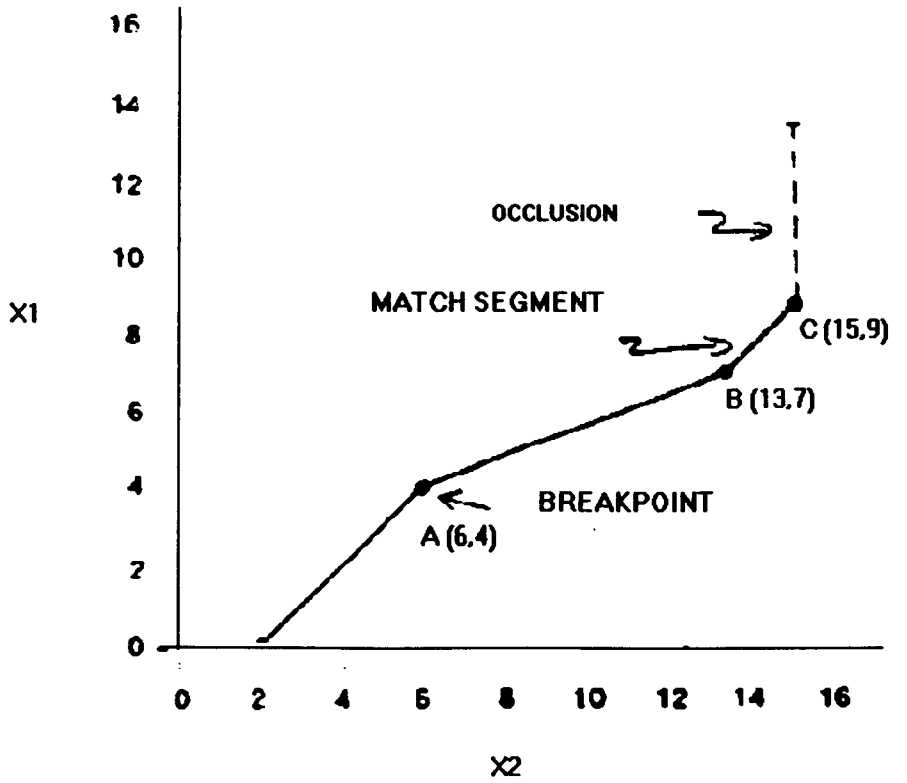

Figure 2.15 Broken Segment Matching for scan line 6.

This method requires an image that can be processed to provide strong and reliable edge detail, so that the edges can be extracted using a process such as Sobel filtering. However in areas where complications such as shadow exist, the process will be entirely dependent on the success of the edge-finding algorithm and will require edge detection in the shadow areas. Shadows are not always a disadvantage, as demonstrated in section 2.4. The success of this method is demonstrated in Figure 2.16. It must be remembered when comparing this technique with others that the computer power available in

1979 was small compared to today, and the simplicity of the image on which it was tested must also be borne in mind.

The resulting model generated by the broken segment technique has captured the broad outline of the test image. An improvement in the result might be achieved by increasing the resolution of the input image and, in addition, low-pass filtering to reduce the noise induced by the edge detection process. It would be interesting to test this technique on a representative urban image, with shadows and a more complex scene. As the method requires good edge extraction for the images to be matched it is likely that it would be limited by the signal/noise ratio of the edge extraction algorithm. Edge extraction, something that the human visual system apparently achieves very easily, is equally something that seems to be very difficult to achieve using image processing techniques. It is the edge extraction difficulties that prompted a further investigation into techniques used in areas associated with the human visual system and hence neural networks.

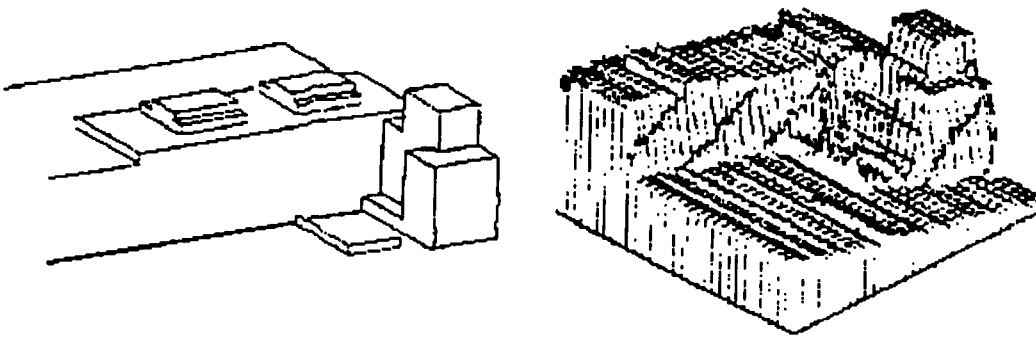

Figure 2.16 Broken segment matching result, left is hand drawn, the right generated by the matching process.

2.6.3 Motion Stereo

As a direct descendant of stereo image processing, Lacina and Nicholson (1979) developed a technique that uses multiple images. The motion of the sensor platform is assumed to be known and a point in one image is located in the subsequent image in the sequence, by using a set of equations that predict the motion of that point. The process requires the definition of a motion vector field which defines the change in position of all points in one image to the next in the sequence. It is not one-to-one since some points will become occluded as the sequence of images progresses and others will be revealed. The equations that are developed to predict the vector field behaviour are time-dependent as the motion of points within the image is directly dependent on the velocity of the sensor platform. It would be possible to produce the motion vector field from just two images; however, a much better result can be achieved by combining the results from several images. The use of multiple

images smoothes out the effects of image noise and inaccurate platform position data.

An inherent problem with low oblique images is the change in scale of objects in the image as the sensor platform approaches. One possible way to minimise this problem is to restrict the areas within the image that are tracked. Corner points of buildings are ideal as the correlation algorithm can work efficiently with the three intersecting edges. This idea of geometry correlation rather than image grey level intensity correlation is important as it overcomes a fundamental problem encountered by any correlation process, namely variability of intensities due to time of day, weather, scale factor and shadows. The difficulty with geometry correlation is that the image will require pre-processing to extract edges and lines with all the associated problems of occluded lines and lines in shadow, as noted above.

This technique involves a considerable computational overhead, which explains why in 1979 it was a military research project; few other agencies would have had the computer power to make it a viable option. However, in 1994, the application of 3-D equations of motion to every pixel in an image is well within the scope of a PC. Limitations of this technique include the requirement to have accurate position, velocity and direction data for the sensor. This restriction might make the technique unsuitable for accurate position calculation that would be required if the data were to be used for

mapping but would be quite suitable for target acquisition in a military application.

Figure 2.17 Motion Stereo, movement of a point.

2.6.4 Shadows

Buildings cast shadows when illuminated which for the operator of an analytical plotter is a distinct disadvantage as details that are in the shadows cast by buildings can be obscured. The shadow of one building may obscure the edges of an adjacent building, making it impossible to clearly define its shape. However, both Huertas and Nevatia (1988) and Liow and Pavlidis (1990) use the analysis of shadows within their methods of building extraction. Hsieh, McKeown and Perlant (1990) and McKeown (1990) also include shadow analysis in their work. Irvin and McKeown (1989) identify shadow analysis as being so important that they develop two specific routines SHADE and SHAVE.

Liow and Pavlidis (1990) make the use of shadows the prime method of extracting information on buildings from aerial images. They used the shadows to improve the detection accuracy and reliability of their classification methods. The high contrast of shadow edges enables edge and line extraction from images with high signal to noise ratio. The method uses edge detection to find boundaries within shadow and a region growing technique to find boundaries without shadow. The technique has been applied in two ways: edge detection then region growing, or region growing then edge detection. These methods, in using images with high shadow contrast, i.e. collected on a very sunny day, restrict them to this type of image and would therefore be degraded by images taken on overcast days when there is little or no shadow. Analysis of the shadow regions in an image is a useful addition to any method but would be very restrictive if used in areas of the world such as Britain where diffuse illumination, due to clouds obscuring the sun, is common. The building types are restricted to those with a flat roof and with little or no texture. Building boundaries are assumed to be straight lines but there is no restriction on shape. The boundaries are confirmed by first checking that the shadow direction is compatible with the known position of the sun and then a search is carried out for pairs of perpendicular lines.

Huertas and Nevatia (1988) use shadows to verify the existence of buildings then link the straight lines that are extracted. The lines form closed polygons and if any lines are missing then line prediction is used. The inclusion of

shadow analysis in their work has assisted in distinguishing between polygons that are flat surfaces, such as car parks, and polygons that are buildings, since only 3D objects cast shadows. The shadow information is also used to estimate height. To make use of the shadows in this way the sun angle must be known and the camera principal ray must point to the centre of the scene. As with the Liow and Pavlidis (1990) method discussed above, the buildings are assumed to have flat roofs and the shadows are cast onto level ground. Liow and Pavlidis also assume that there will be missing lines and corners and developed methods of predicting the position of the missing elements. The alternative approach would be to pre-process the image so that the line and corner detection stage was more successful. It is hypothesised here that a degree of both pre-processing and post-processing is required if all of the building boundaries are to be extracted. Post-processing will require a definition of the expected building shape so that analysis of the resulting lines and corners produces polygons that are buildings. This definition will therefore restrict the types of building detected. Several difficulties are highlighted by Huertas and Nevatia (1988) where buildings are missed either partially or completely, due to poor contrast and texture, small size, complex roof structures and/or the density of the buildings. Better edge detection in areas of low contrast, higher image resolution, improved shadow analysis and the uses of stereo correlation are suggested as areas of further research in the quest to find a solution to the above problems.

The routines developed by Irvin and McKeown (1989) are SHADE and SHAVE, which are shadow detection and shadow verification respectively. SHADE identifies the boundaries of the dark areas of the image, in order to delineate regions that are in shadow (Figure 2.18).

Both smoothing and thresholding of the image are performed. The function of SHAVE is to determine the edges that are adjacent to shadow, delineate the shadow region and make a quality assessment of the area of shadow. Additionally, SHAVE estimates the height of the associated building, which requires the prior knowledge of the sun angle. The results of SHAVE can be seen in Figure 2.19.

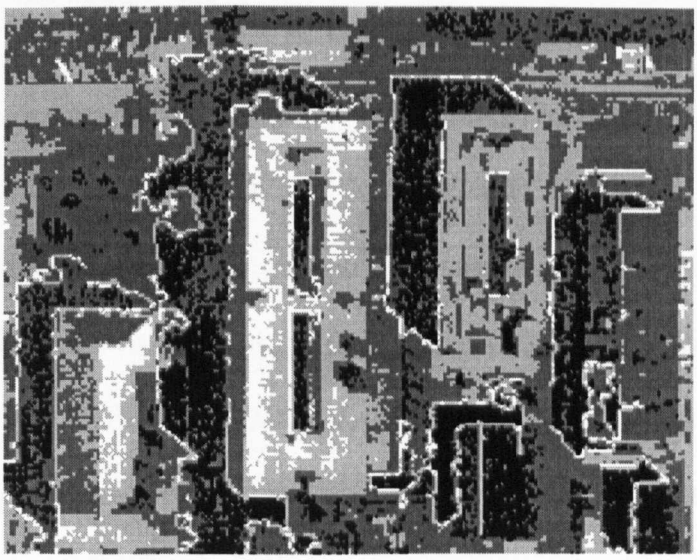

Figure 2.18 Shadow regions found by SHADE.

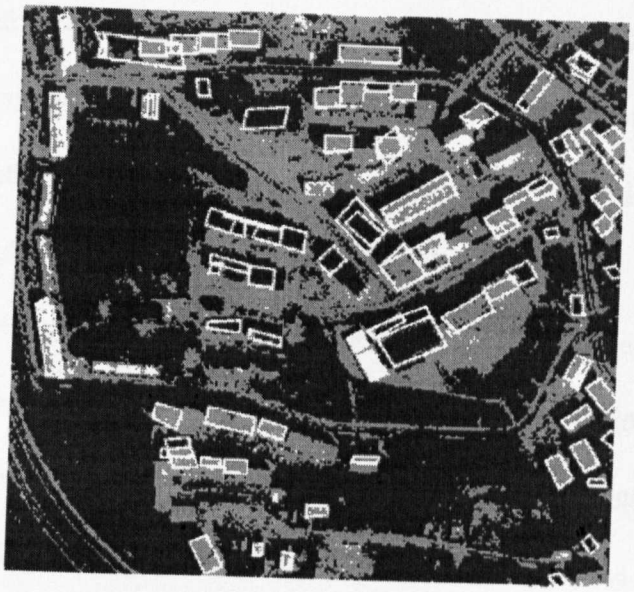

Figure 2.19 Building boundaries extracted by SHADE

These two techniques are integrated into the work of Hsieh, McKeown and Perlant (1990) and McKeown (1990). McKeown (1990) mentions the use of shadows in image registration because the real world position of shadows is independent of the view point. This independence makes the position of a shadow edge or corner useful in stereo correlation. The shadow size may vary due to being occluded by the building in one view relative to the other but the outer corners should fall on the underlying terrain in the same place (in fairly flat areas this will also be the same image position). Restricting the images tested to those with flat roof buildings removes the problem associated with a typical European pitched roof; the shadow cast by the ridge hides the down-sun building edge.

Both Liow and Pavlidis (1990) and Huertas and Nevatia (1988) encountered problems in extracting edges in the non-shadow, low contrast and low signal/noise ratio, areas of images. They state that "there is no algorithm accurate enough to detect all lines perceived by the human visual system" (Huertas and Nevatia 1988). Several methods are developed which partly alleviate this problem, including prior knowledge of the building shape and using only edges belonging to shadow. The result of initial edge extraction by Liow and Pavlidis (1990) is shown in Figure 2.20. Note the building edge lines are in the minority.

Following the edge extraction process, a region-growing procedure is applied followed by verification of building lines by use of the shadow information in the image. The result is shown in Figure 2.21. These results are achieved with requirement for good illumination to produce shadow and flat roofs so that the whole roof has the same texture.

The segmentation process proposed by Liow and Pavlidis (1990) performs well and is probably one of the best to date. However, to achieve the result in Figure 2.21 the original image has to go through 12 or more separate processes. They have taken the path of bottom-up segmentation, extracting primitives then connecting these together to identify the building boundaries. As the process proceeds problems are encountered and a method has to be developed to overcome it. This ad-hoc technique can be rather similar to plugging a hole in

a pipe only to find the increased pressure now makes it leak somewhere else. They have plugged most of the holes, but at the price of a large processing overhead.

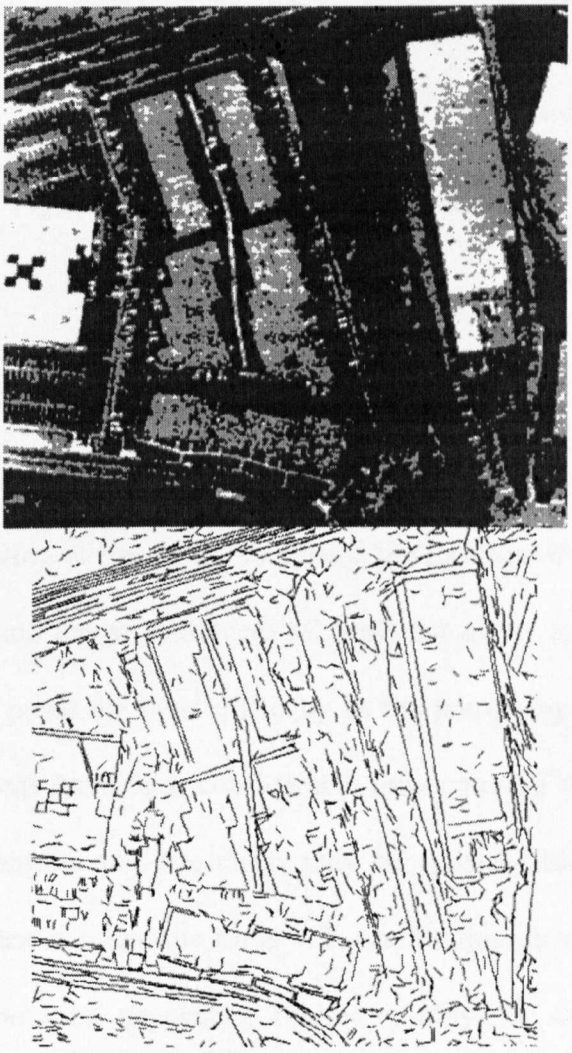

Figure 2.20 Extracted edges using a method developed by Liow and Pavlidis (1990).

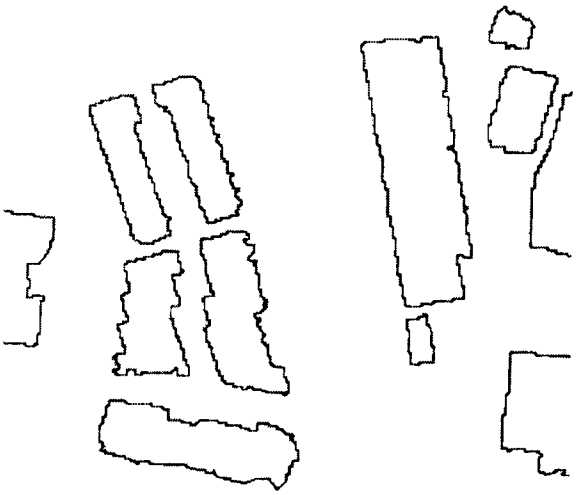

Figure 2.21 Extracted building boundaries.

2.6.5 Stereo Correlation

Marr and Poggio (1976) defined the correlation of two images as comprising three phases: location of a point in one image, the location of the same point in the other image and the measurement of their disparity. It is the second of these phases that poses the most difficulty as the point may not exist in both images or there may have been considerable time between the two exposures and hence the contrast and brightness may be considerably different. The second of these case is often true for SPOT satellite images where there is one or more orbits between exposures. Pridmore, Mayhew and Frisby (1990) developed a method of correlation which is now often referred to as PMF after the authors. They considered Julez (1960) in their work and were in agreement that the human brain has the ability to see depth where there are no monocular clues. However, their research made use of algorithmic image processing

rather than follow the model presented by the brain. It is to be remembered that even in 1990 the more powerful neural network models were still under development and thus not available as they are now to this subject in research.

2.6.5.1 Methods

As noted by McKeown (1990) no one method of stereo image correlation can be used for urban images. He proposed two methods: one area based and the other feature-based. To reduce the computation time a constraint of epipolar alignment of the images pair is required. This reduces the search for matching points to the same scan line in each image, if the alignment is accurate. The alignment is achieved by using unique features such as building edges or shadow. The area base technique uses several levels of resolution starting at coarse and moving to fine in order to cope with large image disparities. The feature based technique compares the grey level along matching epipolar lines. The grey levels are converted to waveforms which when superimposed from one image to the other, if a match exists, can be positioned so that the peaks and troughs align. The results of one scan line are used to guide the next alignment, a technique similar to that used by Henderson *et al.* (1979) in the broken segment matching technique. Figure 2.24 shows an example of the technique I have applied to a scan line in a stereo image pair of a house in Figure 2.22 and Figure 2.23.

McKeown (1990) notes that, although both techniques work, they do not capture the depth discontinuities found in an image with buildings, but define the underlying terrain well. In addition to inter-scanline consistency he also uses intra-scanline consistency, that is continuity along a scanline. However, with buildings in the image there will be disparities caused by occlusions which will require sections of the scanline in one image to be unmatched.

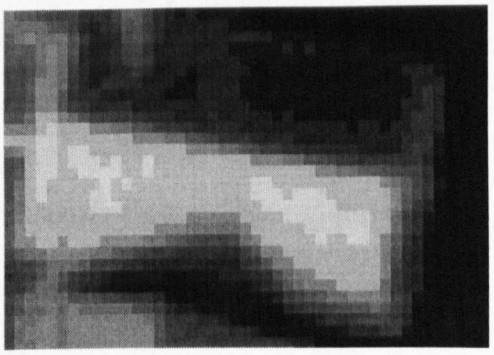

Figure 2.22 House in left image of stereo pair.

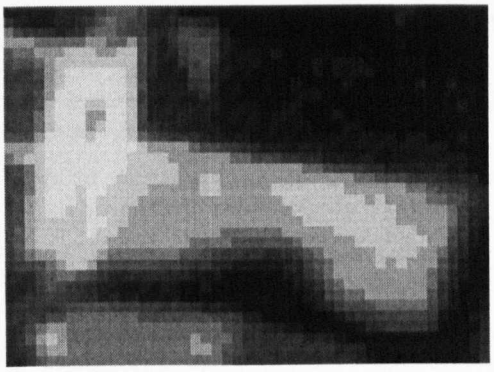

Figure 2.23 House in right image of stereo pair.

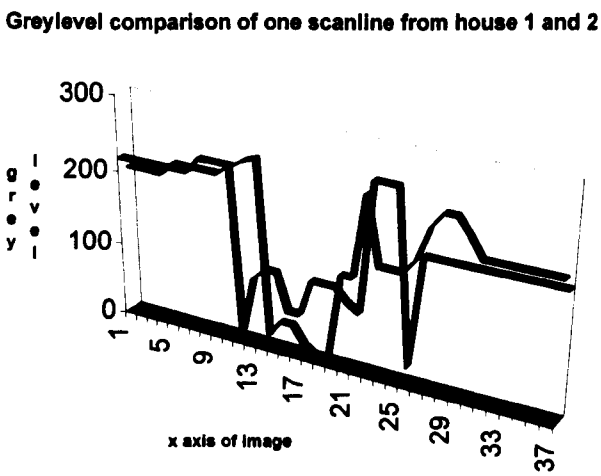

Figure 2.24 Scanline matching.

2.6.5.2 PMF Stereo Correlation

The PMF method, named after its originators, Pridmore, Mayhew and Frisby (1990), follows much previous research into stereo image correlation in identifying a need to limit to the gradient of the disparity between the images. The limiting of the gradient provides a balance between the requirements to remove ambiguity and to deal with real world images, many of which have only moderate disparity gradients. Another problem which arises in stereo correlation is the constraint of uniqueness, which requires that each point in one image must match only one point in the other image. The disparity gradient constraint can help to achieve the uniqueness constraint. An example of the need for these constraints can be seen in images with areas of similar texture. PMF has a degree of the image texture independence. The method may be used to correlate edge, blob like and bar like textures. However, the

built-in independence creates a problem with images having similar horizontal lines or edges as all horizontal lines will match all other horizontal lines in the image (Figure 2.25). This problem arises because lines are treated as a series of points. In later modifications to the method this problem is overcome by identifying lines as single entities. These limitations of the method are of particular importance when application to an urban pair of images is contemplated as these type of images contain many lines, some of which may be near-horizontal.

In the development of PMF the method of calculating the matching strengths between points in a stereo pair of images two strategies were used. The first used a limited matching neighbourhood based usually on the left and the second used a cyclopean image, a monocular fusion of both images. In resolving the ambiguity of matching points the results of experiment showed that the matching strengths of the correct matches were always considerably greater than the incorrect matches.

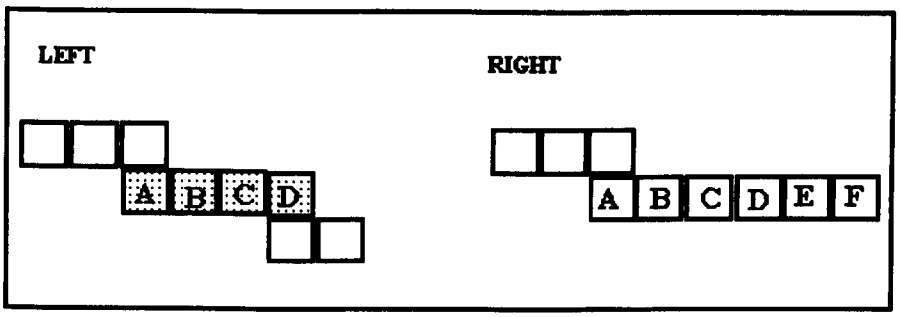

Figure 2.25 PMF Horizontal line ambiguity. Pixel A in the left image will be able to match any of A, B or C in the right.

2.7 Digital Image Correlation

The basic tool required in any stereo image matching system is a correlation function. There are several available all of which are based on relating small areas in each image by comparison of the greylevels. Most correlation functions use the mean greylevel and the deviation in their calculation as described by Gruen (1985). By this means some adjustment for the brightness and contrast difference between the images can be made. In this research probably the most generalised function, but probably not the quickest, as time is not a consideration at the moment. The results are scaled over the range 0..255 so that a perfect match is 255, this makes visualisation of the result possible as an image.

2.7.1 Correlation Method

Correlation is a measure of the similarity of two sets of data, in this case pixel greylevels. The method need not be restricted to just greylevel but can also be

applied to calculated values of the image such as edge gradient. Later in this research the use of calculated image values is used in conjunction with neural networks with a similar aim of measuring similarity, in this case between a set of training values rather than the stereo pair. The correlation method when applied to a greylevel image can be calculated in the simplified form:

$$k = \frac{\sum_{ij}\left[\left(a_{ij} - \bar{a}\right)\left(b_{ij} - \bar{b}\right)\right]}{\sum_{ij}\left(a_{ij} - \bar{a}\right)^2}$$

$$c = \bar{b} - k\bar{a}$$

$$r = \sum_{ij}\left(b_{ij} - \left(ka_{ij} + c\right)\right)^2$$

a, b : pixel greylevels.
$\bar{a}, \bar{b}$: mean pixel greylevels.
i, j : width, height of sampled sub-image.

If r is zero then the sub-images are identical, independent of the contrast difference k and brightness c. This method of reduces the computation time when used for stereo image correlation. The standard correlation coefficient used in statistics has the form:

$$c^2 = \left(\frac{\sum_{ij}\left[\left(a_{ij} - \bar{a}\right)\left(b_{ij} - \bar{b}\right)\right]}{\sum_{ij}\left(a_{ij} - \bar{a}\right)^2}\right)\left(\frac{\sum_{ij}\left[\left(a_{ij} - \bar{a}\right)\left(b_{ij} - \bar{b}\right)\right]}{\sum_{ij}\left(b_{ij} - \bar{b}\right)^2}\right)$$

This is the product of two contrast differences:

$$c^2 = k_a \bullet k_b$$

If the contrast of ***a*** is the inverse of ***b*** then the image patches are only different because of a difference in contrast, as the remaining factor of brightness is a constant, the patches are identical. The square of the coefficient therefore gives a result that is between zero and one. If the image patches are identical then c^2 is equal to one. This method of using least squares approximation requires more calculation than the basic form above but is easier to use as the values it produces are in a distinct range and can be scaled, if for instance the result is required as a percentage.

2.8 Image Classification

Classification is a tool used in thematic mapping which generally assigns each pixel in an image to a group or class. This assumes that there are a distinct and known number of classes that each pixel can be assigned to. However, it is more common to find that some pixels will have to be assigned to an unknown class. The process of classification can be supervised or unsupervised. Supervised methods assign pixels to classes based on the statistical characteristics of samples of pixels representative of a presumed or known number of classes. Unsupervised methods allocate pixels to distinct classes

based on measures of resemblance or similarity. These methods determine non-overlapping groups usually in terms of spectral band values. These groups are then related to actual classes by comparison with known data about the image area. The most common classification techniques allocate class by pixel these are known as 'per-pixel' or 'per-point' methods and are based on spectral band data.

Three types of feature that are most widely used include textural, contextual and spectral. Of these three the most common feature type is spectral. The spectral bands produced in SPOT images provide a ready made data set for land cover classification. For images which only have one spectral band such as greylevel images texture, edge gradient and similar features can be produced from the original single spectral band.

2.8.1 `Geometric Definition of Classification

To illustrate a classification method Figure 2.26 shows two bands of an image shown in graphical form in order to visualise the clustering of pixels into groups or classes. The density and separation of the clusters is intuitive to visual inspection. However, finding a suitable mathematical method to express this is a non-trivial task. One method to define the classes measures the Euclidean distance between pixels in 2D space using Pythagoras's theorem (Figure 2.27). Given the distance from each pixel to every other pixel those

pixels that are close to each other can be assumed to belong to the same class. Computer algorithms to solve the calculation of Euclidean distance for every pixel in a typical 512x512 SPOT image must be optimise for speed since the time factor is not inconsiderable. The method can be generalised and extended to any number of dimensions and therefore spectral bands or features using the generalised form of Pythagoras:

$$d_{ab} = \sqrt{\sum_{i=1}^{p} \left(x_{ia} - x_{ib}\right)^2}$$

where i is the axis number and a and b are the two points between which the distance is to be calculated.

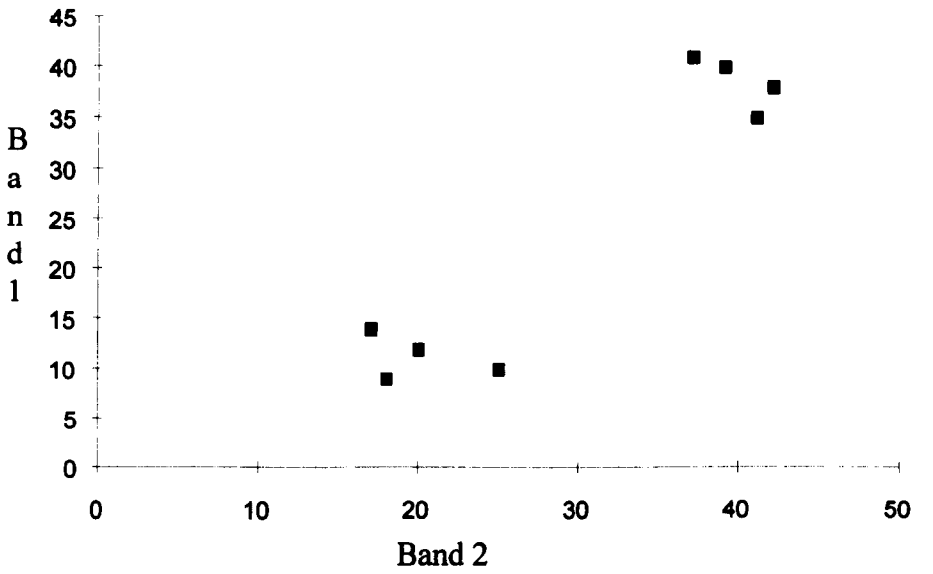

Figure 2.26 Two band image classification by identifying clusters.

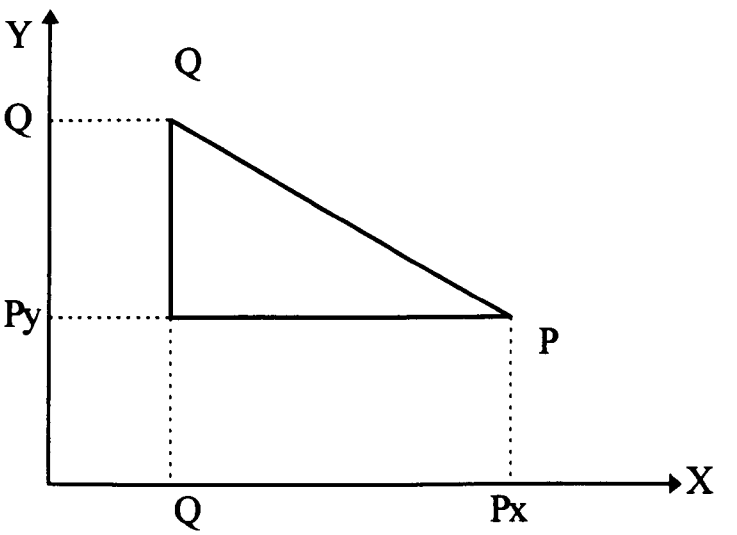

Figure 2.27 Calculation of Euclidean distance QP in two-dimensional feature space, using Pythagoras.

The generalised form of Pythagoras applied to Figure 2.27 becomes:

$$QP = \sqrt{\left((Qx - Px)^2 + (Qy - Py)^2\right)}$$

2.8.2 Classification Methods

Classification methods can be divided into two main types: supervised and unsupervised. In this research the supervised methods are most likely to be suitable as the aim is to identify one particular class. Unsupervised methods , however, can help to give an initial indication of the number of possible classes within a particular image type. Three of the most common supervised classification methods are the centroid, parallelepiped and maximum likelihood classifiers. A central feature of these methods is the definition of the shape of each class in the feature space. In two dimensions these can be thought of as geometrical shapes varying from ellipses to rectangles, n-dimensional forms being extensions of this concept.

2.8.2.1 Centroid

The use of Euclidean distance in classification methods has already been discussed earlier in this chapter. The centroid classification method is based on measuring the Euclidean distance from predefined points in n-dimensional space. These points are called centroids and are defined as the mean centre of each class. N-space is therefore divided up by a set of straight line boundaries, each equidistant from two centroids. The final shape of each region within

which each class falls is dependent on the number and position of the centroids (Figure 2.28).

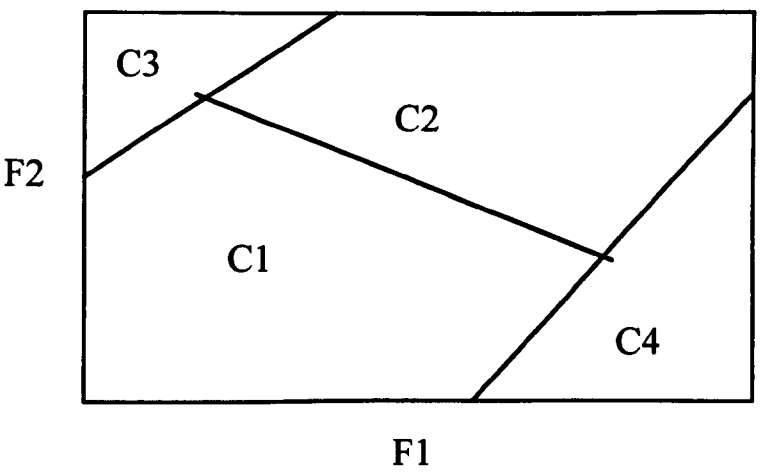

Figure 2.28 Example of centroid classification in a feature space of two dimensions defined by F1 and F2. The class centroids are labelled C2-C4.

2.8.2.2 Parallelepiped

A parallelepiped is an n-dimensional rectangle. The parallelepiped classification method divides the feature space into areas which are parallelepipeds, (Figure 2.29). The method takes each pixel in turn and using the values of each feature and assigns it to the parallelepiped that it lies inside. In this method it is possible for the classes to overlap and a pixel may fall inside one or more parallelepipeds. In this case a decision rule is used to

remove any ambiguity. The simplest rule would assign the pixel to the first parallelepiped or to assign these pixels a more complex rule.

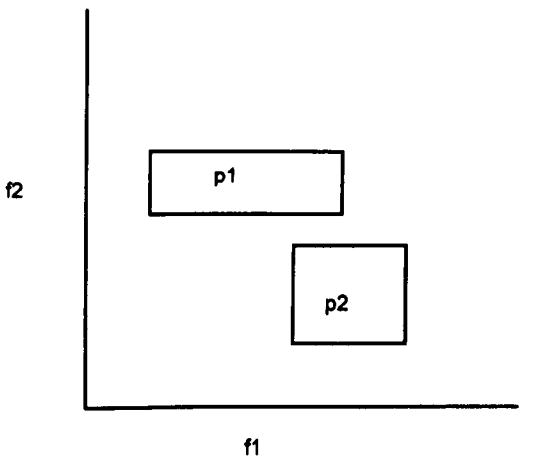

Figure 2.29 Example of parallelepiped classification in a feature space of two-dimensions defined by f1 and f2.

2.8.2.3 Maximum Likelihood

The maximum likelihood classification method describes each class in the feature space as an ellipse (Figure 2.30). The position, size and orientation of the ellipse is defined by the mean vector variances and covariance matrix of the n features defining the feature space. Positive covariance produces a slope of the ellipsoid major axis to the while a zero covariance produces a circle. Sets of concentric ellipses can be thought of as contours of probability of a pixel belonging to that class. This method is restricted by the need to assume that the frequency distribution of the class membership is normally distributed for each class. Although, the assumption of normality is rarely, if ever, satisfied in

practice, the technique is said to be robust, and its widespread use and acceptability implies that results are considered to be reasonable. Neural nets have the advantage that they do not demand conformity to any particular statistical distribution (they are non-parametric) and the measurement scale of the feature is not confined.

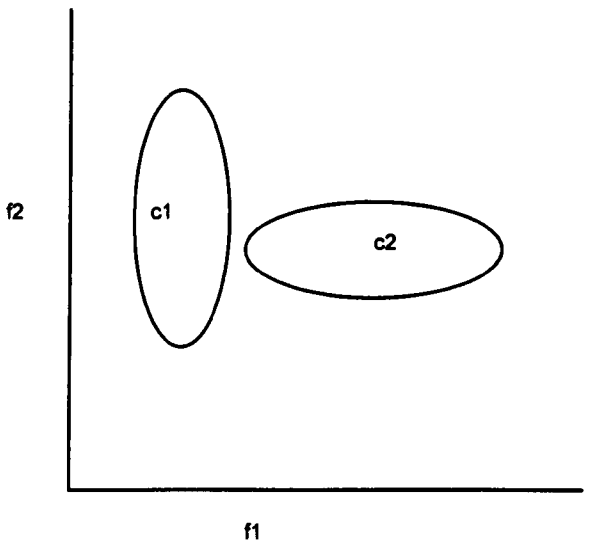

Figure 2.30 Example of maximum likelihood classification in two-dimensions (c = class, f = feature).

2.8.3 Non-spectral Features in Classification

Non-spectral features can be derivations of spectral features or features that are external to the remotely sensed data. Derivations of spectral features include all possible manipulations of the original image data, for example measures of the effect of the neighbourhood around individual pixels. One such neighbourhood feature is texture which in its simplest form can be expressed as

a measure of the variance of the pixel values in a neighbourhood. Non-spectral features that are external to the original data include elevation data. Elevation data could be derived, as has already been described in earlier chapters, by the use of multi-image information such as stereo pairs.

In classifying crops or other types of land cover spectral data alone can provide a sufficient pattern independence to make existing classification methods, as described above, effective. These methods are effective because the classes have a degree of homogeneity over the complete area of the class. An example would be a field of wheat. The field is made up of a repeating spectral pattern which could be a derived fractal as there are several layers of pattern dependant on the resolution of the image. In contrast an urban image has classes of object, buildings for example, which are made up of many different patterns. To classify a building requires more than the identifying one texture or one colour. A building is a complex conjunction of many and diverse features that uniquely define the complete object. A building is a mixture of edge gradient strength, of texture and colour.

If existing classification methods are used on urban images then each section of a building must be allocated to a separate class: the roof, the walls and the edge boundaries. Some of these classes may exist in other aspects of the image such as the footpath or the road. Therefore there are two ways forward - identify the building piecemeal by classifying the different aspects of the building, or

attempting to classify it as a complete entity. Referring back to the discussion of the human visual system shows that both methods are used by humans. First the individual aspects of an object are learnt, then these are seen as groups which define a separate complete object. Using non-spectral features the individual aspects of a building can be identified. These can then be amalgamated to define the complete object. Each aspect of as building can be thought of as a class in the traditional image classification sense and these classes can be further processed to identify objects that are made-up of several classes.

The method presented here can be thought of as extending present classification methods by grouping the results of classification to see if within the resulting classes there exist any composite classes that consist of groups of simple classes. If computing demands are ignored, then a new concept based on grouping basic or fundamental classes to generate new higher level classes can be accepted. A building is a set of classes. The methods required to identify these classes already exist and have been discussed earlier in this research. Neural networks can not only isolate the initial low-level classes but by feeding this low-level class data into another neural network the complete object, the building, can be identified.

2.9 Neural Networks

The 'Knowledge Acquisition Bottleneck' was put forward by Beard (1990) as a driving force behind the need to study biological knowledge systems and from this the need to build artificial neural systems. This bottleneck is caused by real world tasks having great diversity, where rule-based systems are unable to cover all possible definitions of seemingly similar tasks. The two main fields of research which have benefited from the development of artificial neural systems are image and speech recognition.

Neural networks started as models for the human brain and much of the early work on these models was based on the ideas of Freud and James dating to the nineteenth century. The unique feature of neural networks is the ability to learn by example rather than follow a set of instructions as would be the case with a 'conventional' computer. One of the first neural network designs was due to McCulloch and Pitts (1943). Unfortunately there were difficulties with the learning method. A solution to the problem of learning was proposed by Hebb (1949) who introduced a relationship between the weights assigned to a neuron and the activation level of that neuron.

Networks were originally conceived as one or more interconnecting layers of simulated neurones. Most simulations are designed around an input and an output layer with one or more hidden layers between. The human brain may not consist of layers but may be a three-dimensional matrix of neurones.

Simulation of a three-dimensional network in software is possible but difficult with conventional computer hardware since any crossing conductors must be isolated. The possibility of computers using light instead of electricity to transmit information may provide a solution to the three-dimension problem as light paths can pass directly through each other without interference, as suggested by Kosko (1987). This may now become a reality with the advent of protein-based computers as described by Birge (1995).

A hardware neural net can be constructed in a form where each neuron is a basic processor. Processing is then carried out completely in parallel providing a very fast system even with large networks. The speed of such a massively parallel system has obvious advantages in real-time applications, such as speech recognition or robot vision. The cost of such systems is probably the main disadvantage and therefore applications, such as building recognition, can use neural nets which are simulated in software on single processor computers.

The networks designed by Hebb were two-layer networks and were further developed by Rosenblatt (1957). Rosenblatt created a network that adjusted the weight assigned to a link between two neurons in proportion to the error between the two layers. He called his network the Perceptron. He also attempted unsuccessfully to design a three-layer network but was unable to produce a method of adjustment of the weights between the outer and middle

layers. In a three or more layer network the layers other than the input and output layers are called hidden layers.

In the early stages of the development of neural network theory two-layer networks were thought to be limited in their use and therefore great effort was put into the search for a learning rule for three or more layer networks. By the late 1960s the problem was still unresolved and this prompted Minsky and Papert (1969) to declare in their paper *Perceptrons* that two-layer networks were very limited in their use and, since there was no forthcoming method that would enable multi-layer networks to learn, multi-layer networks were not worth further investigation. In line with Minsky and Papert (1969), Aleksander (1966) had devoted his research to single-layer networks, from which emerged the WISARD recognition device (Aleksander, 1984). WISARD is built in hardware around the concept of a single-layer network. It is very fast in operation and can be trained to recognise complex objects such as human faces. The design and abilities of WISARD and other networks designed by Aleksander are considered as a possible method of recognition of buildings.

Two-layer networks studied by Kohonen (1984) produced a network referred to as associative memory network. Associative memory networks use unsupervised learning, the neuron weights being based solely on the input pattern. These networks were further developed by Kosko (1988) and are

known as bi-directional associative memory (BAM). Later work by Kohonen led to the definition of another unsupervised learning network which he called the learning vector quantizer (LVQ). Grossberg (1982) produced a model of the brain, based on what he called Adaptive Resonance Theory (ART), which could cope with information that changes with time. This model provides very compact representations of complex data. Hecht-Nielson (1987) combined the Outstar encoder and the Kohonen LVQ to produce a network called a Counter-Propagation Net (CPN). The CPN design is well suited to the task of pattern recognition and has been used in character recognition systems. CPNs are considered in this research as a possible network design for the recognition of buildings as their generalising capabilities were thought to provide a means to interpolate between known buildings in order to recognise the unknown. Practically this means that the net need only learn a small sub-set of building images, which cuts down the training time and memory requirements.

Werbos (1974) and Parker (1982) independently developed a backpropagation method for allowing multi-layer networks to learn, which meant the problems of multi-layer nets proposed by Minsky and Papert (1969) had been solved. Werbos called his method *Dynamic Feedback* and Parker his method *Learning Logic*. Backpropagation was further developed by McClelland and Rumelhart (1986) and subsequently by others in many different fields of research. These areas include image processing, character recognition, speech recognition and optimisation problems. Stereo image correlation is an optimisation problem

where matching strength and uniqueness of a match have to be resolved. This problem would be suitable for a Hopfield (1985) network which is often cited as an optimal technique for solving the travelling salesman problem. Backpropagation can be very slow as it is a repetitive process in which the neuron weights are adjusted until stable. Backpropagation nets are particularly useful for forecasting events based on multiple loosely connected events, a task which conventional pattern recognition methods would find almost impossible.

Halounova (1995) and Cappellini and Chiuderi (1995) have both carried out research on image classification using neural networks. The work done by Halounova compares the capabilities of a classic maximum likelihood algorithm and a backpropagation network to classify images in urban areas. The results from this classifier were to be used to calculate rainfall runoff in urban areas. Cappellini and Chiuderi used a counterpropagation network to classify non-urban areas in order to determine crop types. In general, remote sensing applications of neural nets for classification purposes have demonstrated a slight improvement in accuracy compared to standard techniques.

This section reviews various neural network models with the aim of choosing a model that is suitable for classifying an urban image into four basic categories: building, road / path, grass and trees. In particular, three neural network models, WISARD by Aleksander (1984), backpropagation by Rumelhart *et al.*

(1986) and counterpropagation by Hecht-Nielson (1987) are investigated in detail.

2.9.1 WISARD

The WISARD recognition device is named after the inventors Wilkie, Stoneham and Aleksander Recognition Device in Aleksander, Stoneham and Wilkie (1982). It is difficult to know whether to class this device as a neural network in the sense that has been discussed above. However, it is variously referred to as a single layer net or self-adaptive network. The device was first alluded to in Aleksander (1966), a paper on self-adaptive logic circuits. These logic circuits were totally hardware based and involved a degree of backpropagation by the introduction of a time delay from output to input.

WISARD was primarily intended for industrial use. The intelligent properties and the basic design of this single-layer network are described in Aleksander (1983) and Aleksander, Thomas and Bowden (1984). WISARD, unlike other neural networks, was designed specifically for image pattern recognition tasks and therefore needs very little adaptation to perform the tasks required for building image recognition. Aleksander (1983) describes this network as consisting of artificial neurons which are defined as bit-organised memory. The memory address is calculated from a constant pseudo-random mapping of groups of pixels from the input image. The use of random mapping of pixels

is important as this gives the net an ability to discriminate between images that may have some areas with a similar pattern. The net is trained by storing a one in memory locations pointed by a random mapping calculation based on a binary image. The net is run by using the same random mapping on the test image. A measure of the similarity the of images is calculated by simply adding the number of ones or zeros found at the calculated addresses. A schematic of the WISARD net is shown in Figure 2.31. A detailed description of WISARD is in Chapter 4 where a modified version of the system, for application to building recognition, is described.

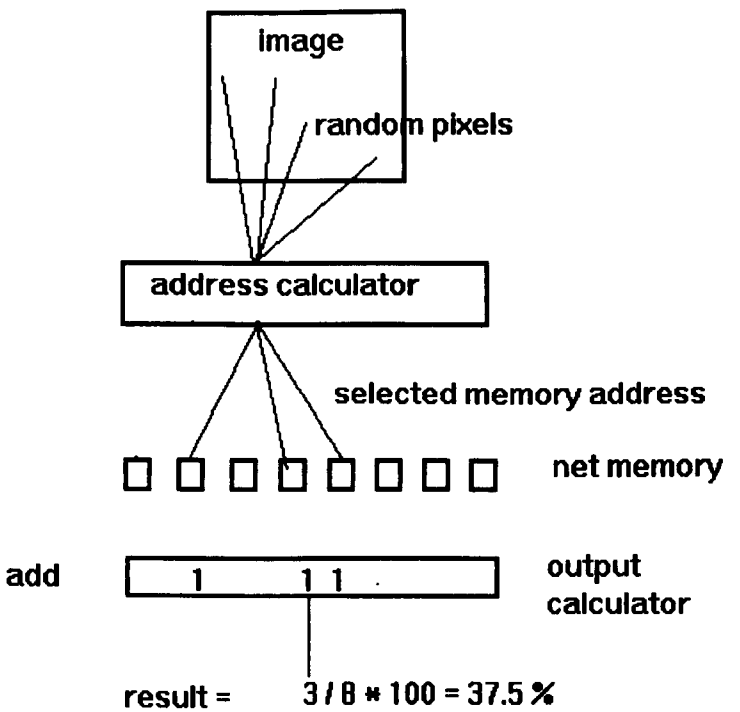

Figure 2.31 Schematic diagram of WISARD.

The design of this single-layer net is ideally suited to direct implementation in hardware and the recognition process is carried-out in a single pass, unlike other networks. These two aspects of the design mean that object recognition can be very fast, of the order of just one memory read/write cycle plus the overhead for adding the result, a total of only a few milliseconds. This makes WISARD ideal for any real-time recognition tasks such as those suggested in Aleksander (1984) which include parts inspection, security and robot vision. One example that is quoted is the discrimination between 4mm and 5mm bolts by training to sets of net memory, one on each size of bolt. Because the net is used in many applications to discriminate between classes of image, the net

memory is referred to as a discriminator. Several discriminators are often used together each trained on a different class. An unknown image is presented to each discriminator in turn and the outputs compared. The difference between the outputs is often a better method of deciding which class the image belongs to rather than just the value of one output. This is defined in Aleksander (1983) as the confidence with which the decision is made:

$$\text{Confidence} = \frac{r_1 - r_2}{r_1}$$

where r_1 is the highest response and r_2 is the next highest response. The emergent intelligent properties of various WISARD system structures are outlined in Aleksander (1983) and are worth repeating here in order to identify which structures might be relevant to building recognition. The net can accept a great degree of diversity. It can be trained on several images which may be quite diverse, such as different rotations or positions of an object within the image. The design of the net makes it sensitive to small differences in the input image; one example was the ability to discriminate between smiling and solemn faces.

The basic network design is modified by Aleksander in several stages which he refers to as level one, two and three structures. Starting with level one structures is modification to improve the degree of confidence in

discrimination between images. This modification involves the use of feedback from the output or responses of the trained discriminators to the input, see Figure 2.32. The net now becomes more sensitive to small differences between the trained and the test image. The improvements quoted by Aleksander are from a confidence of 30%, without feedback, to one of 90% with feedback. The net in this experiment had been trained to discriminate between an image that had a small speck on the left or on the right.

Level two structures produce results similar to associative memory networks. Associative memory networks are discussed later in this chapter. Here instead of producing an output response level to each discriminator, the output is in the form of an image associated with the trained image. If in this case the test image is similar to several of the trained images then the output becomes a mixture of those images. Feedback can be combined with association to produce an ability to decide more accurately between the possible images.

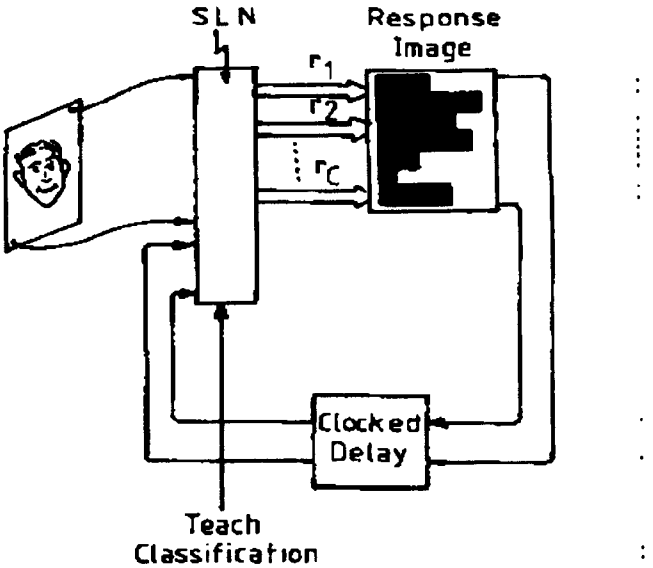

Figure 2.32 WISARD Level 1 Structure.

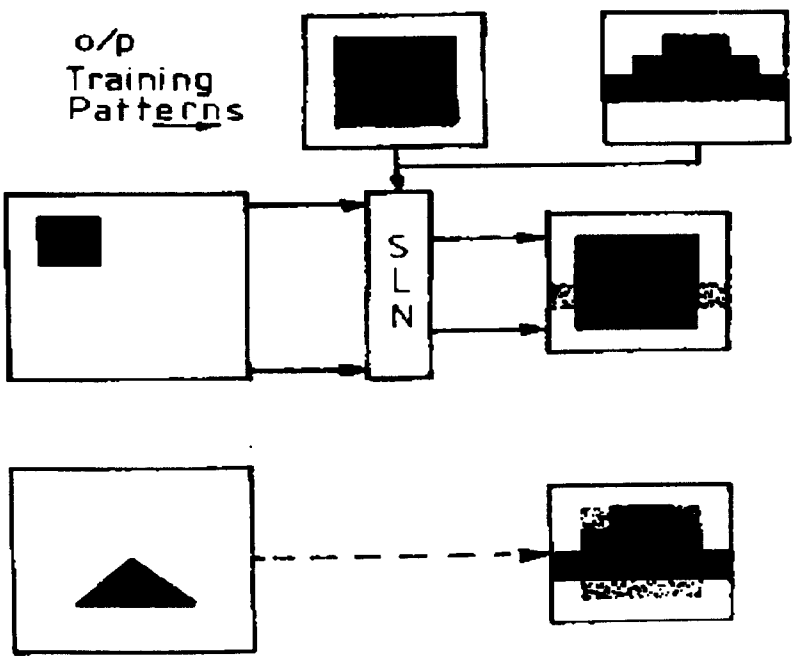

Figure 2.33 WISARD Level 2 Structure.

Level three structures introduce the concept of a high resolution window within the input image, the two images being used to calculate addressing into the network (Figure 2.34). The window is moved around the image manually or otherwise as part of the training of the net. This type of structure is shown by Aleksander to have the property of detecting small objects within a larger image. It can be taught to move around the image following a previously trained path and can recall the sequence of this path. This level three structure shows many promising possibilities for processing aerial images if a stereo pair of images are scanned independently by two windows until a similar feature is detected. The difference between the window positions could possibly be use to provide parallax data and ultimately height data on the images. However, this process was considered to be complex and other development of WISARD were considered to be easy to implement.

Aleksander and Wilson (1985a,b) discuss WISARD single-layer networks, called adaptive windows, that are trained on small patterns, typically 16 by 16 pixels in size, and then tested on a larger unknown image by scanning in pixel size steps. This technique is very similar to many other image processing such as Sobel operators which is compared with WISARD. These adaptive windows are not related to the level 3 structures discussed above. These windows are simple in implementation and can be adapted to the task of building edge finding. However, one disadvantage is the increasing complexity of the method with other than binary (two grey-level) images. For

grey-levels greater than for two the multi-layer networks discussed further on in chapter 2 seem to be more suitable. Two specific capabilities of adaptive windows are discussed by Aleksander and Wilson, namely, edge detection and stereopsis; both of these are of interest to this research.

Edge detection is achieved by training a net with a window that has all pixels except a square in the centre set to zero. This is then tested on an image by scanning in pixel-wide steps. The net produces a high response as the window crosses an edge and the response levels are used to produce the output edge image. The arrangement is shown in Figure 2.35 along with an example response curve.

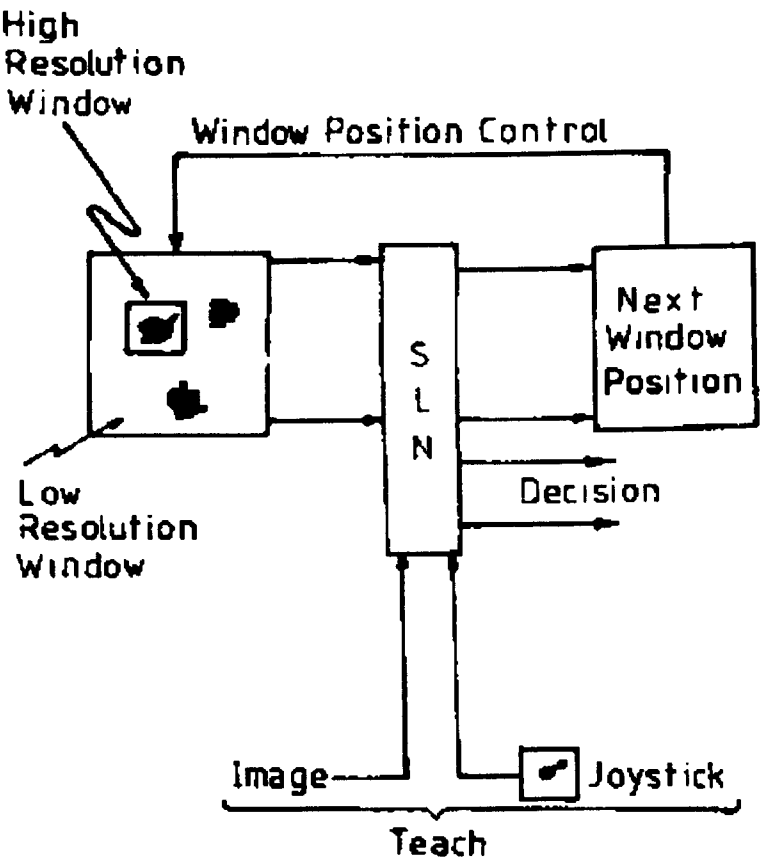

Figure 2.34 WISARD Level 3 Structure.

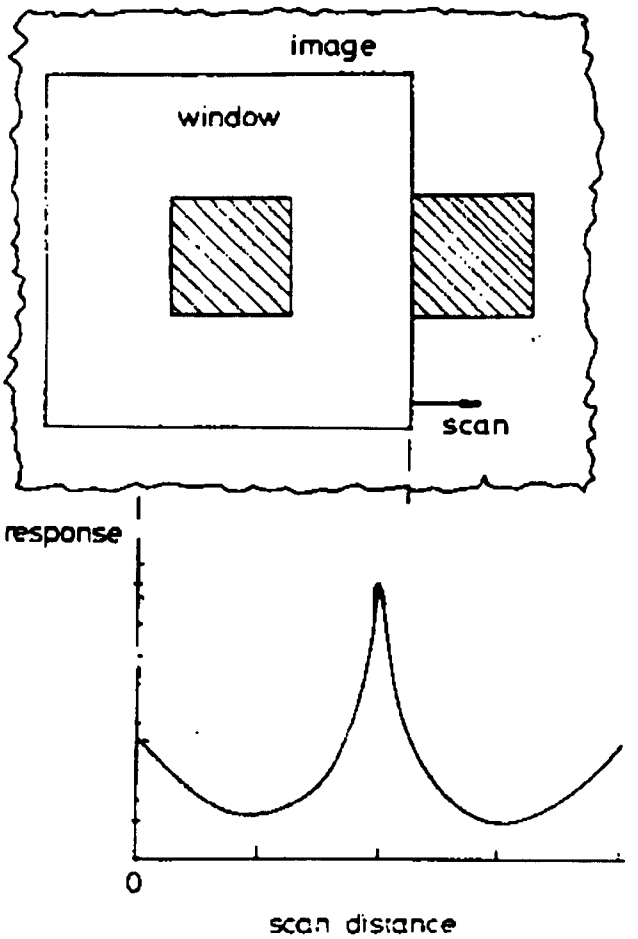

Figure 2.35 Edge detection using adaptive windows.

Interestingly this type of response curve is very similar to that expected of human visual system when considering centre-on cells in the eye, as suggested in Marr and Poggio (1979).

The ability to extract image disparity and hence height differences in an image has always been very important in photogrammetry. The adaptive window technique can be modified to measure image disparity in a stereo pair. The disparity in Aleksander and Wilson (1985a) is assumed to be along the scan

lines and therefore the window need only be one pixel high. This would also assume perfect epipolar alignment of the two images. In a practical implementation of this method it would be necessary for the window to be several pixels high to allow for poor alignment between the images. The net is trained on a set image disparity and when tested on an unknown image will produce a high response when the window is scanned across areas of similar disparity. The most interesting aspect of this use of adaptive windows is that the tests were carried-out by Aleksander on random dot stereograms, discovered by Julesz (1960) and made part of every day life by 'Magic Eye' pictures. Conventional stereo correlation methods are unable to process random dot stereograms.

2.9.2 Multi-layer Networks

Multi-layer networks, unlike single layer networks, can identify classes that are not linearly separable. The network must have n dimensions in order to separate overlapping classes in $n-1$ dimensions. A much quoted example problem solved by a multi-layer network is the XOR problem (Table 2.1). The XOR problem can be solved using a multi-layer neural network with the topology and synaptic weights shown in Figure 2.36 and the following activation function:

$$f(x) = \begin{cases} 1 & x > 0 \\ 0 & x <= 0 \end{cases}$$

$$x = \sum_{i=0}^{1} \mathbf{X}_i \mathbf{W}_i$$

where **X** is the neuron vector and **W** is the weight vector.

Input pattern	Output pattern
00	0
01	1
10	1
11	0

Table 2.1 Input and output vectors for the XOR problem.

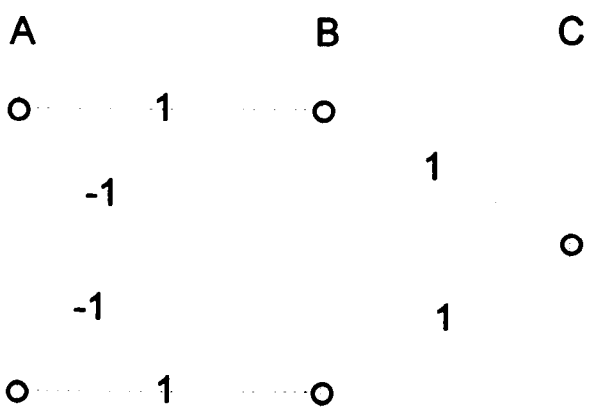

Figure 2.36 Network topology for the XOR problem.

The fundamental building block of any network is the topology of the synapses (weights) and the associated neurons. Figure 2.37 shows a typical layout of a simple two layer network.

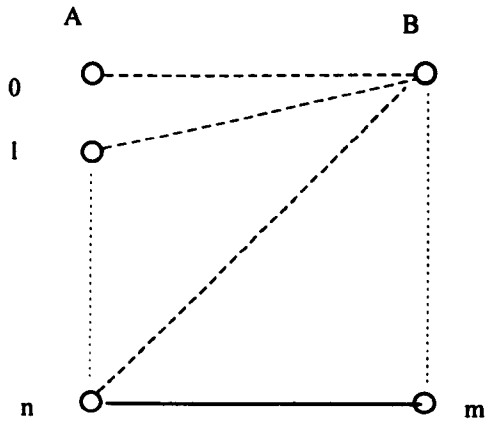

Figure 2.37 Topology of a neural network.

The value assigned to the next layer of neurons in a network are produced by an activation function (Figure 2.38). The patterns learnt by the network are effectively stored in the weights between the neuron layers.

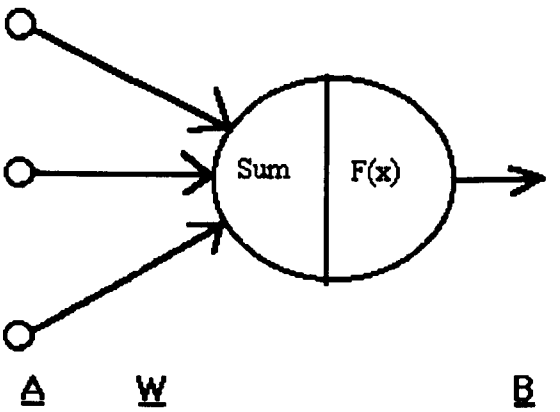

Figure 2.38 Activation function.

The activation function can be of any form dependent on the task that the neural network is to carry out. However, in multi-layer networks the function must be non-linear otherwise the network will display the properties of a single layer network and will be unable to separate classes that are only linearly separable in two dimensions. There are three non-linear threshold functions that are widely used as activation functions:

Step function:

$$f(x > 1) = 1$$
$$f(x = 0) = \text{null}$$
$$f(x < 1) = -1$$

Sigmoid (hyperbolic) function:

$$f(x) = \tanh(x)$$

Sigmoid (logistic) function:

$$f(x) = \frac{1}{1+e^{-x}}$$

where x is the product of the synaptic weight and the neuron activation:

$$x = \sum_{i=0}^{n} \mathbf{A}_i \mathbf{W}_i$$

and n is the size of the vector of neurons **A** and **W** is the vector of synaptic weights.

2.9.2.1 Learning

Learning can be of two kinds: supervised and unsupervised. A supervised learning method is designed to teach a network to associate a given input vector with a given output vector. Unsupervised methods allow the network to derive the output vector without the user's assistance. There are two steps in the learning process: weight adjustment between layers and a learning method that controls the weight adjustment so that the input and output vectors are matched. The weight (synapse) values directly control the value (activation) of

the next layer in the network. For a network to learn, the value of the weights must be adjusted so that a given value in one layer produces the required value in the next layer.

2.9.2.1.1 Supervised Learning

In a supervised network there must be a method of weight adjustment that produces the required output pattern from the given input pattern. Hebb (1949) devised 'Correlation Learning':

$$\Delta \mathbf{w}_{ij} = \alpha \mathbf{a}_{ij} \mathbf{b}_{ij}$$

where α is the learning rate **a** and **b** are the activation values in layer **a** and **b**. He adapted this to produce the 'Sigmoid Hebbian Law':

$$\Delta \mathbf{w}_{ij} = -\mathbf{w}_{ij} + s(\mathbf{a}_i) s(\mathbf{b}_j)$$

where **S(x)** is a Sigmoid function, of the form f(x)=tanh(x).

Supervised learning methods that are mainly based on error correction techniques. Measurement of the error between the required and the calculated output is used to assess the required change in the weights between each layer of the network. The basic error correction method is defined by:

$$\Delta \mathbf{W}_{IJ} = \alpha \mathbf{a}_i \left[\mathbf{c}_j - \mathbf{b}_j \right]$$

where α is the learning rate, **a** is the neuron activation, **b** is the recalled neuron activation and **c** is the required activation. Another method is reinforcement learning, where there is only one error value for each output neuron:

$$\Delta \mathbf{w}_{ij} = \alpha (v - \theta) e_{ij}$$

where V is the total error of the output layer and θ is a output neuron threshold. The degree to which the weight is changed is dependant on the value of e:

$$e = \frac{d \ln g_i}{d \mathbf{w}_{ij}}$$

The value of g varies dependant on the network model and is a measure of probability of the correct output value.

2.9.2.1.2 Unsupervised learning

In unsupervised learning systems the network is self-organising and derives its own set of pattern associations. Unsupervised learning is often used prior to supervised learning in order to give an insight into the scale and nature of intrinsic patterns within the input data set. Two specific methods of

unsupervised learning include additive matrix learning and vector quantizer learning. Additive matrix learning is a simple form of Hebbian learning and is often used in BAMs. The vector quantizer is very similar to nearest neighbour classification in this method one output neuron is chosen and only this neurons weights are adjusted.

2.9.3 Backpropagation Networks

The classic model of a backpropagation network was developed by Rumelhart *et al.* (1986). This type of network has a supervised training algorithm and requires pattern associations between input and output to carry out the training. The relationships between the input and output training examples can be complex and non-linear in nature. Unlike other methods of problem solving, such as expert systems, these networks do not require a rule for every possible combination of input and output. Given sufficient training examples the network will develop the required relationships. The supervised training can be used to advantage in problems such as Stock Market prediction as tolerances on the required results can be established during the training.

In a typical application of backpropagation the network consists of three layers: input, hidden and output (Figure 2.39). Four layers provides a network with arbitrary decision regions and is therefore the maximum size if they are

complex. Each neuron in the network receives input which is summed. The result is used in a threshold function to produce an output value (Figure 2.40).

To train a backpropagation network a cycle of calculations, which consist of a forward phase and then a backpropagation phase, are used. The forward phase, using the input values, calculates the hidden and output layer values. The backpropagation phase calculates the error between the required and the calculated values of the output and hidden layers. These error values are then used to change the synaptic weights. The cycle continues until the error values reach the required tolerance. Following training the network can be run by providing a unknown input values by repeating the same calculations as the forward training phase. The output layer values are then the recalled pattern or values the network has associated with the new input. The input layer and output layer size are determined by the size of the input and output pattern.

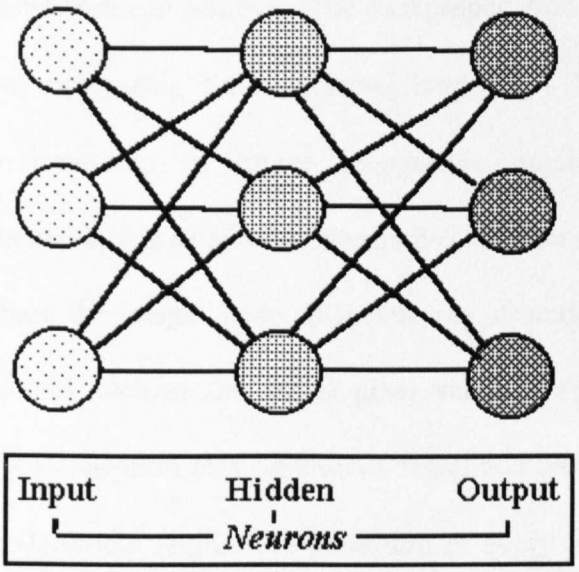

Figure 2.39 Backpropagation network topology

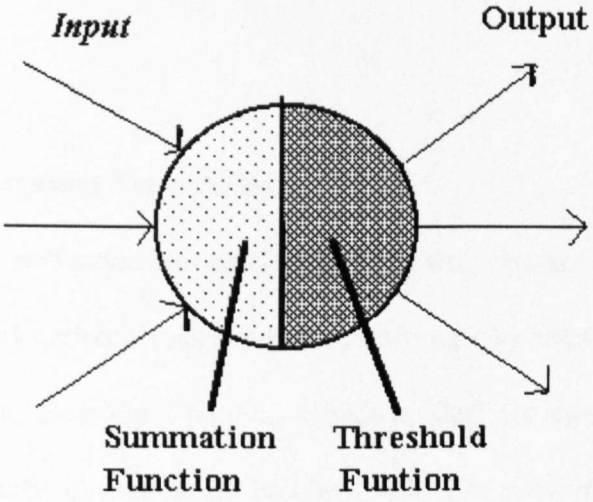

Figure 2.40 Neuron functions.

The hidden layer can be any size but usually is somewhere between the input and output size. The size of the hidden layer will control the degree of

generalisation the network can achieve. The backpropagation process can take a large amount of processing time on serial hardware. Training time is generally not a limitation in image recognition application therefore backpropagation networks are reasonable alternatives to other methods. This is especially true where the images have an underlying structural similarity but variations in attributes such as the actual pixel values. The generalisation capabilities of backpropagation may be able to detect this underlying structure where other methods would require the definition of every possible variation that may be encountered in the images. An example would be an image containing buildings which have the same basic grey levels as the surrounding areas, here the network is required to learn the difference between the hard building edges and the soft contours of the surroundings.

2.9.4 Backpropagation Network Design

Backpropagation networks are applicable to a wide range of classification tasks. This type of network uses supervised training (the network is given the output response to each input pattern) which is ideal for classification of an image where specific classes are to be identified. The network is first trained on known pattern pairs and then run by presenting only the input half of the pattern, the network then producing the output class based on its training.

The network consists of three layers of neurons represented as vectors input (**i**), hidden (**h**) and output (**o**). The hidden layer size is usually between the input and output layer size and should be at least as large as the number of pattern groups that might be formed. Between the layers are related synaptic weights which are stored in two matrices (**w1, w2**). To control the performance of the network there are several constraints imposed consisting of learning rate (α), momentum (θ) and tolerance factors. The learning rate factor controls the amount of change imposed on the synaptic weights. The momentum factor controls the effect previous weight changes have on current weight changes. The tolerance factor puts bounds on the error acceptable in the final output vector.

Backpropagation networks are able to classify complex non-linear relationships, as they are multi-layer in nature. This ability is due to the hidden layer which is often thought of as the generalising layer. This type of network topology is able to store a greater number of pattern associations than the size of the network, which is not the case for all neural network models. However, there is a penalty to pay for these advantages, as a backpropagation often has very long training times and it requires strong pattern associations to operate efficiently. The time factor can be reduced by widening the tolerance of the output error or implementing the algorithms in a parallel hardware architecture.

2.9.4.1 Training

The training process is iterative and consists of a forward calculation to obtain the hidden and output neuron vector activation followed by a backward calculation to obtain new weight matrix values. The weights are initialised by assigning random values. The weight matrix value changes can be made after each pattern pair or after all pattern pairs are presented. The hidden layer neuron activation is described by:

$$\mathbf{h} = \mathrm{f}(\mathbf{iw1})$$

where **h** is the hidden layer neuron vector, **i** is the input layer neuron vector and **w1** is the weight matrix between the two layers. The output layer neuron activation **O** is given by:

$$\mathbf{o} = \mathrm{f}(\mathbf{hw2})$$

where **o** is the output layer neuron vector, **w2** is the weight matrix between the hidden and output layer and **f()** is a Sigmoid function such as the logistic function:

$$\mathrm{f}(x) = \frac{1}{1+e^{-x}}$$

The backward calculation, to obtain the new weight values, starts with calculating the error between the actual output neuron vector and the required output neuron vector:

$$\mathbf{d} = \mathbf{o}(1-\mathbf{o})(\mathbf{o}-\mathbf{t})$$

where **d** is the error vector for each output neuron, **o** is the output layer vector and **t** is the required output layer vector. The hidden layer error is calculated from:

$$\mathbf{e} = \mathbf{h}(1-\mathbf{h})\mathbf{dw2}$$

where e is the error vector for each hidden layer neuron. The weights in the layer between the output and hidden layer are calculated from:

$$\mathbf{w2} = \mathbf{w2} + \Delta\mathbf{w2}_t$$

where $\Delta\mathbf{w2}$ is the change in matrix **w2**, which is calculated from the expression:

$$\Delta\mathbf{w2}_t = \alpha\ \mathbf{hd} + \theta\ \Delta\mathbf{w2}_{t-1}$$

where α is the learning rate and θ is the momentum term. The momentum controls the degree to which previous weight changes effect future weight

changes. The weights in the layer between the hidden and input layer are given by:

$$\mathbf{w1} = \mathbf{w1} + \Delta\mathbf{w1}_t$$

where $\Delta\mathbf{w1}$ similar to $\Delta\mathbf{w2}$ is calculated

$$\Delta\mathbf{w1}_t = \alpha \ \mathbf{ie} + \theta \ \Delta\mathbf{w1}_{t-1}$$

The forward and backward calculation sequence is repeated until the output neuron vector error (**d**) is achieved. Once the sequence is complete the network is deemed to be trained and is ready to be used to classify unknown data.

2.9.4.2 Running

The network is run in a single pass by presenting an unknown input neuron vector **i** and calculating the output neuron vector **o**:

$$\mathbf{h} = \mathrm{f}(\mathbf{w1i})$$
$$\mathbf{o} = \mathrm{f}(\mathbf{w2h})$$

2.9.5 Counterpropagation Networks

Counterpropagation networks (CPN) are a variation on a backpropagation network that were developed by Hecht-Nielsen (1987). The reason for covering them in more detail here is because the CPN method requires a reduced training time. Although training times are not a critical factor in the selection of a network architecture in a research system they may become more important in a PC based system. The CPN is based on the Outstar, Grossberg (1982) and the linear vector quantizer, Kohonen (1984) networks.

A CPN consists of three layers: input, hidden and output. To train the network a vector distance function is used to decide which weights in the hidden layer are adjusted. The choice of weights to be adjusted is based on the difference between the selected weight vector and the input vector. The weights between the hidden and the output layer are then based on the hidden layer and each output layer activation level to recognise an unknown input the strongest activation in the hidden layer is used to select the output pattern. A CPN is therefore similar to a linear vector quantizer followed by a method similar to that used in the Outstar network.

A CPN can be trained in a supervised or unsupervised fashion which is an advantage over the simple backpropagation network. However, the number of patterns that can be learnt is equal to the number of hidden layer neurons. This could be a disadvantage if there is a limitation on computer memory or if the

number of training patterns is large. Kosko (1988) may have provided a solution to this capacity problem in his development of Bi-directional Associative Memory (BAM) networks where a new matrix is created if the original one becomes saturated.

2.9.6 Bi-directional Associative Memory (BAM)

BAMs are two-layer networks which store and recall associations between the two layers. The two layers need not be of the same size but the number of pattern associations is limited to the number of neurons in the smaller layer. The network can be thought of as a flexible surface which is shaped by the pattern associations, each association being related to a particular low point in the surface. Associations are recalled in a similar way in which a ball would roll to the nearest low point on the surface, and the association related to this point provides the recalled pattern. The network is implemented as a simple matrix or set of matrices.

To train the network the input (**A**) and output (**B**) patterns form a vector pair ($\mathbf{A}_n$ $\mathbf{B}_p$), where n and p are size of the respective vectors. The association is stored in the network by adding the vector pair correlation matrix to the network matrix $\mathbf{M}_m$:

$$\mathbf{M} = \sum_{i=1}^{m} \mathbf{A}_i^{\mathrm{T}} \mathbf{B}_i$$

$$m < \min(n, p)$$

BAMs function best if the patterns are binary (0s and 1s) However, the matrix operations are better suited to -1 and +1, hence the first part of the process must map 0s to -1 and 1s to +1. The limit of pattern associations being equal to m is resolved to a degree by using more than one matrix. The training process must therefore include a section which tests the network for the correct recall of all associations. If an association can not be recalled correctly then a new matrix is created and the association is stored there. In the recall process when vector $\mathbf{A}_i$ is used as input the vector $\mathbf{B}_i$ is output and because the network is bi-directional the reverse is also true. BAMs also have a degree of generalisation in that the output will be $\mathbf{B}_i$ when the input vector is similar to $\mathbf{A}_i$.

The limitations of BAM systems do restrict the applications but this has not held back the number of areas of pattern recognition in which they are used, e.g. voice recognition, radar signature recognition and character recognition. Although BAMs usually use binary data it is possible to extend them to image data, especially 256 grey level images, by encoding the greylevel as 8 bits.

2.10 Conclusions

To classify urban images a neural network must be able to cope with buildings in a typical image such as Figure 2.41. There are many difficulties within the image for any classification system, such as shadow, pitched roofs and strong edge features. The requirements can be narrowed down to five basic properties: it must use supervised learning, have a capacity sufficient for the number of classes, have reasonable speed in calculation, interpolate between classes and be resistant to noise. As specific classes are required the model must use supervised learning. The capacity for classes need not be great as only four main classes are required and possibly one or two sub-classes within each class. A typical sub-class might be a feature in and out of shadow. Interpolation is essential as the classes are not uniquely defined. Fortunately, interpolation is an inherent capability in most neural network models. Noise can present a problem to some recognition systems such as edge detection to identify building boundaries. Some neural network models actually require noise to be present in order to function efficiently.

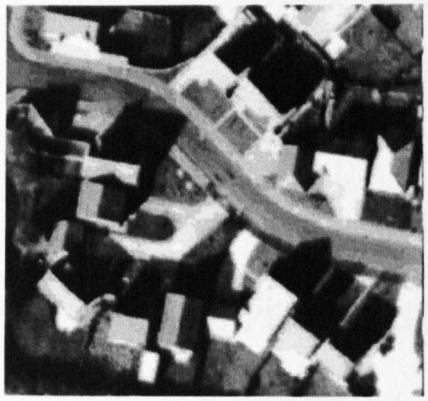

Figure 2.41 Typical urban image with a mixture of building types produced from an Ordnance Survey photograph.

As Paola and Schowengerdt (1995) show backpropagation networks have been extensively and generally successfully used for image classification. Thus, taking into account the features required of a neural network model defined earlier and the research currently proceeding in the field of image classification, the backpropagation model was considered to be the most suitable for the task defined in the aims of this research.

CHAPTER 3

3. CONVENTIONAL IMAGE PROCESSING APPROACH

3.1 Introduction

At the start of this research project the computer processing power required to simulate a neural network was well beyond the capacity of any desktop computer. Although, having studied Aleksander's *et al.* (1984) work in the use of neural networks in pattern recognition and realised that there was a potential use for these techniques in image classification a more conventional approach was the only avenue open. As PCs gained in power to the present generation of Pentium P6 processors the use of neural networks became feasible. This chapter presents the early work produced in those 'pre-Pentium' days. The methods proposed here are based on the processing of the image to identify line features from which the building outlines could be extracted. The techniques include the use of line angle and edge gradient strength.

3.2 Parallel Lines

Parallel lines are abundant in urban images as buildings tend to be constructed of rectangles. Developing a method to identify parallel lines could therefore

provide a means of classifying an image. Parallel lines by definition have the same angle values but their edge gradient strengths may be markedly different. This means that although an image may have many parallel lines it may be difficult to devise an algorithm that can recognise them. Classifying all straight lines in the image, of whatever edge gradient strength will initially simplify the problem, as it is straight parallel lines that may identify the boundaries of a building. Classifying straight line segments can be achieved by using a suitably defined masks convolved across an image that has already been processed to contain only edge gradient strength values. The output values from this process are similar for similar angles, hence parallel lines can be classified.

3.3 Orthogonal Lines and Corners

A line image that has lines of similar angle identified by grey level can be used to find lines that are orthogonal. Orthogonal lines can be defined as lines that are approximately 90 degrees to each other. Orthogonal lines can be used to identify corners of buildings. Corners can be defined as the point at which two orthogonal lines intersect or nearly intersect. The method required to find the corners would scan a parallel line image for points where the difference in the angle (grey level in this case) values of neighbouring pixels are approximately

+/-90°. During this process the original lines could be removed as corners are found and replaced with single pixel wide computer draw lines.

3.4 Vectors

If a vector image of the original raster image can be created then further processing can be carried out using proprietary CAD programs. The increase in speed is due to the process not having to scan every point in the image (usually in the region of 250,000 pixels) but only read a text file containing definitions of the lines, a line being defined by its two end positions. Vector information can also be used in conjunction with GIS software. There are commercial programs that convert between raster and vector graphic file formats. They are usually built-in to the more complex CAD packages. Another advantage of the vector format is the ability to use a plotter to draw accurate straight lines rather than the sometimes jagged lines produced by raster systems.

3.5 Rectangles

Once the vector image is defined then the image can be processed to find possible rectangles. This is making the assumption that a building will be one or more rectangles; that is obviously not always the case but provides a good starting point. If a rectangle is found then the line definitions are removed and

a best fit rectangle is defined. It is assumed that the original lines would not be exactly at 90 degrees to each other and would not necessarily intersect. The result of this process is an image consisting entirely of rectangles defined by the four corner positions. The rectangle image can be refined further in an attempt to identify the buildings by removing isolated rectangles below a threshold size and merging rectangles below the threshold but near enough to another rectangle to be possibly part of the same building.

3.6 Line Extraction (1)

The technique used here to extract lines from the images uses a variation on the Sobel edge detection process. First the horizontal and vertical first differences for the image are calculated then these values are then used to calculate the strength and angle of these differences in the image.

$$S = \sqrt{(V^2 + H^2)}$$
$$A = ATAN\left(\frac{V}{H}\right)$$

where S is the edge strength, A the edge angle, V the vertical difference and H the horizontal difference.

The strength of these differences can be used to detect edges, an edge being assumed to be present if the strength is above a certain threshold. If a suitable

threshold could be chosen then all possible edges within the image would be detected. Since only those edges that may form part of a line segment are required, in order to identify the outline of a building, The assumption is made that a line is constructed from edges of similar strength and angle.

An ideal straight line would have the same greylevel along its length and the angle formed by any set of pixels would be constant. Hence, Figure 3.1 would show an ideal straight line as a single point on the graph, however in real line would be shown as a small cluster of points due to angle and greylevel variations. Unfortunately points on parallel lines will all fall in the same cluster and therefore some method of separating them will be required. The method here compares only neighbouring pixels for similar angle and strength. This method of line extraction does not rely on the need to select a threshold value and is resistant to variations of brightness and contrast between images.

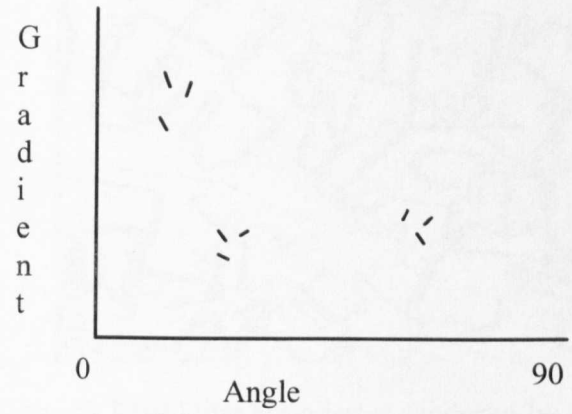

Figure 3.1 Graph of angle versus gradient of line segments within an image.

Figure 3.2 Original Image.

Figure 3.3 Building boundaries produced by method 1.

Figure 3.3 shows the result of applying this method to the image in Figure 3.2. This method has two main limitations: the building boundary lines become thickened and there is a high level of noise. The line thickening could be reduced by using a different shape of mask within which the gradients are measured. Unfortunately this would also make measuring the angle calculation difficult or impossible. The noise could be reduced by various methods of pre and post processing. Method 2 modifies the mask shape and pre-processes the image.

3.7 Line Extraction (2)

To prevent the gradient measurement creating an image with thick lines (2 to 5 pixels) the gradient is measured as the sum of the difference between the adjacent pixels above and to the left of the current pixel. This is successful in only producing lines that are a maximum of two pixels wide.

In order to overcome the noise problem identified in method 1, the image is pre-processed by density slicing. The density slicing is specifically designed to increase the edge gradient of edges in the darkest areas (shadows) of the image, as shown in Figure 3.4. This is achieved by setting dark pixels to high values so in effect they become the brightest part of the image. Because the areas immediately adjacent to the shadow are down-sun, they will be in the darker end of the range and the inversion of the shadow will produce a high gradient value. Since it is mainly the buildings causing the shadows, their down-sun edges can be identified.

Figure 3.4 Shadow region inversion.

The building in the bottom right of Figure 3.4 has the down-sun edge lost in shadow in the left image but it is clearly evident in the right image. This was achieved by the method above. The loss of this edge is due to the pitched roof ridge casting a shadow and the wall of the building casting a shadow in the same direction. Stretching the lower end of the spectrum will give the line extraction technique a chance to identify the lines in shadow and separately stretching the high end of the spectrum will assist in line detection on the building edges which are up sun. Finally, the two line images can be added together. Method 2 produces less noise, here demonstrated by comparing Figure 3.3 with Figure 3.5.

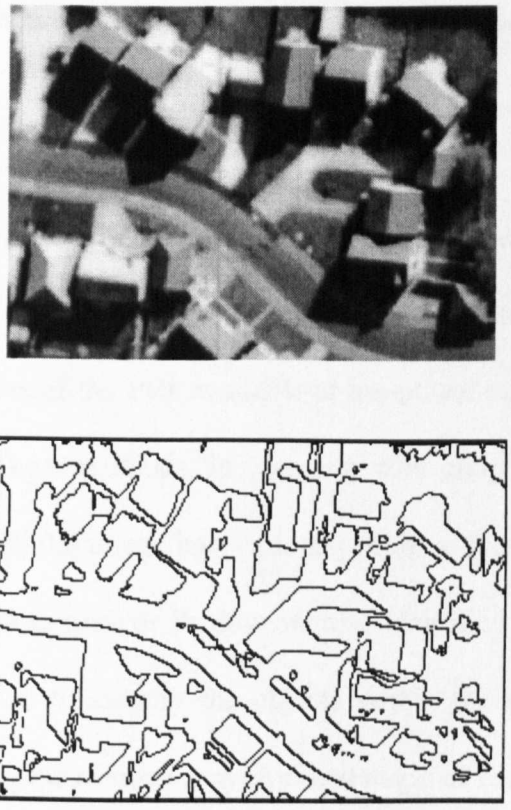

Figure 3.5 Building boundaries produced by method 2.

3.8 Conclusions

The above line and shadow processes do not solve the primary aim of identifying buildings. Some of the building boundaries are successfully identified but so too unfortunately are the lines which belong to the road or path. In addition, some shadows are cast by buildings but some are also cast by other objects such as trees. These are not new problems, as discovered by others including Huertas and Nevatia (1988) and Irvin and McKeown (1989). The image can be processed to a point where other external rules are needed or

user interaction is required to proceed to the next stage where buildings can be identified.

This chapter presents the results of work which attempted to approach urban image classification using conventional image processing techniques that were within the capabilities of the PCs available at the outset of this research (Intel 8086 at 16Mhz). These methods, in common with early attempts at pattern recognition, processed the image into various primitive forms: edges, lines and areas. Attributes such as lines or shadow are also identified so that higher level reasoning can be used to classify the objects within the image. Images are processed to calculate the average greylevel and angle of the strongest gradient, for a fixed window size around each pixel, and use these data to determine whether the pixel is part of a line segment. Line segments are considered to have the same average greylevel and angle within a defined range. Building shadow is also used to filter out non-building lines and a density slicing is used to reveal building detail that would normally be hidden by shadow. These methods could be developed to include AI such as a rule-based filtering process, McKeown *et al.* (1985), to classify the lines. Rule-based methods lack the ability of neural networks to learn by example and could become overwhelmed by the complexity of urban images. Neural networks are employed for land cover classification by Benediktsson, Swain and Ersoy (1990), Bischof, Schneider and Pinz (1992), Civco (1991) and Dreyer (1993). Neural network's non-parametric and fuzzy logic present a possible new

methodology. Therefore, some new methods are developed in the next chapter using neural networks for image classification.

CHAPTER 4

4. ARTIFICIAL NEURAL NETWORK APPROACH

4.1 Introduction

The classification of urban images has in the past relied on the strong presence of edges, lines and shadows in an image. Most methods make use of this fact and use a strategy that extracted the edge, line or shadow information and then processed the resultant image to identify the building boundaries. Some methods rely entirely on image processing while others use various forms of Artificial Intelligence (AI). Rule based methods lack flexibility and can become overwhelmed trying to cover the variations possible in an image. This experience pointed firmly towards the need for a method that could interpolate from given knowledge. Neural networks, with their non-parametric nature and interpolative qualities, present a possible new methodology. The proposal is that a combination of image processing and artificial neural network approaches can be applied to the classification of urban images.

The patterns formed by objects perceived by the human visual system are stored within our neural network. This network has the ability to interpolate and does not seem to be rule based. Artificial neural networks have been

shown to be useful in other areas of research, such as speech and character recognition, due to their ability to learn patterns and find solutions by interpolation when presented with unknown similar patterns. Artificial neural networks have already been used for land cover classification by many researchers, including Benediktsson, Swain and Ersoy (1990), Bischof, Schneider and Pinz (1992), Civco (1991, 1993), Dreyer (1993).

This chapter defines a new technique that uses existing image processing methods to transform a 256 greylevel image into a form suitable for processing by a backpropagation network. Neural networks establish relationships between patterns of data. For the network algorithms to function successfully there must be a strong relationship between the input and output data. In this method the input pattern consists of the pixel values in the original greylevel image and the output pattern contains the required classes of object. It was thought that single pixel greylevels would not provide a suitable input data as there would not be a relationship between these values and the required output class; many classes of object would have the same pixel greylevel. However, a window of pixel values would provide stronger degree of pattern association required to achieve any meaningful classification of the as one of the main means of human recognition is the relationship of groups of pixels. This can be interpreted as the texture of areas of the image. Textural data was used successfully in urban classification by Jensen (1981). For instance a ploughed field is recognised by the distinctive repeating pattern of lines of light and dark

shown to be useful in other areas of research, such as speech and character recognition, due to their ability to learn patterns and find solutions by interpolation when presented with unknown similar patterns. Artificial neural networks have already been used for land cover classification by many researchers, including Benediktsson, Swain and Ersoy (1990), Bischof, Schneider and Pinz (1992), Civco (1991, 1993), Dreyer (1993).

This chapter defines a new technique that uses existing image processing methods to transform a 256 greylevel image into a form suitable for processing by a backpropagation network. Neural networks establish relationships between patterns of data. For the network algorithms to function successfully there must be a strong relationship between the input and output data. In this method the input pattern consists of the pixel values in the original greylevel image and the output pattern contains the required classes of object. It was thought that single pixel greylevels would not provide a suitable input data as there would not be a relationship between these values and the required output class; many classes of object would have the same pixel greylevel. However, a window of pixel values would provide stronger degree of pattern association required to achieve any meaningful classification of the as one of the main means of human recognition is the relationship of groups of pixels. This can be interpreted as the texture of areas of the image. Textural data was used successfully in urban classification by Jensen (1981). For instance a ploughed field is recognised by the distinctive repeating pattern of lines of light and dark

and a building roof ridge is recognised by the single strong edge with two distinct areas either side with little texture forming the roof. There are many more examples and in the case of each object or land cover type in the image this relationship of textures exists. It was therefore decided that the solution to providing the neural network with a suitable input pattern for each class of object lay in the use of texture, edge gradient and other image attributes such as mean greylevel. Human experience of pattern recognition provides a guide to possible neural network methods.

As an introduction to the method the software and hardware tools used in the research are described, followed by two variations on the basic method. The first uses groups (windows) of pixels following Paola and Schowengerdt (1994), Kamata and Kawaguchi (1993) and Ritter and Hepner (1990). They were using multi-spectral data and found that the windows of pixel introduced texture as a factor in the classification process which improved the performance of their methods. This method becomes limiting when large windows of pixels are used, as a 9 by 9 window (for example) requires 81 input neurons in the neural network. One of the factors that controls the training time of a backpropagation network is size, the larger the number of neurons the longer the training time. This fact and the realisation that the method probably only made use of the texture and none of the other attributes such as edge gradient inspired the second method. The second method still uses windows of pixels but processes this window to calculate the following features: mean greylevel,

edge gradient strength and variance. Therefore, regardless of the window size, the number of input neurons is always three. The slight increase in the pre-processing time required to calculate the feature values is greatly offset by the reduction in network training time. The method uses second-order values and therefore is called second-order classification; the first method, using the direct pixel values is therefore called first-order classification. Figure 4.0 provides a guide to this Chapter.

4.2 Tools

A custom Microsoft Windows program (NETIMAGE) was developed to pre-process the images, source input files for the backpropagation network and convert the network output files into image form so that the results could be visualised. An existing backpropagation neural network program (SLUG) was chosen for the ease of use and performance on a Microsoft Windows based PC. Both Windows 3.1 and Windows 95 are compatible with the applications.

4.2.1 Operating System - Microsoft Windows

Microsoft Windows is a multitasking operating system for PCs that allows applications to share the display by using windows. The application must create at least one window to enable user interaction. Windows provides housekeeping information on the state of each window to the individual

applications. Windows applications can have access to several screen 'surfaces' unlike DOS applications. This is especially useful for image processing applications so that several images can be displayed and processed simultaneously.

The application receives input as 'input messages' sent by Windows. Windows receives all input and places it in the appropriate applications 'message queue'; the application then reads these messages from the queue and actions relevant messages, the remainder are returned to the system for other applications to action. There are standard messages that indicate certain actions taken by the user such as pressing a key to which Windows places a WM_KEYDOWN message in the queue. The queue may also have application specific messages to indicate responses to menu selections, for example.

The multitasking environment enables more than one application to run at a time. The memory becomes a shared resource managed by Windows. The memory used by an application is moved and discarded to service the requirements of each application. The hard disc is used in order to create virtual memory above and beyond the physical memory fitted to the computer. This theoretically gives applications access to a memory size equivalent to the amount of free space available on the hard disc. On early PCs memory was

very restricted; one megabyte was common whereas now 16 or 32 megabytes is common.

4.2.2 NETIMAGE - Image Processing Tool for Neural Networks

I developed NETIMAGE to provide functions which would pre-process images for input to a backpropagation neural network and translate the network results back into image form. In addition, the application has some custom and many classic image processing functions. The custom functions include line and shadow processing and the production of false colour images. The false colour images are used for network results and for a unique technique of simultaneous display of multiple image attributes.

NETIMAGE is written in 'C' as a Microsoft Windows application. Windows provides ready-made graphics functions and a common interface to other Windows applications. NETIMAGE is a Multi Document Interface (MDI) application which allows several images to be displayed together. The MDI manages most of the basic operations on the image windows such as size and movement. The image windows are called child windows.

In any image processing application memory is always at a premium as even reasonable size images (512 x 512 pixels) in 256 greylevels require 262,144 bytes. The virtual memory management system of Windows enables a small

PC with only limited memory, typically 8 megabytes, to process virtually any number and size of image. Most image processing tasks, other than on very fast PCs, take several minutes and possibly hours if the image is large or the process complex. Windows multi-tasking allows long processes to be executed as background tasks. Windows 95 is especially well suited to image processing because of its more advanced multi-tasking compared with Windows 3.1.

The aim of the image processing system is to provide the support functions to enable an image to be manipulated by experimental functions. 'C' was chosen because of the mid-level design and the direct compatibility with Windows. It has the standard user interface recommended by Microsoft consisting of a main application window with a menu bar and a client area in which child windows are placed by the MDI. Image files once opened are displayed in child windows. These windows can be changed in size, moved, maximised and minimised. When minimised they are represented as an icon at the bottom of the client window. The image is initially displayed full size. If it is too large for the window, scroll bars are provided so that the complete image can be viewed. The images can also be zoomed to view the complete the image within a window.

Once an image file is open the data in this file can be processed by any routine that is passed its memory handle. The image files are in the standard Windows 256 colour Bitmap (.bmp) format. Other image file formats can be used via

external Windows image format conversion applications. The images are displayed in a device independent form which means that the same code can be used for any device, such as the screen or a printer. The pixel values are accessed via standard Windows Graphic Device Interface (GDI) functions. The use of GDI functions rather directly reading the pixel values from file makes the image processing functions compatible with any hardware configuration supported by Windows. GDI functions also provide for translation of pixel values to RGB format and the reverse, this is used by some of the result images that use RGB colour values. Greylevel images are read and written to devices in RGB format with the red, green and blue set to the original greylevel. By letting Windows do all the work with pixel values NETIMAGE could function with any colour video configuration which supports 256 colours or more.

The images can be cut and pasted via the Clipboard and used by any other Windows application that supports bitmaps. The Clipboard is one of several methods of data exchange within and between applications. The standard Windows help facilities are available from this application and help specific to the image processing. As in all Windows applications the help is context-sensitive

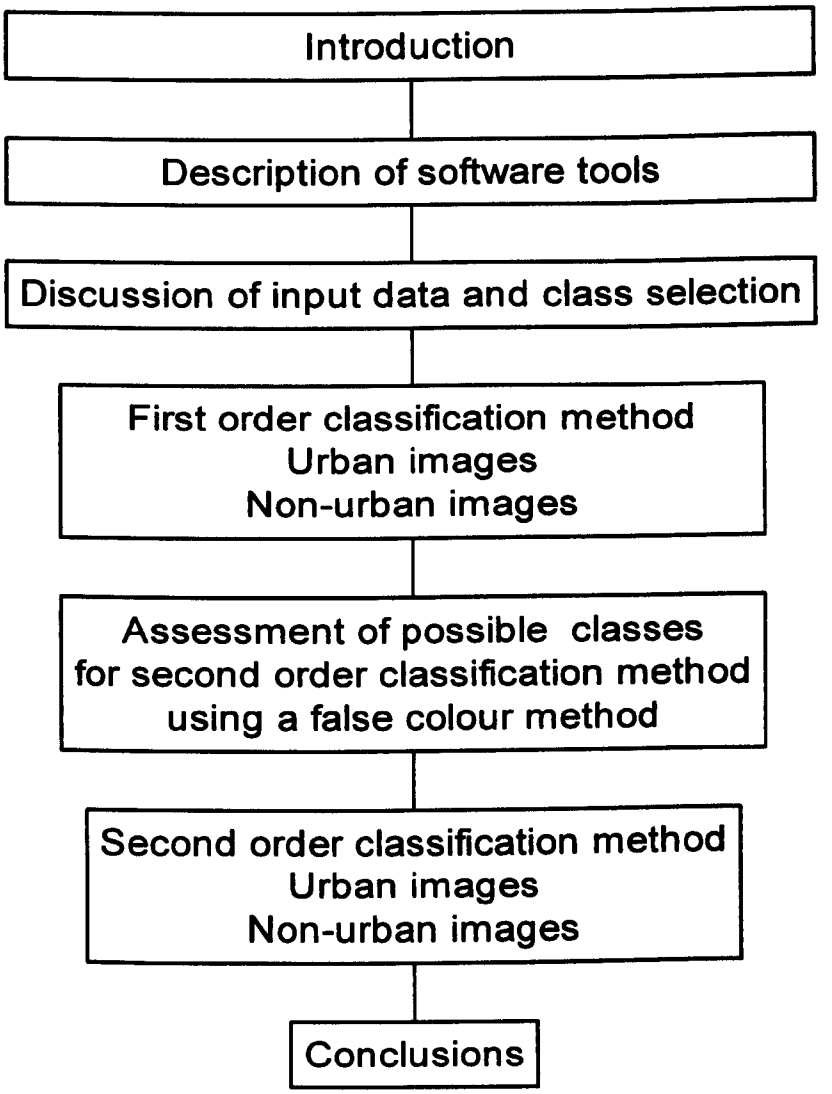

Figure 4.0. Layout of Chapter 4.

4.2.3 SLUG - Backpropagation Network Simulator

SLUG is a commercial program and the copyright of Southern Scientific. It is a backpropagation network with three layers, using a Sigmoid transfer function in the hidden layer and linear transfer functions in the output layer. SLUG uses the steepest descent optimisation method. The number of nodes in each layer, training parameters and transfer functions are user-defined.

A learning rate parameter determines the step size during the descent of the error parameter in weight space. The weights are randomised before training starts. In general, the learning rate should be reduced as the size of the network increases. A maximum error parameter is the convergence criterion:

$$\sum_0^n \alpha^n - \beta^n$$

where α is the desired output, β is the current output and n is the size of the output layer. The maximum iterations parameter specifies the maximum number of times the data set will be presented before the net. A momentum parameter is a measure of how much of a previous training step is retained in the current step and has a value between zero and one.

Training and test data are stored in text files. For SLUG text files (Figure 4.1), two header lines are followed by any number of lines with an input / desired

output vector (floating point) on each line (Table 4.1). Training and test results are stored in log files (Figure 4.1).

```
IO MATRIX
1.0000    1.0000    0.0000
0.0000    0.0000    0.0000
1.0000    0.0000    1.0000
0.0000    1.0000    1.0000

DESIRED MATRIX ---------- the last column of the IO matrix
0.0000
0.0000
1.0000
1.0000

INPUT MATRIX
1.0000    1.0000
0.0000    0.0000
1.0000    0.0000
0.0000    1.0000

Event # 320.883410 ---------- the error at intervals

Network response:
inputvec:  1.00      1.00      response :  0.015
inputvec:  0.00      0.00      response :  0.003
inputvec:  1.00      0.00      response :  1.002
inputvec:  0.00      1.00      response :  1.001

Final Weights
0.0000    0.0000    -5.2722   -1.4644   0.0000    0.0000
0.0000    0.0000    -6.7207   -1.4960   0.0000    0.0000
0.0000    0.0000    0.0000    0.0000    -3.0197   0.0000
0.0000    0.0000    0.0000    0.0000    3.1998    0.0000
0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
0.0000    0.0000    0.7118    1.4773    -0.5769   0.0000
```

Figure 4.1 SLUG backpropagation network training and test log file format.

Rows are origin of connections, columns destination, i.e. in row one, neuron 1 is connected to neurons 3 and 4, and row 5 shows that neuron 5 (the output neuron) does not provide input to anything. The last row is the offset neuron, number 6, which provides inputs to all except the two inputs and itself.

Line							Remarks
1	test						file name
	No. of vectors	Learning rate	Momentum	epoch	Max error	Max no. of iterations	training parameters
2	4	0.5	0.8	0.0	0.1	10000	
3	1	1	0				first IO vector
4	0	0	0				
5	1	0	1				
6	0	1	1				last IO vector

Table 4.1 Example SLUG training file format.

4.2.4 Image Scanning

The images used in all experiments were scanned using a Logitech Scanman 256 greylevel hand scanner. The scanner has several resolution and greylevel options. It uses a single row of charge coupled devices (CCDs). The supporting software is Windows based and has some basic image processing functions. The scanner was calibrated with a standard photographic grey card. The maximum scanning width produces images 1600 pixels wide and any height. The maximum scanning resolution is 160 pixels per centimetre. Given a photograph scale of 1:5000, this means that each pixel is equivalent to approximately 0.3 metres. A typical building in a photograph of this scale has an plan area of 180 sq. metres and therefore, each house is represented by 600 pixels. To produce an image a photograph is illuminated by a built-in light and a row of CCDs detect the reflected light from the photograph. The associated software reduces the detected light intensities to the required number of greylevels.

4.3 Test Images

To simplify the analysis of the results four images of the same area at four different resolutions were scanned and stored as bitmaps. Each image was pre-processed using histogram equalisation to remove any differences in the original spectral values of each pixel and to broaden the range of pixel values to 0..255. The details of each image are in Table 4.2. The image file size had to be less than 16 k bytes because of limitation presented by the backpropagation network software.

File name	Bits per pixel	Width	Height	Resolution	File Size
L100.bmp	8	109	108	100 dpi	11 k
L200.bmp	8	129	121	200 dpi	15 k
L300.bmp	8	130	121	300 dpi	15 k
L400.bmp	8	131	111	400 dpi	14 k

Table 4.2 Test image details.

The neural network results are shown as false colour images with each colour representing a class defined in Table 4.3. Where possible these false colours have been used consistently throughout this chapter. The training data for the neural network is taken from a sub-section of the images and the tested on the training set and then the complete image.

Class No	False Colour	Red	Green	Blue
1	RED	255	0	0
2	GREEN	0	255	0
3	BLUE	0	0	255
4		255	255	0
5	CYAN	0	128	128
6	MAGENTA	255	0	255
7	PURPLE	128	0	128
8	GREY	128	128	128
9	WHITE	255	255	255
10	BLACK	0	0	0

Table 4.3 False colours representing classes in result images.

4.4 Input Data

Image classification using neural networks functions most efficiently when there are strong associations between the input patterns and the output classes. Multi-spectral sensor data can provide the simplest method of obtaining input pattern data for each pixel in an image, each spectral band provides the input data for an input neuron. A network using three band image data would therefore have three input neurons, one per band. SPOT (HRV) in multi-spectral mode provides data in the green, red and near infrared wavebands. The seven bands of the Thematic Mapper satellite data were used by Halounova (1995) compare a maximum likelihood classifier with neural network classifier. The data set consisted of six TM bands, the normalised vegetation index:

$$NDVI = \frac{(R3 - R4)}{(R3 + R4)}$$

where R3 and R4 are the third and forth TM bands and the red, green and blue bands produced by Martin-Taylor Enhancement (MTE).

Foody, McCulloch and Yates (1995) assess the effect of training set size concluding that relative to conventional statistical classification techniques, neural networks are more appropriate where the training sample is small. They also conclude that if one class is more abundant the results will be distorted to this class unless the classes are highly separable. Minimal training sets have been used by Civco (mean vector and 10 samples per class), Hepner *et al.* (1993), Liu and Xiao (1991) and Benediktsson *et al.* (1990). The need for small training sets in order to reduce the training time on slow pre-Pentium PCs confirms the selection of a backpropagation network method rather than a more classical statistical classifier for this application and the need to select classes that are highly separable.

The challenge is to produce suitable input data from a greylevel image that will produce the strong associations required between the input patterns and the output classes. The method adopted involves identifying features held by the required output classes. These features could be texture, edge strength or specific greylevel bands, or other distinctive features that could be isolated within the image using some existing processing technique. Table 4.4 shows

three features: texture, edges and shadow related to four classes: buildings, roads / paths, vegetation and trees, the features that are most evident in each class are indicated with a 'X'. The selection of features must separate the classes, but this does not have to be a linear separation as multi-layer neural networks can generate non-linear class boundaries in feature space. Figure 4.2 shows a possible network topology.

Where three input neurons are used in a network such as Figure 4.2 the input data can be visualised in a RGB image format. Each one of the colour components of the image relating to an input neuron. For example, in Figure 4.3 the red component represents texture, the green component edge and the blue component shadow.

Classes / Features	Texture	Edges	Shadow
Buildings		X	X
Roads / Paths		X	
Vegetation	X		
Trees	X		X

Table 4.4 Possible combinations of input features and output classes for image classification.

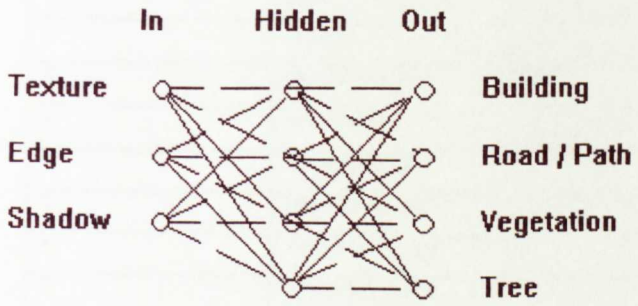

Figure 4.2 Example network topology.

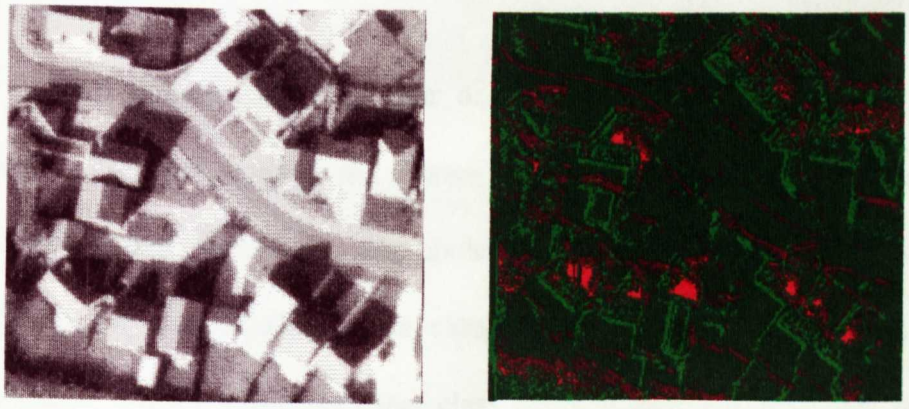

Figure 4.3 False colour image to display backpropagation output class.

4.5 Output Classes

Halounova (1995), who used the classification results to calculate run-off, selected a set of specific output classes shown in Table 4.5. Visualisation of the output classes can be achieved most easily by assigning each output class a colour.

Class	Type
1	Asphalt
2	Industrial roofs
3	House type 1
4	House type 2
5	Old built-up area
6	Vegetation type 1
7	Vegetation type 2
8	Vegetation type 3
9	Vegetation type 4
10	Bare soil type 1
11	Bare soil type 2
12	Water

Table 4.5 Output classes defined by Halounova (1995).

For the test images used here four or more classes are defined for each example. For each example the classes and their respective false colour are defined in an associated table. The choice of classes has a direct effect on the results of the classification. The classes chosen were by inspection and experiment, and the choice of other class types or number of classes would change the accuracy of the classification.

4.6 Classification Accuracy

The classification results are displayed as false colour images for easy visualisation. In addition there is a need for a numeric assessment of the accuracy of the result. This is achieved by using a method based discrete multi-variate analysis (DMVA). DMVA is a common method for quantifying the error in this type of problem. The results of the process are shown in a k x

k error matrix, where k is the number of classes. The elements of column i show the number of pixels which the reference data defines as class i and which the neural network has classified as class 1 to k. The diagonal of the matrix therefore shows the number of pixels that have been correctly classified by the neural network and the other elements of the column show the number of pixels that have been incorrectly classified. The overall classification accuracy (Ω) is therefore defined as:

$$\Omega = \frac{\sum_{i=1}^{k} \mathbf{A}_{ii}}{\sum_{i=1}^{k} \sum_{j=1}^{k} \mathbf{A}_{ij}}$$

where $\mathbf{A}$ is a $k \times k$ matrix. The overall classification accuracy is the sum of the diagonal elements divided by the sum of all of the elements of the matrix. Individual class accuracy is the diagonal element for that class divided by the sum of the corresponding row. The reference data are manually defined by allocation of a false colour to the original reference image. Any pixels in the reference image or the results of the neural network classification that are unclassified (do not belong to any of the trained classes) are allocated a special class number with a false colour of black. The DMVA calculation does not include the unclassified pixels and therefore reduces any confusion in the

results this might produce. Table 4.6 is an example of the results of classification shown in DMVA form.

Network Class	Reference Class 1	2	3	4	5	Class Accuracy
1	100	2	0	4	1	93.5%
2	2	85	2	3	5	87.6%
3	8	10	200	2	7	88.1%
4	4	12	8	350	0	93.6%
5	6	11	12	2	120	79.5%
				Overall Accuracy =		89.4%

Table 4.6 Example of classification results in Discrete Multi-variate Analysis form.

4.7 Class Selection

The results of the neural network classification are stored in a text file in which each line shows the network response and gives each input vector and respective output vector. The class selection process is carried out by NETIMAGE and the result can be displayed as a false colour image (a separate colour for each class) or as a text file of class numbers for each pixel in the classified image. The most basic way to select a class from the output vector is to take the highest value (Cromp 1991, Mulder and Spreeuwers 1991, Benediktsson *et al.* 1990). This simple arrangement was modified by Key *et al.* (1989) by a threshold. The threshold can be applied in two ways: the

neuron value must be the highest and above the defined threshold, or the highest value must exceed the next highest by the threshold. There is also another possible method of selection that was used by Aleksander for WISARD:

$$X = \frac{r_1 - r_2}{r_1}$$

where X is a measure of confidence in the selection of class $\mathbf{r_1}$, where $\mathbf{r_1}$ is the highest and $\mathbf{r_2}$ is the second highest output neuron. A threshold value on X then determines whether class $\mathbf{r_1}$ is selected. NETIMAGE is designed to use the threshold method of class selection for speed reasons. However, with the advent of the Pentium based PCs Aleksander's method could be employed with no appreciable speed reduction.

4.8 First Order Classification

In this research greylevel images are used rather than multi-spectral images as the most common low altitude aerial survey photography is black and white and was widely available at the start of this research. During the period of this research more satellite data has become available, and in the near future high resolution panchromatic and multi-spectral data with resolutions of 1-4m is planned. Several research groups have used multi-spectral data with neural

networks to classify land cover (Paola and Schowengerdt 1995), whereas very little work using neural networks has been carried out on panchromatic images. Multi-spectral data provide a natural set of data as input to a neural network: the individual bands form the input vector without modification. Panchromatic images by definition have only one spectral band which would result in only one input neuron if the technique adopted for multi-spectral images was used. Additionally, there is usually no direct relationship between the greylevel in a panchromatic image and the output class. The first step in the use of greylevel images with neural networks is therefore the creation of a suitable input vector that has a strong relationship with the output classes. First order image data, the raw greylevels, of an image do not generally have a strong enough relationship with the required output classes to be classified with a backpropagation network. The first order method presented here uses a window of pixels as an input vector to a backpropagation network. By using a group of pixels rather a single pixel value the network is presented with a primitive description of the texture of the image within the window selected.

With a feature such as texture the window size will directly affect the type of texture detected. Research in the area of fractals may solve the problem of texture scale. Fractals have not been discounted but left as a possible future project to enhance the results reported here. The window would need to be small enough to describe the texture of the object within one class but not too large as to spill over into another class. The effect of variation in window size

is demonstrated. There are several established measures of texture the most common being variance and standard deviation, used by many including Haralick *et al.* (1973) and Logan *et al.* (1979). The assumption is that those areas with the highest values of variance are assumed to have the roughest texture. More complex filters have been used by Wechler and Citron (1980) for texture classification. Here in the method using first order image data a simple rectangular window is used to provide the input vector to the neural network. The use of variance as a texture measure is considered later. The first requirement for using a neural network is to train the network which can only be achieved once a suitable set of training data has been created. The selection of training data for the first order method is discussed next.

4.8.1 First Order Trained Network using Minimal Training data and Four Classes

Minimising training time is a high priority when using a backpropagation network on a PC. Probably the most dominant factor in the training time is the size of the training data set: the larger the set the longer the training time. Reducing the size of the training data set is therefore an important factor when using a desktop PC if the training times are not to extend into days. Minimal training sets were used by Hepner *et al.* (1990). In the first example of first order training a training set of only 12 vectors, 3 vectors per class, is used. This abnormally small data set was chosen for the initial experiment so that

results could be achieved in less than an hour of processing time using an 80486 33Mhz PC. A more standard training data set size would be between 200 and 20,000 patterns. The second experiment has a larger training set as this was processed using a Pentium Pro PC. The training data was collected using NETIMAGE.

The network for this first example had an input vector size of nine neurons, a hidden layer of seven neurons and four output classes (Table 4.8). The transfer function used was a Sigmoid function and the momentum and learning rate were set to 0.5. The input vector was the first order greylevel values from a 3 x 3 window convolved across the image in Figure 4.4. The image was histogram equalised prior to processing to use the entire range of values from zero to 255. This was especially important as the values are scaled to the range zero to one prior to training. The four classes were chosen and the feature allocated to each class is shown in Table 4.7.

Class No	Feature
1	Light side of building roofs.
2	Vegetation.
3	Shadow side of building roofs.
4	Paths.

Table 4.7 Class allocation for first order training data.

PIXEL VALUE (SCALED 0..1)									CLASS No./FALSE COLOUR			
1	2	3	4	5	6	7	8	9	1 red	2 green	3 blue	4 yellow
081	020	007	091	028	008	084	015	005	1	0	0	0
067	071	072	067	072	082	067	071	069	1	0	0	0
071	075	016	072	074	016	075	079	017	1	0	0	0
019	011	023	036	032	027	024	026	020	0	1	0	0
030	024	025	040	032	024	030	024	017	0	1	0	0
026	042	030	020	020	028	011	026	044	0	1	0	0
072	082	089	067	080	085	065	085	090	0	0	1	0
084	075	071	080	076	074	080	076	071	0	0	1	0
083	074	073	082	073	073	081	075	075	0	0	1	0
044	048	040	045	042	042	045	045	047	0	0	0	1
055	047	046	050	044	051	046	045	042	0	0	0	1
053	044	042	050	042	038	047	040	037	0	0	0	1

Table 4.8 First order training data.

4.8.1.1 Results

The network response to training is shown in Table 4.9. The number of cycles required to train the network to an error of less than 0.1 was 1090. It can be seen that even with a minimal training set the network was able to produce a distinct class separation. This discrimination is shown by the values for each original trained class being very close to unity. Figure 4.4 is the result of a backpropagation network trained using first order data. This result is from a network trained using not only a minimal training set but also a small 3x3 window size. This result shows that the network is acting as an edge detector at this resolution. The light and dark sides roof classes outline the edges in the image and the other classes show areas of smoother texture such as the road and field.

This first example shows that the class separation seems to be dependent on texture and as expected, the mask size will have a direct effect on classes that

are extracted in the resulting image. Hence, the next example (Figure 4.5) uses a 5x5 mask. The classes chosen are similar to the previous example except class four was used for non-roof shadow. The result is that the light side of the roofs are clearly defined by class one (red) but the dark side of the roof, class three (blue), is now confused with the dark vegetation. The non-roof shadow, class four (yellow), is successful but the remaining class for vegetation shows only the lighter vegetation.

Class	Input vector								
1	0.81	0.20	0.07	0.91	0.28	0.08	0.84	0.15	0.05
1	0.67	0.71	0.72	0.67	0.72	0.82	0.67	0.71	0.69
1	0.71	0.75	0.16	0.72	0.74	0.16	0.75	0.79	0.17
2	0.19	0.11	0.23	0.36	0.32	0.27	0.24	0.26	0.20
2	0.30	0.24	0.25	0.40	0.32	0.24	0.30	0.24	0.17
2	0.26	0.42	0.30	0.20	0.20	0.28	0.11	0.26	0.44
3	0.72	0.82	0.89	0.67	0.80	0.85	0.65	0.85	0.90
3	0.84	0.75	0.71	0.80	0.76	0.74	0.80	0.76	0.71
3	0.83	0.74	0.73	0.82	0.73	0.73	0.81	0.75	0.75
4	0.44	0.48	0.40	0.45	0.42	0.42	0.45	0.45	0.47
4	0.55	0.47	0.46	0.50	0.44	0.51	0.46	0.45	0.42
4	0.53	0.44	0.42	0.50	0.42	0.38	0.47	0.40	0.37

Class	Output Vector			
1	0.990	0.010	0.000	0.013
1	0.949	0.000	0.029	0.033
1	0.994	0.000	0.002	0.000
2	0.000	0.995	0.000	0.003
2	0.000	0.982	0.000	0.006
2	0.000	0.991	0.000	0.024
3	0.028	0.000	0.979	0.000
3	0.029	0.000	0.981	0.000
3	0.018	0.000	0.988	0.000
4	0.006	0.017	0.000	0.977
4	0.028	0.005	0.000	0.985
4	0.010	0.014	0.000	0.973

Table 4.9 Network response to first order training.

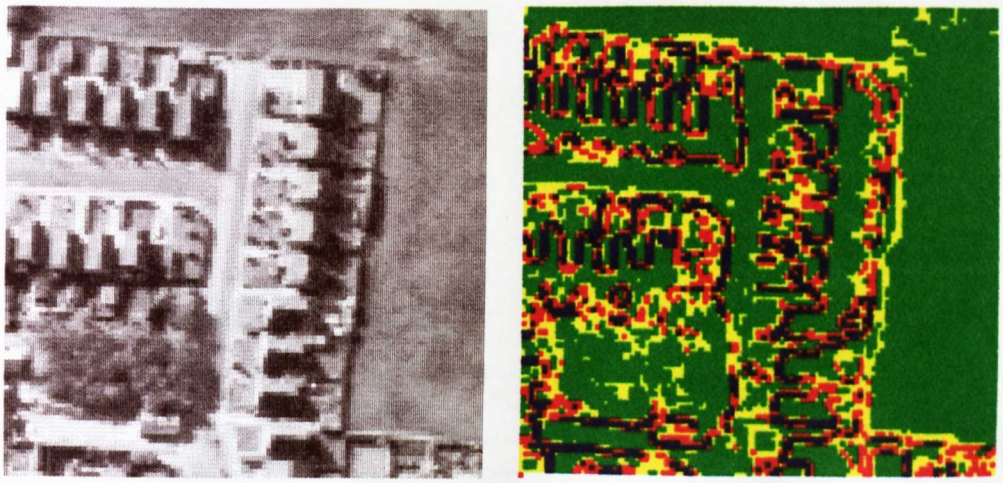

Figure 4.4 First order training results 3x3 mask (Z01).

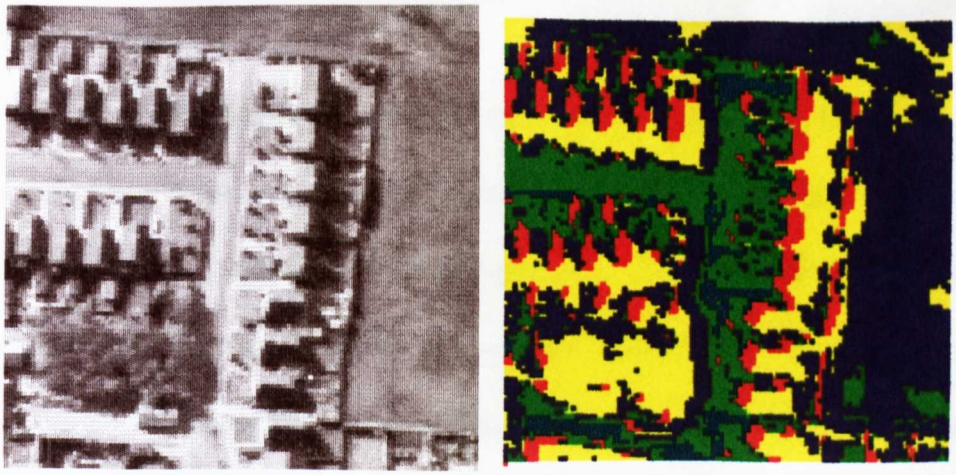

Figure 4.5 First order training results with 5x5 mask (Z02).

4.8.2 First Order Trained Network using Five Classes

The second set of results are produced with first order data uses a larger training set and five classes to allow a direct comparison of results with the later second order method which uses the same class allocation. The feature allocation to class numbers is shown in Table 4.10. As the training data set is

much larger, in this second example, it is not included but consists of an average of 10 patterns per feature compared to three in the first example of the method.

Class No	Feature
1	Road
2	Light side of roof
3	Shadow
4	Field
5	Trees

Table 4.10 Allocation of features to class numbers using five classes.

4.8.2.1 Results

The results of the second example of first order training using a larger training set and five classes is shown in Figure 4.6 and Figure 4.8, their respective DMVA are shown in Figure 4.7 and Figure 4.9. Two window sizes are used 3x3 and 5x5. The results of the 3x3 window example are very encouraging, however, the 5x5 example are poor. The possible reason for the poor results of the 5x5 window example may be due to the window size covering an area that contains more than one class. Whereas the 3x3 window remains within a single class and therefore provides the network with a training set that has greater class separation. This problem is, to a certain extent, avoided in non-urban images where the number of pixels allocated to one class may be greater than a window of 25x25, depending on the resolution of the image. Partly as a demonstration of this the next section digresses from urban to non-urban images. This set of example networks shows that textural information alone

may be successful if the window size is carefully chosen. A method that includes additional feature attributes is presented later.

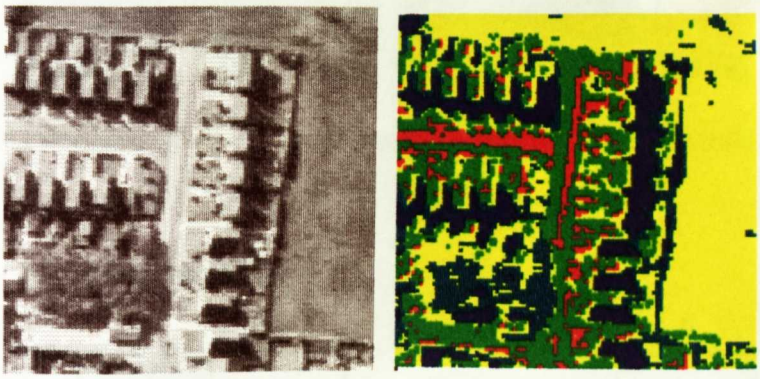

Figure 4.6 First order, five class, trained network result using a 3x3 window (Z01a).

		Reference Class					
		1	2	3	4	5	Class Accuracy
Network Class	1	404	107	0	0	0	79.1%
	2	780	330	28	16	20	28.1%
	3	0	3	583	0	56	90.8%
	4	25	229	63	2751	320	81.2%
	5	0	26	293	67	554	58.9%
						Overall Accuracy =	69.5%

Figure 4.7 DMVA of result image Z01a.

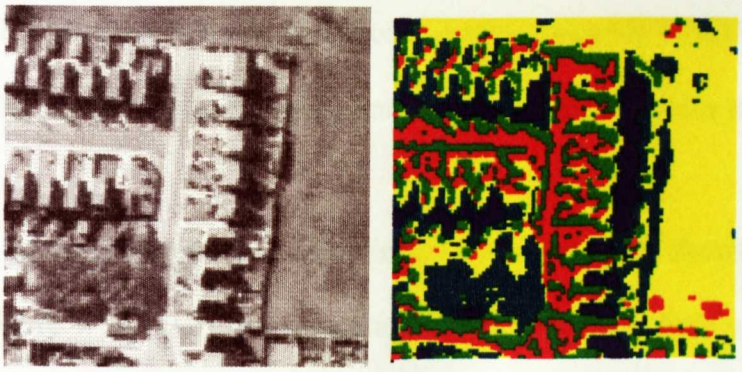

Figure 4.8 First order, five class, trained network using a 5x5 window (Z02a).

		Reference Class 1	2	3	4	5	Class Accuracy
Network Class	1	221	108	159	267	122	25.2%
	2	187	108	137	307	138	12.3%
	3	42	54	79	171	44	20.3%
	4	524	258	348	1407	408	47.8%
	5	177	167	205	682	214	14.8%
					Overall Accuracy =		31.1%

Figure 4.9 DMVA of result image Z02a.

4.8.3 Classifying a Non-Urban Image using a First Order Trained Network

Before moving on to look at unsupervised class selection the first order method is applied to non-urban images to observe the behaviour of the technique with a less cluttered image and for possible comparison with existing multi-spectral

methods. The training data was collected as before using NETIMAGE and the same training parameters as the urban images above were set in the network

The results of the first order non-urban image classification as shown in Figure 4.10 and Figure 4.13. The class separation is good with very little noise but as there was not a separate class for trees and shrubs there is some confusion in these areas. In Figure 4.10 and Figure 4.12 the tree shadows have however been identified quite successfully (cyan) and the sharp edge detail of the buildings is also slightly confused with other classes, whereas in Figure 4.13 and Figure 4.15, using a slightly different set of classes, the water and the hedges / trees fall into the same class (cyan), as the greylevel value rather than the texture has predominated. Also the road has been identified as scrub (red) probably because of the high frequency of the texture in the scrub being similar to the road edge texture.

Pixel greylevels									Class				
									1	2	3	4	5
0.573	0.616	0.580	0.580	0.580	0.620	0.620	0.588	0.624	1	0	0	0	0
0.553	0.553	0.553	0.561	0.549	0.525	0.565	0.569	0.588	1	0	0	0	0
0.584	0.604	0.604	0.584	0.592	0.612	0.580	0.604	0.588	1	0	0	0	0
0.608	0.573	0.596	0.604	0.584	0.576	0.616	0.576	0.584	1	0	0	0	0
0.573	0.573	0.565	0.569	0.584	0.569	0.592	0.576	0.573	1	0	0	0	0
0.576	0.608	0.604	0.573	0.600	0.612	0.600	0.604	0.576	1	0	0	0	0
0.549	0.541	0.514	0.537	0.525	0.537	0.525	0.518	0.522	1	0	0	0	0
0.549	0.518	0.533	0.506	0.518	0.478	0.494	0.525	0.471	1	0	0	0	0
0.518	0.510	0.533	0.518	0.506	0.522	0.541	0.525	0.537	1	0	0	0	0
0.533	0.514	0.510	0.514	0.482	0.471	0.541	0.525	0.514	1	0	0	0	0
0.584	0.580	0.573	0.573	0.573	0.549	0.545	0.561	0.545	1	0	0	0	0
0.518	0.525	0.537	0.549	0.545	0.553	0.529	0.573	0.584	1	0	0	0	0
0.541	0.529	0.525	0.533	0.525	0.525	0.541	0.525	0.529	1	0	0	0	0
0.478	0.522	0.506	0.486	0.525	0.522	0.486	0.537	0.529	0	1	0	0	0
0.451	0.416	0.420	0.451	0.416	0.424	0.435	0.420	0.420	0	1	0	0	0
0.443	0.424	0.427	0.443	0.424	0.424	0.467	0.427	0.427	0	1	0	0	0
0.427	0.416	0.420	0.431	0.412	0.416	0.447	0.416	0.416	0	1	0	0	0
0.494	0.514	0.518	0.494	0.522	0.522	0.490	0.510	0.518	0	1	0	0	0
0.475	0.435	0.455	0.490	0.451	0.443	0.494	0.459	0.439	0	1	0	0	0
0.459	0.463	0.471	0.447	0.459	0.459	0.439	0.451	0.459	0	1	0	0	0
0.475	0.490	0.467	0.475	0.494	0.478	0.467	0.459	0.475	0	1	0	0	0
0.463	0.455	0.459	0.459	0.447	0.447	0.451	0.447	0.439	0	1	0	0	0
0.427	0.459	0.471	0.412	0.447	0.455	0.416	0.447	0.455	0	1	0	0	0
0.502	0.482	0.459	0.498	0.494	0.471	0.518	0.506	0.471	0	1	0	0	0
0.671	0.667	0.671	0.667	0.655	0.659	0.655	0.663	0.678	0	0	1	0	0
0.671	0.608	0.620	0.647	0.612	0.592	0.651	0.600	0.580	0	0	1	0	0
0.651	0.635	0.604	0.663	0.616	0.612	0.655	0.624	0.631	0	0	1	0	0
0.635	0.635	0.682	0.643	0.635	0.686	0.627	0.643	0.659	0	0	1	0	0
0.580	0.627	0.675	0.576	0.604	0.694	0.588	0.659	0.694	0	0	1	0	0
0.647	0.667	0.631	0.675	0.659	0.624	0.667	0.639	0.643	0	0	1	0	0
0.541	0.549	0.561	0.561	0.549	0.573	0.561	0.565	0.596	0	0	1	0	0
0.573	0.584	0.561	0.592	0.576	0.514	0.549	0.525	0.529	0	0	1	0	0
0.631	0.639	0.600	0.604	0.592	0.616	0.675	0.675	0.682	0	0	0	1	0
0.596	0.624	0.565	0.647	0.682	0.635	0.686	0.592	0.682	0	0	0	1	0
0.620	0.573	0.749	0.604	0.557	0.816	0.592	0.576	0.835	0	0	0	1	0
0.651	0.643	0.690	0.671	0.690	0.616	0.639	0.584	0.647	0	0	0	1	0
0.384	0.475	0.561	0.329	0.439	0.451	0.400	0.537	0.510	0	0	0	0	1
0.416	0.349	0.325	0.345	0.376	0.435	0.561	0.439	0.498	0	0	0	0	1
0.416	0.420	0.439	0.412	0.467	0.475	0.490	0.533	0.427	0	0	0	0	1
0.365	0.294	0.392	0.325	0.388	0.443	0.325	0.337	0.400	0	0	0	0	1
0.561	0.439	0.498	0.451	0.420	0.522	0.510	0.302	0.314	0	0	0	0	1
0.498	0.424	0.439	0.522	0.467	0.541	0.314	0.463	0.561	0	0	0	0	1
0.537	0.549	0.596	0.306	0.251	0.278	0.365	0.365	0.439	0	0	0	0	1
0.620	0.541	0.537	0.325	0.345	0.306	0.494	0.584	0.365	0	0	0	0	1

Table 4.11 First order training data for non-urban images.

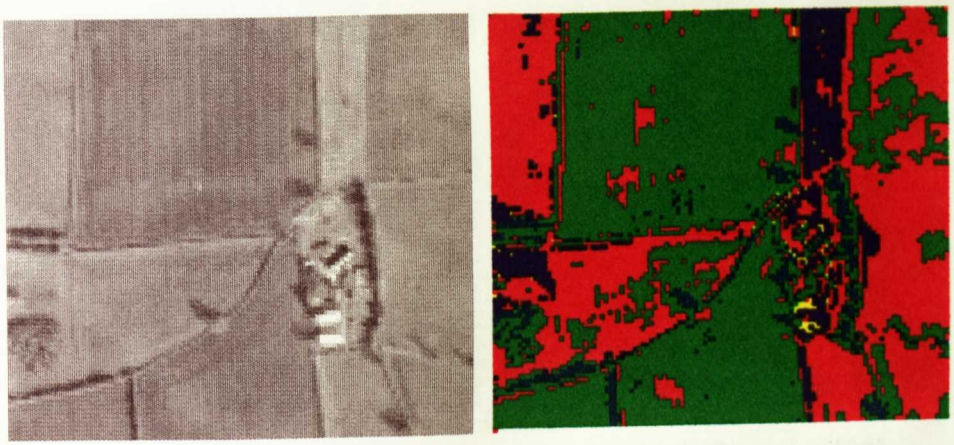

Figure 4.10 First order non-urban image example 1 result (Z03).

CLASS No.	FALSE COLOUR	CLASS TYPE
1	(Red)	Meadow.
2	(Green)	Ploughed.
3	(Blue)	Scrub.
4	(Yellow)	Building.
5	(Cyan)	Shadow

Figure 4.11 Class designation for non-urban image example 1.

	Reference Class					
	1	2	3	4	5	Class Accuracy
Network Class 1	2583	247	353	5	138	77.7%
2	856	3842	41	8	277	76.5%
3	112	14	542	13	18	77.5%
4	1	1	12	28	7	57.1%
5	3	44	1	6	311	85.2%
				Overall Accuracy =		77.2%

Figure 4.12 DMVA of result image Z03.

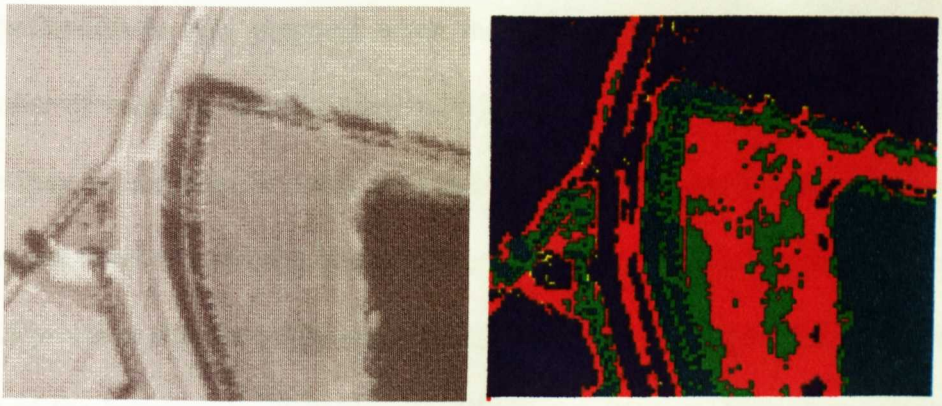

Figure 4.13 First order non-urban image example 2 result (Z04).

CLASS No.	FALSE COLOUR	CLASS TYPE
1	(Red)	Scrub.
2	(Green)	Hedges.
3	(Blue)	Ploughed.
4	(Yellow)	Building.
5	(Cyan)	Water

Figure 4.14 Class designation for non-urban images example 2.

		Reference Class					
		1	2	3	4	5	Class Accuracy
Network Class	1	2316	187	73	10	51	87.8%
	2	648	308	0	0	72	30.0%
	3	124	67	3289	62	3	92.8%
	4	5	2	1	13	1	59.1%
	5	23	575	0	0	1231	67.3%
					Overall Accuracy =		79.0%

Figure 4.15 DMVA of result image Z04.

It is possible from these examples to show that the successful use of panchromatic images is possible if texture alone is sufficient for the network to separate the required classes. Also this method of single spectral image

classification is only practical when the mask size is small, and hence the number of values in the input vector is low, but in urban images a large mask size may be required in order to select an area of the training image that is representative of a specific object in the image. Typically a tree might consist of 25 pixels or a house might easily be 200 pixels which may be the limit of the size of the input vector if the training time on a PC is to be kept reasonable. Here the only second-order feature is texture. At a later stage the use of edge gradient strength and the mean grey level are used in an attempt to reduce some of the problems encountered with the use of first order pixel values alone. It will be seen that the input vector can be no greater than three for any size of mask which provides constant training regardless of the mask size.

With a supervised training system as here the selection and number of classes is critical to the results, as seen by the overlapping of classes in these examples. Hence the next section diverges slightly to look at the problem identifying the number of possible classes when three features, (mean greylevel, edge gradient and texture) are used.

4.9 Unsupervised Classification to Assess the Number of Classes

One method of assessing the number of possible classes in an image classified using three second-order features, mean greylevel, edge gradient and texture,

represents the strength of each of these features at a point by a colour value. If the primary colour red, green and blue are assigned to the three features the resulting image will vary in colour directly in proportion to the calculated value of each of the features. Figure 4.16 and Figure 4.18 show an original greylevel image and the false colour result, in which red represents the mean greylevel, green represents the Laplacian or Sobel values and blue represents the variance in the original image. The brightness of each primary colour varies from 0 - 255. These three features can also be thought of as the low and high frequency components and the texture within the original image. The resulting image gives a visualisation of the number of classes in the image. To help interpret these false colour images Table 4.12 gives the perceived colours produced by mixing the three primary colours with intensity values from 0 to 255.

Red *Mean* Low frequency	Green *Laplacian* High frequency	Blue *Variance* Texture	Perceived Colour
255	0	0	red
0	255	0	green
0	0	255	blue
255	255	0	yellow
0	127	127	cyan
255	0	255	pink
127	0	127	purple
127	127	127	grey
255	255	255	white
0	0	0	black

Table 4.12 RGB Colour perception and related second-order value.

Resolution also plays a large part in the results from this process. Figure 4.19 is an enlarged section of Figure 4.16. The enlarged section of the image shows results in the three bands that are not visible in the original image. In Figure 4.19 all houses apart from two, can be identified by the class that is bright red, whereas in Figure 4.16 the distinction between the houses and some of the vegetation is not so clear.

Figure 4.16 False colour image of mean (red), Laplacian (green) and variance (blue) (Z05).

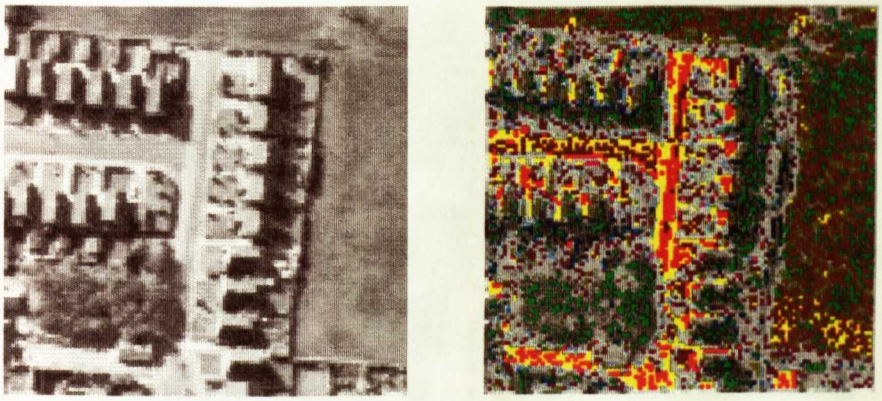

Figure 4.17 False colour images using mean (red), Laplace (green) and variance (blue) (Z06).

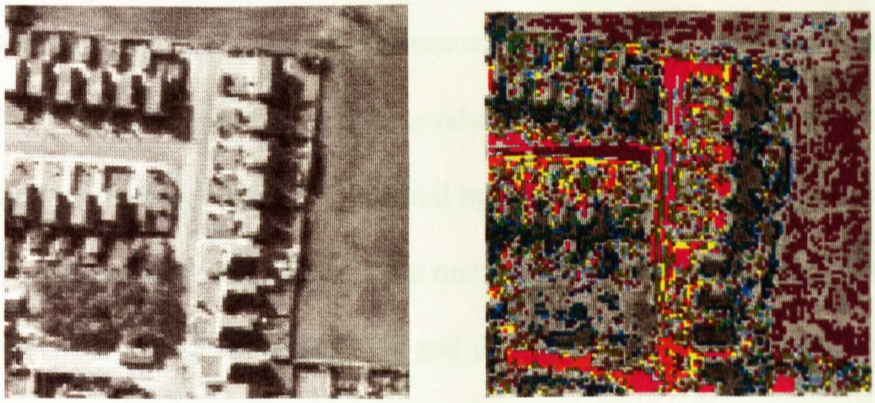

Figure 4.18 False colour image using mean (red), Sobel (green) and variance (blue) (Z07).

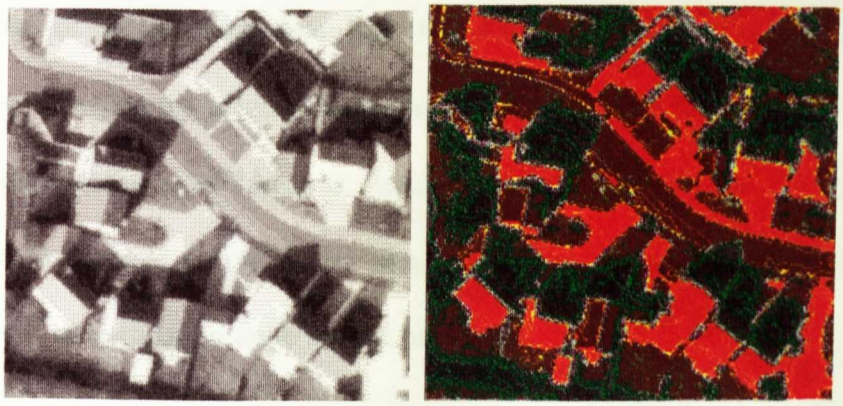

Figure 4.19 False colour image mean (red), Laplace (green) and variance (blue) (Z08).

False colour representations of second-order values using three features is the simplest translation to false colour because of the direct relationships defined between the RGB values and the second-order values. However, this can be extended to as many images as required by sub-dividing the range of values in each primary colour. For example the number of second-order values could be doubled by using the lower (0..127) and upper (128..255) ranges of values in each primary colour band. These features were chosen initially as representative of the dominant features in these urban images. The next section uses the three features mean, Laplacian and variance as input vectors for a backpropagation network.

4.10 Neural Network Classification using Second-order Values

Second-order data can be thought of as primary data processed to reveal certain aspects of the original image. In a typical urban image there are various

categories of second-order data: high frequency, low frequency and textural. By pre-processing the original image to produce these categories of data they can be used as the input vector to a neural network. Hence from a single band image a multi-band second order image has been created and therefore the multi-band classification methods can be applied.

The three bands of a SPOT HRV image represent different spectral bands, first order values, and these values can be used to classify the land cover using techniques such as nearest neighbour classification. The values can also be used as an input vector to a backpropagation network to classify the land cover in the image. If the original image only has one spectral band, such as a black and white Ordnance Survey aerial photograph, as in this research, these techniques cannot be used directly. The image must first be processed to produce two or more second-order values representing some feature of the original greylevel image. These values can then be used much in the same way as the traditional multi-band techniques.

Images of the same land area may have differing greylevel values depending on illumination and season so it seems that the use of greylevel itself is insufficient. However, extremes of greylevel such as in shadow may provide a key input for classification. In addition mean greylevels can be reasonably constant between images and can provide a clue to an object's class. Certain objects in an image also tend to occupy distinct areas of the greylevel spectrum,

such as certain types of building roofs and tarmac road surfaces. A density sliced image would separate these objects into distinct greylevel slices and may be used as an input value for the network in conjunction with other features.

Shape can provide one of the largest and most consistent cues to an object's class. Shape, unfortunately, is orientation-sensitive and therefore the network training set would have to take into consideration all possible orientations of the classified objects. Fortunately, neural networks can interpolate and therefore the number of orientations of a training set would not need to be large as shown by Aleksander (1984). In this research shape has not been used directly due to the difficulty in defining shape as a single value in order to be compatible with the features that are used which can be defined by a single number. However, because the input data is calculated from an area of the image shape is included, to a degree, within the data.

As can be seen from the results derived by first order method, texture is a strong identifying feature in an image and a suitable measure of texture is variance. As the images to be classified are urban in nature, edge gradient is also an important identifying feature for classes such as roads and buildings.

With consideration to the above discussion the values of the mean greylevel, Laplacian and variance were chosen as the input vector for the neural network. The number of possible classes has already been demonstrated in the section on

on unsupervised classification. As an example Figure 4.20 has four output classes.

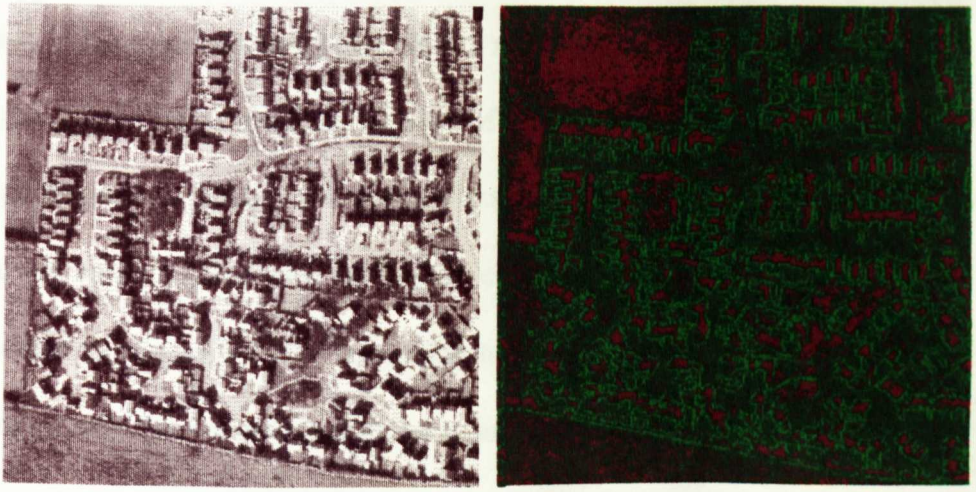

Figure 4.20 Second-order classified image (Z09).

4.10.1 Identifying Second-order Class

Before attempting to classify features in an image the relationship between the second-order values and the features is investigated. This relationship is thought of as identifying the second-order classes within the image. Once the possible second-order classes have been identified then the feature classes are identified. The training data in the first example was designed to relate mean, Laplacian and variance directly to an output class rather than a feature extracted from the image. This is a direct follow on from the unsupervised classification method show earlier but in this case using a neural network. The training parameters shown in Table 4.13 and the training data is shown in

Table 4.14. The result of this method (Figure 4.21) shows how the three basic classes of mean, Laplacian and variance are distributed in this type of image.

In Figure 4.21 areas of the image with high greylevel values are in class 1 (red). This is as expected since class 1 represents the mean greylevel. Class 2 (green) is the high frequency sections of the image such as the vegetation. The high frequency sections are within class two even in the shadow areas which are not easily visible with the human eye. The areas of the image that have greatest edge gradient have been allocated to class 3 (blue). Here, with a mask size of 3x3, the variance has acted as an edge detector. It is seen in the next example where there mask size is increased that the class relationship changes.

Parameter	Value
Transfer function (hidden layer)	Sigmoid
Transfer function (output layer)	Sigmoid
Learning rate	0.5
Momentum	0.5
Max error specified	1.0
Final error achieved	0.9978
Number of iterations	174
Initial value of weights	Random

Table 4.13 Second-order class training data training parameters.

Mean	Laplace	Variance	Class 1 red	Class 2 green	Class 3 blue
0.705882	0	0	1	0	0
0.745098	0.019608	0.015686	1	0	0
0.72549	0.039216	0.011765	1	0	0
0	0.72549	0	0	1	0
0.007843	0.745098	0.015686	0	1	0
0.019608	0.705882	0.019608	0	1	0
0	0	0.498039	0	0	1
0.027451	0.019608	0.509804	0	0	1
0.011765	0.031373	0.54902	0	0	1

Table 4.14 Second-order class training data.

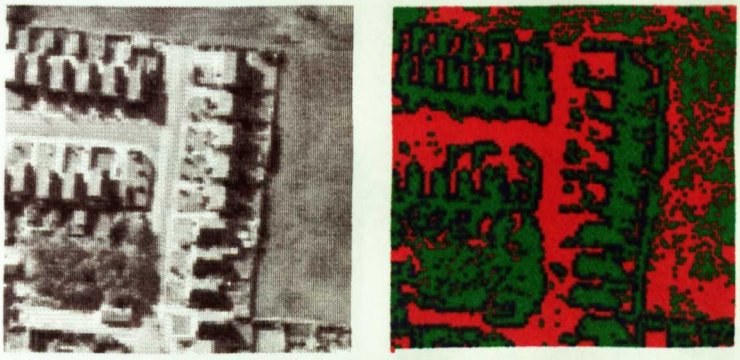

Figure 4.21 Second-order class result image (Z10).

The mask size is the number of pixels, in the form of a square, which are used to calculate the mean, Laplacian and variance values. The possible mask sizes available from the image processing software are 3x3, as in the above results, 5x5, 7x7 and 9x9. The results in Figure 4.22 are based on the same network training method and parameters as the previous example with mask sizes of 5x5, 7x7 and 9x9. The effect is very similar to using a mean or median filter, the amount of noise reducing (i.e. smoothing increasing) with the increase in

the mask size. Class 3 (blue) increases in area around the building boundaries and class 1 (red) predominates over class 2.

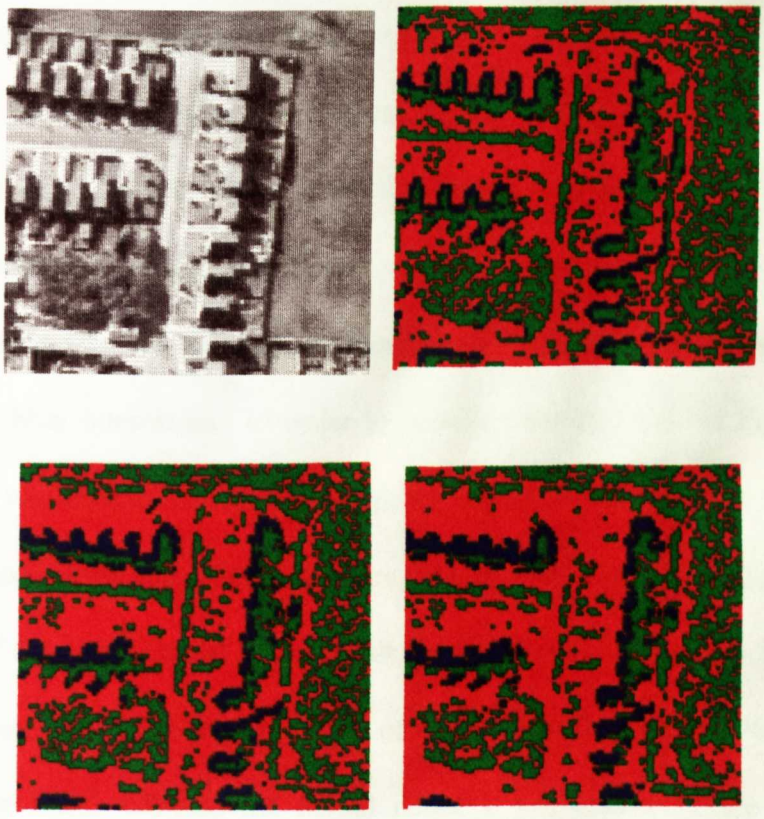

Figure 4.22 Second-order class results using mask sizes 5, 7 and 9 (Z11..Z13).

In the earlier unsupervised examples the output classes (colours) are based on combination of the three primary colours. As a comparison the next example uses a network trained on 10 classes, each class being represented in the result as a distinct combination of the three primary colours. The network is a backpropagation network with an input layer size of three, a hidden layer size of seven and an output layer size of 10. The training parameters are shown in

Table 4.15. The backpropagation network was trained on the data in Table 4.16. The network results are shown in Figure 4.23.

The results in Figure 4.23 should be directly comparable with those in Figure 4.17. Figure 4.17 is produced directly from the values produced by processing the image for mean, Laplace and variance. The values directly represent the red, green and blue intensities in the test image, whereas Figure 4.23 is the result of training a backpropagation network with a restricted synthetic training set representing the ten basic colours that can be produced by variations of red, green and blue intensities. In order to compare the two results Figure 4.24 shows the two images together. It is apparent from the two images that there is less noise in the backpropagation result as would be expected since the number of colours (classes) in the result is limited to ten. In more traditional image processing techniques the range of values allocated to each class would have to be fixed, whereas the neural network provides a degree of fuzzy decision boundaries. The class allocation in Figure 4.23 is not based here on actual classes but has been chosen to provide the greatest class separation, which in turn gives a faster training backpropagation training time. It is interesting to note that the objects in the original image are still identifiable even with such a restricted training set.

The next stage is therefore to use real image data to provide a training set that identifies classes that are related to features. These features will be a

combination of the three input values. The results of this section will provide cues as to the types of combination of mean, Laplacian and variance that certain image features possess.

Parameter	
Transfer function (hidden layer)	Sigmoid
Transfer function (output layer)	Sigmoid
Learning rate	0.5
Momentum	0.8
Max error specified	0.5
Final error achieved	0.95
Number of iterations	785
Initial value of weights	Random near zero

Table 4.15 Network definition for 10 output classes.

Mean	Laplace	Variance	1 red	2 green	3 blue	4 yellow	5 cyan	6 magenta	7 purple	8 grey	9 white	10 black
1.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	1.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1.00	1.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.50	0.50	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00
1.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00
0.50	0.00	0.50	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00
0.50	0.50	0.50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00
1.00	1.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00

Table 4.16 Training data for 10 second-order classes.

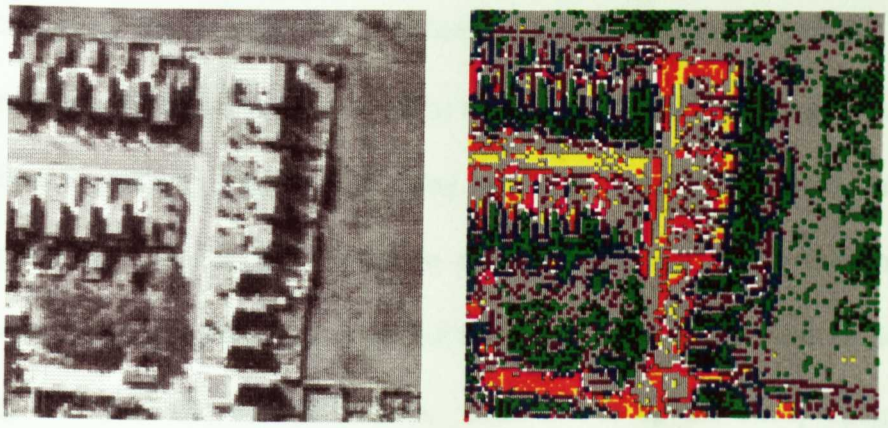

Figure 4.23 Second-order class result with 10 classes (Z14).

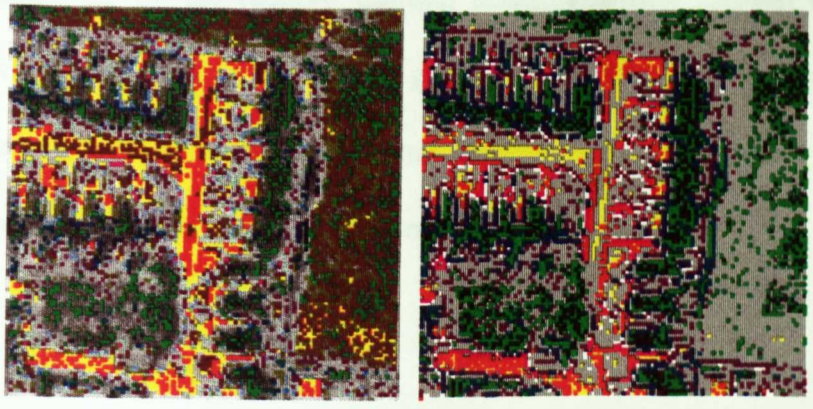

Figure 4.24 Comparison of unsupervised classified (left) and backpropagation network (right) results.

4.10.2 Identifying Feature Class

The previous section identified the second-order classes within an image this section now moves on to use that knowledge to identify features. Identifying features is, of course, the aim of the research and as with the first order classification method both urban and non-urban images are used. Training times using a backpropagation network can be very long and methods to

improve them are amongst the forefront in neural network research. This research uses PC-based software and the need for reasonable training times was therefore important. The idea of using artificially created minimal training sets provided a means with which to restrict the length of the training by removing training examples that are not typical of the assigned class or do not belong to any of the defined classes. This speeds up the backpropagation process as there are fewer false inputs and thus false minimum points for it to descend in to. The reduction in generalisation of the network was than compensated for by the reduced training times. The first method uses the unsupervised classification results as a guide the second method uses real data which is averaged to create the minimal training data. The accuracy of the result images is assessed using discrete multi-variate analysis.

4.10.2.1 Minimal Training Data

This method creates minimal training data by analysis of real image data. Values were captured using the 'NETIMAGE' program and the results reduced to the values shown in Table 4.17. A minimal set of input data with values spread randomly around the average is then created. The class separation is shown in Figure 4.25 as a plot of the average values of mean, Laplacian and variance for each class in Table 4.17. Figure 4.25 shows visually how similar classes one and two are. Another solution would be to use different second-order data. However where two classes are actually similar in reality the only

means of separation may be to use data derived from other factors in the image. The training data derived from the real image data is shown in Table 4.18 and Table 4.19.

The result of the classification is shown in Figure 4.26 and Figure 4.27. The most efficiently isolated class appears to be the field (yellow) and trees (grey). The hedges tend to be classified as trees, as might be expected. The shadow class (blue) is also fairly distinct. However, the road and roof classes are mixed which is as predicted due to the similarity of the classes, shown in the graph in Figure 4.25.

3x3 mask	Value		MEAN			LAPLACE			VARIANCE	
Class No.	Class Name	low	high	ave	low	high	ave	low	high	ave
1	ROAD	178	192	185	102	139	120.5	12	32	22
2	ROOF	163	210	186.5	109	151	130	20	32	26
3	SHADOW	23	32	27.5	104	164	134	32	64	48
4	FIELD	89	109	99	78	177	127.5	12	40	26
5	TREES	60	74	67	138	168	153	56	80	68

Table 4.17 Values derived from real image data.

Mean	Laplace	variance	1	2	3	4	5
178	102	12	1	0	0	0	0
192	139	32	1	0	0	0	0
185	120	22	1	0	0	0	0
163	109	20	0	1	0	0	0
210	151	130	0	1	0	0	0
186	130	26	0	1	0	0	0
23	104	32	0	0	1	0	0
32	164	64	0	0	1	0	0
28	134	48	0	0	1	0	0
89	78	12	0	0	0	1	0
109	177	12	0	0	0	1	0
99	127	26	0	0	0	1	0
60	138	56	0	0	0	0	1
74	168	80	0	0	0	0	1
67	153	68	0	0	0	0	1

Table 4.18 Minimal training data.

Mean	Laplace	variance	class 1	class 2	class 3	class 4	class 5
			road	roof	shadow	field	tree
			red	green	blue	yellow	grey
0.698	0.400	0.047	1.000	0.000	0.000	0.000	0.000
0.753	0.545	0.125	1.000	0.000	0.000	0.000	0.000
0.725	0.471	0.086	1.000	0.000	0.000	0.000	0.000
0.639	0.427	0.078	0.000	1.000	0.000	0.000	0.000
0.824	0.592	0.510	0.000	1.000	0.000	0.000	0.000
0.729	0.510	0.102	0.000	1.000	0.000	0.000	0.000
0.090	0.408	0.125	0.000	0.000	1.000	0.000	0.000
0.125	0.643	0.251	0.000	0.000	1.000	0.000	0.000
0.110	0.525	0.188	0.000	0.000	1.000	0.000	0.000
0.349	0.306	0.047	0.000	0.000	0.000	1.000	0.000
0.427	0.694	0.047	0.000	0.000	0.000	1.000	0.000
0.388	0.498	0.102	0.000	0.000	0.000	1.000	0.000
0.235	0.541	0.220	0.000	0.000	0.000	0.000	1.000
0.290	0.659	0.314	0.000	0.000	0.000	0.000	1.000
0.263	0.600	0.267	0.000	0.000	0.000	0.000	1.000

Table 4.19 Minimal training data (scaled 0..1).

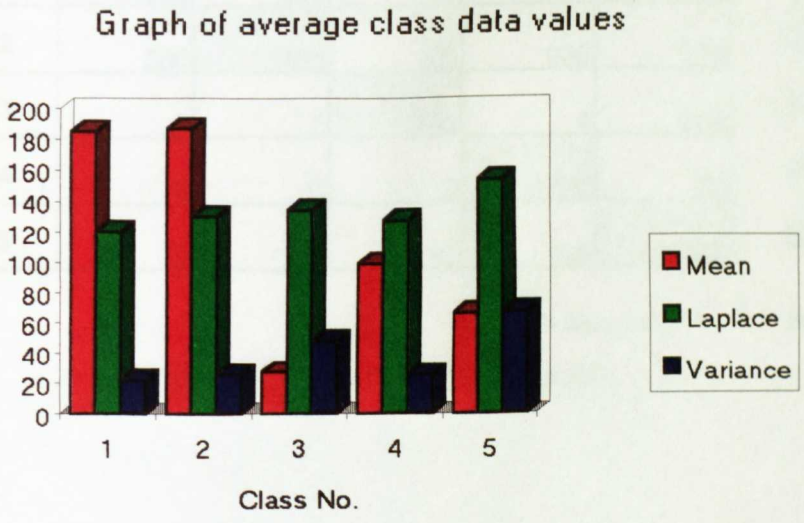

Figure 4.25 Graph of the average values for each class.

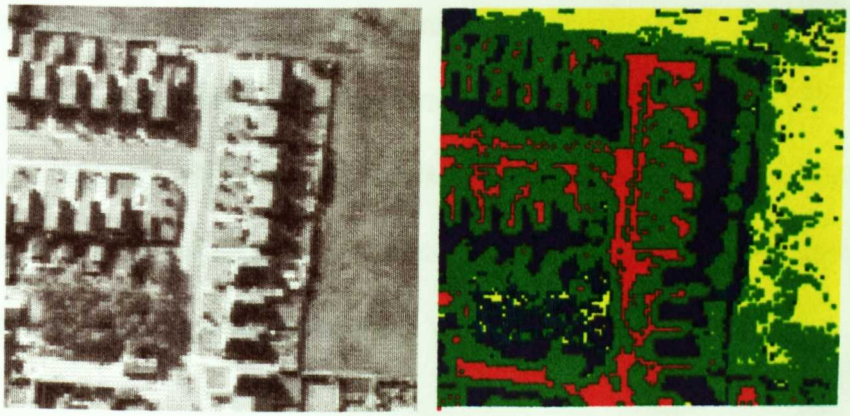

Figure 4.26 Result image from minimal training data (Z15).

	Reference Class					
Network Class	1	2	3	4	5	Class Accuracy
1	679	205	0	1	0	76.7%
2	530	483	138	936	276	20.4%
3	0	7	742	4	209	77.1%
4	0	0	0	1770	68	96.3%
5	0	0	87	123	397	65.4%
				Overall Accuracy =		61.2%

Figure 4.27 DMVA of result image Z15.

4.10.2.2 Real Training Data

This section again uses a minimal training set of data but in this case the it is raw data extracted directly from the training image. The same classes, network design and parameters as the previous example are used. The size of the training set is increased from 15 to 60 patterns: this is still considered a minimal size compared with other research in neural network classification.

The result is shown in Figure 4.28. Compared with previous example in Figure 4.26, which used artificially created training data, the use of real data and an increase in the size of the training data set has improved class separation. Class separation has improved best in classes 3, 4 and 5 (shadow, field and trees). The field and tree class are particularly well classified.

The remaining difficulty still lies in classifying classes 1 and 2 (road and roof). Although the road is well classified, parts of the roofs are classified as road and the roof class spreads into the gardens and hedges. A solution may be to increase the number of classes in line with the results from the unsupervised classification examples. Another may be to increase the size of the mask and identify a unique area of the roofs which significantly differs from other classes. A possible area on the roof is the ridge.

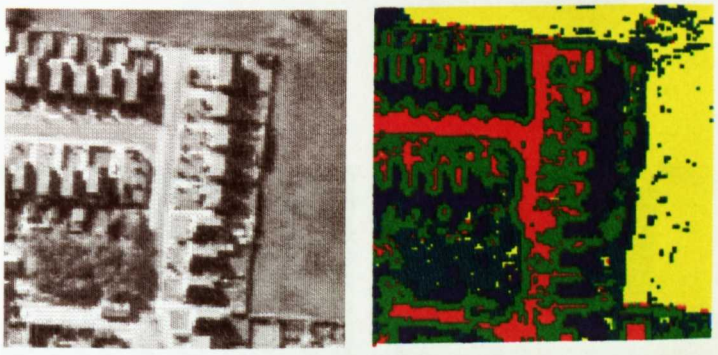

Figure 4.28 Result image second-order training data 3x3 mask (Z16).

Network Class	Reference Class 1	2	3	4	5	Class Accuracy
1	869	226	0	17	5	77.8%
2	340	456	72	18	86	46.9%
3	0	6	734	0	137	83.7%
4	0	0	1	2478	45	98.2%
5	0	7	160	321	677	58.1%
				Overall Accuracy =		78.3%

Figure 4.29 DMVA of result image Z16.

The next example uses a larger mask size (7x7) and shows an increase in classification accuracy with the increase in mask size. The results are shown Figure 4.30 and Figure 4.31. The increase in accuracy may be due to the increase in mask size giving a better definition of the characteristics of each class and smoothing away local variability. This effect is also produced in the non-urban images in the following section.

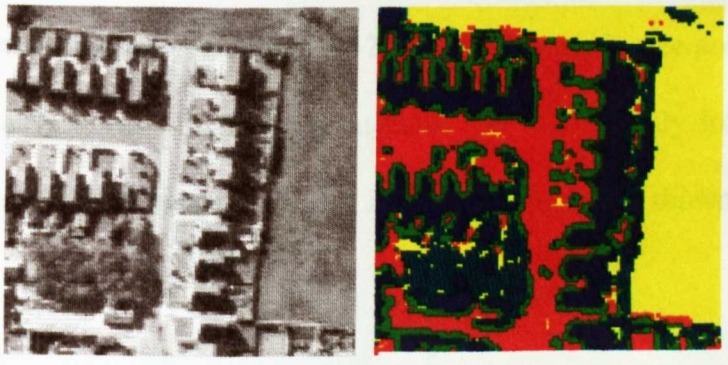

Figure 4.30 Result image second-order classification 7x7 mask (Z17).

Network Class \ Reference Class	1	2	3	4	5	Class Accuracy
1	1133	396	1	11	13	72.91%
2	58	273	64	4	54	60.26%
3	0	15	809	0	164	81.88%
4	17	0	2	2737	79	96.54%
5	1	11	91	82	640	77.58%
				Overall Accuracy =		84.03%

Figure 4.31 DMVA of result image Z17.

4.10.2.3 Non-Urban Images

As with the first order classification at the beginning of this chapter the next examples of second-order classification use non-urban images. The image chosen was manually classified as having five classes of cultivation and the neural network was trained on the real image data as shown in Table 4.20. The trained network was then used to classify the image in Figure 4.32. The training required 20000 iterations, with the learning rate and momentum set at 0.5. The result is shown in Figure 4.32 using a 3x3 mask size and in Figure 4.34 using a 9x9 mask. The change in mask size significantly improves the classification accuracy as demonstrated in the DMVA results shown in Figure 4.33 Figure 4.35.

The non-urban images show a much greater degree of classification accuracy to an extent because each class covers a much larger and more consistent area of the image than in urban images. In urban images there can be several classes very close to each other and only covering a small area of the overall image. Because of the small size of the classes in urban images the choice of mask size or image resolution plays a much greater part than in non-urban images.

Mean	Laplace	Variance	1	2	3	4	5
0.616	0.510	0.063	1	0	0	0	0
0.592	0.553	0.047	1	0	0	0	0
0.596	0.486	0.031	1	0	0	0	0
0.596	0.573	0.031	1	0	0	0	0
0.569	0.537	0.031	1	0	0	0	0
0.588	0.498	0.031	1	0	0	0	0
0.490	0.545	0.110	0	1	0	0	0
0.435	0.486	0.047	0	1	0	0	0
0.525	0.498	0.016	0	1	0	0	0
0.447	0.467	0.031	0	1	0	0	0
0.518	0.486	0.047	0	1	0	0	0
0.541	0.541	0.031	0	1	0	0	0
0.498	0.475	0.047	0	1	0	0	0
0.482	0.459	0.016	0	1	0	0	0
0.451	0.510	0.047	0	1	0	0	0
0.455	0.475	0.016	0	1	0	0	0
0.447	0.518	0.016	0	1	0	0	0
0.651	0.635	0.110	0	0	1	0	0
0.671	0.518	0.016	0	0	1	0	0
0.663	0.557	0.063	0	0	1	0	0
0.647	0.565	0.063	0	0	1	0	0
0.584	0.537	0.047	0	0	1	0	0
0.545	0.451	0.063	0	0	1	0	0
0.518	0.494	0.016	0	0	1	0	0
0.565	0.416	0.047	0	0	0	1	0
0.486	0.529	0.031	0	0	0	1	0
0.545	0.502	0.063	0	0	0	1	0
0.557	0.467	0.031	0	0	0	1	0
0.533	0.494	0.016	0	0	0	1	0
0.514	0.506	0.016	0	0	0	1	0
0.392	0.506	0.251	0	0	0	0	1
0.349	0.475	0.157	0	0	0	0	1
0.435	0.412	0.267	0	0	0	0	1
0.455	0.286	0.220	0	0	0	0	1
0.384	0.682	0.157	0	0	0	0	1
0.455	0.518	0.251	0	0	0	0	1
0.455	0.078	0.471	0	0	0	0	1

Table 4.20 Real training data for non-urban image analysis.

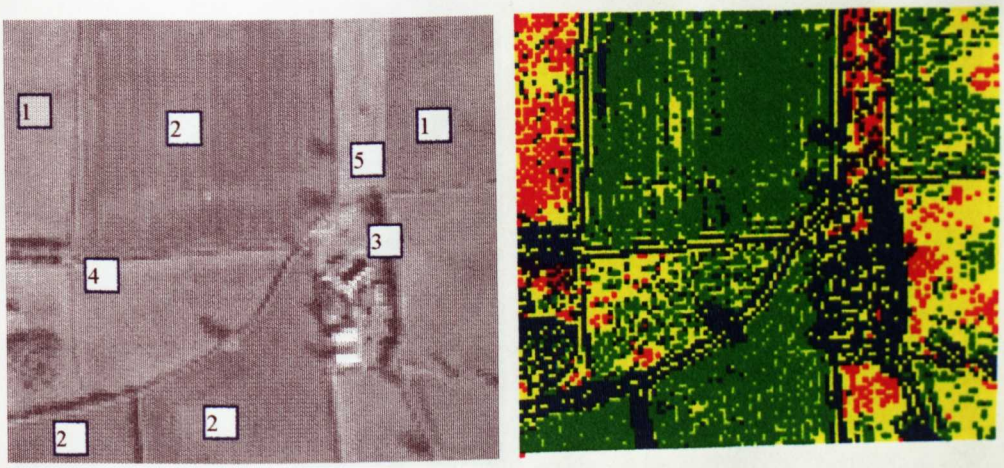

Figure 4.32 Second-order non-urban image result 3x3 mask (Z18).

Network Class	Reference Class 1	2	3	4	5	Class Accuracy
1	306	1	6	3	252	53.9%
2	564	2431	298	145	82	69.1%
3	4	0	1	0	121	0.0%
4	1459	547	182	346	710	10.7%
5	268	181	711	381	778	33.5%
				Overall Accuracy =		39.5%

Figure 4.33 DMVA of result Z18.

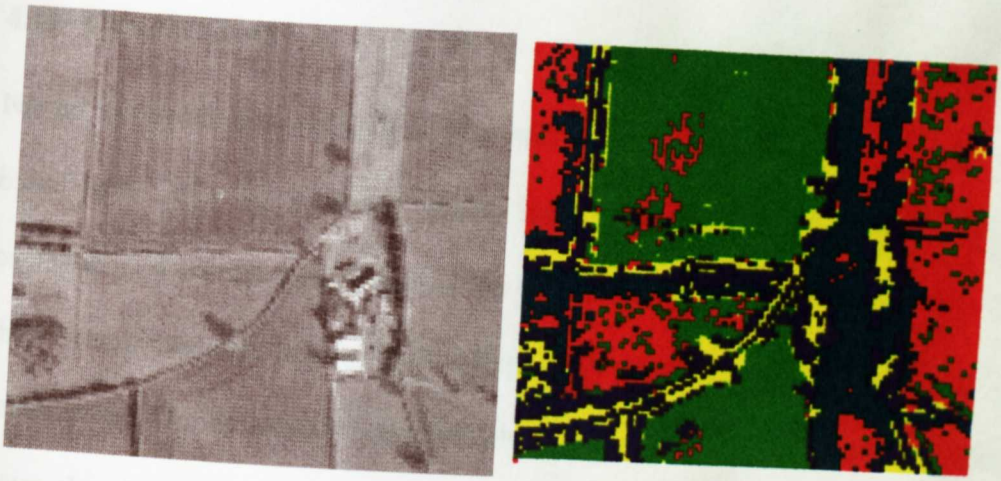

Figure 4.34 Second-order (9x9 mask) non-urban result image (Z19).

No.	False Colour	Type
1	Red	Pasture
2	Green	Ploughed
3	Blue	Dark Vegetation (Trees)
4	Yellow	Light Vegetation (Hedges)
5	Cyan	Scrub

Table 4.21 Class definition for Z18 and Z19.

Network Class \ Reference Class	1	2	3	4	5	Class Accuracy
1	1933	132	0	3	249	83.4%
2	336	2943	0	0	1	89.7%
3	104	47	1171	1	42	85.8%
4	42	38	0	834	3	90.9%
5	186	0	27	37	1648	86.8%
					Overall Accuracy =	87.2%

Figure 4.35 DMVA of Z19.

4.11 Conclusions

Neural networks were identified as a method of avoiding the pitfalls of rule based and other AI methods which lacked flexibility and could become overwhelmed by the complexity of an urban image. Neural networks, with their non-parametric nature and interpolative qualities, present a feasible methodology for the classification of urban images. Artificial neural networks have already been used for land cover classification by Benediktsson, Swain and Ersoy (1990), Bischof, Schneider and Pinz (1992), Civco (1991, 1993), Dreyer (1993).

This chapter proposes new methods of classifying single spectral band images such as panchromatic images using a neural network. The methods use existing image processing methods to transform a 256 greylevel image into a form suitable for classification by a backpropagation network. The methods provide input data that have the strong relationship between the input and output data required for successful training of a backpropagation network. All of the methods are implemented on a PC running Windows using a commercial neural network called SLUG and image processing software developed in-house specifically for pre- and post-processing image data for the neural network.

The test images chosen are representative of urban areas in Great Britain rather than the images used in some previous research which include flat roof

buildings and grid-shaped road systems. The training data was initially reduced to a minimal size in order to allow reasonable training times as the PCs used in the early stages of this research were not sufficiently powerful to cope with large data sets. The size of the training set influences the quality of the results, as shown by Civco (1991) where only one mean vector was used for each class, because a degree of randomness must be presented to the network at the training stage if the generalisation is to occur. Foody, McCulloch and Yates (1995) assess the effect of training set size and conclude that relative to conventional statistical classification techniques, neural networks are more appropriate where the training sample is small. They also conclude that if one class is more abundant the results will be distorted unless the classes are highly separable. Minimal training sets have been used by Civco (mean vector and 10 samples per class), Hepner *et al.* (1993), Liu and Xiao (1991) and Benediktsson *et al.* (1990). The training sets here vary from one to 10 samples per class. With the advent of fast Pentium PCs the size could be increased to a 100 samples per class and still maintain reasonable training times.

Class selection was carefully considered to ensure that they were as separable as possible. Therefore some unsupervised classification using a novel image processing technique was used so that the images could be assessed to see which objects in the images could be selected as a class object. However, in

order to achieve the aim certain objects had to be selected as classes such as roofs, building edges and roads.

The first classification method uses a window of pixels as the input vector and thus a measure of texture for a given class (Kamata and Kawaguchi 1993, Hepner *et al.* 1990 and Ritter and Hepner 1990). The main limitation of this method is its sensitivity to the spatial resolution of the image, in much the same way as the texture changes with changes in the number of pixels selected. This method produced a classification accuracy around 78% for non-urban images and at best 69% and worst 31% for urban images. The main limitation of this method is the size of the input vector, as to obtain a reasonable measure of the local texture from a window of raw pixel values requires a minimum of a 3x3 window and a maximum that could be as much as 21x21 depending on the resolution of the image. As the resolution decreases the textural information in the image is lost; however, the size of the window can be reduced as a smaller window will cover an area of representative texture within a class. For each type of image there is probably an ideal balance between window size and resolution. The main disadvantage of large window sizes is the increased training times, which on a PC become unacceptable above 9x9 window sizes which generate an input vector 81 neurons in size. There are possible additional factors involved in this method such as rotational sensitivity and uneven lighting of similar texture. Both of these should be removed by using a training set that includes these variations, but this then works against the need

for small sets to reduce training times. Would it be possible to design a method that could have a window of any size, with limited increase in processing time, and a constant size input vector to the neural network?

The second method, that is demonstrated here uses second-order values, fulfils the requirement of constant input vector size, and hence produces predictable and small training times, and variable window sizes without a corresponding increase in processing. This is achieved by pre-processing the training data to produce a three-neuron input vector to the network. In the first method texture is the only image attribute characteristic of the window of pixels. Therefore two other attributes are considered, namely, mean greylevel and a measure of the high frequency content of the class sample. The Laplacian value of the window is chosen as a suitable measure of the high frequency content and the variance is chosen as a measure of texture. Hence the input vector consists of mean, Laplacian and variance, which were considered as representative of the classes of object required for classification. Window sizes, using this method, could be increased as the processing time was several orders of magnitude lower than presenting this window size using the first method as an input vector to the network. Window size with this method therefore no longer becomes a limitation to using neural networks for classification. The window size is demonstrated to play an important role in the classification accuracy, increasing from 78% to 84% with an increase in the window size from 3x3 to

7x7. Using the second order method this is achieved with no increase in neural network training time and minimal increase in overall processing time.

The third method discussed here is a proposal to use more than one network. This hierarchical system classifies class primitives then uses these as the input to the next network where the classes are defined in terms of these primitives. This arrangement is used by Kanellopoulos *et al.* (1991) but there was not sufficient time to develop this method beyond the initial stages here. Hierarchical processing is almost certainly a factor in human pattern recognition as context and the relationship of certain image attributes provide keys to recognition.

The development of these methods has highlighted many factors that affect the use of neural networks in the classification of urban images. The second order method has proved to be applicable to non-urban images and could be adapted for use with multi-spectral image data. Other image attributes could be used than the mean, Laplacian and variance, possibly to increase the accuracy by improving class separation. The final chapter continues this discussion by looking at possible future applications and draws together some overall conclusions on this research.

CHAPTER 5

5. FUTURE DEVELOPMENTS AND APPLICATIONS

5.1 Future Developments

Most research using neural networks for image classification has found that the technique performed as well as or better than other existing classification methods (Benediktsson *et al.* 1990, Bischof *et al.* 1992, Blonda *et al.* 1993, Frierens *et al.* 1994, Heermann and Khazenie 1992, Kanellopoulos *et al.* 1993, Key *et. al.* 1989, 1990, Kiang 1992, Li *et al.* 1993, Liu and Xiao 1991, Medina and Vasquez 1991, Short 1991, Paola and Schowengerdt 1994 and Yosida and Omatu 1994).

Standard image classification methods use single pixel values as input. This research uses a window of pixels in order to calculate second-order spectral or spatial values. Several researchers (Kamata and Kawaguchi 1993, Hepner *et al.* 1990, Paola and Schowengerdt 1994 and Ritter and Hepner 1990) have found that the use of a window of pixels with standard neural network methods allowed the texture of the window to be included as a factor in the input vector, with an improvement in the accuracy of the image classification. It was

assumed by these groups that the use of a windowing procedure brought textural information into the training and classification process.

Hepner *et al.* (1990) and Ritter and Hepner (1990) used, with success, smaller training sets in the neural network methods than had been previously used in standard methods such as maximum likelihood classification. The research reported here also used small training sets in order to reduce processing time and simplify the collection of training data. In addition, completely synthetic (artificially created) training sets were also used. Paola and Schowengerdt (1994) put forward an explanation for the success of small training sets in comparison to standard methods.

The success of the methods proposed in this research is dependent on texture and the scale of the texture. Choosing the correct window size with which to sample the image for texture and other attributes is discussed. Curran (1988) demonstrates the use a semi-variogram, an important element of geo-statistics, to determine the optimal window size in such problems. Figure 5.1 shows a graph of variance against pixel distance. The pixel distance is the distance from the centre pixel of a window of pixel values that are used to calculate the variance. Curran (1988) found that for each texture type that there was a distance at which the variance no longer changed or changed only very slowly, as can be seen in the graph. The lessons learnt from Curran's (1988) work

could be applied to the methods in this research in order to optimise the window size that is used to create neural network training sets.

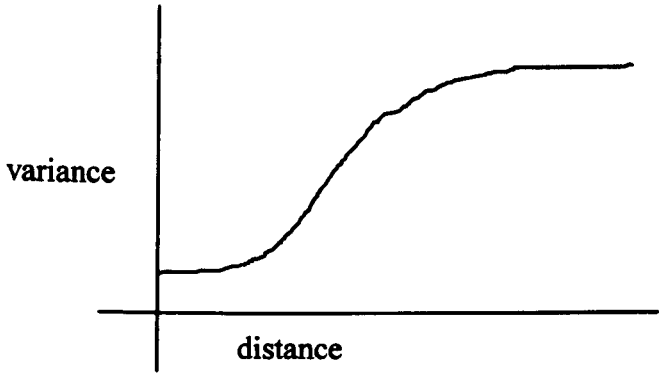

Figure 5.1 Semi-variogram graph.

Foody (1995) proposes the use of prior knowledge in neural network classification with minimal training sets. Urban images often have class distributions that are not equivalent as there may be large areas of gardens and roads but only a small number of pixels belonging to the building class. This uneven distribution may have a lesser effect on neural network classifiers than statistical classifiers but will still reduce the accuracy of the results, the use of prior knowledge is shown by Foody (1995) to increase the classification accuracy. The results of the first and second level classification methods may benefit from the employment of a hierarchical set of neural networks by using primitive image classification results to provide input to a network that is trained on combining these primitives. A possible network design shown in Figure 5.2 might be used to generate 'third order' classes to provide input to a

second network. An example result is shown in Figure 5.3 with a key to the output classes in Table 5.1.

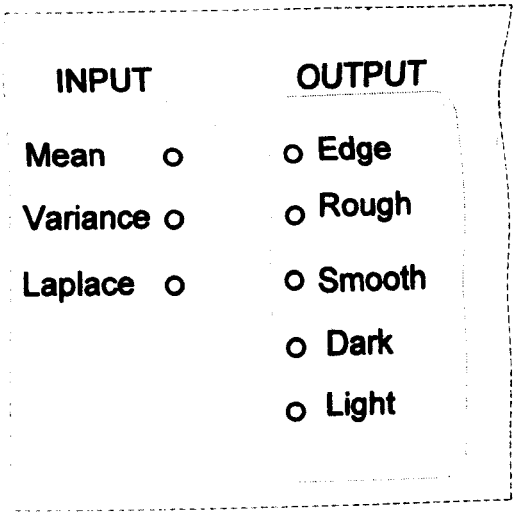

Figure 5.2 Structure of network trained to output primitive second-order spectral classes

Output Colour	Third order type
red	light smooth
green	dark edge
blue	dark rough
yellow	light edge
cyan	
magenta	light rough
purple	medium rough
grey	
white	
black	dark smooth

Table 5.1 Key to image of third order classes.

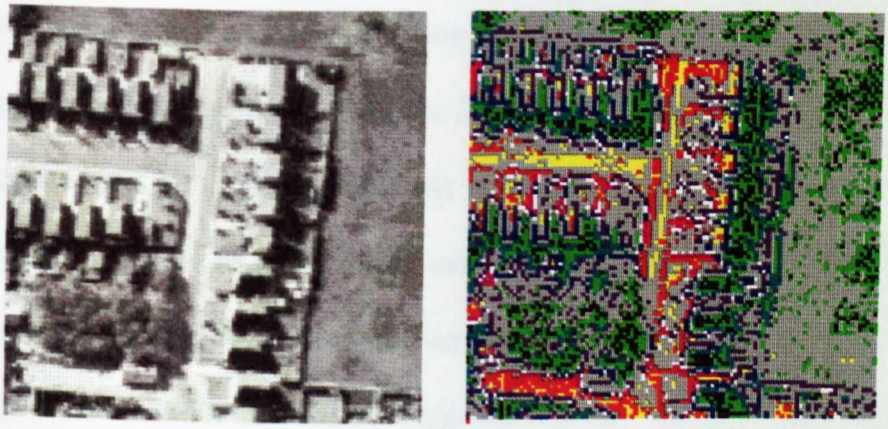

Figure 5.3 Example image of third order classes (Z20)

The main disadvantages of using neural networks for classification fall into two categories. Firstly, they have a complicated training process involving trial and error method for setting parameters such as: learning rate, momentum and hidden layer size. Secondly, the training time is directly proportional to training set size, the topology of the network and the size of input and output vector. These disadvantages are mainly outweighed by the fast classification time, as once the network is trained it is a single pass calculation. Hence once trained the training time reduces, in proportion to the overall classification time, as the size of the image increases. Also the advantage that neural networks are able to generalise to the extent that they can classify images on which they are not trained. Neural networks are very amenable to implementation in parallel processing systems. The current software (NETIMAGE) could be developed to assist the user with gathering training and image data by removing the need to create intermediate files and by passing the training and run time data directly to the network. The network could be

trained over a period of time with more and varied data so that the ability to interpolate and make fuzzy decisions is optimised. Adapting some of the techniques used by Aleksander (1984) in WISARD may be a way forward especially in relation to stereo image analysis. Stereo disparity might be used as another image feature in the input vector to the neural network. Non-urban images contain classes that form contiguous regions. Whereas urban images have rapid changes of class over relatively small areas. A pixel by pixel matching process, as used in non-urban image classification, may not be suitable for urban images. Therefore, a possible way forward may be to interpret the class membership results not on a pixel by pixel basis but to evaluate windows of pixels to find the areas of the image where membership of a certain class is strongest. This method would identify the centroid of a class rather than class of every pixel in the image.

5.2 Future Applications

The proposed new satellite, to be launched by the OrbImage company, has sensors that will produce panchromatic imagery resolutions of one metre. The techniques developed in this research are designed for single-band images and may have a direct application to OrbImage images in the future. The high resolution of the images is ideal for urban images where commercial

applications of remote sensing may require buildings to be designated to an accuracy of one metre.

The methods developed here could be applied to greylevel images produced by the Ordnance Survey for land cover classification at a much higher resolution than is now possible with only satellite data. This may be particularly applicable to precision farming or survey of land use.

REFERENCES

The following references are specifically cited in the text.

Aleksander, I. (1966) Self adaptive logic circuits, *Electronic Letters,* **2(8),** August 1966, 321-322.

Aleksander, I. (1973) Random Logic Nets: Stability and Adaptation, *International Journal of Man-Machine Studies,* **5,** 115-131.

Aleksander, I. Stoneham, T.J. and Wilkie, R.A. (1982) Computer Vision Systems for Industry, *Digital Systems for Industrial Automation,* **1(4), 305-320.**

Aleksander, I. (1983) Emergent Intelligent Properties of Progressively Structured Pattern Recognition Nets, *Pattern Recognition Letters,* **1, 375-384.**

Aleksander, I., Thomas, W.V. and Bowden, P.A. (1984) WISARD; A Radical Step Forward in Image Recognition, *Sensor Review (GB),* **4** (3), July 1984, 120-124.

Aleksander, I. and Wilson, M.J.D. (1985a) Prospects for Adaptive Window Architectures, *SPIE architectures and Algorithms for Digital Image Processing,* **596,** 74-81.

Sensing Data, *IEEE Transactions on Geoscience and Remote Sensing*, **28**, 540-552.

Birge, R.R. (1995) Protein-Based Computers, *Scientific American*, March 1995, 66-71.

Bischof, H. Schneider, W. and Pinz, A.J. (1992), Multispectral Classification of Landsat Images using Neural Networks, *IEEE Transactions on Geoscience and Remote Sensing*, **30**(3), 482-490.

Blonda, P., La Forgia, V., Pasquariello, G. and Satalino, G. 1994, Multispectral Classification by a Modular Neural Network Architecture, *Proceedings of the International Geoscience and Remote Sensing Symposium (IGARSS 94)*, Pasadena CA, 8-12 August 1994, 1873-1876.

Cappellini, V., Chiuderi, A. and Fini, S. (1995) Neural Networks in Remote Sensing Multisensor Data Processing, *Sensors and Environmental Applications of Remote Sensing*, Askne(ed.), 1995 Balkema, Rotterdam, ISBN 90 5410 5240, 457-462.

Carpenter, G.A. and Grossberg, S. (1991) *Pattern Recognition by Self-Organising Neural Networks*, The MIT Press.

Chan, M.H. and Tsui, H.T. (1989) Recognition of Partially Occluded 3D Objects, *IEE Proceedings*, **136E**(2), 124-141, March 1989.

Civco, D.L. (1993), Artificial Neural Networks for Land Cover Classification and Mapping, *International Journal of Geographic Information Systems*, **7**, 173-186.

Civco, D.L. (1991), Landsat TM Image Classification with an Artificial Neural Network, *Proceedings ASPRS-ACSM Annual Meeting*, Baltimore MD, **3**, 67-77.

Cooper, P.R., Friedmann, D.E. and Wood, S.A. (1986) The Automatic Generation of Digital Terrain Models from Satellite Images by Stereo, *SPIE*, **660**, 124-135.

Curran, P.J. (1986) *Principles of Remote Sensing*, Longman Scientific and Technical.

Curran, P.J. (1988) The Semi-Variogram in Remote Sensing, *Remote Sensing of the Environment*, **24**, 493-507.

Day, T. and Muller, J-P. (1988) Digital Elevation Model Production by Stereo Matching SPOT Image Pairs: A Comparison of Algorithms, *Proceedings 3rd Alvey Vision Club*, Manchester UK, 31 August - 2 September 1988.

Day, T. and Muller, J-P. (1988) Quality Assessment of Digital Elevation Models Produced by Automatic Stereo Matchers from SPOT Image Pairs, *Photogrammetric Record*, **12**(72), 797-808.

Dayhoff, J. E. (1990) *Neural Network Architectures*, Van Nostrand Reinhold, New York.

Dowman, I.J., Gugan, D.J., Muller, J.P., O'Neil, M. and Paramananda, V. (1988) Digital Processing of SPOT Data, *Fast Processing of Photogrammetric Data WG II/2*, 1-13, University College London.

Dreyer, P. (1993), Classification of Land Cover using Optimised Neural Nets on SPOT Data, *Photogrammetric Engineering and Remote Sensing*, **59**, 617-621.

Faust, H-W (1989) Digitisation of Photogrammetric Images, *Proceedings of the 42nd Photogrammetric Week*, Stuttgart University.

Grimson, W.E.L. (1980) Computing Shape using a Theory of Human Vision, *Unpublished PhD Thesis*, Mathematics Department, Massachusetts Institute of Technology, Cambridge MA.

Grossberg, S. (1982) *The Adaptive Brain*, Reidel Press, Boston, MA.

Gruen, A.W. (1985) Adaptive Least Squares Correlation: A Powerful Image Matching Technique, *South African Journal of Photogrammetry, Remote Sensing and Cartography*, **14**(3), 175-187.

Hahn, M. (1989) Automatic Measurement of DTM by Means of Image Matching Techniques, *Proceedings of the 42nd Photogrammetric Week*, Stuttgart University, Germany.

Halounova, L. (1995) Comparison of Neural Networks and Maximum Likelihood Classifications in an Urban Area, *Sensors and Environmental Applications of Remote Sensing*, Askne(ed.), 1995 Balkema, Rotterdam, ISBN 90 5410 5240, 463-468.

Hannah, M.J. (1979) Computer Matching Areas in Stereo Images, *AI Laboratory Stanford University Memo*, **AIM-239**, 1974.

Haralick, R.M. *et al.* (1973) Textural Features for Image Discrimination: A Statistical and Conditional Probability Study, ***Remote Sensing of Environment*, 1**, 131-142.

Hebb, D. (1949) *Organisation of Behaviour*, John Wiley and Sons, New York.

Hecht-Nielsen, R. (1987) Counterpropagation Networks, ***Proceedings of the IEEE First International Conference on Neural Networks,*** 1987.

Henderson, R.L., Miller, W.J. and Grosch, C.B. (1979) Automatic Stereo Reconstruction of Man-made Targets, *SPIE Proceedings, Digital Processing of Aerial Images*, **186** , 240-248.

Heerman, P.D. and Khazenie, N. (1992), Classification of Multispectral Remote Sensing Data using a Back-Propagation Neural Network, *IEEE Transactions on Geoscience and Remote Sensing*, **30**, 81-88.

Hinton, G.E. (1985) Learning in Parallel Networks, *Byte,* April 1985, 265-273.

Hopfield, J. and Tank, D. (1985) Neural Computation of Decisions in Optimisation Problems, *Biological Cybernetics*, **52**, 147-152.

Hsieh, Y.C. McKeowen, D. and Perlant, F. (1990) Recovering 3D Information from Complex Aerial Imagery, *Proceedings of the Image Understanding Workshop,* September 1990, 670-691.

Hsieh, Y.C. McKeowen, D. and Perlant, F. (1990) Performance Evaluation of Scene Registration and Stereo Matching for Cartographic Feature Extraction, *Technical Report Carnegie Mellon University,* CMU-CS-90-193, November 1990.

Huertas, A. and Nevatia, R. (1988) Detecting Buildings in Aerial Images, *Computer Vision, Graphics and Image Processing,* **41,** 131-152

Irvin, R.B. and McKeown, D.M. (1989) Methods for Exploiting the Relationship between Buildings and their Shadows in Aerial Imagery, *IEEE Transactions on Systems, Man and Cybernetics,* November / December 1989, **19**(6), 1564-1575.

James, M. (1987) *Pattern Recognition,* BSP Professional Books, Oxford.

Jenson, J.R. (1981) Urban Change Detection Mapping using Landsat Digital Data, *The American Cartographer*, **8**(2), 127-147.

Jones, W.P. and Hoskins, J (1987) Back-Propagation: A generalised Delta Learning Rule, *Byte,* October 1987, 155-162.

Julesz, B. (1960) Binocular Depth Perception of Computer-Generated Patterns, *Bell Systems Technical Journal,* **39**, 1125-1162.

Julesz, B. (1963) Towards the Automation of Binocular Depth Perception (AUTOMAP-1), *Proceedings of the IFIPS Congress,* Munich 1962 (ed. C.M. Popplewell), Amsterdam, Holland.

Kanellopoulos, I. (1992) Land-cover Discrimination in SPOT HRV Imagery using an Artificial Neural Network - A 20 Class Experiment, *International Journal of Remote Sensing*, **13**(5), 917-924.

Kanellopoulos, I. Wilkinson, G.G. and Megier, J. (1993), Integration of Neural Networks and Statistical Image Classification for Land Cover Mapping, *Proceedings of the International Geoscience and Remote Sensing Symposium*, Tokyo, Japan, August, 1993, 511-513.

Key, J. Maslanik, J.A. and Schweiger, A.J. (1989), Classification of Merged AVHRR and SMMR Arctic Data with Neural Networks, *Proceedings of the International Geoscience and Remote Sensing*, **55**, 1331-1338.

Key, J. Maslanik, J.A. and Schweiger, A.J. (1990), Neural Network vs. Maximum Likelihood Classifications of Spectral and Textural Features in Visible, Thermal and Microwave Data, *Proceedings of the International Geoscience and Remote Sensing Symposium*, College Park MD, May, 1990, 1277-1280.

Kohonen, T. (1984) *Self-Organisation and Associative Memory,* Springer-Verlag, Berlin.

Kosko, B. (1987) Constructing an Associative Memory, *Byte,* September 1987, 137-144.

Kosko, B. (1988) Bi-directional Associative Memories, *IEEE Transactions Systems, Man, Cybernetics,* **L8**, 49-60.

Lacina, W.B. Nicholson, W.Q. (1979) Passive Determination of 3D Form from Dynamic Imagery, *SPIE,* **186**, 178-189.

Liebes, (1981) Geometric Constraints for Interpreting Images of Common Structure Elements: Orthogonal Trihedral Vertices, *SPIE,* **281**.

Liow, Yuh-Tay and Pavlidis, T. (1990) Use of Shadows for Extracting Buildings in Aerial Images, *Computer vision, Graphics and Image Processing*, **49**, 242-277.

Lippmann, R.P. (1987) An Introduction to Computing with Neural Nets, *IEEE ASSP Magazine*, April 87, 4-22.

Liu, Z.K. and Xiao, J.Y. (1991) Classification of Remote-Sensed Image Data using Artificial Neural Networks, *International Journal of Remote Sensing*, **12**(11), 2424-2439.

Li, H. Liu, Z. and Sun, W. (1993), A New Approach to Pattern Recognition of Remote Sensing Image using Artificial Neural Networks, *Proceedings of the International Geoscience and Remote Sensing Symposium*, Tokyo Japan, August, 1993, 713-715.

Logan, T.L., Strahler, A.H. and Woodcock, C.E. (1979) Use of a Standard Deviation Based Texture Channel for Landsat Classification of Forest Strata, *Machine Processing of Remotely Sensed Data Symposium*, Purdue University, Indiana, 395-404.

Lukes, G.E. (1981) Computer Assisted Photointerpretation Research at USAETL, *SPIE*, **758**, 172-176.

Machuca, R. and Gilbert, A.L. (1979) Finding Edges in Noisy Scenes, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI 3(1), 103-111, January 1981.

Marr, D. and Poggio, T. (1976) Co-operative Computation of Stereo Disparity, *Science*, N.Y., **194**, 283-287.

Marr, D. and Poggio, T. (1979) A Computational Theory of Human Stereo Vision, *Proceedings of the Royal Society London*, **204**, 301-328.

Mather, P.M. (1985) A Computationally Efficient Maximum Likelihood Classifier Employing Prior Probabilities for Remotely Sensed Data, *International Journal of Remote Sensing*, **6**(2), 369-376.

Mather, P.M. (1987) *Computer Processing of Remotely Sensed Images*, John Wiley and Sons, Chichester.

Maurer, H. (1974) Quantification of Textures-Textural Parameters and their Significance for Classifying Agricultural Crop Types from Colour Aerial Photographs, *Photogrammetria*, **30**, 21-40.

McCulloch, W. and Pitts, W. (1943) A Logical Calculus of the Ideas Immanent in Nervous Activity, *Bulletin of Mathematical Biophysics,* **5,** 115-133.

McKeown, D.M. (1984) Digital Cartography and Photo Interpretation from a Database Viewpoint, *Department of Computer Science Carnegie-Mellon University, New Applications of Databases,* ISBN 0-12-2755502, 19-42.

McKeown, D.M. Havery, W.A. and McDermott, J. (1985) Rule Based Interpretation of Aerial Imagery, *IEEE Transactions on Pattern Analysis and Machine Intelligence,* PAMI7(5), 570-585, September 1985.

McKeown, D.M. (1990) Toward Automatic Cartographic Feature Extraction, *Mapping and Spatial Modelling for Navigation,* NATO ASI Series, **F65, 149-180.**

McKeown, D.M. (1991) Feature Extraction and Image Data for GIS, *Proceedings of R.S.S. Conference, Spatial Data 2000,* Oxford University, 17-20, September 1991.

McKeown, D.M. Harvey, W.A., Wilson, L.E. and Wixson, L.E. (1989) Automated Knowledge Acquisition for Aerial Image Interpretation, *Computer Vision, Graphics and Image Processing,* **46,** 37-81.

Minsky, M.L. and Papert, S. (1969) *Perceptrons: An Introduction to Computational Geometry*, M.I.T. Press, Cambridge, M.A.

Mohan, R. Medioni, G. and Nevatia, R. (1989) Stereo Error Detection, Correction and Evaluation, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **2**, 113-120.

Mohan, R. and Nevatia, R. (1989) Using Perceptual Organisation to Extract 3-D Structures, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **11**(11), November 1989, 1121-1139.

Morris, A.C., Stevens, A. and Muller, J.P. (1988) Ground Control Determination for Registration of Satellite Imagery using Digital Map Data, *Photogrammetric Record*, **12**(72), 809-822.

Muller, J-P. (1988) *Digital Image Processing in Remote Sensing*, (Computing Issues in Digital Image Processing in Remote Sensing), Taylor and Francis, London, 1988.

Muller, J-P. (1988), Key Issues in Image Understanding in Remote Sensing, *Phil. Transactions Royal Society, London*, **324**, 381-395.

Muller, J-P. (1992) Global Topography and Surface Roughness from Automated Stereo Matching in Understanding the Terrestrial Environment In: Mather, P.M. (1992) *TERRA-1 Understanding the Terrestrial Environment - The Role of Earth Observations from Space,* Taylor and Francis, London, 1992

Muller, J-P., Collins, K.A., Otto, G.P. and Roberts, J.B.G. , (1988) Stereo Matching using Transputer Arrays, *Invited Paper: ISPRS 16th Congress, Kyoto, Japan,* 27(b3), 559-586, July 1-12 1988.

Nishihara and Larson (1979) Towards a Real-time Implementation of the Marr and Poggio Stereo Matcher, SPIE, **281**.

Otto, G.P. (1988) Rectification of SPOT Data for Stereo Image Matching, *International Archives of Photogrammetry and Remote Sensing,* 27(B3), 635-645.

Otto, G.P. and Chau, T.K.W. (1988) A "Region Growing" Algorithm for Matching of Terrain Images, *Proceedings of the Fourth Alvey Vision Conference University of Manchester,* 31 August - 2 September 1988, 123-128.

Paola, J.D. and Schowengerdt, R.A. (1995) A Review and Analysis of Backpropagation Neural Networks for Classification of Remotely-Sensed Multi-spectral Imagery, *International Journal of Remote Sensing*, **16**(16), 3033-3058.

Parker, (1982) Lemma Logic, *Paper*, Stanford University, 1982.

Peacegood, G. (1989) Stereo Matching of Aerial Photography - Data Capture and Restitution, *Alvey Deliverable M22d2ps Internal Report, Department of Photogrammetry and Surveying*, February 1989.

Perlant, F.P. and McKeown, D.M. (1990) Scene Registration in Aerial Image Analysis, *Photogrammetric Engineering and Remote Sensing*, **56**(4), 481-493, April 1990.

Perlant, F.P. and McKeown, D.M. (1990) Improved Disparity Map Analysis through the Fusion of Monocular Image Segmentation, *Proceedings of IAPR Workshop on Multisource Data Integration in Remote Sensing*, College Park, MD June 14-15, 1990, NASA Conference Publication.

Pollard, S.B. (1985) Identifying Correspondences in Binocular Stereo, *Unpublished PhD Thesis*, University of Sheffield, 1985.

Pollard, S.B. Mayhew, J.E.W. and Frisby, J.P. (1985) A Stereo Correspondence Algorithm using a Disparity Gradient Limit (PMF), *Perception*, **14(4), 449-470.**

Pridmore, T., Mayhew, J.E.W. and Frisby, J.P. (1990) Exploiting Image-plane Data in the Interpretation of Edge-based Binocular Disparity, *Computer Vision, Graphics and Image Processing*, **52**, 1-25.

Rosenblatt, F. (1957) The Perceptron: A Perceiving and Recognising Automaton, Cornell Aeronautical Laboratory, Cornell University.

Rumelhart, D.E. and McClelland, J.L. (1986) eds. *Parallel Distributed Processing*, MIT Press, Cambridge, MA.

Rumelhart, D.E. (1986) Learning Internal Representations by Errors Propagation, In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volume 1: Foundations*, edited by Rumelhart, D.E. and McClelland, J.L. (Cambridge, MA: The MIT press), 318-362.

Singh, A. and Shneier, M (1990) Grey Level Corner Detection: A Generalisation and a Robust Real Time Implementation, *Computer Vision, Graphics and Image Processing*, **51**, 54-69.

Trivedi, M. and Harlow, C.A. (1985) Identification of Unique Objects in High Resolution Aerial Images, *Optical Engineering,* **24**(3), 502-506.

Tsui, H.T. and Chan, C.K. (1989) Hough Technique for 3D Object Recognition, *IEE Proceedings,* **136E**(6), 565-568.

Webster (1979) Layered Relaxation Network for Object Detection, *SPIE Image Understanding Systems,* **205.**

Wechler, H. and Citron, T. (1980) Feature Extraction for Texture Classification, *Pattern Recognition,* **12,** 301-311.

Werbos, P. (1974) Beyond Regression: New Tools for Prediction and Analysis in the Behavioural Sciences, *Unpublished PhD Thesis,* Harvard University, Cambridge MA.

Wolf, P.R. (1974) *Elements of Photogrammetry,* McGraw-Hill, New York.

BIBLIOGRAPHY

The following references were used as background material and are not specifically cited in the text.

Arsenault, H.H., Hsu, Yuan-Neng. and Yang, Y. (1982) Incoherent Method for Rotation-Invariant Recognition (Ground Scene Image Processing for Air Photography), *Applied Optics* , **21**(4), 610-615.

Bartelt, H., Edl, P., Gotz, J. and Lohmann, A.W. (1985), Pseudo Stereo Images, *Journal of the Optical Society of America* , **2**(3), 386-392, March 1985.

Gega, J.V. Le Thach, C. and Sailer, C. T. (1992), Neural Net Spectral Pattern Recognition, *SPIE, Neural and Stochastic Methods in Image and Signal Processing*, **1766**, 609-618.

Gibson, G.J. and Cowan, C.F.N. (1990), On the Decision Regions of Multilayer Perceptrons, *Proceedings of the IEEE*, **78**, 1590-1593.

Hepner, G.F., Logan, T., Ritter, N. and Bryant, N. (1990), Artificial Neural Network Classification using a Minimal Training set: Comparison to Conventional

Supervised Classification, *Photogrammetric Engineering and Remote Sensing*, **56**, 469-473..

Inoue, A., Fukue, K., Shimoda, H. and Sakata, T. (1993), A Classification Method using Spatial Information Extracted by Neural Network, *Proceedings of the International Geoscience and Remote Sensing Symposium*, Tokyo, Japan, August, 1993, 893-895.