THE DESIGN AND IMPLEMENTATION OF A PURELY DIGITAL STEREO-PHOTOGRAMMETRIC SYSTEM ON THE IBM 3090 MULTI-USER MAINFRAME COMPUTER

By

ALI AZIZI

A Thesis Submitted for the Degree of
Doctor of Philosophy (Ph.D.)
in Photogrammetry

UNIVERSITY OF GLASGOW

March 1990

To

My father

A C K N O W L E D G E M E N T S

The author wishes to express his sincere gratitude to his supervisor, Professor G. Petrie, for suggesting this research topic and for his continuous encouragement and advice throughout the period over which this research was carried out. Indeed without his continuous help and supervision, especially during the writing-up stage, this project could not be as complete as it is.

The author is also extremely grateful to his other supervisor, Mr. B.D.F. Methley, for his help during the initial writing-up stage of the work and for his continuous advice and encouragement.

The author's sincere thanks also go to his research colleague, Mr. Zhilin Li, for his suggestion to use the PANACEA contouring package for testing the accuracy of the generated data and for all his constructive discussions. Sincere thanks are also due to his research fellow students Mr. A. Abdallah and Mr. S. AlKazan.

Special thanks go to Mr. D.A. Tait for his help and encouragement. The help received from the other Topographic Science staff members is also gratefully acknowledged, in particular, that from Mr. I. Gordon and Mr. J. Shearer.

The freedom to use the Departmental facilities granted by the previous and the present Heads of the Department, Professors J. Tivy and I. Thompson is gratefully acknowledged.

Thanks are also due to the following individuals:

The Chief Technician, Mr. I. Gerrard, and his assistant, Mr. L. Hill, for their assistance in photographing the 5080 graphics system and some of the figures in this thesis, and to the Senior Cartographer, Mr. M. Shand, for his help in printing some of the diagrams used in this thesis.

The staff in the Computing Centre for their help and advice, in particular, Dr. P. Rosenberg for his help with the vectorization of the programs.

Dr. H. MacGillivray and Dr. Q. Parker for digitizing the analogue aerial photos with the COSMOS scanning microdensitometer located at the Royal Observatory in Edinburough.

Sincere thanks are also due to author's friends, in particular, Mr. K. Eftekhari, Mr. S. Handoyo, Dr. A. Elniweiri, Dr. M. Disfani, Mr. M. Mowlazadeh and Mr. M. Sadeghi.

The author would also like to express his deep gratitude to his mother, sister and brothers who financially supported him; their encouragement and support never failed.

Finally, the author is extremely grateful to his wife, Mrs. Zohreh Azizi, for her patience and assistance through all the stages of this research work. Indeed, without his wife's substantial help in drawing the diagrams and typing some parts of the thesis, this work would not easily have come to an end. In addition, her continuous encouragement, which had a great impact on the progress of the work, never stopped.

S U M M A R Y

This thesis is concerned with an investigation into the possibilities of implementing various aspects of a purely digital stereo-photogrammetric (DSP) system on the IBM 3090 150E mainframe multi-user computer. The main aspects discussed within the context of this thesis are:-

(i) Mathematical modelling of the process of formation of digital images in the space and frequency domains.

(ii) Experiments on improving the pictorial quality of digital aerial photos using Inverse and Wiener filters.

(iii) Devising and implementing an approach for the automatic sub-pixel measurement of cross-type fiducial marks for the inner orientation, using the Gradient operator and image modelling least squares (IML) approach.

(iv) Devising and implementing a method for the digital rectification of overlapping aerial photos and the formation of the stereo-model.

(v) Design and implementation of a digital stereo-photogrammetric system (DSP) and the generation of a DTM using visual measurement.

(vi) Investigating the feasibility of stereo-viewing of binary images and the possibility of performing measurements on such images.

(vii) Implementing a method for the automatic generation of a DTM using a one-dimensional image correlation along epipolar lines and experimentally optimizing the size of the correlation window.

(viii) Assessment of the accuracy of the DTM data generated both by the DSP and the automatic correlation method.

(ix) Vectorization of the rectification and correlation programs to achieve higher speed-up factors in the computational process.

CONTENTS

CHAPTER ONE

Introduction

1.1 Background and Statement of the Problem

The recent improvements in the speed, size and cost effectiveness of digital computers have had a major impact on the processing of large sets of numerical data. These improvements have encouraged those who deal with large and continuously varying quantities to convert these functions into digital (discrete) form. This allows the data to be processed by digital computers. In the field of electronic engineering, for example, the power and basic simplicity of digital signal processing over analogue methods is so great that "whenever possible the present analogue systems are being converted to an equivalent digital power" (Hamming, 1983). In other fields, digital manipulation and processing of the data has also found a significant role. Examples are: computer vision and pattern recognition for the robot industry; the detection of cracks and flaws in machine parts from industrial radiographs; the formation and enhancement of bio-medical imageries, including radiographs, thermographs, and nuclear scanned images; motion detection; object detection and feature classification and extraction using satellite images, etc. (Rosenfeld and Kak, 1976; Pratt, 1978; Castleman, 1979; Moik, 1980; Mather, 1987).

Photogrammetrists were also amongst those who realised the potential power inherent in a digital representation of functions (i.e. images). As early as the 1960's, pioneering work was carried out by Sharp (Sharp et. al, 1965); this was followed by independent work carried out by Helava and Masry who investigated the correlation of two overlapping digitized aerial photos. This was done using electronic scanning along epipolar lines for automatic measurement of the x-parallaxes (Helava and Chapelle, 1972; Masry, 1974).

The availability of satellite imagery, acquired directly in digital form from the mid 1970's onward, and the gradual and rapid advancements in computer technology, further accelerated the interest of photogrammetrists to shift from analogue to digital techniques. These interests were focused mainly on DTM generation from digitized

aerial photos using automatic correlation algorithms (Scarano and Brumm, 1976; Wolf and Dewitt, 1982; Claus, 1984); fully automatic digital rectification and orthophoto production (Keating and Boston, 1979; Konecny, 1979; Wiesel, 1985); and automatic measurement of signalized points, fiducial marks and control points (Thurgood and Mikhail, 1982; Zhou, 1986).

There is yet another line of research in utilizing the digital image, initially suggested by Sarjakoski (1981). The idea is to replace the analogue images with digital data and hence introduce the possibility of substituting the mechanical parts of the plotters with the computer's capability to carry out image processing and scrolling. In other words, the idea is to produce a stereo-plotter with no mechanical components for the movements of the images. The idea would eventually lead to a new generation of photogrammetric plotter called the **Digital Stereo-photogrammetric Plotter** or DSP (Cogan et. al., 1988).

The various investigations mentioned above, clearly demonstrated the fact that all photogrammetric operations are equally applicable to the domain of the digital image. This view has led to a new era of so-called **digital image photogrammetry**. Helava (1988b) has reviewed this field, including the development of a fully automated process for the interior and exterior orientation of photographs; the generation of DTMs and orthophotos for topographic mapping applications; and the determination of three-dimensional coordinates of objects in close range photogrammetry for industrial applications. He considers these to be well within the state of the art of digital image photogrammetry. Regarding the implementation of a DSP, there are clear signs that such a system can fully realize all the features of analogue and analytical plotters (Gugan and Dowman, 1986) and therefore, a DSP can also be regarded as well within the state of the art of digital image photogrammetry.

However, it has been pointed out (Torlegård, 1986) that digital image photogrammetry is still in its infancy, even though some systems are

already in experimental use. The major problems are as follows[1]

(a) The **data storage requirements** for digital images are massive. For example, a single aerial photo digitized with a 20 μm pixel size contains 132,250,000 pixels. Konecny (1979) estimates that a standard 8-track CCT, 720 metres long can store the image data from a standard-format (23X23 cm) metric photograph digitized with 25 μm pixel size. On the other hand, a standard 24-track high density digital tape (HDDT), 2,800 metres long, can store 900 such photographs with a 12.5 μm pixel size. However, these are held sequentially as a form of mass storage. Turning to the fast access type of storage device required during actual computer operations, an 88 Mbytes disk is required to store a single photograph with a 25 μm pixel size (Petrie, 1983).

The above mentioned problem is directly connected to the pixel size of the digital image. Hence, the storage requirement may be reduced dramatically if the digital image has a relatively large pixel size. Unfortunately this has the consequence of bringing about a reduction in the information contained in the original photograph as well as introducing the problem of **sub-pixel measurement**.

(b) The **CPU times** required for the digital processing of images including automatic DTM generation, are at present excessive; consequently this makes the operation costly and often uneconomic.

(c) There is the problem of **automatic image correlation** for metric information extraction in an area with low signal content; and in terrain with geometrical complexities such as break lines, etc.

(d) Complications arise in the detection, recognition and isolation of different features (e.g. roads) because of their non-deterministic behaviour. This makes the fully automatic realization of feature

1 *A purely digital solution is considered here as opposed to a digital analytical system, which is a hybrid solution to photogrammetric problems.*

extraction an impractical task at the present time.

(e) At present, there is no direct digital camera which can match the resolution and fidelity of a film camera. Therefore, a digitizing process for the acquisition of digital images seems inevitable. The digitizing process cannot be error-free; errors which originate from imperfections of the mechanical and optical construction of image digitizers and the finite dimensions of pixels, reduce the achievable accuracy.

Regarding the non-automatic (manual) aspect of digital photogrammetry, the technical problems of real-time image roaming and scrolling and other similar operations seem to have been coped with (Gugan and Dowman, 1986) and thus, as stated above, a DSP is well within the state of the art. However, regarding the manual measurements carried out on a display device and the density range requirement of display devices, the following points are worth mentioning:

(i) The achievable accuracy of pointing to the finite size pixel of an image displayed on the display device using a cursor or cursors with monoscopic or stereoscopic viewing, still requires further investigation.

(ii) The digital image is almost a continuous tone image and consequently needs a display device with a range of memory buffers commensurate with the density range of the image. It is interesting and worthwhile to assess the effect that the reduction of the density ranges of the display device has on metric accuracy and stereo-viewing perception. In particular, it is interesting to explore the potential application of the recently developed algorithms for producing binary images to photogrammetric applications such as stereo-viewing and measurement.

In the present project, an attempt has been made to implement different procedures required for a fully digital solution of

photogrammetric operations. In order to exhibit the potential of such a system, a powerful mainframe computer with vector facilities has been used. The use of such a machine also helps to cope with some of the problems mentioned above. A mainframe-based system offers the two distinct advantages of large storage capacity and speed --the latter produced by the vector hardware of the machine. Unfortunately, it also has a disadvantage in delaying interaction with the photogrammetrist due to the multi-user environment of the machine.

1.2 Research Objectives

At the University of Glasgow, the existence of the Kelvin Project for numerically intensive computation on the IBM 3090-150E mainframe machine with vector facility, provided a unique opportunity for the author to investigate the feasibility of devising and implementing a mainframe-based digital image photogrammetric system and hence to be able to investigate some of the problems mentioned above. The efforts are concentrated on certain of the automation and manual aspects mentioned above. On the **automation** side, target measurement with a view to sub-pixel mensuration; digital rectification of overlapping aerial photos and model formation, hence producing a 3-dimensional viewing of the image on the display unit; and an improvement in the CPU time required for one-dimensional image correlation along epipolar lines have all been investigated. On the **manual** side, the achievable accuracy for monoscopic and stereoscopic measurements carried out on a high-resolution display device, and the effect of using binary images in stereo-viewing and measurement have been investigated. To carry out stereoscopic measurements on the digital images, a digital stereo-photogrammetric system has been designed and implemented using a general purpose image processing system (IAX) available on the IBM 3090 mainframe computer. This has also allowed an investigation into the feasibility of using a general purpose multi-user machine for manually generating a DTM and contour lines.

In total, the following features of digital image photogrammetry have been covered in the present project:

- Mathematical modelling of the imaging process in the space and frequency domains.
- Some analysis of digital pictorial rectification in the frequency domain.
- Automatic measurement of fiducial marks using the Gradient operator and correlation techniques.
- Digital rectification of overlapping aerial photos and model formation.
- Stereo-viewing of the left and right rectified digital images on a display device.
- Design and implementation of a digital stereo-photogrammetric system to generate a DTM and subsequently derive contour lines.
- Automatic generation of a DTM using correlation algorithms applied in one dimension to the digitally rectified images.
- Vectorization of the rectification and correlation programs to achieve higher speed-up factors in the computational process.
- And finally assessing the accuracy of all measurements carried out in this project.

Thus the objectives of the thesis have three main dimensions:

1. Examining the **automation aspects** of digital image photogrammetry for target measurement, y-parallax and associated x-parallax removal for the entire model, DTM generation, and the vectorization potential offered by the IBM 3090 to speed up the operation.
2. Considering the **manual operations** of digital image photogrammetry, the following features have been examined:
   The feasibility of implementing a purely digital stereo-photogrammetric system to generate a DTM and contour lines in a multi-user environment and the effect of using binary images generated by a so-called "error diffusion function algorithm", on stereoscopic viewing and stereoscopic measurement.

3. Assessing the **achievable accuracy** for the generated data both automatically and manually; in particular, the accuracy of the measurements carried out directly using cursors on the displayed digital image (i.e. on the data generated by the DSP).

1.3 Outline of the Thesis

There are ten chapters in addition to this introduction. Apart from Chapter 10, each chapter covers a single phase of a whole cycle (system). It starts with the first phase of the formation of the digital image which has inherent pictorial degradations and metric displacement and distortions. This will be followed by the second and third phases of the pictorial and metric rectifications respectively. The fourth phase deals with the model reconstruction and finally the cycle ends with the last stage of DTM generation. These related sequential phases may be represented as a system with the terrain as input and the model and the generated DTM (i.e the reconstructed terrain) as the output. This is shown schematically as follows (Fig. 1.1):

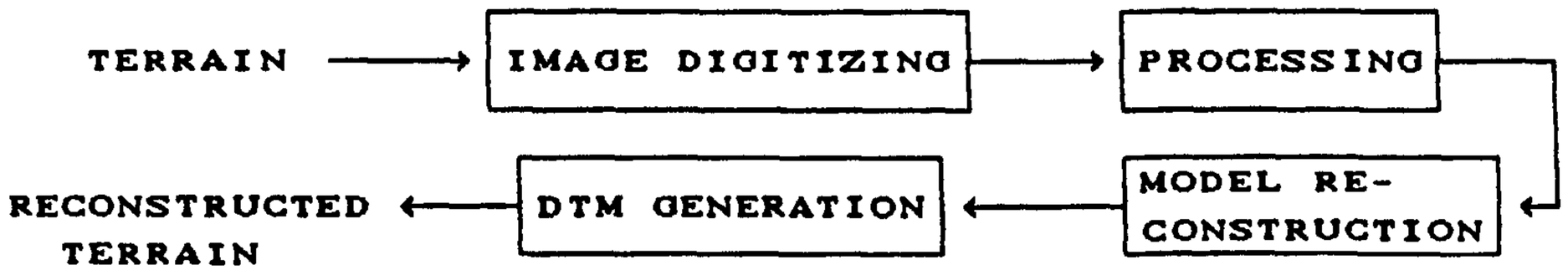


Fig. 1.1

More specific descriptions of the chapters are as follows:

- **Chapter 2** deals with the mathematical preliminaries and mathematical modelling of the digital image formation using the linear system theory, considering both space and frequency domains, followed by a discussion about the pictorial degradations and the metric displacements and distortions which are imposed on the digital image by the imaging camera and the image digitizer.

- As mentioned in Section 1.1, film cameras reign supreme for image recording at present and hence in this project, the digital data are acquired by digitization of a complete pair of aerial photos using an accurate scanning micro-densitometer. Therefore, **Chapter 3** is devoted to the actual procedures required for image digitization and the construction of the microdensitometer instruments used for this process. This is followed by a description of the flatbed microdensitometer used in this project for image digitization and the structure of the recorded data.
- Having acquired a digital image with inherent pictorial and metric degradations, **Chapter 4** deals with the correction of the pictorial degradations using the frequency domain approach.
- Geometric displacements with a view to digitally removing the y-parallaxes and associated x-parallaxes from the entire model, are dealt with in **Chapter 5**. This chapter covers the procedures involved in measuring the fiducial marks; in analytical rectification; and in image resampling using different interpolation techniques. This is followed by the description of the cross target measurement and rectification programs developed for these purposes. The chapter concludes with the presentation of the rectified images.
- Having produced a pair of rectified left and right images free of y-parallax and associated x-parallax, in **Chapter 6**, the reconstruction and measurement of the model for DTM generation by designing and implementing a purely digital stereo-photogrammetric system using the IAX general purpose image processing package in conjunction with the IBM 5080 graphics display system, is presented and discussed. The chapter ends by presenting the generated DTM and contour lines and the assessment of the accuracy of the generated data.
- **Chapter 7** describes the stereoscopic measurements carried out on binary images using the implemented digital stereo-photogrammetric system. This is followed by the

presentation of the results of the measurement.

- **Chapter 8** is devoted to the theoretical basis and the state of the art of digital image correlation.
- The implementation of automatic image correlation techniques carried out in this research work both for target measurement and DTM generation and the accuracy estimation of the output data are presented in **Chapter 9**.
- After an introduction to parallel and vector computers, the vectorization of the rectification and the correlation programs is presented in **Chapter 10**. The achieved CPU times for image rectification and correlation are given and the results are compared with those acquired using the scalar mode.
- Finally **Chapter 11**, the closing chapter, draws some general conclusions regarding the present state of digital image photogrammetry, and makes recommendations for future work.

CHAPTER TWO

Mathematical Modelling of the Digital Image Formation

2.1 Introduction

Photogrammetry deals with measurements of a three-dimensional model formed from overlapping photographs. When a photograph is made, the lens collects the electromagnetic energy emitted or reflected by objects in the object space and focuses it on the image plane. One can say, therefore, that the energy appearing on the image is a function of the three-dimensional object e(X,Y,Z) being photographed. Therefore, the image may be regarded as a two-dimensional light intensity function f(x,y), where the value of the function at the position defined by coordinates x,y gives the intensity of the image at that point. However, the main theme of this thesis, as mentioned in Chapter 1, is to manipulate and handle this function (i.e. the recorded image) by a digital computer in order to perform different photogrammetric tasks. Therefore a necessary precursor to the computerized manipulation of the function f(x,y) is the conversion of this function into its corresponding digital form g(x,y).

As we shall see in the following sections, and also in Chapter 3, this conversion is carried out by means of a so-called "microdensitometer" device. Clearly, the recorded image of the three-dimensional object does not have the same geometrical and intensity characteristics as the incoming electromagnetic radiation which is emitted or reflected by the object. Neither does the image, converted into digital form, have the exact geometric and intensity relationship with the recorded image f(x,y). Using the "Information Theory" approach, the camera and its accessories and the microdensitometer may be thought of as a "system" or systems that exhibit some sort of "response" when a "stimulus" is applied. The stimulus (e.g. the analogue image) is referred to as the input to the system (e.g. the microdensitometer system), whereas the response is known as its output (e.g. the digitized image).

However, formation of the analogue image by the camera imaging system is not the main concern here; instead, the primary concern of this chapter is to obtain some detailed insight into the procedures involved in the formation of a digital image and some of the related mathematical aspects from the "Information Theory" point of view.

Hence the main objective is to develop a theoretical basis for the chapters that follow later.

In this chapter, answers are sought to the following questions:-

- what is a digital image and how is it formed ?
- how can the digitization process be described mathematically ?
- how can the image digitizer interact with the input and produce the output ?
- under what conditions can the pictorial information of the aerial photo be fully transformed into digital form and how can the original input image be reconstructed from the digitized image ?
- what kind of distortions are transferred to or created in the digitized aerial photos ?

2.2 Digital Image Definition

As mentioned earlier, analogue images are not directly amenable to computer analysis. Since computers work with numerical rather than pictorial data, an image must be converted into numerical form before processing. A numerical or **digital image** is essentially a two-dimensional matrix defined by:

$$f(x,y) = \begin{bmatrix} f(0.0) & f(0,1) & \dots & f(0.N-1) \\ \vdots & \dots & \vdots & \dots & \vdots \\ f(M-1,0) & f(M-1,1) & \dots & f(M-1,N-1) \end{bmatrix} \qquad (2.1)$$

The above MXN array of numbers represents what is called a digital image, and each element of the matrix is called an **image element**, (or a **picture element**, or a **pel**, or a **pixel**). Each pixel in the array represents a **density value** or a **grey level** corresponding to a point in the analogue image.

For a digital image, the origin of the coordinate system is defined in the upper left corner of the digital image matrix, which agrees with the commonly used conventions for matrix element indexing. It is worth noting that this is different from common photogrammetric practice.

Thus the coordinates for pixel f(m,n) located at the row m and column n of the digital image represented by the matrix in Eq. 2.1 have discrete values given by,

$$\begin{cases} x(m,n) = x_o + D_x\, n \\ \\ y(m,n) = y_o + D_y\, m \end{cases} \qquad (2.2)$$

where x(m,n), y(m,n) are the x and y coordinates of the pixel f(m,n); D_x and D_y represent the digitizing interval; and x_o, y_o are constant values representing the coordinates of the origin of the image.

A digital image is produced by means of **digitization** which is discussed in the following section.

2.2.1 Digitization

The word "digitization" as applied in this thesis has the following meaning:

> Digitization is the process of converting a two-dimensional continuous function (in this case the analogue image) into digital (discrete) form.

Since continuous tone images have continuity both in position (i.e. a plane with continuous variation of x and y) and in tone (i.e. the continuous grey level variation), therefore, to digitize, it is necessary to discretize these two aspects of the image, namely the position and the density. In other words, digitization consists of **sampling** the grey level in an analogue image at an MXN matrix of points, and of **quantizing** the continuous grey levels at the sampled points into Q usually uniform intervals. The finer the sampling and the quantization, the better the approximation of the original image. Thus the aim of sampling and quantization is to represent a continuous tone image by an array of numbers (samples), such that a comparable quality of image can be reconstructed from the samples.

Image digitization may be accomplished by means of scanning micro-densitometers (sometimes referred to as scanners or scanning digitizers or simply as image digitizers). The scanning digitizers

will scan systematically through a hard-copy negative image or a diapositive using a sampling spot to isolate a small part of the image for intensity measurement. The sampled data are then recorded on magnetic tape for subsequent processing. A detailed description of the principle and construction of microdensitometers will be given in Chapter 3.

Sampling and quantization are discussed in more detail in the following sections.

2.2.1.1 Sampling

As mentioned before, for generating a digital image, the pixels are collected at intervals D_x and D_y. Usually $D_x = D_y = P$, where P is normally a constant called the **sampling interval**. In practice, each pixel is often assumed to represent an area equal to D_x X D_y. The size of the pixel is therefore assumed to be the same as the sampling interval.

The choice of the sampling interval (i.e. the pixel size) plays an essential role in the transformation of the information from the analogue image into digital form. If the sampling interval is too long (resulting in a large pixel size), some image resolution (image information) is lost. The image is then said to be **under-sampled** (Fig. 2.1a). More details regarding the under-sampling problem are given in Section 2.7.8. If the sampling interval is too short (i.e. there is too small a pixel size), then too many digital samples are made and the image is said to be **over-sampled**.

2.2.1.2 Quantization

As mentioned above, the grey levels of a continuous tone image must be divided into discrete values in a digital image. In most of the cases, uniform quantization levels are used. Each quantized pixel is represented by a binary word. If a natural binary code is used and the word length is b bits, then the number of quantization levels Q is:

$$Q = 2^b \qquad (2.3)$$

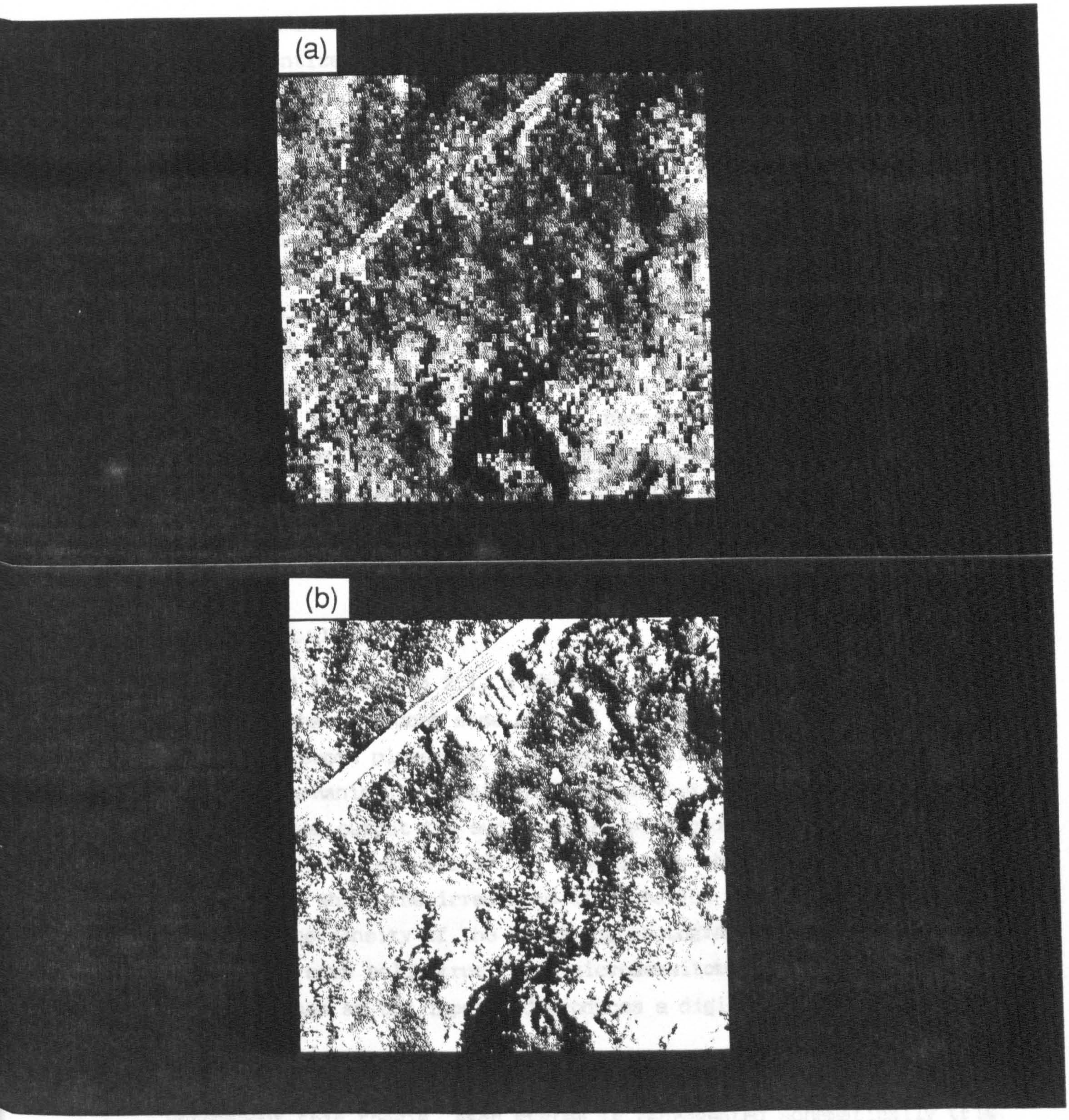


Fig.2.1 2.1a Effect of reducing sampling interval.
2.1b Effect of reducing quantization levels.
(compare these with the original image given in Fig. 2.11(a1)).

The word length is usually chosen to be 6, 7, or 8 bits, resulting in 64, 128, or 256 quantization levels, respectively (Moik 1980). When an image is quantized with an inadequate number of bits, the image will have a degradation in its grey levels (see Fig. 2.1b). Clearly, if an image is quantized with a larger number of bits than required, there will be some unnecessary density values which appear as noise.

From what was mentioned in this and the preceding sections, it can be concluded that, when an analogue image is digitized, two critical decisions concerning the sampling interval (pixel size) and the quantization level should be made in order to have a digital image with the minimum amount of grey level degradation. For aerial photos, 256 quantization levels are considered to be adequate. To find out the optimum sampling interval for transforming the full information from analogue to digital form, it is necessary first to derive a mathematical expression for the image digitization process from which the sampling theorem can be derived. In order to model the imaging process mathematically, the camera and the image digitizer are viewed as a linear system. This is discussed in the following sections.

2.3 Physical Systems

As mentioned in section 2.1, a physical system may be thought of as any device that produces some sort of response when a stimulus is applied. A record player is an example of such a system, the input being the mechanical vibrations of the stylus in the grooves of the record and the output the audio signal produced by the loudspeakers. A camera is another example of a system (in this case, an optical system), for which the irradiance distribution of some object is the input and the density of the resulting photographic negative is the output. The same holds true for a microdensitometer which receives the analogue image as the input and produces a digital image as output.

Irrespective of the inherent physical structure of a system, throughout this thesis, each system is represented schematically (Fig. 2.2) as a box with input f(x,y) and output g(x,y),

Fig. 2.2

or simply as,

$$f(x,y) \longrightarrow g(x,y)$$

Systems are normally classified into **linear** and **non-linear** systems. Linear systems are also categorized into **shift-variant** and **shift-invariant** systems. The camera and microdensitometers may be regarded, with a good approximation, as linear shift-invariant systems, and are considered as such in the sections to follow. The analysis of non-linear and the shift-variant linear systems do not lie within the scope of this thesis.

The next section will give the definition and a brief introduction to the properties of the linear shift-invariant system. It will then be demonstrated that, with some approximations, the camera and micro-densitometer systems may be regarded as a linear shift-invariant system. Finally it will be shown how the behaviour of such systems can be modelled mathematically, which makes it possible to reconstruct the original input using the known behaviour of the system.

2.3.1 Linear Shift-Invariant (LSI) System

The word linear suggests something pertaining to a straight line relationship. Thus, a system which mainly satisfies the following requirement is referred to as a linear system: (Gabel and Roberts, 1973).

1. In the one-dimensional case, if the input f(x) gives rise to the output g(x), then af(x) gives rise to ag(x). i.e.,

$$f(x) \longrightarrow g(x) \qquad (2.4a)$$

then

$$af(x) \longrightarrow ag(x) \qquad (2.4b)$$

This property, called **homogeneity**, is a property of all linear systems.

2. A linear system must also possess the property of **superposition**. That is, if (see Fig. 2.3)

$$f_1(x) \longrightarrow g_1(x)$$

and

$$f_2(x) \longrightarrow g_2(x)$$

then

$$f_1(x)+f_2(x) \longrightarrow g_1(x)+g_2(x) \qquad (2.5)$$

Combining Relations 2.4b and 2.5 yields

$$af_1(x)+bf_2(x) \longrightarrow ag_1(x)+bg_2(x) \qquad (2.6)$$

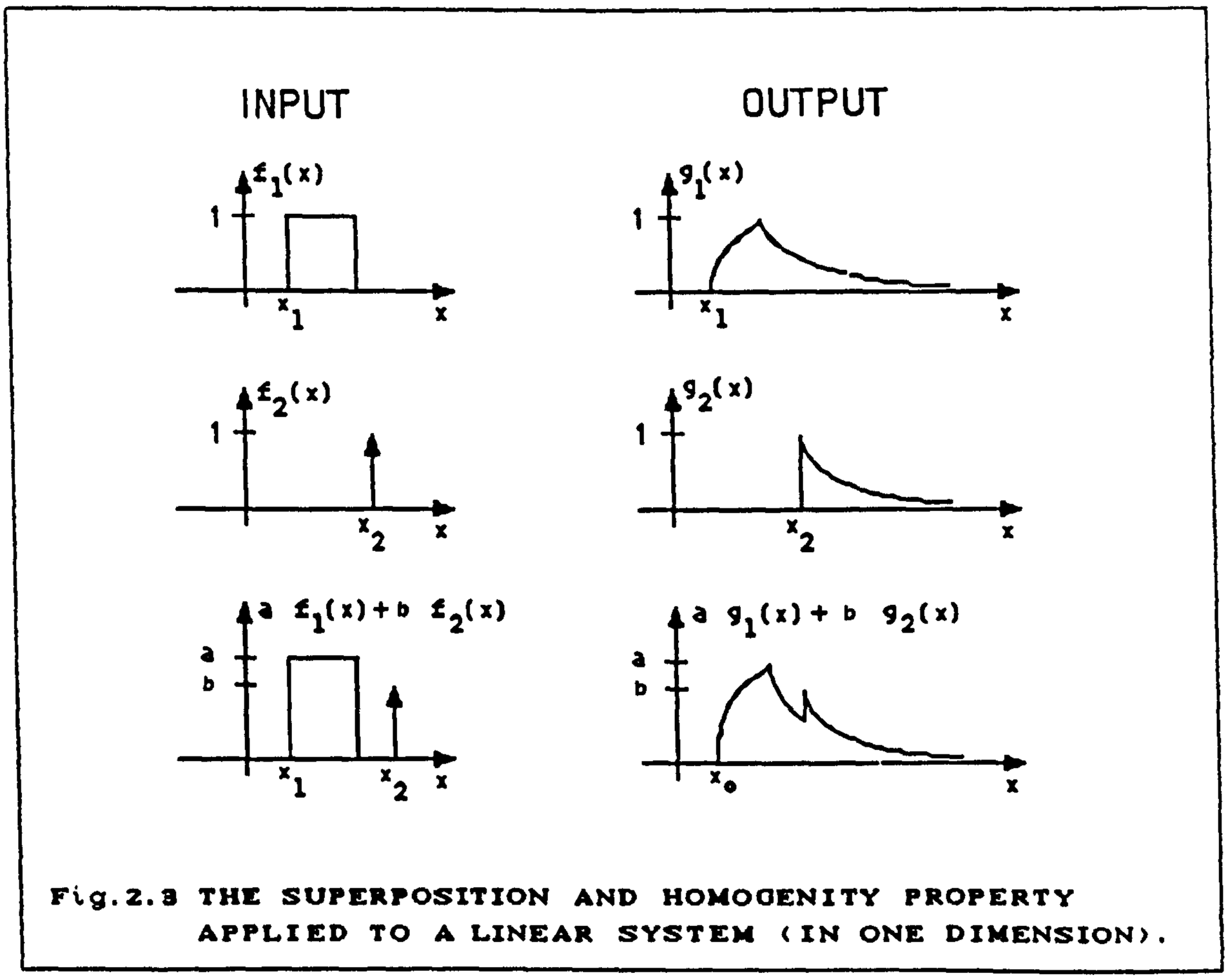


Fig.2.3 THE SUPERPOSITION AND HOMOGENITY PROPERTY APPLIED TO A LINEAR SYSTEM (IN ONE DIMENSION).

Thus a system is linear if and only if superposition and homogeneity hold. A convenient notation for the arrow of Relation 2.4a is to use functional notation and represent the transformation of input into output by the operator $\mathbb{T}$, i.e.,

$$g(x) = \mathbb{U}\left\{f(x)\right\}$$

and similarly for two-dimensional functions,

$$g(x,y) = \mathbb{U}\left\{f(x,y)\right\} \tag{2.7}$$

A system is said to be shift-invariant (fixed, stationary, time invariant, space invariant, isoplanatic) if the only effect caused by a shift in the position of the input is an equal shift in the position of the output (Gaskill, 1978). In other words, if a system is shift-invariant and if (Fig. 2.4),

$$g(x) = \mathbb{U}\left\{f(x)\right\}$$

then

$$g(x-x_o) = \mathbb{U}\left\{f(x-x_o)\right\}$$

for two dimensional functions,

$$g(x-x_o,y-y_o) = \mathbb{U}\left\{f(x-x_o,y-y_o)\right\}$$

where x_o and y_o are real constants. Thus the magnitude and shape of the output are unchanged as the input is shifted along the x and y axes; only the location of the output is changed.

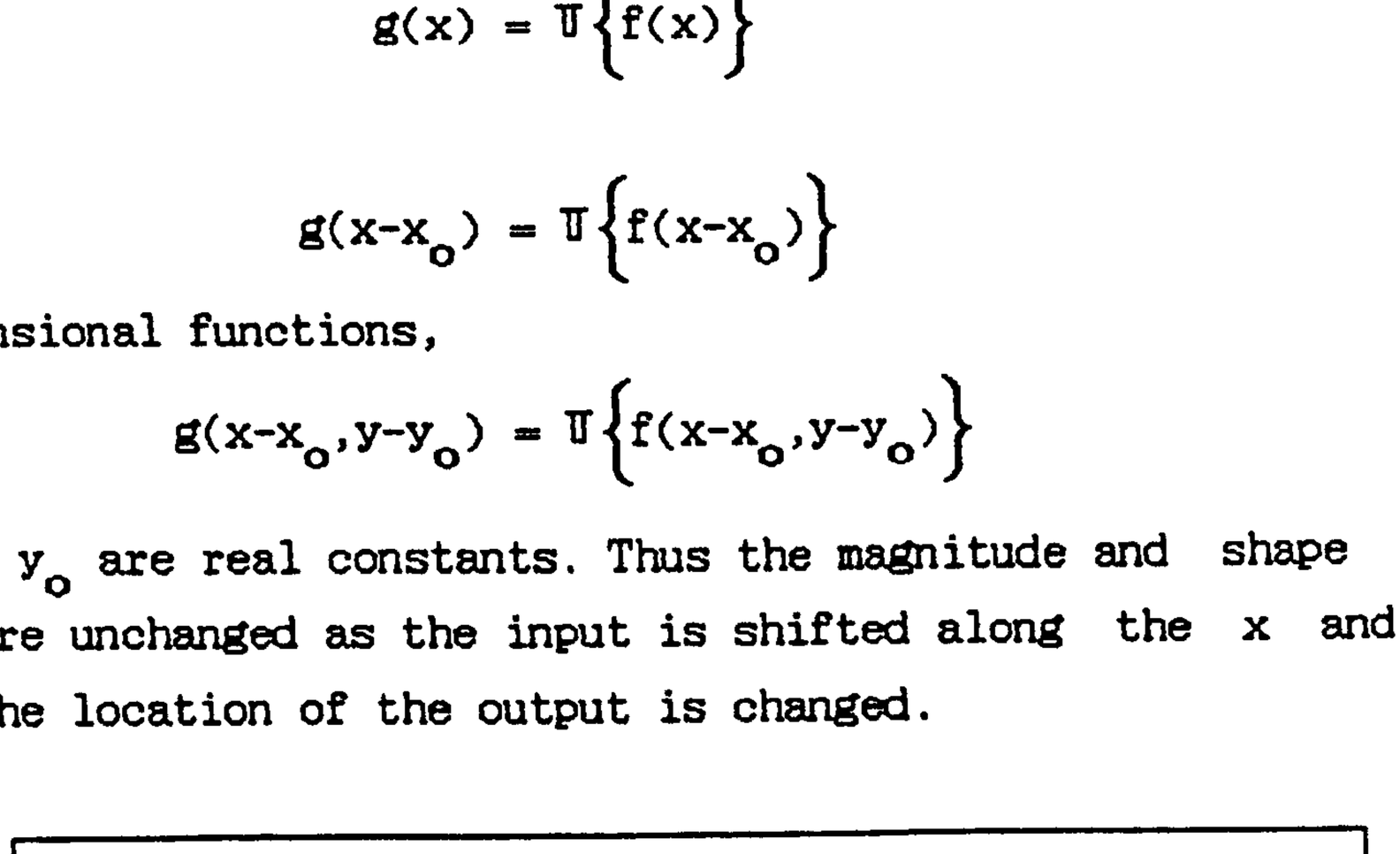


Fig.2.4 INPUT-OUTPUT RELATIONSHIP FOR A SHIFT-INVARIANT SYSTEM (IN ONE DIMENSION).

Having in mind these definitions presented above, next an attempt can be made to investigate and find whether or not the camera and microdensitometer can be regarded as an LSI system.

2.3.2 Camera and Microdensitometer as an LSI System

To investigate whether the camera can be regarded as an LSI system, first the homogeneity property will be considered. Suppose a is the image of a small object A on the ground. Now if the object A is replaced by the object B which reflects the light n times as high as the object A, then, in an ideal case, b the image of the object B will also have an intensity n times higher than the image a, i.e. if,

$$a = \mathbb{U}\{A\}\ ,\quad b = \mathbb{U}\{B\}\quad \text{and}\quad B = nA\ ,$$

then

$$na = \mathbb{U}\{nA\}$$

Similarly, it can be shown that, in an ideal case, the superposition property also holds true, i.e.

$$a+b = \mathbb{U}\{A+B\}$$

It can also be demonstrated easily that in an ideal case (i.e. using a diffraction-limited (aberration-free) lens and for any reasonable off-axis distance, etc.) the camera is a shift-invariant system, i.e., if A is shifted with an amount equal to x_o, then

$$a+x_o = A+x_o$$

where the image scale is 1. Thus essentially the shape of the image undergoes no change.

It can be shown similarly that, in an ideal case, the microdensitometer is an LSI system. However, in reality, there are different factors which influence the imaging and digitizing procedures. For the camera, these are mainly the atmospheric effect which varies in time, the sensitivity variation and non-linear response of the emulsion, etc., and for the microdensitometer, they include the non-linear response of the photo cell, etc. These factors affect the linearity of the system so that the system characteristic as a whole varies in time. Nevertheless, as far as aerial cameras and reasonably well designed microdensitometers are concerned, the system

non-linearities are negligible, and to a good approximation, the camera and the microdensitometer may be regarded as a linear system. This may be represented by the following diagram (Fig. 2.5):

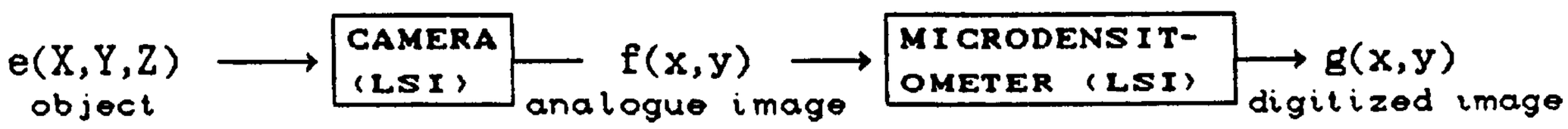


Fig. 2.5

The situation for satellite imagery acquired with Multi- Spectral Scanners (MSS) and linear array devices is different in that the atmospheric effect, the sensor non-linear response and hence the noise, etc., become more pronounced than those encountered in aerial photography. In these situations, non-linear and non-deterministic systems may be used for more accurate modelling of the behaviour of such systems.

2.4 Digital Image Formation (Mathematical Approach)

Having in mind the assumption of the microdensitometer as an LSI system, in the next sections, an attempt is made to model the characteristics of the microdensitometer mathematically and to subsequently form the digital image.

2.4.1 Sampling Spots

Consider a two-dimensional function f(x,y) representing the analogue image. After digitization, this function will be transformed into another function g(x,y). In an ideal case (i.e. having a perfect digitizing microdensitometer), the function g(x,y) would be the exact duplication of the function f(x,y). In other words, if the sampling spot had an infinitely small dimension with the intensity of unity, and if the number of such sampling spots was equal to the analogue image resolution, then the microdensitometer would behave as a linear system which multiplies the input image by unity and hence gives the output as an exact replica of the input (see Sections 2.4.3 and 2.6.1). Such an ideal sampling spot is called the "**Dirac delta function or impulse**" $\delta(x)$ and can be modelled as the limit of a narrow

rectangular pulse (Castleman, 1979),i.e.,

$$\delta(x) = \lim_{a \to 0} \frac{1}{a} \prod \left(\frac{x}{a} \right) \qquad (2.8)$$

where $\prod$ denotes a rectangular pulse (Fig. 2.6(a1), see Section 9.2.1). In Eq. 2.8, as a becomes smaller, the pulse becomes narrower but taller to maintain unit area. In the limit, the pulse becomes infinitely tall with infinitesimally small width and with the area equal to unity (Fig. 2.6(a2)). i.e.,

$$\delta(x) = \begin{cases} \infty , & x = 0 \\ 0 , & \text{otherwise} \end{cases} \qquad (2.9)$$

The **shifted impulse** $\delta(x-u)$ is defined as, (Fig. 2.6(a3))

$$\delta(x-u) = \begin{cases} \infty , & x = u \\ 0 , & \text{otherwise} \end{cases} \qquad (2.10)$$

Relations 2.9 and 2.10 imply,

$$\int_{-\infty}^{\infty} \delta(x)dx = \int_{-\varepsilon}^{\varepsilon} \delta(x)dx = \int_{-\infty}^{\infty} \delta(x-u)dx = \int_{u-\varepsilon}^{u+\varepsilon} \delta(x-u)dx = 1. \qquad (2.11)$$

where ε is an infinitesimally small limit of integration. The impulse in two dimensions is given by:

$$\int_{-\varepsilon}^{\varepsilon} \int \delta(x,y)dxdy = 1 \qquad (2.12)$$

Clearly, the Dirac delta function does not exist in reality. Instead, the sampling spots have a finite size with varying intensity. Such sampling spots may be approximated by a sinc^1 or sinc^2 functions (Figs. 2.6b and 2.6c). Strictly speaking, the sinc function occurs on the axis of an aberration-free optical system at perfect focus; diffraction effects due to the wave nature of light are always present and are beyond our control. Hence sometimes the sinc function is

---

1 *The sinc function (pronounced "sink") is defined as,*
$$sinc(x) = sin(\pi x)/(\pi x)$$
The square of this function is referred to as the $sinc^2$ *function.*

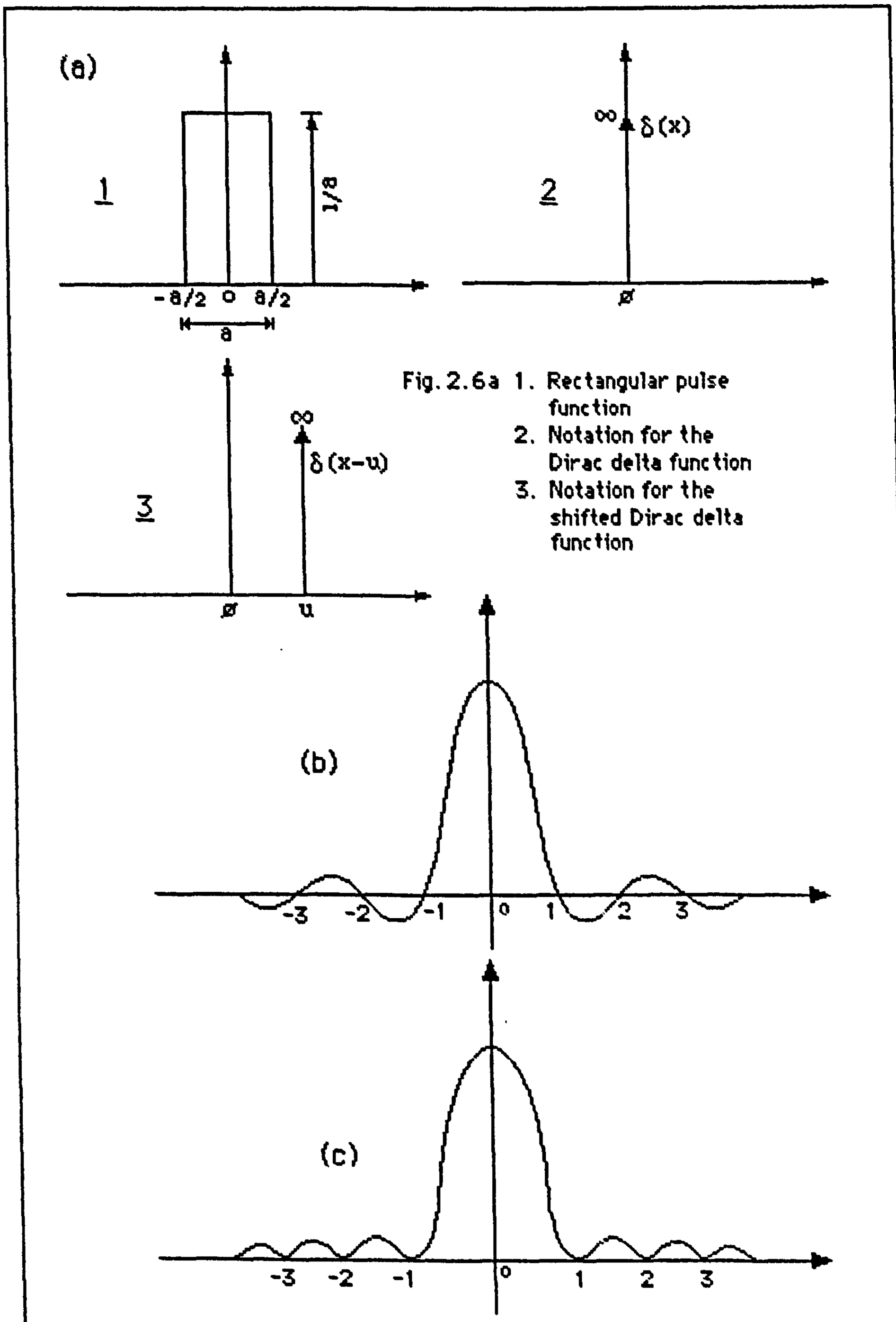


Fig. 2.6a 1. Rectangular pulse function
2. Notation for the Dirac delta function
3. Notation for the shifted Dirac delta function

Fig. 2.6b Sinc function

Fig. 2.6c $Sinc^2$ function

referred to as an ideal sampling spot and is usually used to model the optical system. However, many electro-optical image digitizers function with a Gaussian spot which has the property that the intensity at a distance r from the centre is

$$I(r) = I(0) \exp -(r/k)^2 \qquad (2.13)$$

where I(0) is the intensity in the centre, k is a constant, and r is the radial distance from the centre (Helava. 1982).

In the next section, it will be demonstrated how a knowledge of the sampling spot is used to model the behaviour of the linear system; subsequently it will be shown how the system interacts with the input to produce the output.

2.4.2 Point Spread Function

The effects of the physical system during the imaging process, such as optical diffraction -- resulting in the sampling spots such as those mentioned in the preceding section -- as well as technical imperfections such as aberrations of the camera lens, lack of perfect focusing, and graininess of the film, etc., produce jointly and separately what is called a **point spread function (psf)**. In other words, the psf is the image of a "perfect point" or "point source" (the Dirac delta function). If any arbitrary picture f(x,y) could be considered to be a sum of point sources, then the knowledge about the output of an input point source could be used to determine the output for the function f(x,y). For example, if an image digitizer with a Gaussian scanning spot is used, the image of a point source will appear as a finite sized circle exhibiting a Gaussian intensity variation. This finite circle has the same size and grey level characteristics as the scanning spot itself. For this reason, the psf is also called an **impulse response**. This is discussed in more detail later and represented mathematically by Eq. 2.16. Hence, a knowledge of the mathematical function representing the characteristics of the scanning spot may be used as a psf of the image digitizer system. It can be used to determine the output for any particular input. Alternatively a knowledge of the input and output may be used to determine the psf. This latter case is explained in more detail in

Section 4.8.

One can, therefore, say that the psf represents the mathematical behaviour of the system. Hence, in Eq. 2.7, the operator $\mathbb{T}$ may be thought of as the psf of the LSI system.

2.4.3 Convolution Operation

The next intention is to demonstrate how the psf of a LSI system interacts with the input to give the output. This interaction may be expressed mathematically by means of a so-called **convolution** operation (also called composition product, superposition integral, running mean, Duhamel integral, etc.). As will be shown in the following paragraph, the output of an LSI system is given by the convolution of the input with the system's psf (impulse response).

The convolution of two real-valued, two-dimensional functions f and h, is defined by,

$$g(x,y) = \int_{-\infty}^{\infty}\int f(u,v)h(x-u,y-v)dudv \qquad (2.14)$$

or using the usual shorthand notation for convolution operation:

$$g = f * h$$

where * denotes the convolution operation.

Equation 2.14 may be viewed simply as one of finding the area of the product f(u,v) and h(x-u,y-v) as x and y are allowed to vary systematically. Note that for x=0 and y=0, the function h(0-u,0-v) is merely h(u,v) rotated 180° about its origin (mirror image) and that h(x-u,y-v) is translated so as to move the origin of h to the point x,y (Gaskill,1978; Castleman,1979). The function h is shifted along the x and y directions while it retains its shape which clearly demonstrates an LSI system.

Equation 2.14 may be thought of as a mathematical model expressing, for instance, the digitization carried out by a microdensitometer, where h represents the microdensitometer's sampling spot; and f and g

are the input and output images respectively.

Fig. 2.7 demonstrates the result of the convolution of a two-dimensional rectangular function with a Gaussian psf. Fig. 2.7a and 2.7b represent the original input and the psf respectively and the resulting output is given in Fig. 2.7c. Since the digitization process is always accompanied by noise, Fig. 2.7d represents the output contaminated with random noise. Fig. 2.7 as a whole is a simulated example of the digitization process modelled by the convolution operation.

A discrete two-dimensional convolution is given by,

$$g(i,j) = f * h = \sum_{m=0}^{K-1} \sum_{n=0}^{L-1} f(m,n)h(i-m,j-n) \qquad (2.15)$$

where

$$i= K , K+1 ,..., M-1 ; j= L , L+1 ,..., N-1$$

The dimensions of the input image f, the psf h and the output image g are as follows:-

$$f(1:M,1:N) , h(1:K,1:L) , g(1:M-K,1:N-L)$$

If the input to an image digitizer system is a "perfect point" (i.e. a Dirac delta function), the output of the system can be derived by the following convolution,

$$\int_{-\infty}^{\infty}\int \delta(u,v)f(x-u,y-v)dudv = f(x-u,y-u)\Big|_{u = 0,\ v = 0} = f(x,y) \qquad (2.16)$$

where f(x,y) denotes the system psf. As Eq. 2.16 shows, the output of the system at u=0 and v=0 is f(x,y). This means that the Dirac delta function is the identity function under convolution. For this reason, the characteristic function of the linear system (i.e. psf) is called the **impulse response** of the system (recall Section 2.4.2).

In addition to what was mentioned in the preceding paragraphs, the convolution operation has applications in digital filtering operations, feature enhancement, noise removal and other image

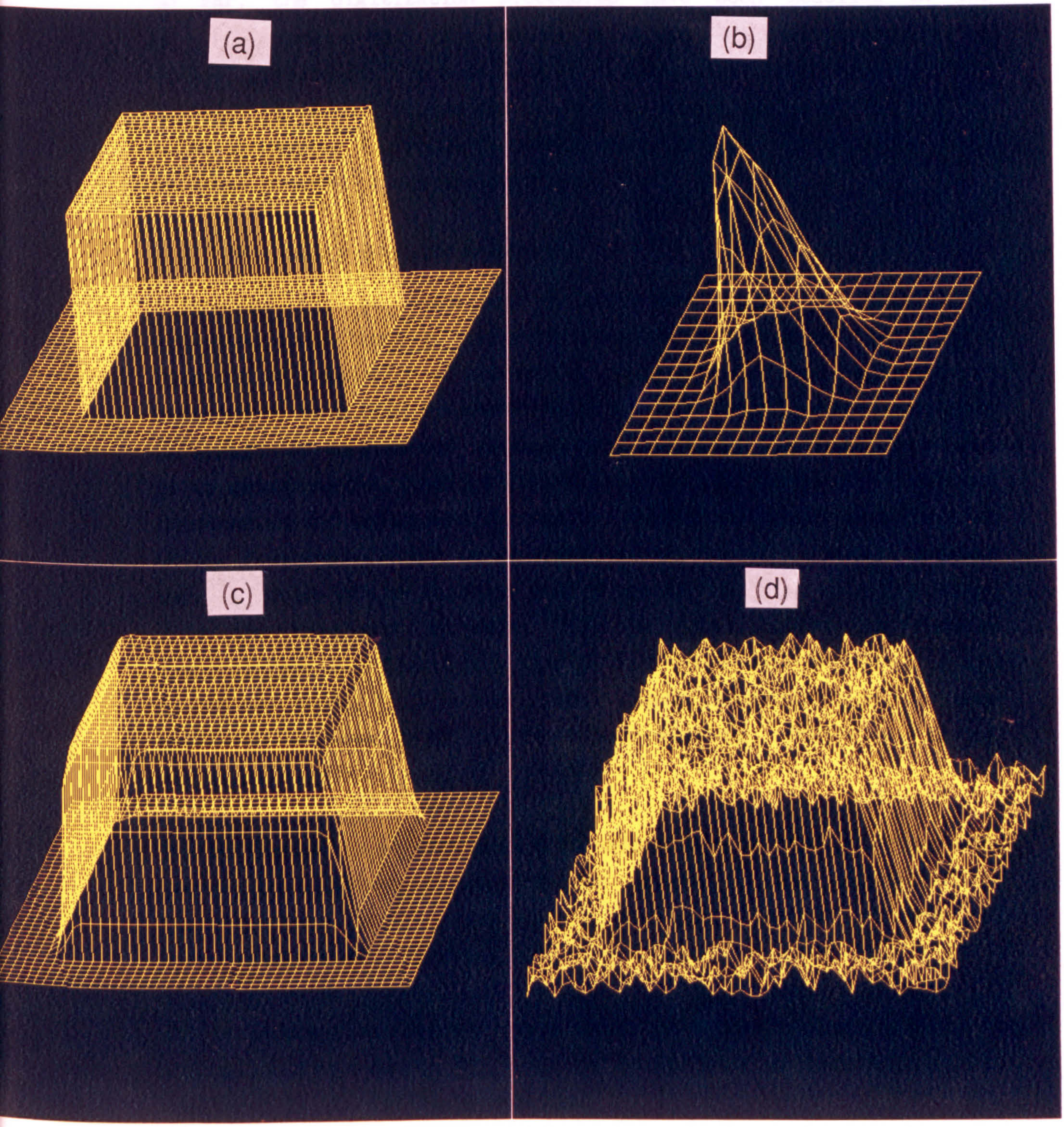


Fig.2.7 a. Two-dimensional rectangular function (input).
b. Gaussian psf.
c. The convolved output.
d. The output plus noise.

processing operations which will be discussed in detail in Chapter 4.

So far, the digitization procedures have been modelled in the so-called space domain (see below). It is now necessary to consider a second approach for the modelling of the digitization procedure. This approach is called "frequency, or Fourier domain analysis". As will be shown in Chapter 4, frequency domain analysis is a very powerful tool in image processing and analysis operations.

2.5 Frequency Domain Analysis

A function may be described in different ways. For example, a one-dimensional trace of a microdensitometer d=f(x) gives the density values d as a function of successive x coordinates. This approach is referred to as a **space domain** approach. The second method, which may be thought of as a second language for describing the same function, is to decompose the function f(x) into sinusoidal waves with different frequencies and amplitudes and expressing the amplitudes as a function of the frequencies. This is referred to as the **frequency domain** representation of the function f(x) which may be shown by F(V), where V is the frequency variable. F(V) is often called the Complex Temporal-frequency Spectrum or simply the **frequency spectrum** of f(x). As denoted above, f(x) is the space domain representation of some quantity, whereas F(V) is the corresponding frequency domain representation.

The motivation to perform such a transformation is that it allows more insight to be gained concerning the nature of the digital image formation (see Section 2.6.1). In addition, certain mathematical operations are greatly simplified (see Section 2.5.5 and Chapter 4).

The transformation from the space domain into the frequency domain is accomplished by means of the "**Fourier transform**" process which will be discussed in the following sections.

2.5.1 Periodic and Non-Periodic Functions

Before considering the Fourier transform, it is necessary to define

periodic and non-periodic functions. A **periodic function** f(x) has the property that, for all x values

$$f(x) = f(x+nT) \tag{2.17}$$

where n is an integer and T is a real positive constant known as the period of f(x) --see Fig.2.8c. From this expression, it is clear that a periodic function repeats itself exactly after fixed intervals of nT. The reciprocal of the period is called the **fundamental frequency** of the function, i.e., $V_o = 1/T$. The fundamental spatial frequency of a periodic function is measured in units of cycles/metre (or line pairs per millimetre "lp/mm"), as used to specify the resolution of the camera lens or the lens/film combination. An example of a periodic function is the sinusoidal function (Fig. 2.8b) given by:

$$f(x) = A\sin(2\pi V_o x - \theta)$$

where A is the amplitude; and θ denotes the phase shift.

Functions that do not satisfy the Equation 2.17 are called **non-periodic or aperiodic functions**, such as the Gaussian function (Fig. 2.8a).

Clearly, digital images belong to the non-periodic functions. However, it will be demonstrated in Section 2.5.3, that non-periodic functions can be decomposed into sinusoidal functions by means of the Fourier transform which is the topic of the next sections.

2.5.2 The Fourier Series

To realize the concept of the Fourier transform, it is convenient to start with a description of the **Fourier series**:

Provided it satisfies a set of so-called Dirichlet conditions, a periodic function f(x) with a period $T=1/V_o$, can be decomposed into a sum of sinusoidals known as a Fourier series,

$$f(x) = A_o/2 + \sum_{n=1}^{\infty} \left[A_n \cos(2\pi n V_o x) + B_n \sin(2\pi n V_o x)\right] \tag{2.18}$$

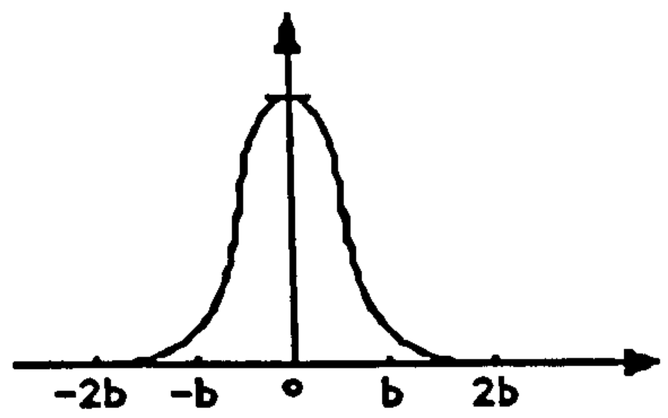


Fig. 2.8.a The Gausian function is a non-periodic function.

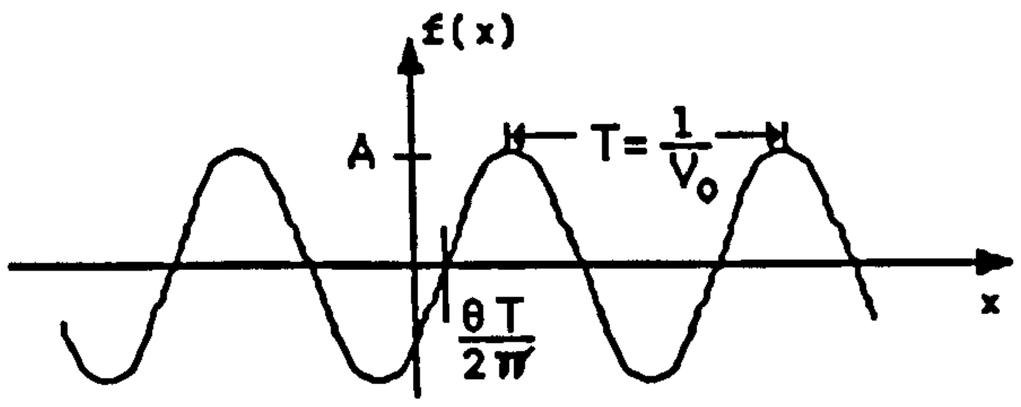


Fig. 2.8 b Sinusodical function of amplitude A, frequency V_o, and phase shift θ.

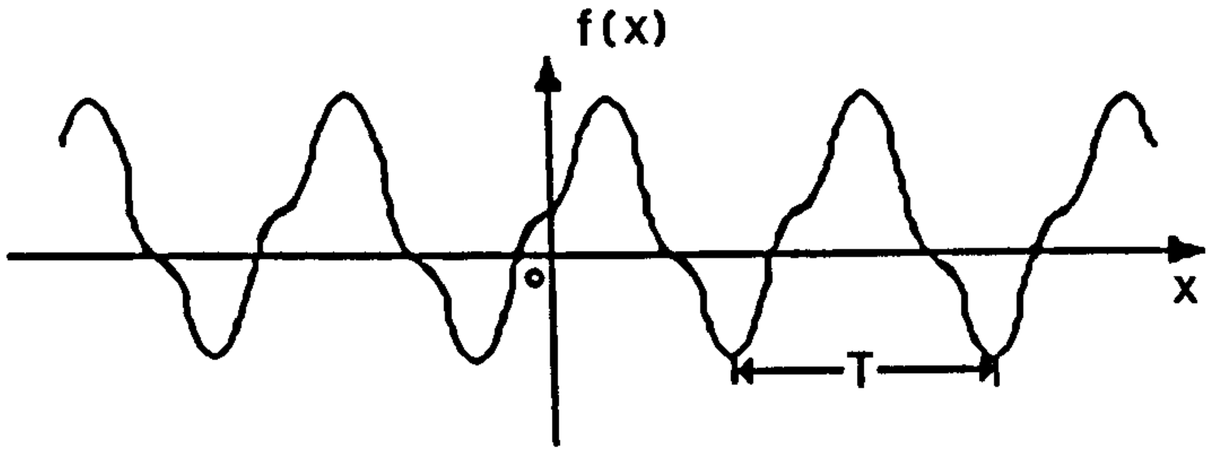


Fig. 2.8.c Periodic function of period T.

where

$$A_o = 2/T\int_{-T/2}^{T/2} f(x)dx$$

$$A_n = 2/T\int_{-T/2}^{T/2} f(x)\cos(2\pi nV_o x)dx$$

$$B_n = 2/T\int_{-T/2}^{T/2} f(x)\sin(2\pi nV_o x)dx$$

Consider the Euler formulae,

$$e^{i2\pi nV_o x} = \cos(2\pi nV_o x)+i\sin(2\pi nV_o x)$$

and

$$e^{-i2\pi nV_o x} = \cos(2\pi nV_o x)-i\sin(2\pi nV_o x)$$

where e is the base of the Naperian logarithm; and $i^2 = -1$.

These relations imply,

$$\cos(2\pi nV_o x) = 1/2\left[e^{i2\pi nV_o x} + e^{-i2\pi nV_o x}\right] \qquad (2.19)$$

and

$$\sin(2\pi nV_o x) = 1/2\left[e^{i2\pi nV_o x} - e^{-i2\pi nV_o x}\right] \qquad (2.20)$$

Substituting Eqs. 2.19 and 2.20 in 2.18,

$$f(x) = A_o/2+\sum_{n=1}^{\infty} 1/2(A_n+iB_n)e^{-i2\pi nV_o x} + \sum_{n=1}^{\infty} 1/2(A_n-iB_n)e^{i2\pi nV_o x}$$

and in more compact form (--see O'Neil, 1963),

$$f(x) = \sum_{n=-\infty}^{\infty} D_n e^{i2\pi nV_o x} \qquad (2.21)$$

where

$$D_n = 1/T \int_{-T/2}^{T/2} f(x) e^{-i2\pi n V_o x} \, dx$$

From Eq. 2.21, it can be seen that the Fourier series decomposes a function into a linear combination of complex exponentials with the complex valued coefficient D_n assigning the proper weight to each term. The terms of the series for n=±1 may be combined to form a sinusoidal function of frequency V_o, called the **fundamental component** of the original function. In general, the terms for n=±m comprise the mth harmonic component of the function, with frequency $\pm mV_o$. For smoothed and continuous functions, the coefficient D_n decreases in magnitude faster than discontinuous functions, such as a rectangular-wave function. This means that the smoother the function, the faster its Fourier series converges (Gaskill, 1978).

2.5.3 The Fourier Transform

A digital image, clearly, cannot be considered as a periodic function, since it cannot satisfy the Eq. 2.17. Therefore, the Fourier series cannot be applied to this kind of function. To represent a non-periodic function in terms of its sinusoidal components, the **Fourier transform** may be used. If f(x) represents a non-periodic function (e.g. a microdensitometer trace), and it satisfies the Dirichlet conditions, with the strengthened requirement that it must be absolutely integrable on the infinite interval (-∞ ,+∞), Eq. 2.21 may be written as the following integral:

$$f(x) = \int_{-\infty}^{\infty} F(V) e^{i2\pi V x} \, dv \qquad (2.22)$$

where V is the frequency variable; and

$$F(V) = \int_{-\infty}^{\infty} f(x) e^{-i2\pi V x} \, dx \qquad (2.23)$$

F(V), as mentioned before, is the frequency spectrum of f(x). It is also called the Fourier transform of f(x) and is usually shown by:

$$\mathbb{F}\left\{f(x)\right\} = F(V)$$

where $\mathbb{F}$ denotes the Fourier transform operator.

The inverse Fourier transform ($\mathbb{F}^{-1}$) of F(V) is defined by Eq. 2.22 and hence f(x) and F(V) are called Fourier transform pairs. In Eq. 2.22, F(V) plays the same role as the coefficients of the Fourier series D_n in Eq.2.21, assigning the proper weights to the various components, but, in general, it is a piecewise-continuous function of the frequency variable V (Gaskill, 1978).

The frequency spectrum of an arbitrary function f(x) is in general complex-valued, i.e.,

$$F(V) = R(V)+iI(V)$$

where R(V) and iI(V) denote the real and imaginary parts respectively. As a result, it is often advantageous to put F(V) in the form

$$F(V) = |F(V)|\exp\left[-i\phi(V)\right]$$

where $|F(V)|$ is known as the **amplitude spectrum** of f(x) and $\phi(V)$ is called its **phase spectrum**. Both $|F(V)|$ and $\phi(V)$ are real-valued functions and calculated as (Gaskill, 1978; Moik, 1980),

$$|F(V)| = \left[R(V)^2+I(V)^2\right]^{1/2} \quad , \quad \phi(V) = \tan^{-1}\frac{I(V)}{R(V)}$$

The Fourier transform of a non-periodic function exhibits the same important property as the Fourier series i.e., the smoother a function is, the more rapidly its transform will approach zero with increasing frequency. This fact is demonstrated in Figs. 2.9a, b and c which represent three functions in the space domain and their corresponding functions in the frequency domain. It can be seen that the frequency domain representation of the function given in Fig. 2.9a, approaches zero, with increasing frequency, more rapidly than the two other functions (Figs. 2.9b and 2.9c) which have coarser structures in the space domain. As we will see in Chapter 4, this property is a very important factor in digital filtering and noise removal procedures.

The Fourier transform of a (psf) is called a **transfer function**. Its

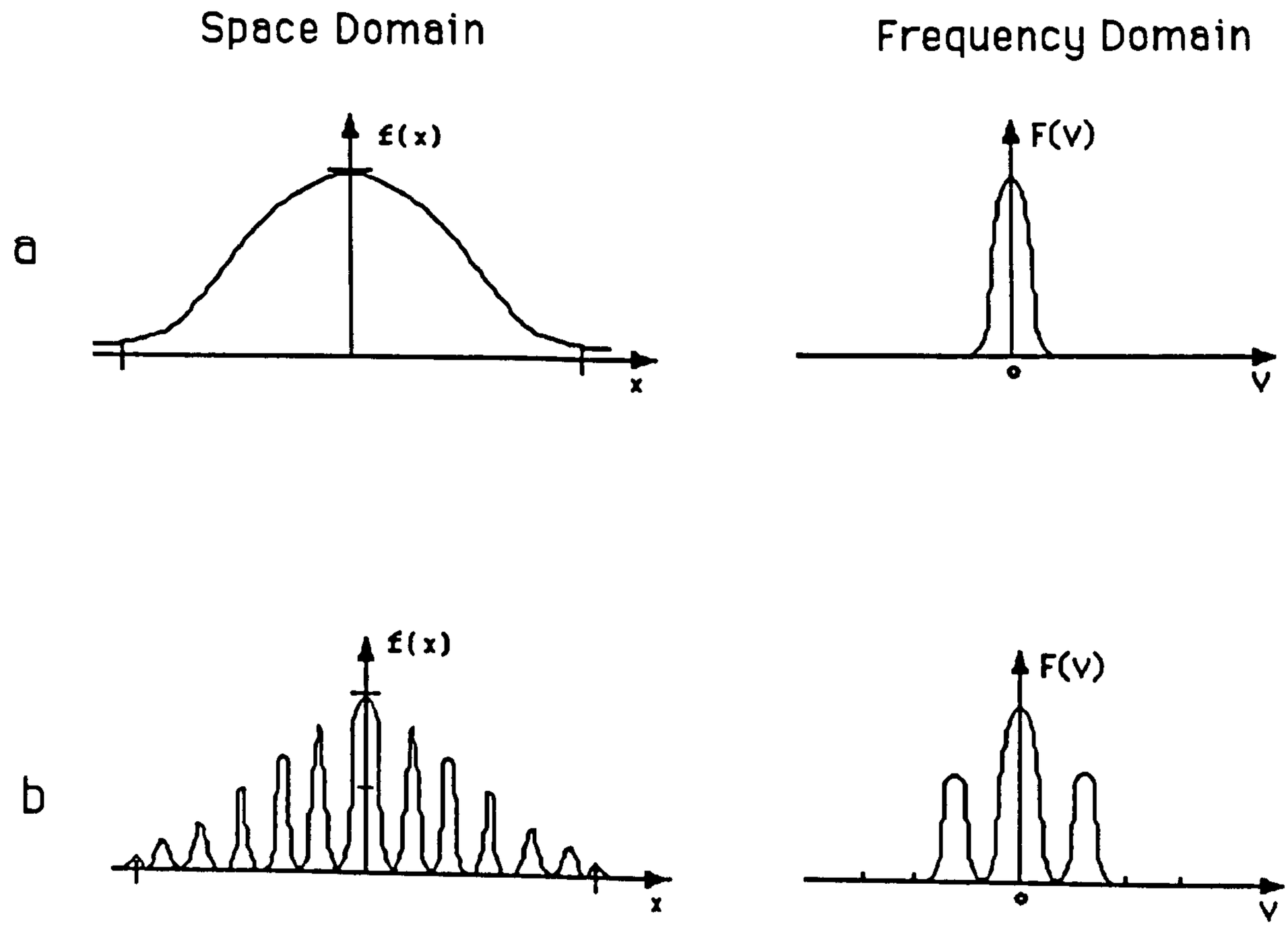


Fig. 2.9 Relationship between the fine structure of a function and the nature of its spectrum.

normalized form is usually known as the **Modulation Transfer Function (MTF)**. Hence the psf and the transfer function may be referred to as **Fourier transform pairs**.

The graphical representations of the Fourier transform of some frequently used functions in image processing (i.e., Gaussian, sinc, and $sinc^2$ functions) are given in Fig. 2.10. As the Figs. 2.10a, b and c show, the Fourier transform of a Gaussian function is also a Gaussian function; the Fourier transform of a $sinc^2$ function is a triangular function; and the Fourier transform of a sinc function is a rectangular function.

2.5.4 The Fourier Transform in Two Dimensions

The Fourier transform G(U,V) of a two-dimensional function g(x,y) is given by:

$$G(U,V) = \int_{-\infty}^{\infty}\int g(x,y)e^{-i2\pi(Ux+Vy)}\,dxdy \qquad (2.24)$$

where U and V are frequency variables along the x and y axes respectively.

For a digital image g(x,y) of the size N by M, a discrete Fourier transform may be applied as follows:

$$G(m,n) = \frac{1}{MN}\sum_{j=0}^{M-1}\sum_{k=0}^{N-1} g(j,k)\,\exp\left[-i2\pi\left(\frac{jm}{M}+\frac{kn}{N}\right)\right] \qquad (2.25a)$$

where m = 0,1,...,M-1 and n = 0,1,...,N-1

The discrete inverse Fourier transform is given by

$$g(j,k) = \frac{1}{MN}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} G(m,n)\,\exp\left[i2\pi\left(\frac{jm}{M}+\frac{kn}{N}\right)\right] \qquad (2.25b)$$

where j = 0,1,...,M-1 and k = 0,1,...,N-1.

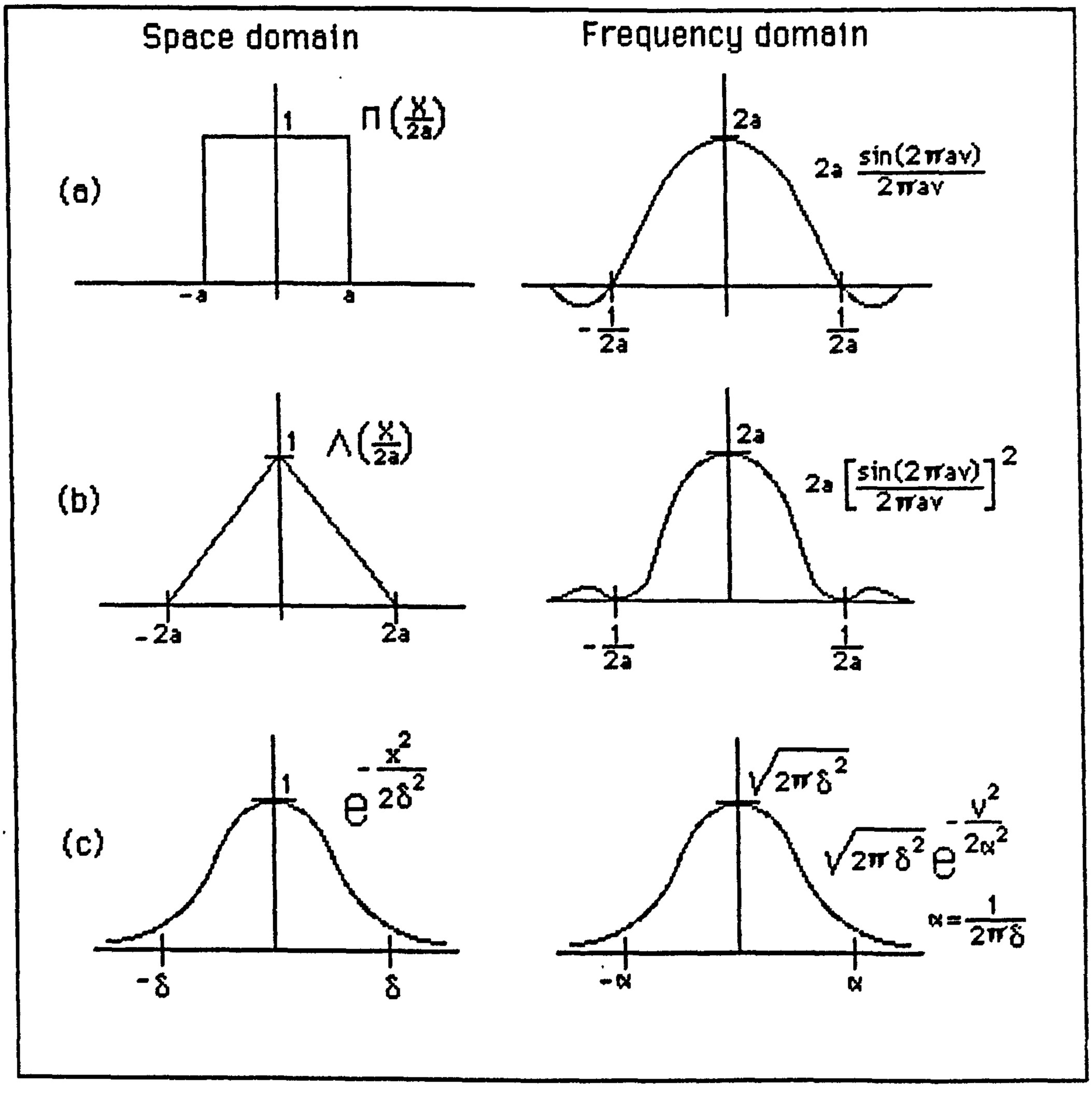


Fig. 2.10 Scanning spots and their Fourier transforms.

The amplitude spectrum of part of the digital image used in this project is presented in Fig. 2.11. Figs. 2.11(a2) and 2.11(b2) represent a single microdensitometer trace along a scan line and its corresponding amplitude spectrum respectively. Figs 2.11(a1) and 2.11(b1) show the digital image (i.e., a two-dimensional function in the space domain) and the amplitude spectrum of the same function respectively. Both amplitude spectra (Figs. 2.11(b1) and 2.11(b2)) have their origin at the centre of the input data. These Fourier transformations are performed using an equation similar to Eq. 2.25a. The two-dimensional representation of the amplitude spectrum of Fig. 2.11(a1) is given in Fig. 2.12.

2.5.5 Properties of the Fourier Transform

The Fourier transform has several associated properties such as the autocorrelation theorem; the similarity theorem; the shift theorem; the convolution theorem; etc. Some of these properties (similarity, shift, and convolution) are used in the subsequent analysis and therefore a brief review of each is given in the following paragraphs:

- **The similarity theorem** states that narrowing a function broadens its Fourier spectrum and vice versa. i.e. if,

$$\mathbb{F}\left\{f(x)\right\} = F(V)$$

then

$$\mathbb{F}\left\{f(ax)\right\} = \frac{1}{|a|} F\left(\frac{V}{a}\right) \qquad (2.26)$$

- **The shift theorem** states that shifting a function introduces a complex exponential coefficient into its Fourier spectrum. i.e.,

$$\mathbb{F}\left\{f(x-a)\right\} = e^{-i2\pi aV} F(V) \qquad (2.27)$$

where a denotes the amount of shift.

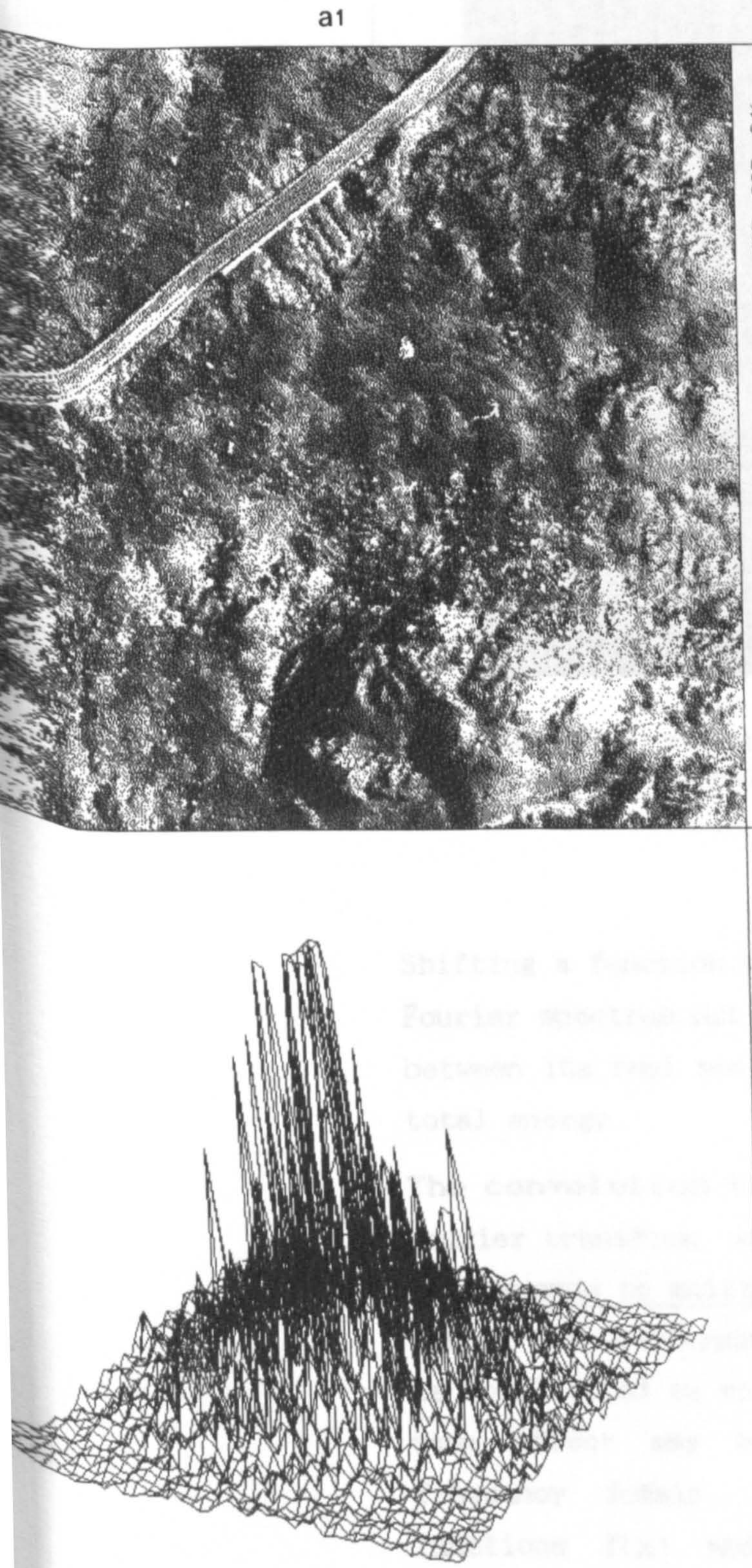


Fig.2.11 a1. The image in space domain.
b1. The amplitude spectrum of a1.
a2. A single microdensitometer trace (space domain).
b2. The amplitude spectrum of a2.

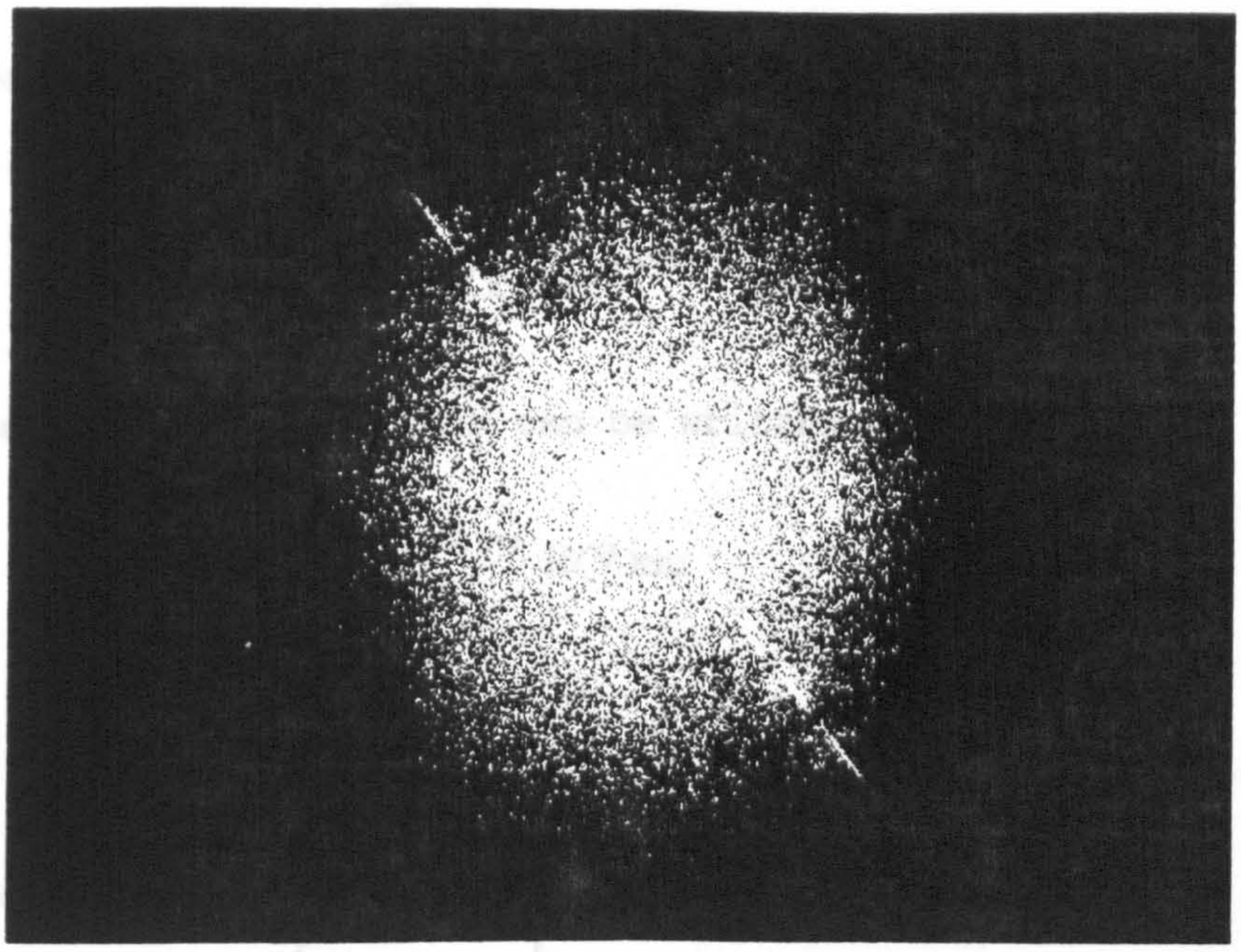


Fig. 2.12 TWO-DIMENSIONAL REPRESENTATION OF THE AMPLITUDE SPECTRUM OF Fig. 2.11(a1).

Shifting a function does not change the amplitude of the Fourier spectrum but does alter the distribution of energy[2] between its real and imaginary parts without changing the total energy.

- **The convolution theorem** points out a major benefit of the Fourier transform which is: Convolution in one domain corresponds to multiplication in the other domain. Therefore, rather than performing convolution in the space domain, which is complicated to visualize and expensive to implement, the same effect may be achieved by multiplication in the frequency domain. To prove this property, consider two functions f(x) and h(x). The Fourier transform of the convolution of these functions --see Eqs. 2.14 and 2.23-- may

[2] *The energy of a function is given by:* $\int_{-\infty}^{\infty} |f(x)|^2 dx$, *and according to Rayleigh's theorem:* $\int_{-\infty}^{\infty} |f(x)|^2 dx = \int_{-\infty}^{\infty} |F(V)|^2 dV$.

be expressed as:

$$\mathbb{F}\left\{f(x)*h(x)\right\} = \int_{-\infty}^{\infty}\left[\int_{-\infty}^{\infty} f(u)h(x-u)du\right]e^{-i2\pi vx}\,dx$$

$$= \int_{-\infty}^{\infty} f(u)\left[\int_{-\infty}^{\infty} h(x-u)e^{-i2\pi vx}\,dx\right]du$$

By the shift theorem, this can be written as

$$= \int_{-\infty}^{\infty} f(u)e^{-i2\pi vu}\,H(V)du$$

$$= \left[\int_{-\infty}^{\infty} f(u)e^{-i2\pi vu}\,du\right]H(V)$$

which means

$$\mathbb{F}\left\{f(x)*h(x)\right\} = F(V)H(V) \qquad (2.28)$$

where F and H are the Fourier transforms of f and h respectively. It also follows that

$$\mathbb{F}\left\{F(V)*H(V)\right\} = f(x)h(x)$$

That is, convolution in one domain corresponds to multiplication in the other domain. Therefore, the relationship developed in Eq.2.14 may be written as follows,

$$G = FH \qquad (2.29a)$$

where H represents the system transfer function.

It also follows that the convolution of a function f with the Dirac delta function in the frequency domain is,

$$\mathbb{F}\left\{f(x) * \delta(x)\right\} = F.1 = F \qquad (2.29b)$$

where F is the Fourier transform of f; and the Fourier transform of the Dirac function is 1 (Castleman, 1979).

2.6 Sampling Theorem

The models for the imaging process which were developed in the space and frequency domains in the preceding sections, will be used in the next sections to derive the optimum sampling interval for image digitization.

2.6.1 Modelling the Sampling Procedures in the Space and Frequency Domains

The problem of the sampling interval (pixel size) for image digitization was briefly considered in Section 2.2.1.1. In this section, the problem is approached using the linear system theory and digital image formation developed in the previous sections. The sampling is initially modelled in one dimension for an ideal case, using a series of Dirac delta functions, which leads to the determination of the optimum sampling interval. Image sampling with a real scanning spot in two dimensions will be discussed in Section 2.7.

As mentioned in Section 2.2.1.1, images are usually sampled at fixed intervals of $D_x = D_y = P$. Hence, in a perfect sampling system, a sampling function, which is referred to as a "comb" function or a "shah" function (Gaskill, 1978; Castleman, 1979), may be defined as follows: A **comb function** consists of an array of unit amplitude Dirac delta functions spaced one unit apart (Fig. 2.13). This can be shown mathematically as,

$$\text{comb}(x) = \sum_{P=-\infty}^{\infty} \delta(x-p) \qquad (2.30)$$

where P takes only integer values.

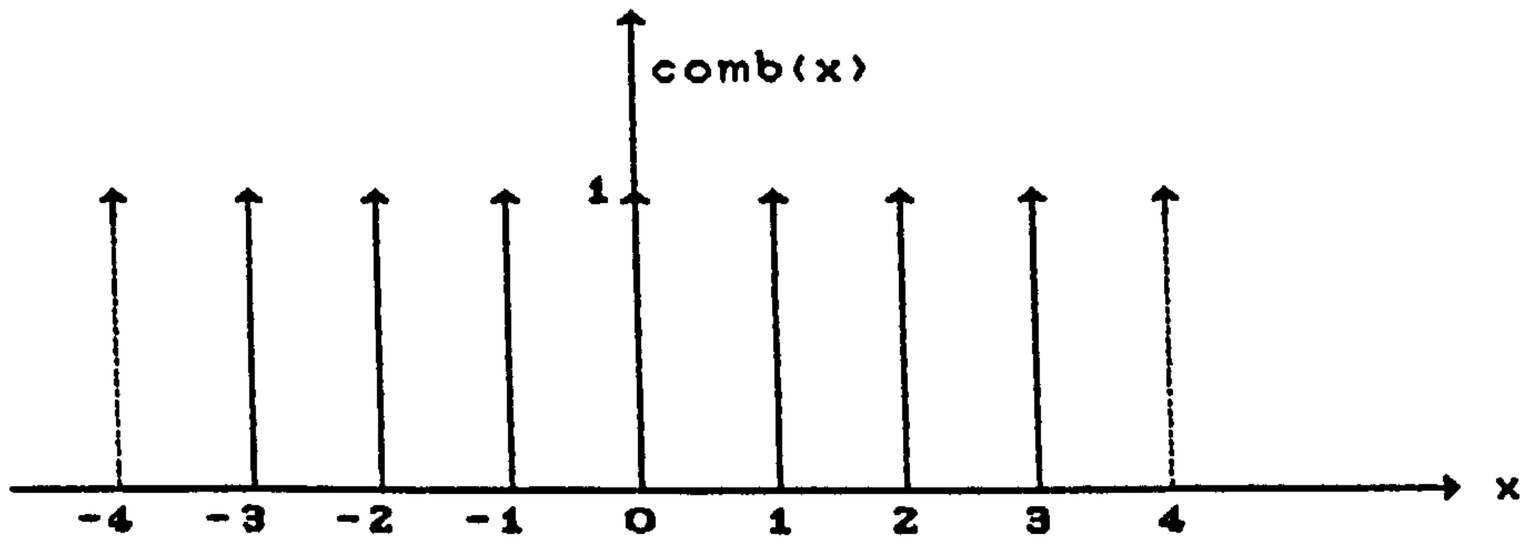


Fig. 2.13 GRAPHICAL REPRESENTATION OF THE COMB FUNCTION

To sample the continuous function f(x) by the comb function in the space domain, the comb function is multiplied by f(x), resulting in the discrete sampled function as the output. Now, if instead of a unit sampling interval, an arbitrary sampling interval b is used, then the output g(x) --see Figs. 2.14(a1), 2.14(b1) and 2.14(c1)-- is given by,

$$\text{comb}(x/b) \;.\; f(x) = g(x) \tag{2.31}$$

where comb(x/b) denotes the scaled version of the comb function. It means that, comb(x/b) is an array of delta functions spread |b| units apart.

Considering the fact that the comb function is its own Fourier transform (Castleman, 1979), Eq. 2.31 can be written in the frequency domain (recall the convolution theorem) as

$$\text{bcomb}(bV) * F(V) = G(V) \tag{2.32}$$

where bcomb(bV) is the Fourier transform of comb(x/b) (recall the similarity theorem[3]); and F(V) and G(V) are the Fourier transforms of the original and the sampled functions respectively. Thus the convolution in the frequency domain replicates the spectrum F(V) every 1/b along the V axis, and therefore results in a periodic spectrum (see Figs. 2.14(a2), 2.14(b2), 2.14(c2)).

2.6.2 Reconstruction of the Original Image

Now that the function f(x) has been sampled, it is necessary to reconstruct the original function using this sampled function. The original function can be reconstructed by simply eliminating all the replicas of F(V) --see Fig. 2.15-- except the one centred on the origin. One way to do this is to multiply the entire spectrum by a rectangular function which has elements of zero and one. This leaves only the central spectrum and reduces the others to zero. From Fig. 2.15, it is clear that the isolation of the central spectrum is only possible if two requirements exist. These requirements are namely:

3 *This means that the Fourier transform of the comb function of Dirac functions which are separated by b, is a series of Dirac functions separated by 1/b (see Fig. 2.14(a2)).*

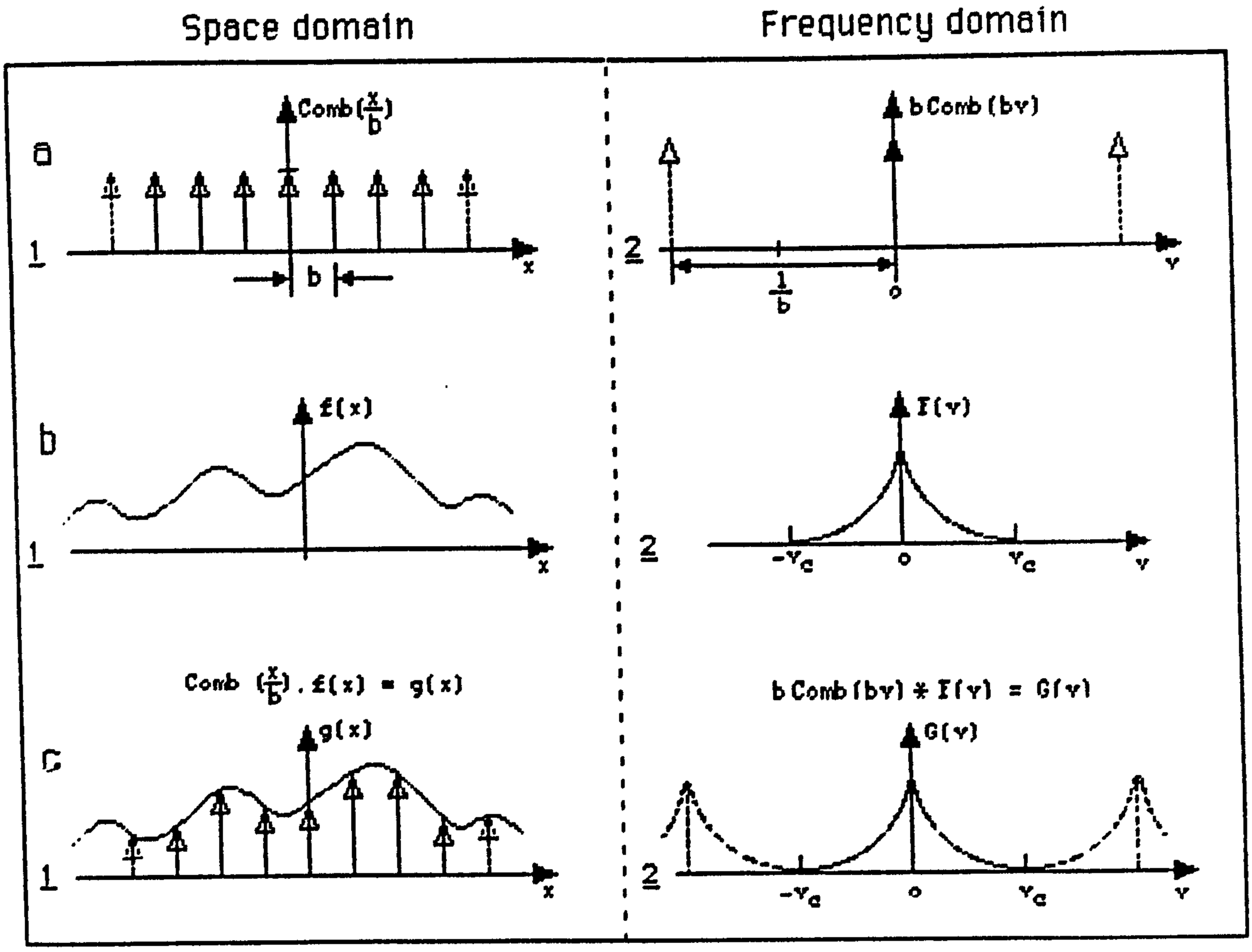


Fig 2.14 Sampling in space and frequency domains, using comb function.

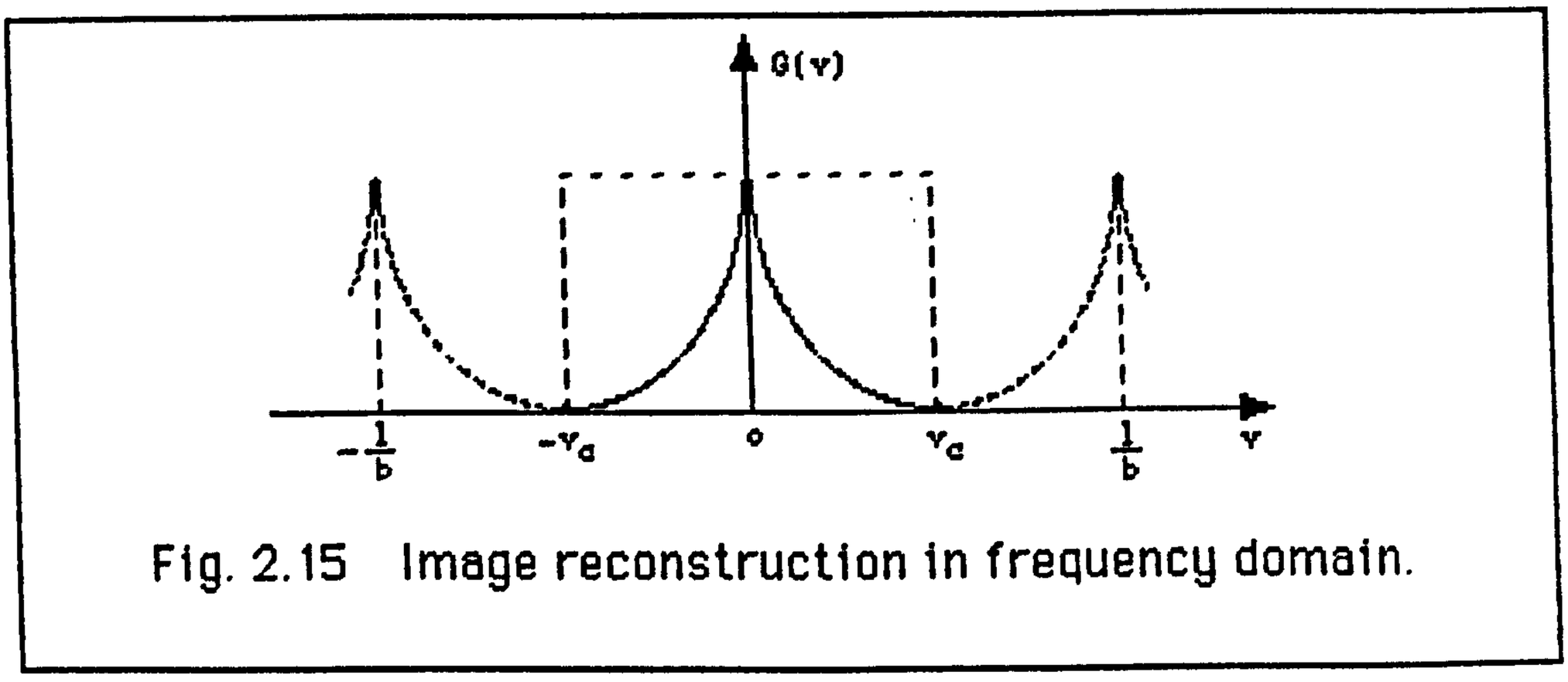


Fig. 2.15 Image reconstruction in frequency domain.

1. The original function f(x) is band-limited at a frequency V_c; that is

$$F(V_c) = 0$$

where V_c is a frequency beyond which the function F(V) is equal to zero.

2. The relationship between the sampling interval b and the frequency V_c must satisfy the following condition that

$$b \leq \frac{1}{2V_c}$$

The term 1/(2b) is sometimes referred to as the **Nyquist frequency**.

Hence the reconstruction may be achieved by multiplying the function F(V) by the rectangular function $\prod(V/2V_c)$,

$$F(V) = G(V)\prod\left\{\frac{V}{2V_c}\right\} \qquad (2.33)$$

The original input function can then be calculated by the inverse Fourier transform of Eq. 2.33 which yields,

$$f(x) = \mathbb{F}^{-1}\left\{F(V)\right\} = \mathbb{F}^{-1}\left\{G(V)\prod(V/2V_c)\right\} \qquad (2.34)$$

Applying the convolution theorem to the right hand side of Eq. 2.34 gives,

$$f(x) = g(x) * 2V_c \frac{\sin(2\pi V_c x)}{2\pi V_c x} \qquad (2.35)$$

This means that, f(x) can be perfectly reconstructed from the digitized function g(x) by convolving the sampled function with an interpolatory function of the form sin(x)/x.

The above mentioned theory of reconstructing the original function is usually referred to as the **sampling theorem** and is generally attributed to Whittaker and to Shannon, and therefore is sometimes referred to as the Whittaker-Shannon sampling theorem (Gaskill, 1978).

The theorem states (O'Neill, 1963) that it is sufficient to sample a function with an interval $b \leq 1/2V_c$. Such sampling permits the complete reconstruction of the original function and no information is lost.

The discussion presented above is equally applicable for the two-dimensional case. It goes without saying that to sample a two-dimensional function, a two-dimensional array of Dirac delta functions (i.e. a two-dimensional comb function) is required.

2.7 Image Sampling with a Real Scanning Spot

In reality, sampling is carried out with a Gaussian or similar kind of sampling spot, with different levels of noise always associated with the sampling procedures. The image spectrum, therefore, contains high frequency noise. Therefore, for reconstructing the original image, the central spectrum cannot be fully isolated (Fig. 2.16a).

As has already been seen, another problem which usually arises in real situations is the choice of the sampling interval. If the sampling interval is too large, 1/b will be too small which means the replicated spectra are too close to each other and hence, they overlap. This causes a problem which is usually referred to as an **aliasing problem**. Again the main spectrum cannot be isolated (Fig. 2.16b). This problem can cause distortion and artifacts in the digital image (Helava, 1982). Those images with strong periodic components, such as streets in high-altitude photographs of urban areas, may give rise to a number of aliased frequencies, the presence of which in the reconstructed picture creates a **Moiré pattern**. (Rosenfeld and Kak, 1976).

The sampling theorem discussed above may now be used to determine the minimum sampling interval for the digitization of an aerial photo. According to the sampling theorem, in order to digitize an image which has the resolution of 40 line pairs/mm (i.e. a typical figure for the mapping camera system resolution), the image must be sampled with a sampling spot which is 12.5 μm. However, as mentioned above, in practice, the digitization of data is influenced by Moiré effects which are dependent on the phase position of sampled points in

relation to the sampling spot. Konecny, et. al (1979) have reported the following relationship between the pixel size (sampling interval) and the line width:

$$\frac{2L}{b} = 2.50 \text{ to } 3.08 \qquad (2.36)$$

where L is the minimum resolvable line width; and b is the sampling interval (pixel size). The above relation is in agreement with the factor determined empirically by Kell in 1934. Using the Kell factor, the figure mentioned above should be modified to give a 10 μm sampling spot. Therefore, to transfer the entire information on aerial photos independently of the phase position, a 10 μm sampling interval is required.

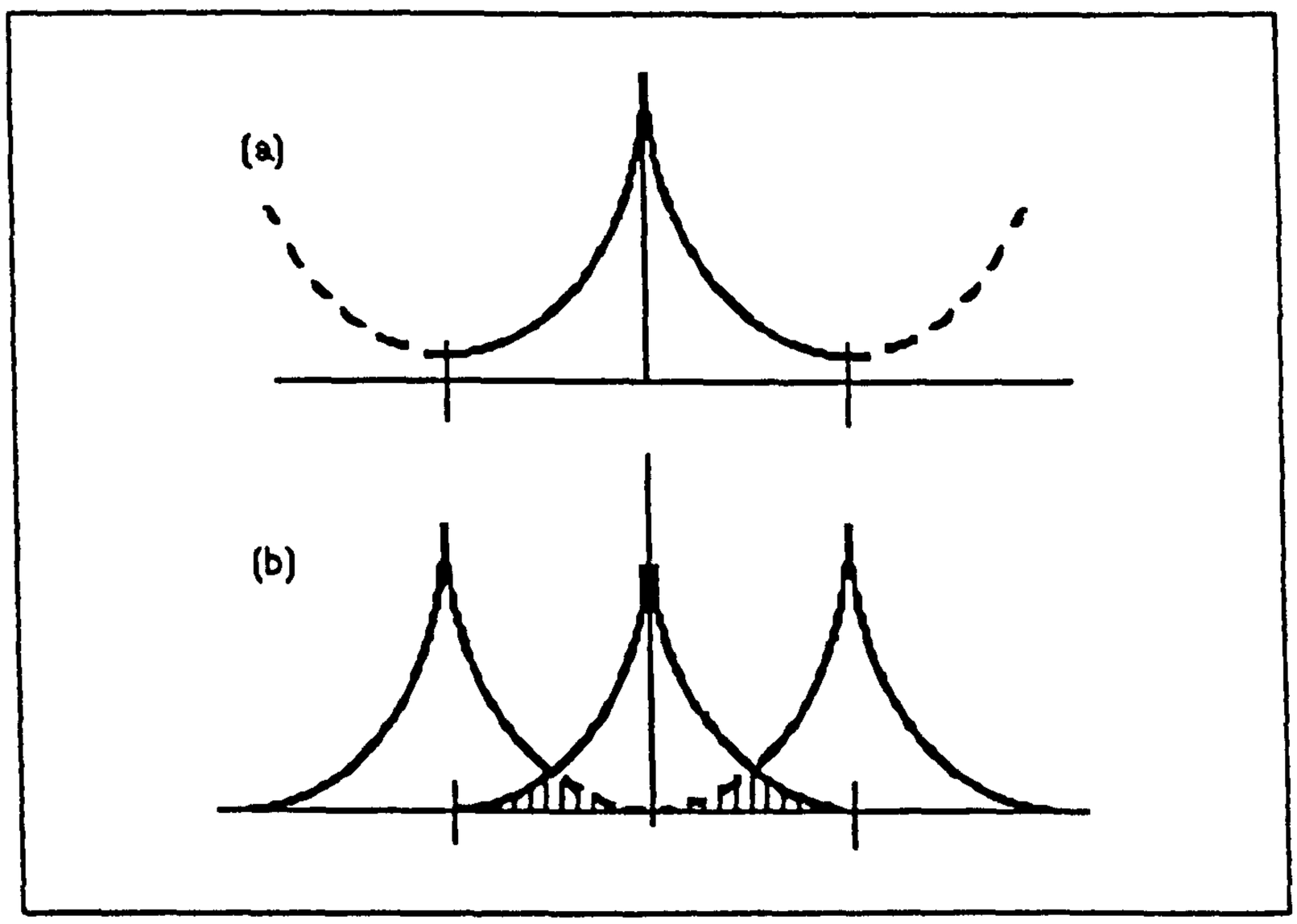


Fig. 2.16 EXAMPLES OF IMAGE SAMPLING PROBLEMS
a- NOISE PROBLEM
b- ALIASING PROBLEM (AFTER HELAVA, 1982)

2.8 Metric and Pictorial Information of Digital Images

Considering the aerial photograph as a two-dimensional light intensity function (see Section 2.1), it is possible to differentiate easily between the two sets of information present in it. The first set of information, which corresponds to the grey level variations, is referred to as **pictorial information**, and the second set which is related to the position of each image point, is known as **metric** or

geometric information. So far, a description has been given of the image formation, in so far as the pictorial information and their degradations are concerned. However, due to the imperfections in the mechanical construction of image digitizers, positional or metric degradation is also introduced to the digitized image. To be more precise, one should say that the digitized image bears two sets of metric degradations --one due to the mechanical construction of the image digitizer and the other inherited from the metric displacements and distortions already present on the analogue image, due to factors such as deviation of the camera axis from the vertical, and so on.

2.8.1 Factors Causing the Metric and Pictorial Degradations on Aerial Photographs

In general, digital image formation can be described mathematically by a transformation that maps a point f(x,y) from the analogue image coordinate system into the recorded digitized image g(x' ,y'), i.e.,

$$g(x',y') = \mathbb{U}\left\{f(x,y)\right\} + n \qquad (2.37)$$

where $\mathbb{U}$ is an operator which represents the image degradations, both of a pictorial and geometrical nature, generated by the image digitizer; and n is the noise which is nearly always associated with the digitization process. However, in this research project, the operator $\mathbb{U}$ in Eq. 2.37, is extended to cover the pictorial degradations and metric displacements/distortions which are already present on the analogue image in addition to those degradations generated by the microdensitometer. The same holds true for the variable n, which represents the noise present on the analogue image and the noise introduced by the image digitizer (see Section 4.2). This means that f(x,y), in Eq. 2.37, represents the image of a point which is free from the pictorial degradations, the metric displacements/distortions and the noise.

The mathematical presentation given in Equation 2.37 may be simplified by separating the degradations (i.e. the operator $\mathbb{U}$) into pictorial degradations (operator $\mathbb{U}_p$) and metric displacements/distortions (operator $\mathbb{U}_g$). i.e.,

$$g(x',y') = \mathbb{U}_p \; \mathbb{U}_g \left\{ f(x,y) \right\} + n \qquad (2.38)$$

The pictorial degradations and metric displacements/distortions are two different but related concepts. The former is associated with the **recognizability**, while the latter affects the **locatability**. In other words, pictorial degradations affect the ability to distinguish two closely spaced objects, whereas metric displacements/distortions limit the accuracy in estimating the distance between two (resolvable) objects.

Pictorial degradations caused by the **camera** as an imaging system are mainly the result of a combination of the following components:

aerial camera lens resolution; the film emulsion granularity; and any image motion occurring during exposure time. The effect of each of these factors may be combined to give the imaging system psf, i.e.,

$$2\sigma = 2(\sigma^2 \text{lens} + \sigma^2 \text{film} + \sigma^2 \text{image motion})^{1/2}$$

where 2σ is the distance between the two points of inflection of a Gaussian-shaped spread function. The MTF of a typical mapping camera system is given in Fig. 2.17.

Pictorial degradations caused by the **image digitizer system** are mainly the result of a combination of the following factors:
the shape and the size of the microdensitometer scanning spot (Sections 2.2.1.1 and 2.4.1); and the quantization levels (Section 2.2.1.2). A combination of these factors gives the total psf or MTF of the image digitizer.

The **geometric displacements** caused by the camera system are well known to photogrammetrists. For example, they may result from the deviations of the camera optical axis from the vertical during exposure time; terrain relief displacement; Earth curvature and atmospheric refraction. The **geometric distortions** are introduced by the imperfections of the camera system. These are mainly caused by lens distortion and the instability or lack of flatness of the film base. Since the geometric displacements and the geometric distortions are

both related to the metric aspects of the imaging system, in contrast with the pictorial degradations which are related to the pictorial side, throughout this research project, they are represented as a single operator (i.e. the operator U_g).

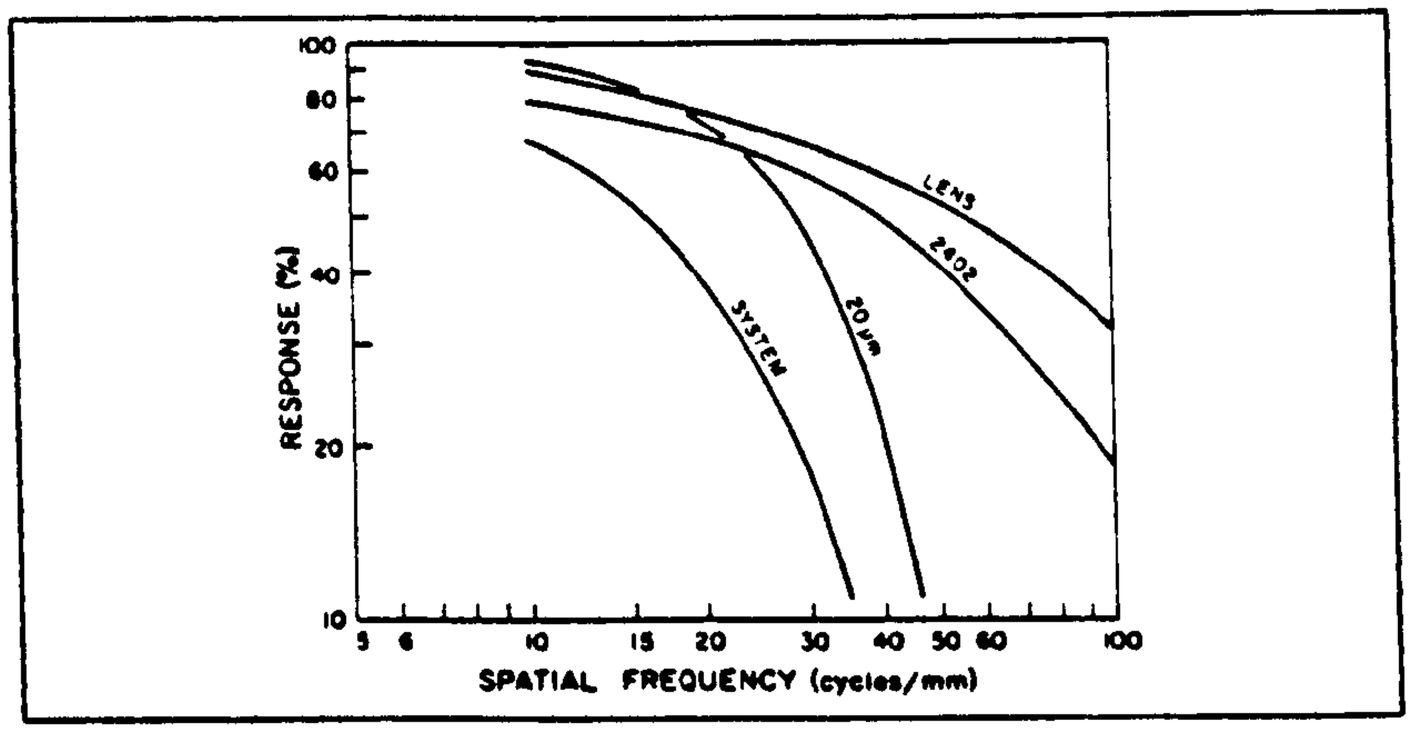


Fig. 2.17 MTF'S FOR LENS, EMULSION, 20 μm IMAGE MOTION AND THE CORRESPONDING SYSTEM MTF. (AFTER MANUAL OF PHOTOGRAMMETRY, 1980)

As mentioned earlier, these geometric displacements/distortions on the analogue image are present before digitization. Imperfections in the construction of the microdensitometer cause further metric distortions. Indeed when ordinary microdensitometers are used, these distortions may be quite significant. For example, Wolf and Dewitt (1984) used a drum-type microdensitometer (see Chapter 3) to digitize aerial photos having a series of drilled holes. The coordinates of the holes were determined from the scanned image and compared with the coordinates of the same points measured by a monocomparator and the discrepancies between the two sets of measured points were calculated. According to their report, these discrepancies were quite significant with a highly systematic shearing pattern. The discrepancies were more significant (rmse of ±193 μm) along the y-direction (i.e. the drum rotation direction) than in the x-direction (rmse of ±48 μm).

Chapters 4 and 5 will present a thorough investigation as to how the pictorial degradations and the metric distortions and displacements can be eliminated.

2.9 Conclusion

In this chapter, a mathematical model has been developed in the space domain as well as in the frequency domain to describe the procedures of digital image formation and, according to that mathematical model, the optimum sampling interval has been calculated. The pictorial degradations and the metric displacements and distortions which are inevitable during the imaging process have also been discussed briefly from the photogrammetric view point. In the next chapter, consideration will be given to the practical procedures involved in image digitization; and the design and construction of the microdensitometer used in this project to digitize a pair of aerial photos, will be described.

CHAPTER THREE

Digital Image Formation Using the COSMOS Scanning Microdensitometer

3.1 Introduction

In the preceding chapter, the theoretical principles involved in the formation of digital images were considered. As mentioned, the actual formation of a digital image from an aerial photograph is accomplished by means of an image digitizer. The characteristics of image digitizers will be discussed below.

3.2 Principles and Classification of Image Digitizers

In general, the digitization process can be realized by two different types of instruments which use two completely different digitization principles:

(a) The first type of image digitizer is referred to as a **scanning image digitizer or scanning microdensitometer**. As the name suggests, instruments belonging to this category use a raster scanning mechanism to cover the whole area of the photograph with a single element detector used to convert the brightness of each pixel into electrical signals which in turn can be converted to a digital value (Fig. 3.1a).

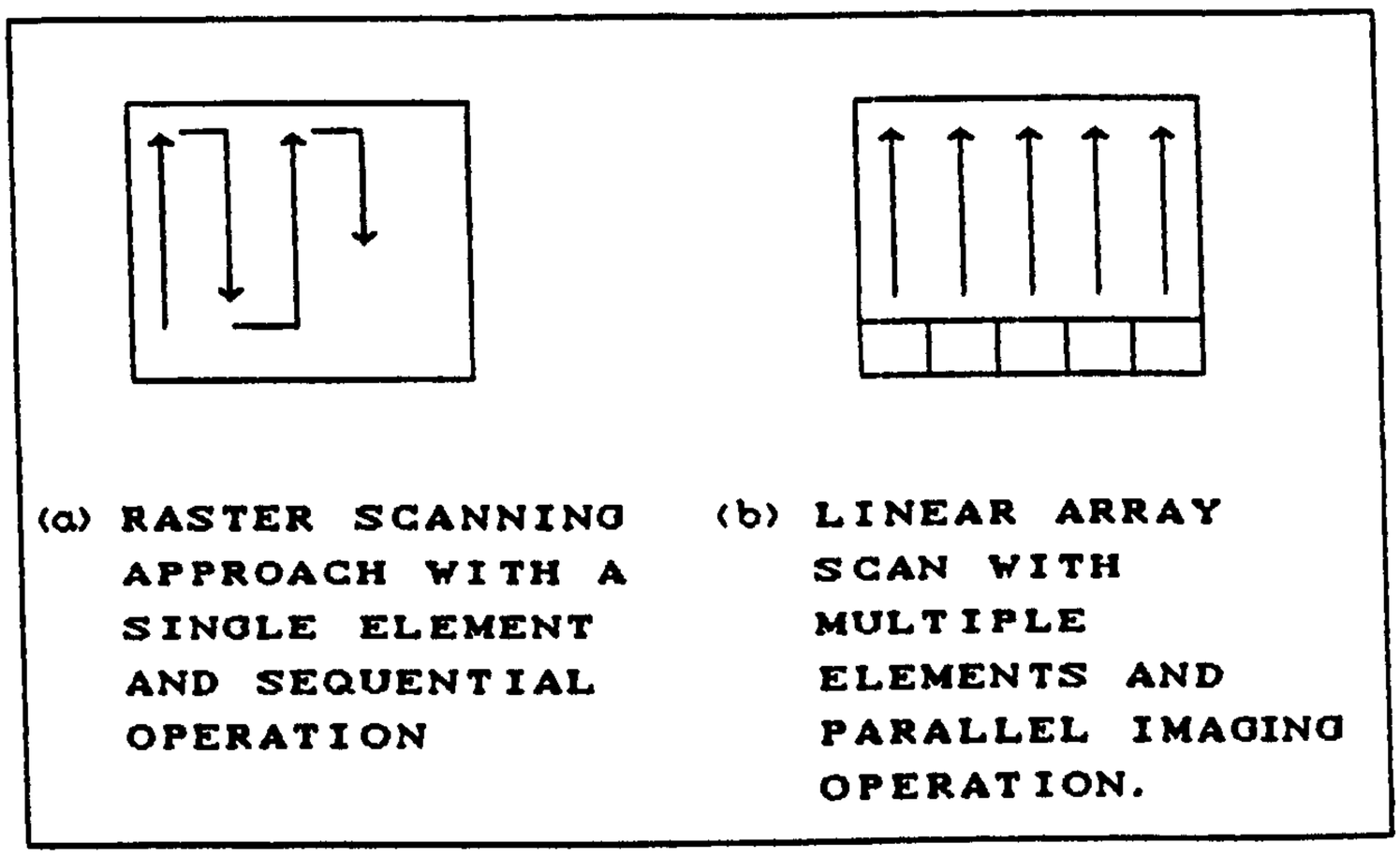


Fig. 3.1

Scanning microdensitometers, currently available in the market, may be grouped into two different types, namely, flatbed and drum-type microdensitometers.

(i) **Flatbed** microdensitometers allow the image which is to

be scanned to be laid flat on a fixed measuring stage. In terms of geometrical accuracy, this class of microdensitometer ranks much higher than the other kinds of image digitizers and hence they are much more expensive than these other types. Currently there are only a few of these machines available in the U.K. These include the Automated Photographic Measuring System of the University of Cambridge (Kibblewhite, 1981) and the COSOMS microdensitometer of the Royal Observatory in Edinburgh (MacGillivray and Stobie, 1984). The latter has been used in this project and will be described in some detail later in this chapter.

(ii) The **drum-type** microdensitometers use a rotating drum to produce a raster scan. This type of microdensitometer seems to be less appropriate for photogrammetric applications than the first group, since they seem to have geometrical instability in the direction of the drum rotation (see Chapter 2.8.1). There are a number of drum-type microdensitometers available on the market, e.g. the Scandig device from the Vickers company and various devices made by Optronics. This category of scanning image digitizer will also be considered in more detail later in this chapter.

(b) Unlike the first category which uses a raster scanning mechanism and a single element detector, the second type of image digitizer employs an **array of solid-state detectors** (also called Charged Coupled Devices, or CCDs). These image digitizers can use either a linear or an areal array of detectors (Amin, 1986).

(i) An image digitizer employing a **linear array** uses a number of discrete detector elements which are oriented across a line. Thus the device digitizes a single line simultaneously via a projection lens, with the length of the array and the number of detectors corresponding to the width of the photograph being digitized (Fig. 3.1b). The scanning motion employs a precision stepping motor and lead screw assembly to drive the linear array of detectors across the focal plane of the image

digitizer so that the whole area of the photograph can be digitized (Fig. 3.2a).

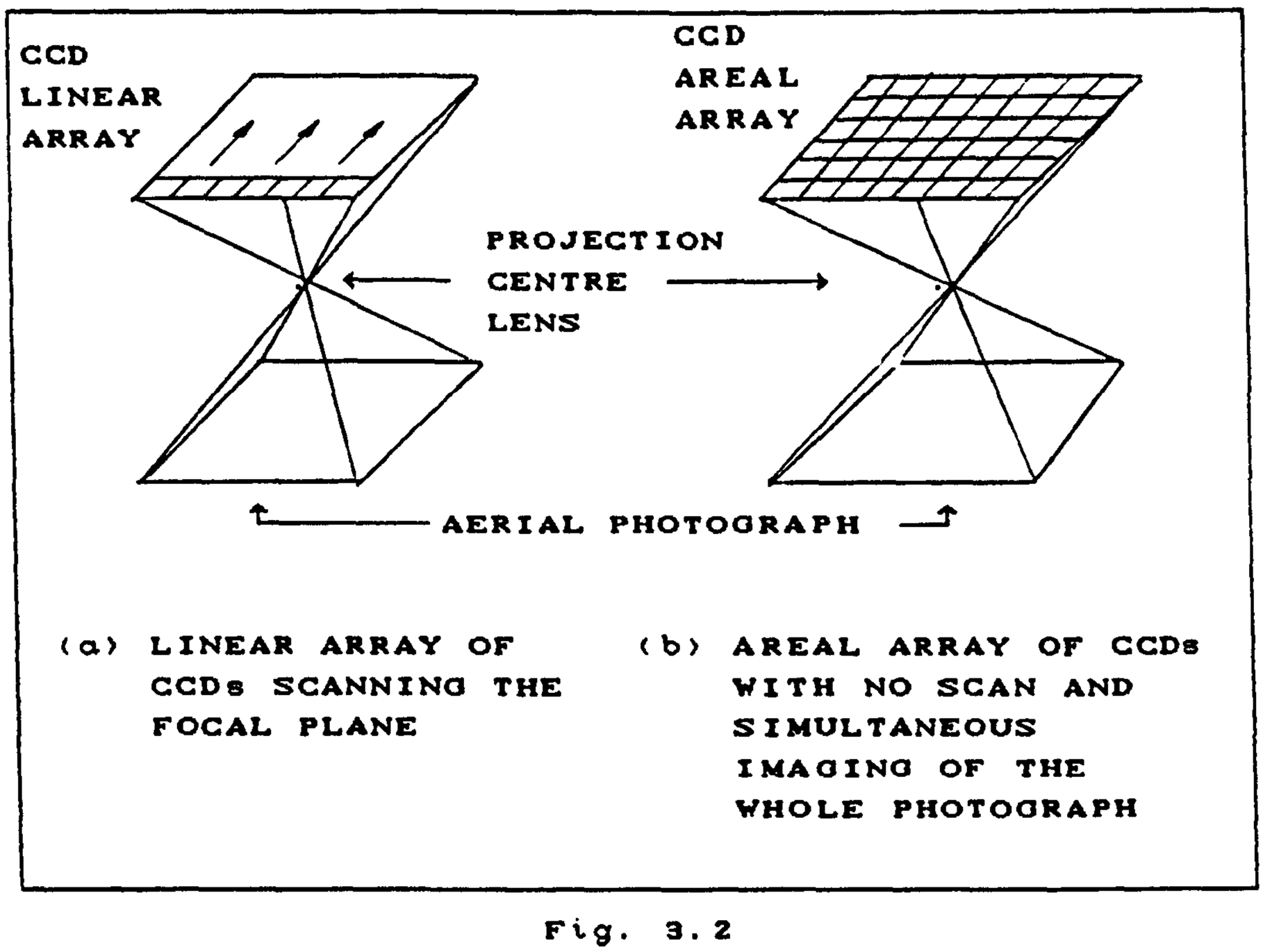


Fig. 3.2

There are several commercially available image digitizers which use this approach. Examples are the Eikonix instruments from the Kodak company and the Japanese-made Kimoto digital imaging camera systems. This type of digitizer may have a comparatively high resolution, e.g. the Kimoto model 620F instrument has a resolution of (3,456X4,472=15,455,232 pixels) which corresponds to a 65X50 μm^2 pixel size when used with an aerial photograph of the standard size 23x23 cm^2. However, the geometric fidelity of this category of image digitizer seems not to be too consistent in the direction of movement of the linear array scanning across the focal plane.

(ii) Those devices which use an **areal array**, arrange the solid-state detectors in a **matrix form** so that the whole of the image is digitized simultaneously via a projection lens (Fig. 3.2b). In geometric terms, the imaging process is

similar to that of a photographic camera, except that the photographic emulsion is replaced by the matrix array of detectors. However, due to the cost of the solid-state detectors and also the problem of calibration of the response of the huge number of detectors, existing areal array image digitizers employ a limited number of detectors and hence have a rather low resolution. At the present time, the highest solid-state frame camera using an areal array of CCDs has a resolution of 1,000X1,000 pixels, giving 1M pixels. This means that there is a very long way to go before an all-digital frame camera appears which gives the same resolution and format size as the present highly developed photogrammetric cameras (Petrie, 1990). Also when such a device is used as an image digitizer, it falls some considerable way short of achieving a comparable resolution to that achieved with an instrument based on the use of a linear array (1M pixels v. 15.4M pixels).

On the basis of having a rather limited geometric accuracy and resolution, the solid-state image digitizers were rejected in this research work for the digitization of the aerial photographs which contain a high degree of pictorial and geometric information (see Chapter 2). Instead, as mentioned earlier, a very accurate scanning microdensitometer (the COSMOS machine) was employed to digitize the aerial photos. The first part of this chapter describes the principle and construction of scanning microdensitometer instruments in general, while the second part deals with the principle and construction of the COSMOS microdensitometer.

3.3 Components of Scanning Image Digitizers

An image digitizer must be able to divide an image into pixels, measure the grey level of the image at each pixel, quantize that measurement to produce an integer which is proportional to the measured density of the pixel and finally write the generated digits in computer-compatible form on a data storage device. To accomplish this, a scanning image digitizer must have the following elements:

- **a sampling aperture (sampling spot)**, which allows the digitizer to isolate the pixels individually while ignoring the remainder of the image;
- **a mechanism for scanning the image**, which allows the sampling aperture or spot to move over the image in a pre-determined pattern and to address each pixel in turn;
- **a sensor**, which measures the brightness of each pixel and converts it into an electrical voltage;
- **an analogue-to-digital converter (A/D converter)**, which converts the continuous output of the sensor into an integer value; and
- **the output medium**, which stores the integer value in an appropriate format for subsequent processing; this output medium may be omitted if the image is being processed "on-line".

Before considering these components in some detail, it should be mentioned that there are two alternative digitizing approaches which can be employed in a scanning microdensitometer. As specified by Castleman (1979), these are "scan-out" digitizing and "scan-in" digitizing. In a **scan-out system**, the entire image is illuminated continuously, and the sampling aperture allows the sensor to sense one pixel at a time. In a **scan-in system**, only one small spot of the image is illuminated and the transmitted light is collected for the sensor. In this case, the image is scanned with the illuminating beam. There is a third approach which is a combination of the previous two. In a **"scan-in/scan-out" system**, the object is illuminated by a moving spot and sampled through a moving aperture that follows the spot.

As discussed above, an image digitizer must have a light source, a light sensor, a scanning system, and a digital-to-analogue converter. The sections which follow give a brief description of each of these components.

3.3.1 Light Sources

The light for an scanning microdensitometer can be generated in different ways. The primary source is usually an incandescent lamp, a

laser, a light emitting diode (LED) or the electron beam generated by an electron gun.

Incandescent lamps are convenient for the general illumination of the image being digitized and hence are suitable for use in scan-out systems. For scan-in systems, the filament of a small bulb can be imaged with a lens to form a small bright spot. However, a more efficient source is the **laser**. Highly concentrated beams can be produced with a laser for use in scan-in digitizer systems. The laser generates a narrow, intense, coherent beam of light by first raising the atoms of an active material (neon, chromium, etc.) to a high energy state. This situation gives rise to a beam of coherent light of high intensity that is easily focused and deflected. An alternative solution is to use a **light emitting diode (LED)** which is capable of producing a tiny and rather stable source of light. This may be used in a scan-in system. Finally, the electron beam generated by a Cathode Ray Tube (CRT) may be used as a light source. Certain phosphors emit light when irradiated with the electron beam generated by a (CRT). The generated light spot can be focused by a lens onto the aerial photograph and may be used for scan-in or scan-out digitizer systems.

3.3.2 Light Sensors

Light sensors produce an electrical signal proportional to the amount of light falling upon them. Three physical phenomena give rise to three types of light sensors. These are :

(i) **photoemissive substances** such as oxides of the alkaline metals (e.g. silver, caesium, etc.), which emit electrons when irradiated with light;

(ii) **photovoltaic substances**, such as silicon and selenium solar cells, which generate an electrical potential when exposed to light; and

(iii) **photoconductors**, such as cadmium sulphide and cadmium selenide, that show a drop in their electrical resistance when exposed to light.

The microdensitometer used in this project employs photoemissive

devices. Hence brief descriptions of the **phototubes** and the **photomultiplier tubes**, which use photoemissive substances, are given in the following paragraphs.

The **phototube** (Fig 3.3) has a positively charged anode and a negatively charged cathode coated with layers of oxides of certain alkaline metals. When photons of sufficient energy (wavelength less than about 1 μm) strike the photocathode, electrons are freed from the surface. Under the influence of the electric field, they migrate to the anode, producing a current flow through the device. The amount of current produced is proportional to the photon flux which is incident on the photocathode.

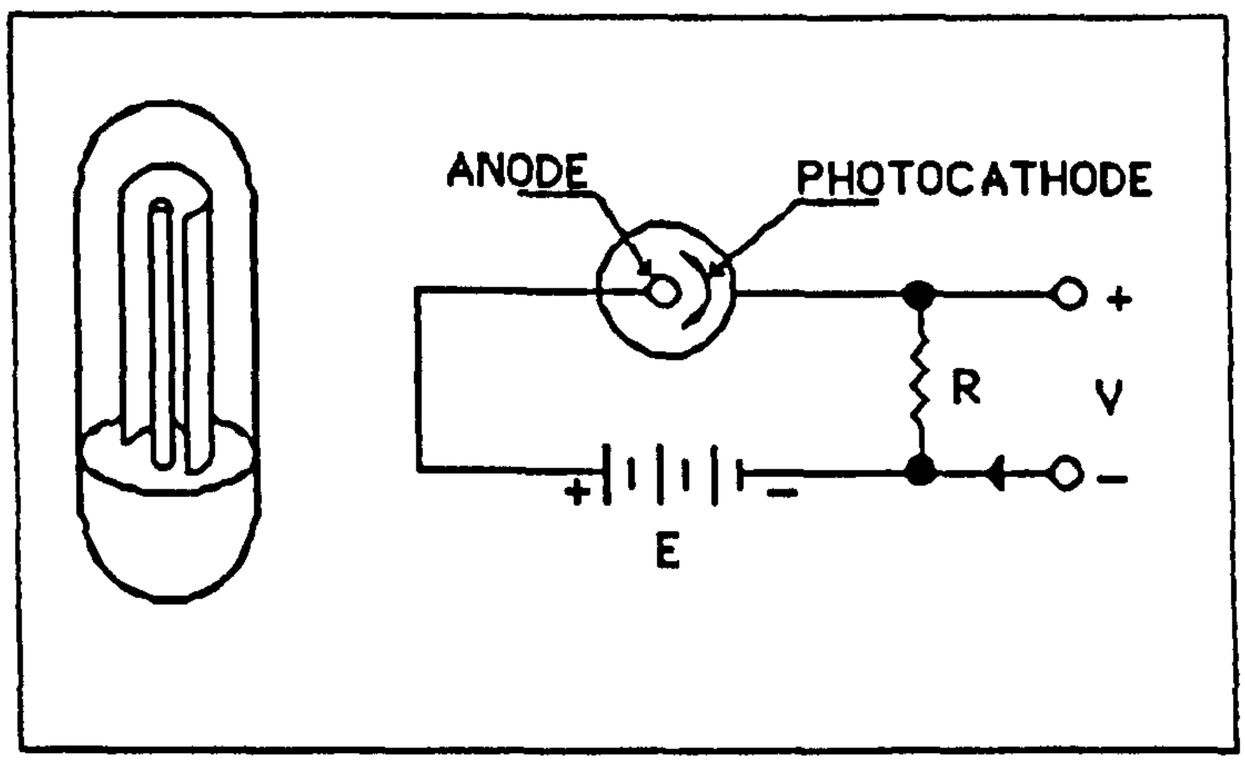


Fig. 3.3 THE PHOTOTUBE

The **photomultiplier tube** (Fig 3.4) has a photoemissive face that forms a semi-transparent photocathode. Behind the cathode are a series of dynodes charged with progressively higher positive voltages. Primary electrons freed from the photocathode are accelerated toward the first dynode. The impact on the first dynode frees secondary electrons, producing a multiplying effect. The process continues until the electrons from the last dynode are collected on the anode to produce a current in the external circuit. Hence the photomultiplier tube behaves similarly to the phototube except that it is more sensitive to the low light levels because of the multiplying effect of the dynodes.

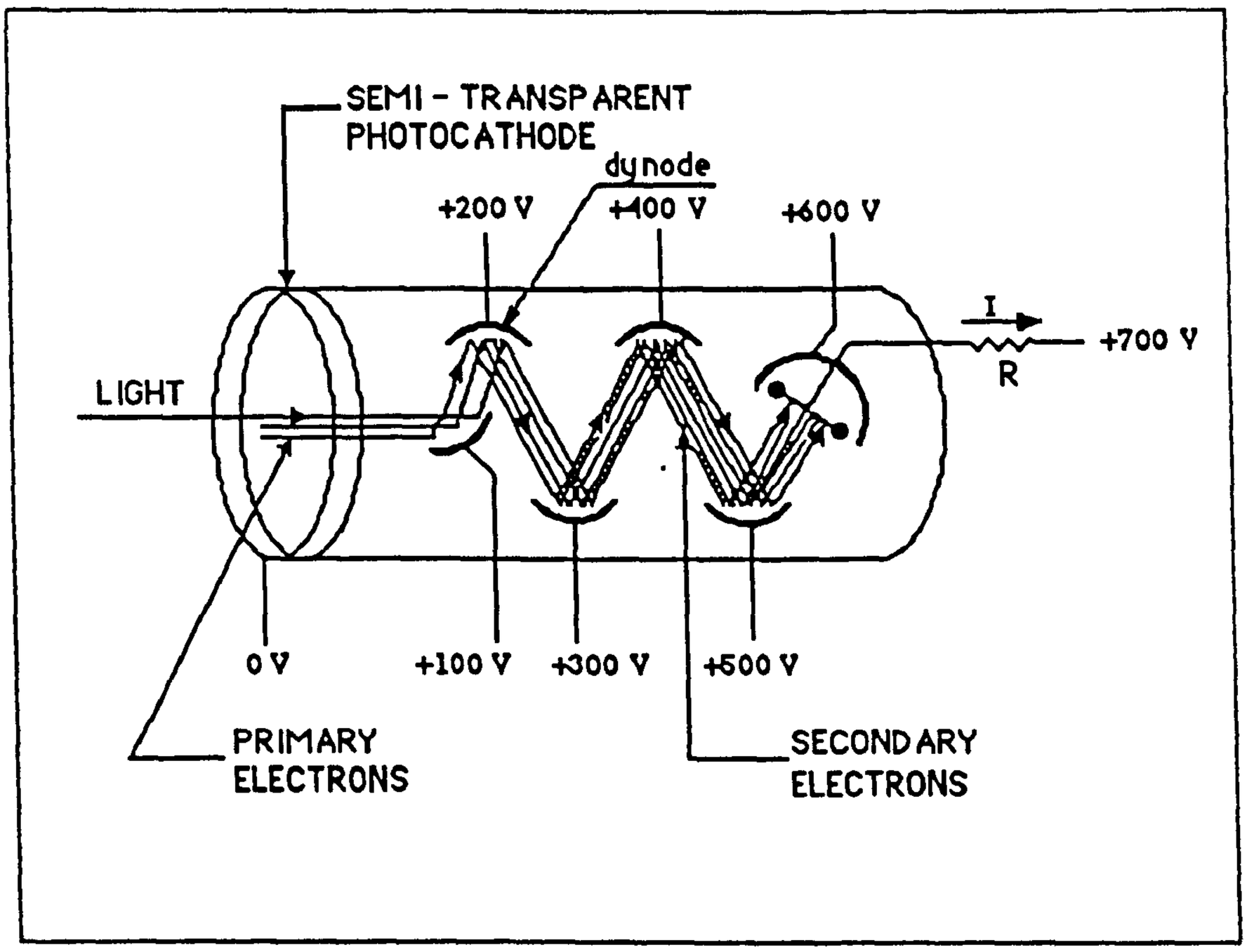


Fig. 3.4 THE PHOTOMULTIPLIER TUBE

3.3.3 Scanning Mechanism

From the photogrammetric point of view, the scanning mechanism of the image digitizer is probably the most important component, since any non-linearity of the scanning will introduce geometrical distortions. There are three general scanning mechanisms implemented in micro-densitometers. These are:

a. Moving-Mirror Scanning

In this approach, the motion of an oscillating mirror, mounted on a galvanometer motor, causes the scanning spot to move across the image (see Fig. 3.5) The major disadvantage of the oscillating mirror is the low speed of operation. More importantly, the result is subject to geometric distortion, because the displacement of the scanning spot is not proportional to the deflection angle of the mirror but rather to the tangent of that angle. This produces non-uniform pixel spacing, resulting in a so-called "barrel" distortion of the

digitized image (Fig. 3.6a).

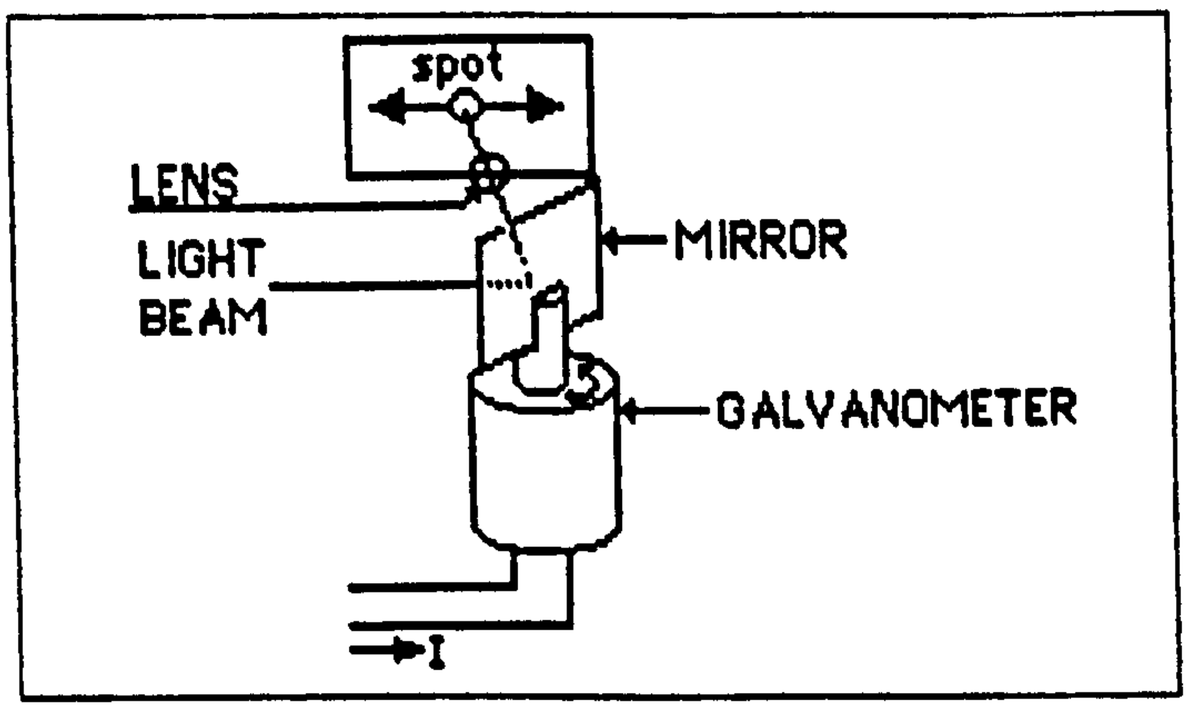


Fig. 3.5 MOVING MIRROR SCANNING

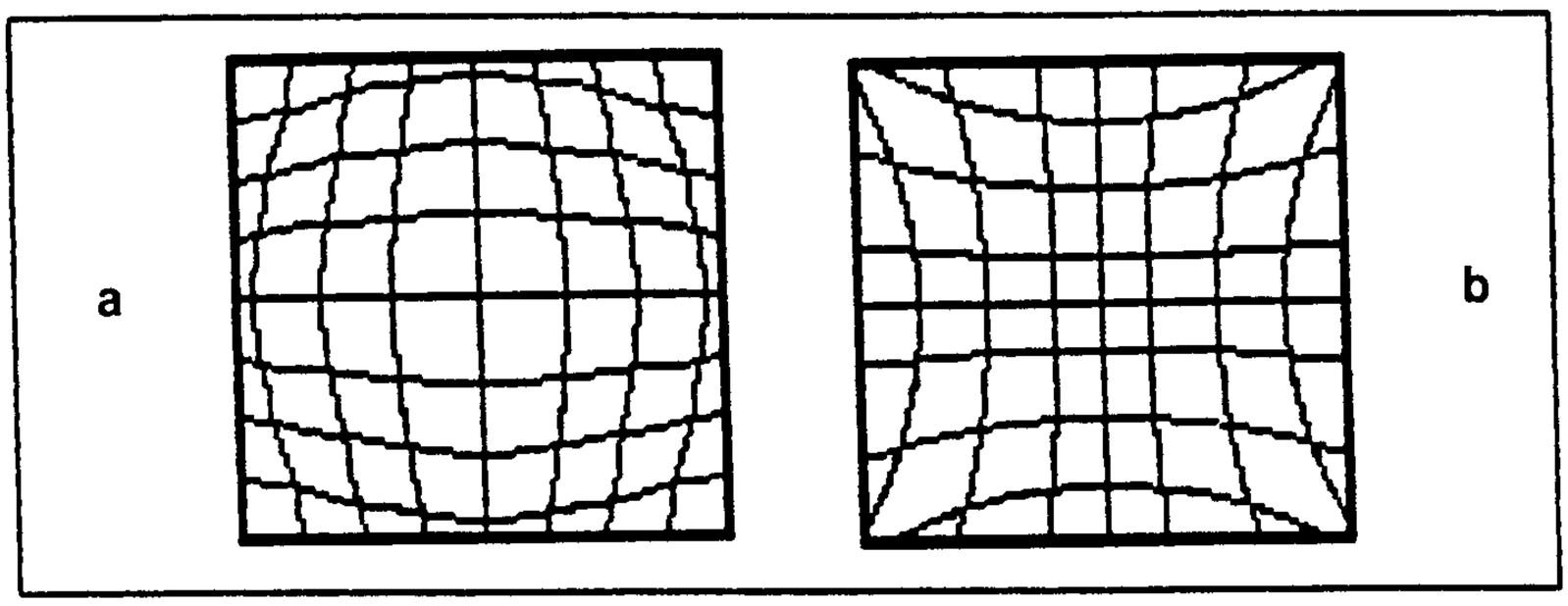


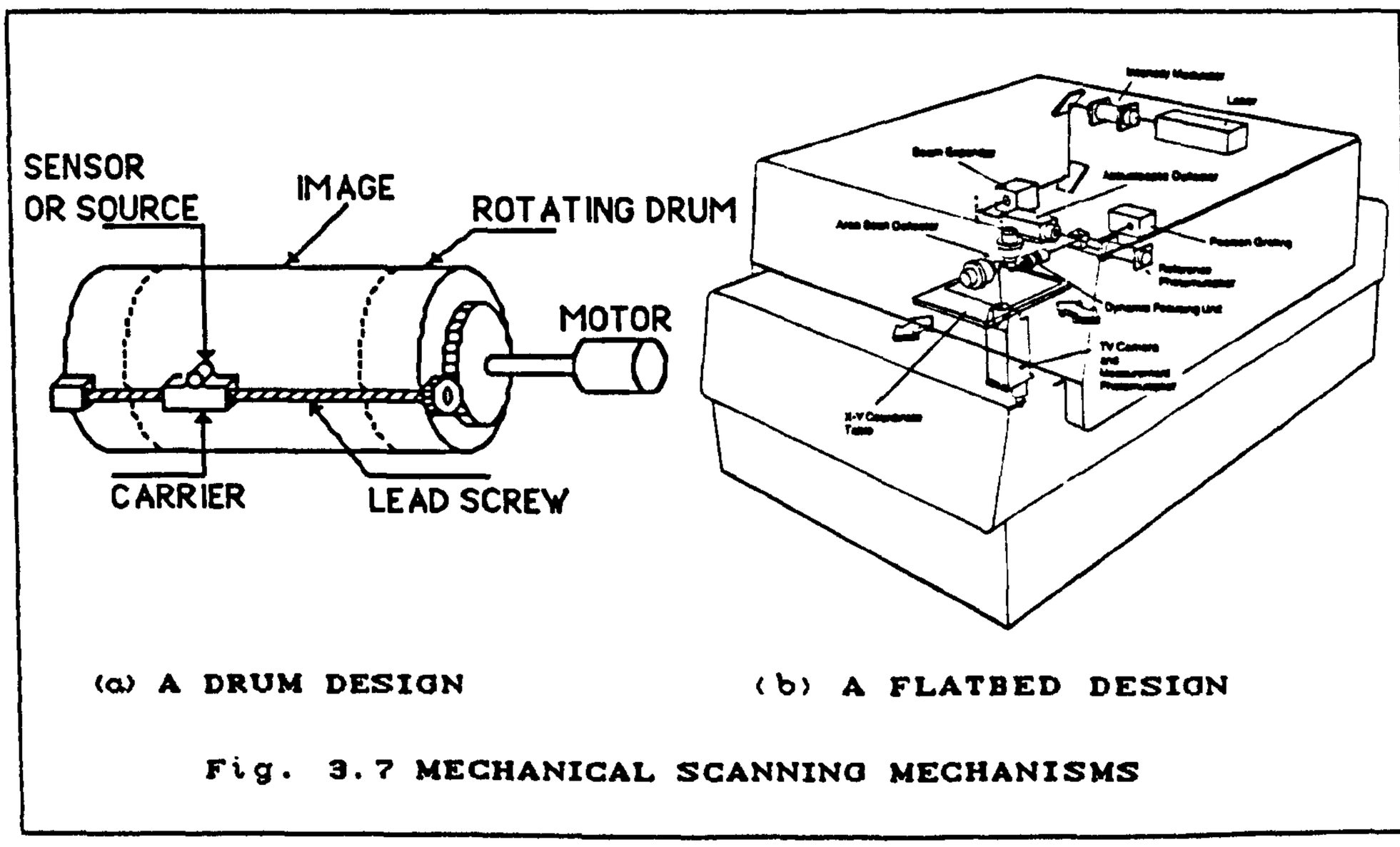
Fig. 3.6 (a) BARREL DISTORTION
(b) PINCUSHION DISTORTION

b. Mechanical Scanning

Mechanical scanning is usually implemented in one of two basic ways:

1. "Drum" designs involve film diapositives which are fastened onto a rotating cylinder. The scanning aperture or the sensor is mounted on a lead screw (Fig 3.7a). In operation, the drum rotates at a constant speed (i.e. there is a constant y-motion) below the sensor which is moved continuously forward in steps along the axis of the drum (i.e. x-direction). In this way, a two-dimensional image scanning is produced. As mentioned earlier, this scanning mechanism may produce distortions, especially in the direction of drum rotation.

2. **"Flatbed" designs** enable a glass or film diapositive to be laid flat on the fixed measuring stage. A raster scan is accomplished by movement of the carriage containing the light source and sensor in the x-direction and, after the completion of each line, it moves one step in the y-direction. Alternatively, the stage supporting the plate may be moved in both the x and y directions against a fixed position of the light source and sensor --though this is less common. However, the flatbed scanning design may also be implemented by a combination of the plate movement and a scan deflector such as that of the rotating mirror, CRT, etc. (Fig. 3.7b). The moving mirror or CRT produces a scan parallel to the x-direction, while the carriage supporting the photograph moves in the y-direction. More detail of this type of scanning mechanism is given in Section 3.4.2 where the general description of the COSMOS microdensitometer is given.



(a) A DRUM DESIGN (b) A FLATBED DESIGN

Fig. 3.7 MECHANICAL SCANNING MECHANISMS

c. **Electron Beam Scanning**

In this approach, an electron beam, generated by the electron gun in the base of the CRT, is deflected to form a raster scan by using electrostatic deflection plates, so that the electric field exerts a force on the electrons, changing their direction of travel. The generated electron beam strikes the glass screen of the CRT which is coated with a thin layer of phosphor. At this position where the beam of high speed electrons strikes the phosphor layer, a series of electrons is displaced so that this point on the layer becomes positively charged and glows brightly. This bright spot is then projected on to the image to be digitized via a focusing lens (Fig. 3.8). Due to the non-linear movement of the scanning spot, the electron beam scanning may produce a distortion pattern which is usually referred to as a pincushion distortion (Fig. 3.6b).

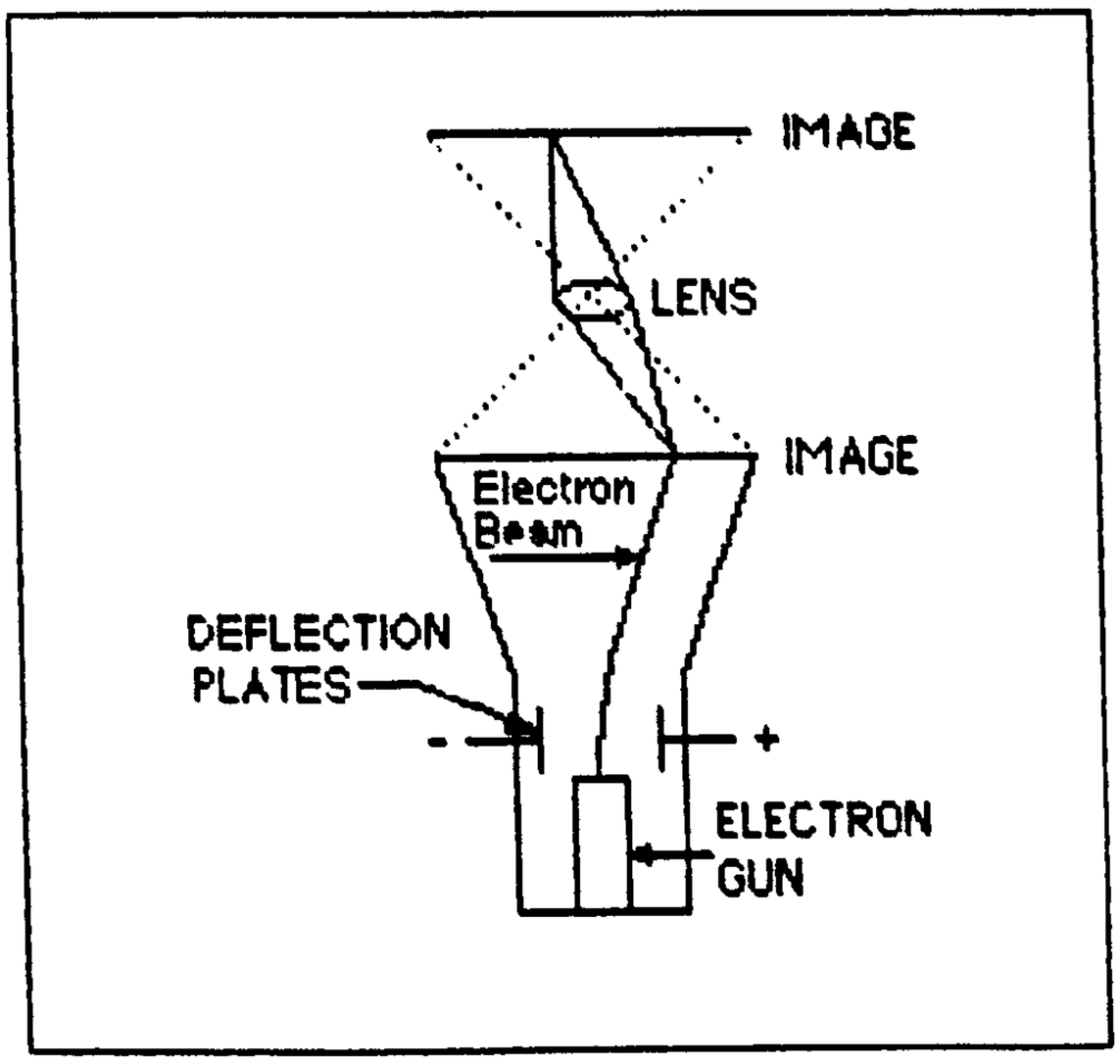


Fig. 3.8 ELECTRON BEAM SCANNING

3.3.4 Analogue-to-Digital (A/D) Conversion

In order to digitize an image with grey levels proportional to the

density[1] of the diapositive, it is necessary to quantize a signal that is proportional to the negative of the logarithm of transmittance[2]. This task is realized in the image digitizer by an **analogue-to-digital (A/D) conversion circuit** (Fig 3.9) which is described below.

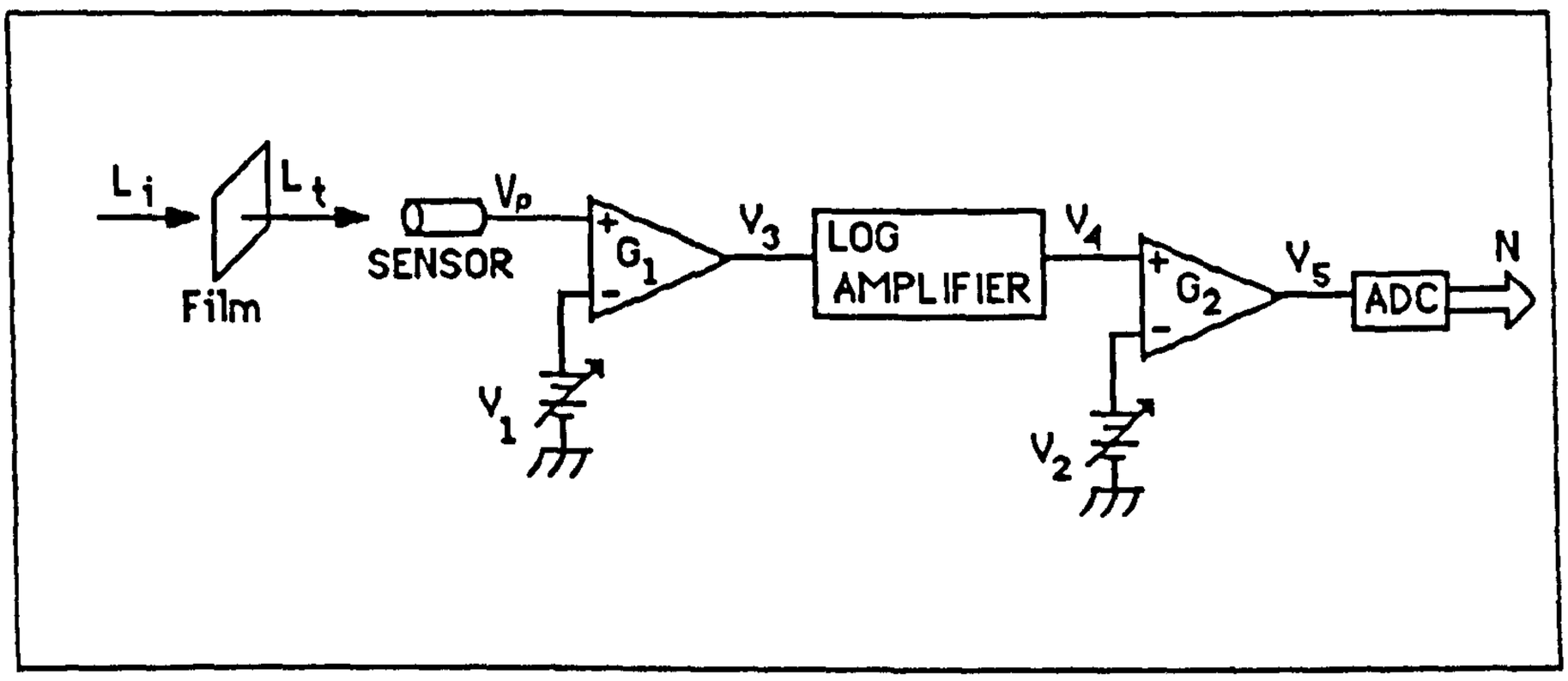


Fig. 3.9 ANALOGUE-TO-DIGITAL CONVERSION CIRCUIT

Suppose each pixel location on the film is illuminated with a beam of intensity L_i. The film attenuates the beam intensity to L_t before it strikes the photomultiplier tube. The output signal generated by the photomultiplier tube is given by:

$$V_p = V_d + K L_t$$

where V_d is the dark current component and K is the sensitivity coefficient. The signal then passes into a differential amplifier which produces the output,

$$V_3 = G_1(V_p - V_1)$$

where the gain G_1 and offset voltage V_1 are adjustable. The logarithmic amplifier then produces an output,

$$V_4 = \mathrm{Log}(V_3)$$

Finally, a second differential amplifier produces,

---

1 *The density is the logarithm of the ratio of the incident light to the transmitted light.*

2 *The transmittance is the ratio of the transmitted light to the incident light.*

$$V_5 = G_2 (V_4 - V_2)$$

where G_2 and V_2 are again adjustable. The A/D converter can then be adjusted to produce an integer N in response to V_5 such as,

$$N = \text{integer } (25.5\ V_5)$$

i.e., N goes from 0 to 255 linearly provided that V_5 is adjusted to vary from 0 to 10 volts.

Having described the components of the scanning microdensitometer, the next section deals with the construction of the microdensitometer used to digitize the aerial photos for this project.

3.4 The COSMOS Microdensitometer

COSMOS (Co-ordinates, Sizes, Magnitudes, Orientations and Shapes) is a highly accurate computer-controlled flatbed scanning microdensitometer located at the Royal Observatory, Edinburgh. It is intended primarily to detect and measure the positions and structural properties of the many faint stars and galaxies appearing on photographic plates taken by astronomical telescopes (Williams and Pratt, 1975). However, the machine was used in this project to digitize a pair of aerial photos in order to perform digital photogrammetric operations. The description of the machine is given in the sections which follow.

3.4.1 General Configuration of the COSMOS Machine

The general configuration of the COSMOS machine is shown in Fig. 3.10. The machine has the following components and specifications:

- A **flying spot** is generated by a CRT and focused on the emulsion. The scanning spot has a Gaussian profile in the core. An autofocus mechanism enables the machine to follow the "sag" in thin large-format glass plates which can be as much as 250 μm from the centre to the edge of the plate.
- The machine has a **compound x/y carriage** with digital readout generated from Moiré fringe gratings to a resolution of 0.5

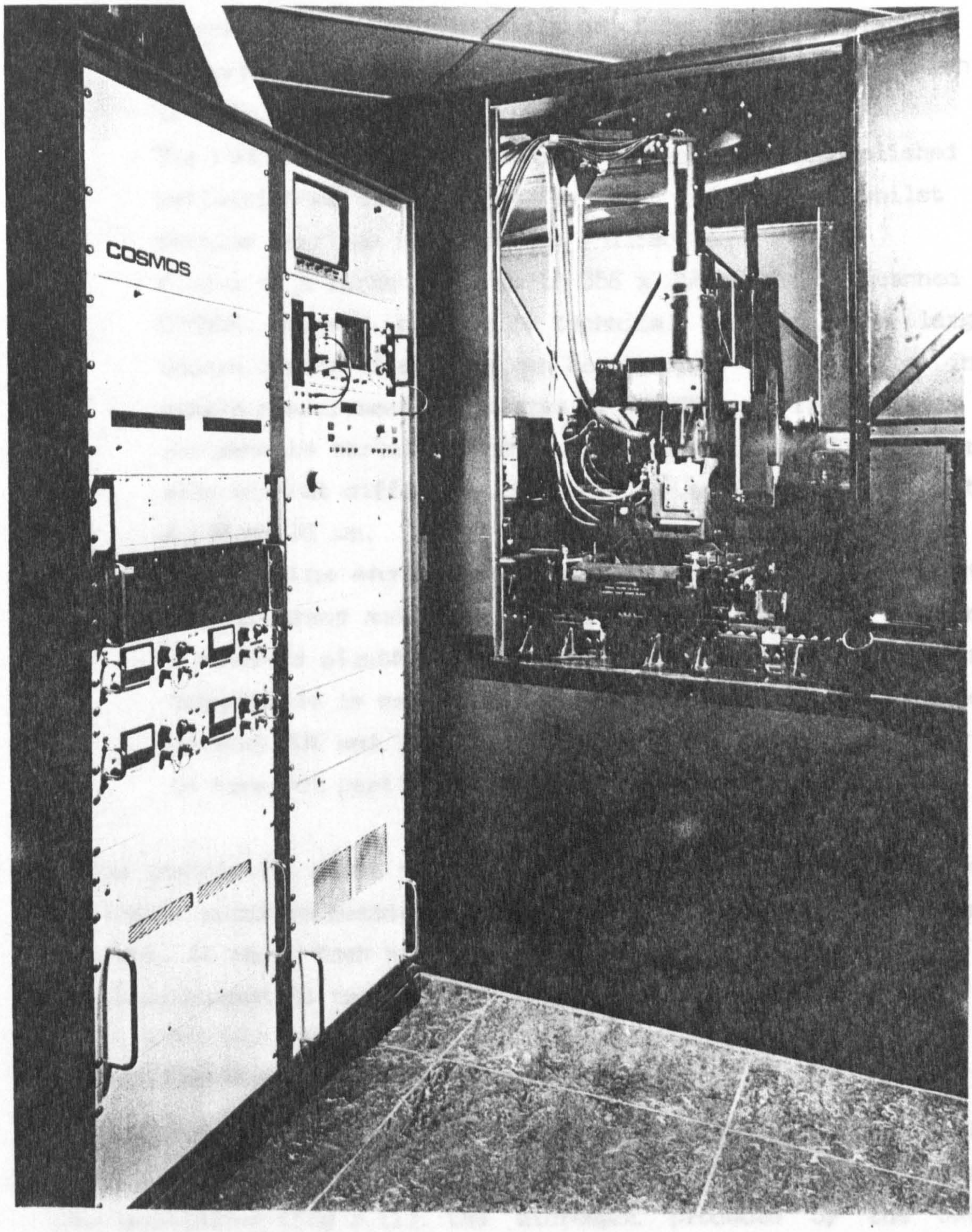


Plate I:- A view of the COSMOS machine showing the electronics racks (left), plate carriage (centre) and scanning assembly (above centre).

Fig. 3.10

μm. The position of the carriage is controlled via a closed-loop servo action, maintaining the position to an accuracy of 0.5 μm. The plate or film transparency to be measured is clamped, emulsion up, in the plate holder, which is then mounted horizontally in the machine.

- The **raster scan** in the COSMOS machine is accomplished by deflection of the CRT spot in the x-direction whilst the machine carriage moves in the y-direction.
- Plates of a format size up to 356 x 356 mm can be scanned on COSMOS. However, because of technical details, the largest square region of a plate can be scanned in practice in a single measurement is an area of 287x287 mm. Therefore it can accommodate aerial photos in the standard 230x230 mm format size without difficulty. The pixel size can be set at either 8, 16 or 32 μm.
- The **machine environment** is strictly controlled to maintain high accuracy and stability. The machine itself is mounted on a concrete plinth which is embedded in solid rock. The room temperature is maintained at 20°± 0.5°C and the humidity between 30% and 50% . In addition, air filtration is provided to take out particles with a size greater than 2 μm.

As the description given above shows, the COSMOS machine is intended for highly accurate measurements of analogue images and, in terms of accuracy, it ranks much higher than the normal microdensitometers used for photogrammetric tasks.

3.4.2 The Working Principle of COSMOS

The machine operates by focusing light from a CRT onto the emulsion and recording the transmitted light with a "main" channel photomultiplier (Fig 3.11). The microspot produced by the CRT is deflected in the x-direction, the x-position control within the scan being achieved by a grating channel signal fed into a phase-locked loop circuit which provides feedback to the CRT deflection coils. The length of the deflection is 128 pixels (i.e. 1, 2 or 4 mm for the 8, 16 and 32 μm increment sizes respectively). The microspot is deflected in the x-direction, while the carriage is shifted in the y-direction

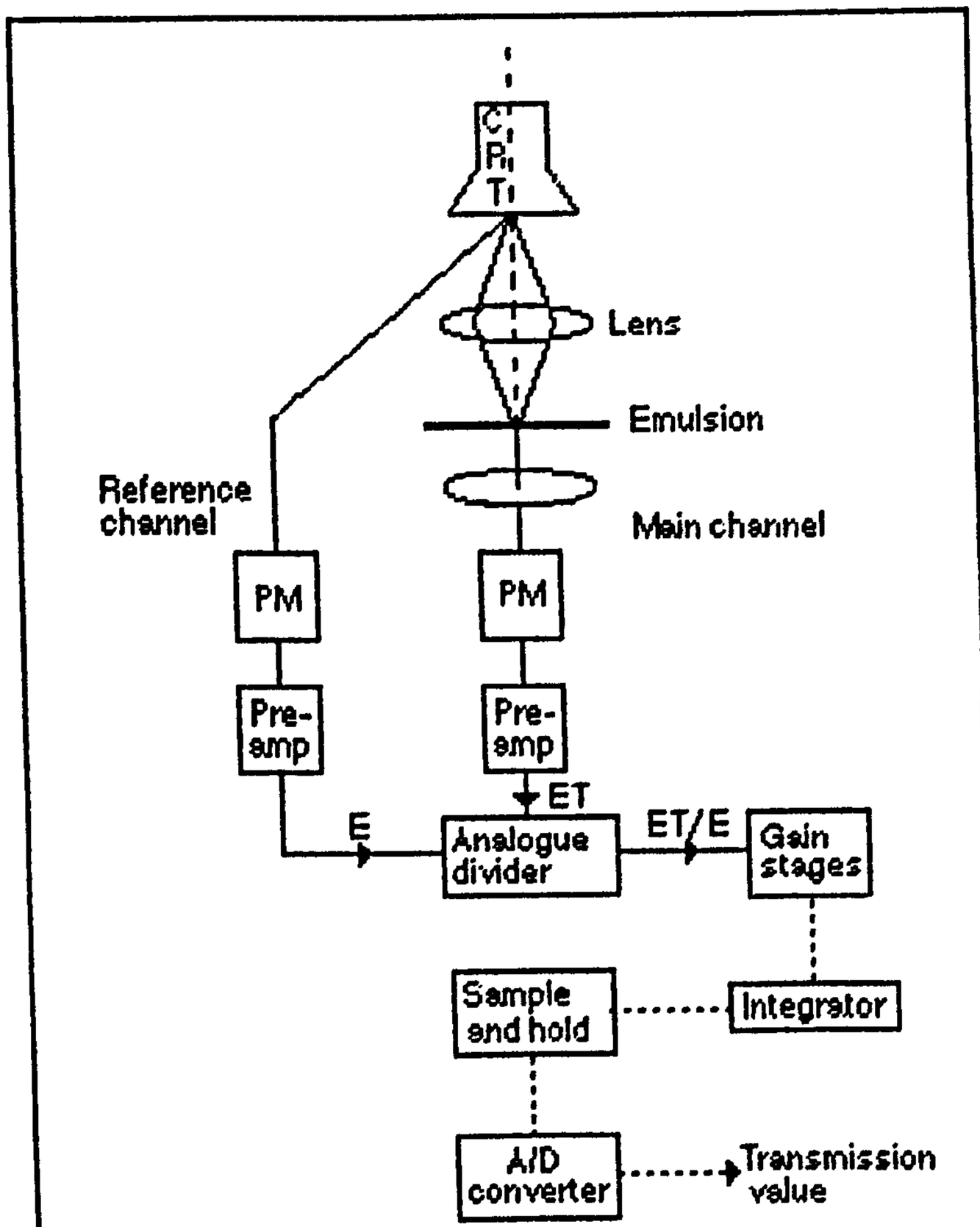


Fig. 3.11 Schematic plan of the scanning/detection system on COSMOS.

by means of hydraulic rams. Thus a complete measurement of the plate is built up from scans in "lanes" of 128 pixels width.

The light transmitted through the plate is detected by the main channel photomultiplier and the output current is converted into a voltage. A second photomultiplier (the "reference channel") looks obliquely at the CRT phosphor to monitor the spot brightness and uses this information together with the main channel signal in a divider loop to normalize the spot brightness. This allows the elimination of effects such as variations in spot brightness with time and variations in response across the face of the CRT. Finally the integrated signal is fed into an A/D converter giving as output an 8 or 14-bit transmission value (i.e. either 256 or 16,384 density levels).

COSMOS operates in either one of the two available measuring modes, i.e. either in **Mapping Mode (MM)** or **Threshold Mapping (TM)** mode. In the (MM), mode COSMOS simply digitizes a complete area and retains the information on every pixel. In the TM mode, the measurement proceeds as in the normal MM mode, but instead of outputting the transmission value of every pixel, the machine determines (for astronomical plates) a smoothed local sky background on the photograph and applies a threshold at a fixed percentage level in the intensity space above this background. Only pixels of intensity above this level are output. This second mode is designed specifically to suit astronomical applications.

3.4.3 The Control and Data Processing Computers

A block diagram of the computer configuration associated with COSMOS is shown in Fig 3.12. Control of the machine is carried out by means of a DEC (Digital) PDP 11/24 mini-computer. The operator interacts with the machine by means of a VT100 terminal which has a graphics capability. The PDP 11 organises the read-out of data from a FIFO (First In First Out) buffer by means of a DMA (Direct Memory Access) device either onto 1,600 b.p.i magnetic tapes or via a high speed data transfer link to the dedicated processing computer, which is a DEC (Digital) VAX 11/750 mini-computer equipped with a floating point accelerator and 8 Mbytes of memory.

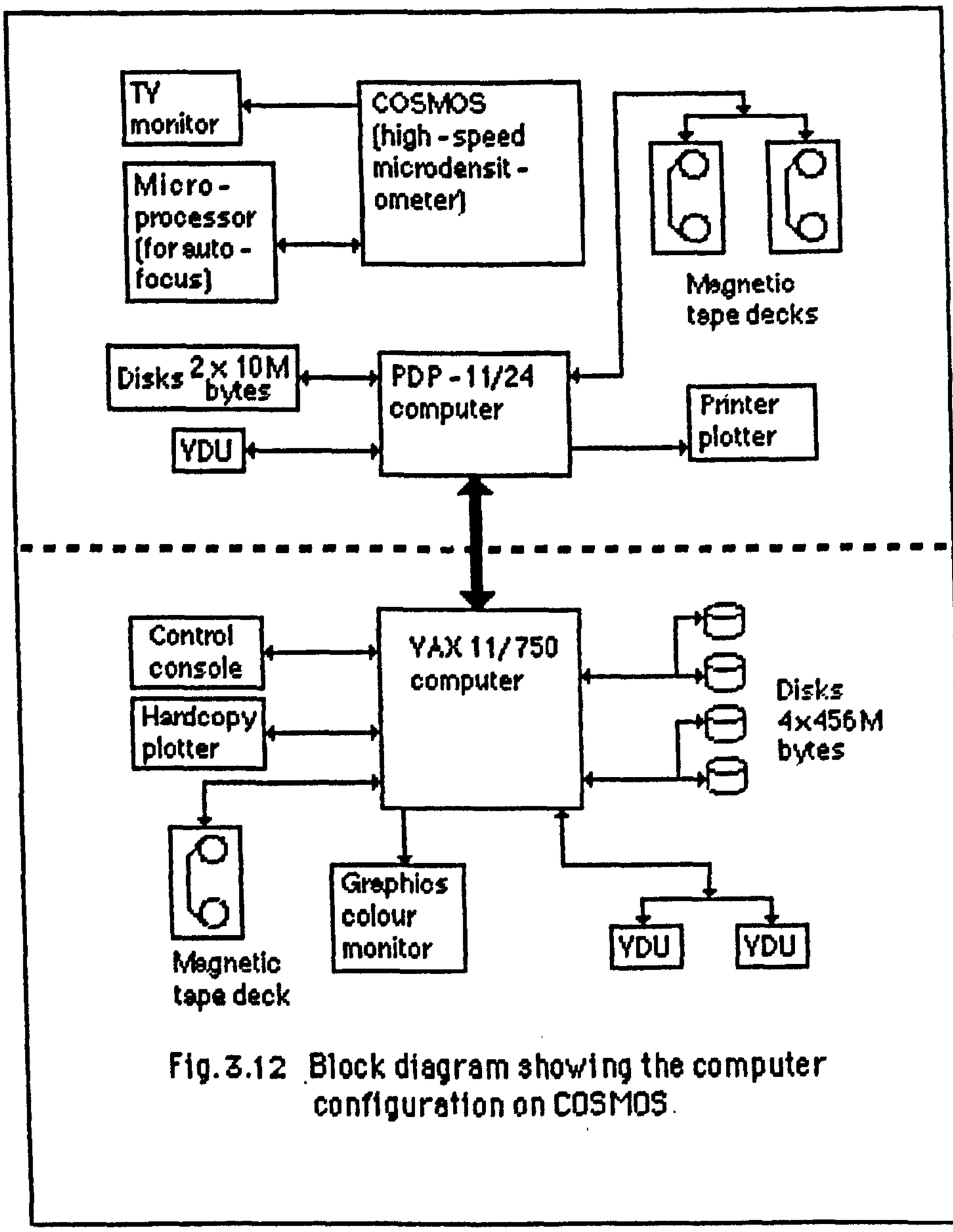


Fig. 3.12 Block diagram showing the computer configuration on COSMOS.

3.5 Data Structure and Storage

In order to implement a purely digital photogrammetric system, as mentioned before, a suitable pair of aerial photos covering a mountainous area in North Wales with an approximate scale of 1:18,000 was selected from the different sets of aerial photos available in the Department. The selected photos were the only ones which seemed to be really suitable for the subsequent image processing stages. The entire area of each photo was then digitized in MM mode by the COSMOS machine.

As discussed in Chapter 2, the image should ideally be digitized with a minimum of 10 μm pixel size if the entire information available on the analogue photo is to be transformed into digital form. However, digitization at such a level would involve a huge storage requirement, as indicated by the amount of digital image data with respect to pixel size set out in Table 3.1. Since there were limitations on available disk storage on the IBM 3090 computer used for the project, it was decided to digitize the image using a pixel size larger than 10 μm.

PIXEL SIZE (μM)	PIXEL PER LINE FOR 23 CM	PIXEL PER IMAGE FOR (23*23 CM)	IF IMAGE IS QUANTIZED AT 8 BIT PER PIXEL (BITS)
10	23,000	529,000,000	4,232,000,000
20	11,500	132,250,000	1,058,000,000
50	4,600	21,160,000	169,280,000

TABLE 3.1 DIGITAL IMAGE DATA PRODUCED BY DIGITIZING A METRIC FRAME CAMERA PHOTOGRAPH (23*23 CM FORMAT) (AFTER PETRIE, 1990)

Trinder (1987) compared the MTF of an original photograph with the MTF of the images digitized with sampling spots varying from 12.5 to 100 μm. He showed that there is a rather small deterioration in the image quality when the image is digitized with pixel sizes up to about 25 μm. However, the image quality deteriorated significantly for the images digitized with 50 and 100 μm sampling spots respectively (Trinder, 1987). On the basis of this experimentation, it was decided

to digitize the photographs with the largest COSMOS sampling spot --namely 32μm. This also gave the opportunity to investigate the effects of such a pixel size on the accuracy of the geometric information (described later in Chapters 6 and 9). Nevertheless, even 32 μm pixel size produced a digital image of about 10^8 digits corresponding to 100 Mbytes of storage. The digitized transmission values were encoded into 14-bit transmission values. These were stored as 16-bit words in COSMOS data files which were recorded on two 6,250 b.p.i. magnetic tapes, each tape holding a separate photo.

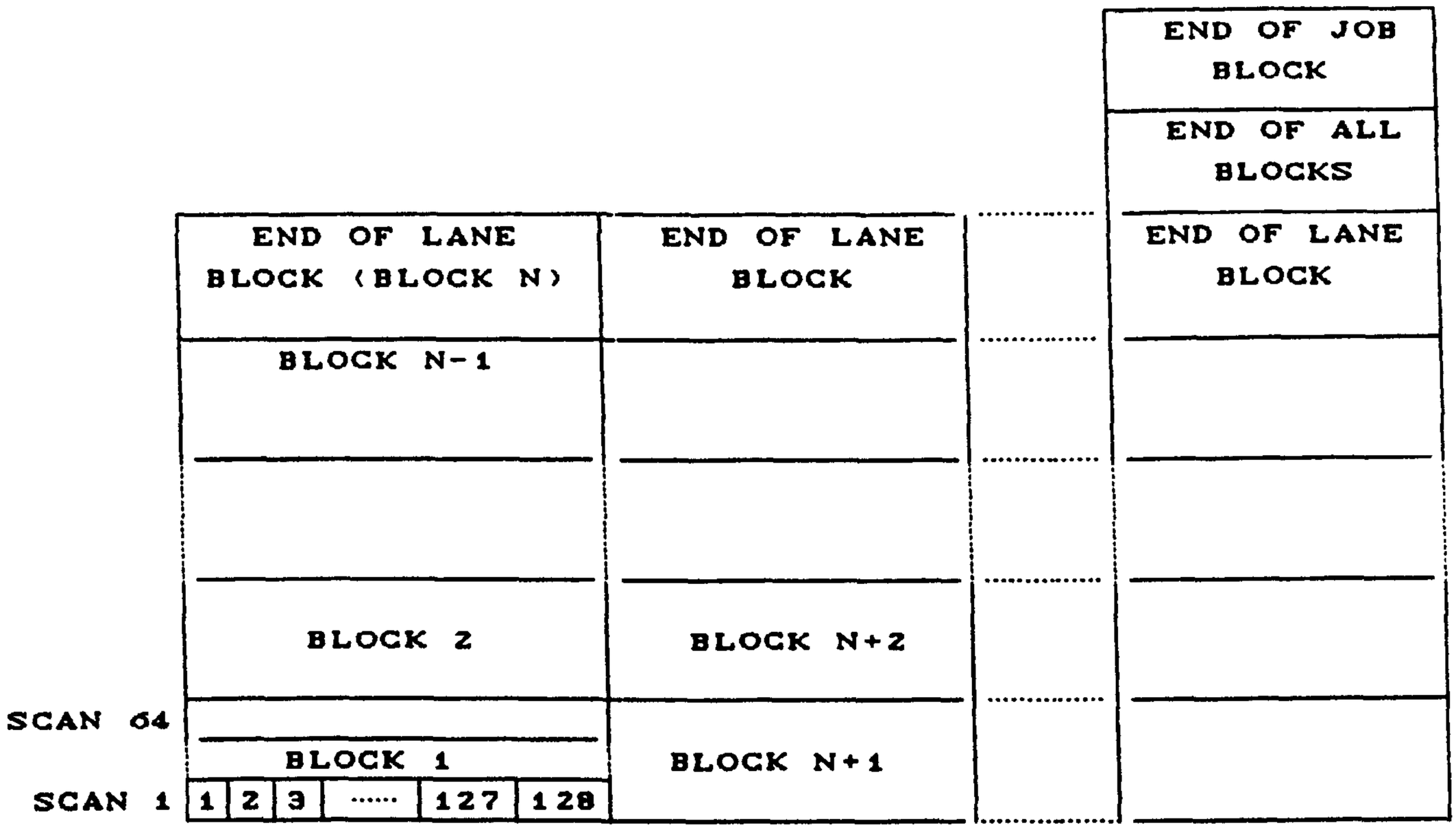
3.5.1 The File Structure

The structure of the data file (see Fig 3.13) is as follows:-

Block1	:	Housekeeping block
Block2	:	Data block for lane 1
"	:	"
"	:	"
Block n	:	End of lane block for lane 1
Block n+1	:	Data block for lane 2
"	:	"
"	:	etc... until the end of the file
"	:	"
Block p	:	End of lane block for last lane
Block p+1	:	End of area block
Block p+2	:	End of job block

3.5.2 The Data Block Format

As mentioned before, COSMOS scans a plate in lanes which span the whole y dimension of the plate. Each lane has a width of 128 pixels which is measured in the x-direction and is known as a scan. In mapping mode, the grey levels for 64 of these scans are stored consecutively in each data block as 8,192 words. The next data block in the file contains grey levels for the next 64 scans and so forth until the last block of the lane is reached (Fig 3.13).



EACH BLOCK CONTAINS 128 PIXELS*64 SCAN LINES=8,192 PIXELS

Fig. 3.13 THE STRUCTURE OF THE AERIAL PHOTO DATA DIGITIZED BY THE COSMOS MACHINE.

3.6 The Storage Hierarchy

To meet the requirement for quick access to the large volume of digitized data needed for processing, a hierarchical scheme was implemented. The scheme is shown in Fig 3.14. According to this scheme, magnetic tapes (at 6,250 b.p.i. density) were used as mass storage. Each holds approximately 50 Mbytes of the data. Since the data access time on magnetic tape is relatively long and more immediate access was required, the data were loaded onto the IBM 3090 hard disk. For this project, 50 Mbytes of disk storage was allocated to store the digital image data.

Even though the data on the disk can be reached in a few milliseconds, the access time is still too long for many of the digital image processing operations. Therefore, a rather large random access memory is necessary. This was either 10 Mbytes of RAM on the IBM 3090 or about 1 Mbytes of RAM in the memory buffer of the IBM 5080 graphics system (see Chapter 6).

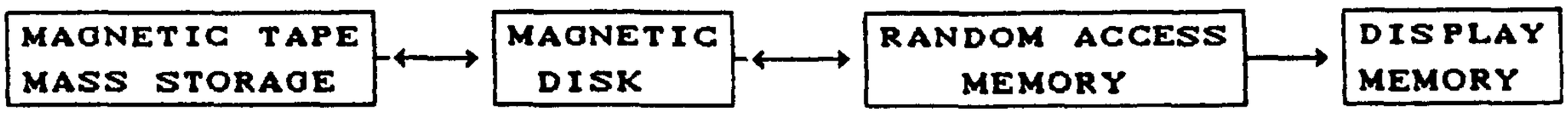


Fig. 3.14 STORAGE HIERARCHY

3.7 Conclusion

In the preceding chapter, the mathematical theory of image digitization was presented, while in this chapter, the actual procedures involved in the digitization of a pair of aerial photos by the COSMOS microdensitometer have been described. As mentioned in Chapter 2, the digital image inherits both pictorial degradation and geometric displacements/distortions. In the next chapter, an investigation into the characteristics and possible removal of the pictorial degradations is presented. Geometric distortions and displacements are considered later in Chapter 5.

CHAPTER FOUR

Experiments on Preprocessing and Pictorial Quality Improvement of Digital Aerial Photos

4.1 Introduction

Chapter 2 described how digital images are always degraded to some extent because of the effect of the camera imaging system and the digitizing microdensitometer. In the same chapter, these degradations were grouped into pictorial (grey level) degradation and metric displacements and distortions. Photogrammetrists are basically concerned with the geometric information which is mainly affected by the geometric displacements/distortions. However, the pictorial degradations might also affect the extraction of the metric information by reducing the interpretability of the image and hence the precision of measurement. The pictorial degradations may be introduced by factors such as the lens and the microdensitometer scanning spot; noise; change of terrain radiance with view angle; atmospheric effects; etc. Therefore, reliable metric information extraction may not be possible without a prior pictorial rectification which removes the grey level degradations and enhances the interpretability of the image.

In non-digital photogrammetry, pictorial rectification and geometric correction are carried out using analogue photographic processing techniques and stereo-plotting instruments respectively. However, in digital photogrammetry, these rectifications are dealt with quite differently, i.e, by implementing computerized digital solutions.

In the present project, the digital pictorial and metric rectifications, which are sometimes referred to as **image restoration**, were carried out with a view to reconstructing a three-dimensional stereoscopic model and subsequently extracting three-dimensional metric information using the reconstructed model surface. This chapter is devoted to a description of the digital pictorial rectification and some related experiments. Geometric rectification will be discussed in Chapter 5.

4.2 Digital Image Restoration

The problem of image restoration is to determine a correct image $f'(x,y)$ from the degraded recorded image $g(x,y)$ that is as close as possible, both geometrically and pictorially, to the ideal image

f(x,y) which is free from pictorial and metric degradations. The factors which introduce pictorial degradations on the image were discussed in Chapter 2. These are mainly the blurring effect of the camera and microdensitometer systems plus the influence of external factors such as the atmospheric effects; the change of terrain radiance with view angle, etc. which introduce further pictorial degradations to the image. Moreover, the images are always subject to some levels of noise. This may be regarded either as random noise or as non-random noise -- see Section 4.3. The relationship developed in Section 2.8.1 may be recalled.

$$g(x',y') = \mathbb{U}_p\mathbb{U}_g\left\{f(x,y)\right\} + n$$

This expression is a model for the imaging process decomposed into pictorial degradation (operator $\mathbb{U}_p$) and metric displacements/ distortions (operator $\mathbb{U}_g$); where the additional term n denotes the noise.

An estimate f'(x,y) of the original ideal non-degraded image f(x,y) may be obtained by:

$$f'(x,y) = \mathbb{U}_p^{-1}\mathbb{U}_g^{-1}\left\{g(x',y')\right\} - n \qquad (4.1)$$

where $\mathbb{U}_p^{-1}$ and $\mathbb{U}_g^{-1}$ are the inverse operators for the pictorial and metric rectifications respectively. Since, in the particular case of this research work, the data have not been acquired directly in digital form (i.e. the analogue image has been formed first and has then been digitized), as mentioned in Chapter 2, the pictorial degradations of the digital image are considered to be the result of two different convolution operations:

1. the convolution of the incoming signal to the camera system with the camera system psf; and
2. the convolution of the input analogue image with the microdensitometer psf.

These two convolution operations may be combined to give a single convolution operation. Throughout the remainder of this chapter, the operator $\mathbb{U}_p$ represents a combination of the two convolutions mentioned above. The operator $\mathbb{U}_p$ also represents the external sources of pictorial degradation such as the change of terrain radiance, etc.

Before describing how the inverse operator U_p^{-1} is implemented and how n is removed, it is necessary to examine in some detail the types and characteristics of the noise present in the image.

4.3 Noise

Generally speaking, an actual image consists of a set of non-random grey level variations which have been contaminated by additional signals called noise. A measure of the contamination of the actual image may be estimated using the **signal-to-noise ratio** which may be calculated as,

$$SNR = \sigma_g^2 / \sigma_n^2 \qquad (4.2)$$

where σ_g^2 and σ_n^2 are the actual image and the noise variances respectively. The higher the value of the σ_n^2 , the smaller the signal-to-noise ratio and the lower the image quality.

Images are subject to many different types of noise. Some of these are dependent on the image signal (**signal dependent noise**). In general, the noise generated during image sensing and recording, is signal dependent (Moik, 1980). Noise due to photographic graininess may be quoted as an example of signal dependent noise (Rosenfeld and Kak, 1976).

Another kind of noise is **signal independent noise**. The noise generated by a vidicon television camera when an image is scanned, is usually independent of the strength of the image signal (Rosenfeld and Kak, 1976) and hence it may be considered as signal independent noise.

For signal dependent noise, one can write, g = f + nf, where f and g are the actual and the contaminated images respectively. However, to facilitate the mathematical treatment in this thesis, noise is assumed to be signal independent and additive, i.e. g = f + n.

As mentioned earlier, noise may also be classified into **uncorrelated random(or, so-called "white") noise** and **coherent noise**. Periodic noise may be mentioned as an example of coherent noise. It appears as

periodic patterns, with varying amplitude, frequency and phase, superimposed on the image.

4.4 Preprocessing and Deconvolution Operations

The sources of pictorial degradations in digitized aerial photos do not have the same significance as those in images acquired by airborne or spaceborne sensors using solid-state detectors such as MSS and linear array sensors. Nevertheless, depending on the quality of the analogue photo and the quality of the digitizing microdensitometer, the removal of the pictorial degradations from the digitized aerial photo becomes an essential step in the digital image restoration procedure.

Pictorial rectification may be performed in two stages:

1. Removing the atmospheric effects; different terrain radiance with view angle; and coherent noise. In this thesis, this stage is referred to as a **preprocessing** operation.
2. Removing the blurring effect of the camera and microdensitometer systems and random noise. This stage is referred to as a **deconvolution** operation.

Since in preprocessing and deconvolution operations, filtering techniques are implemented, before considering these operations, a brief introduction to digital filters needs to be given.

4.5 Digital Filtering

Filtering is a basic image processing operation used in the reduction or elimination of pictorial degradation and in enhancement of the images. A linear filter is an LSI system (Section 2.3.1) that modifies the spatial frequency characteristics of an image. As was demonstrated in Section 2.4.3, the effect of any LSI system can be represented by a convolution operation. Therefore, linear filtering may be described in the space domain by Eq. 2.14 which is repeated here for convenience,

$$g(x,y) = \int_{-\infty}^{\infty}\int f(u,v)h(x-u,y-v)dudv$$

or in the frequency domain by

$$G = FH$$

where f is the input image with Fourier transform F; h is the impulse response of the filter with Fourier transform H (called the filter transfer function); and g is the output (filtered image) with Fourier transform G. Filters are conceptually easier to specify and apply in the frequency domain rather than in the space domain, because their representation in the frequency domain is simpler and convolution is replaced by multiplication (Section 2.5.5). Therefore, all of the filtering operations carried out in the experiments described in this thesis have been performed in the frequency domain.

In order to gain more insight into the digital filter's characteristics, some important filters are briefly described. These are: **Ideal low-pass, Ideal high-pass, Ideal band-pass and Notch filters**

a. Ideal Low-Pass Filter

An ideal low-pass filter transfer function (Fig. 4.1a) is defined by

$$H(U,V) = H(W) = \begin{cases} 1 \text{ for } W \leq W_c \\ 0 \text{ for } W > W_c \end{cases} \qquad (4.3)$$

where H(U,V) is a circularly symmetric low-pass filter; and H(W) is the same low-pass filter in a polar coordinate system with $W=(U^2+V^2)^{1/2}$ This filter suppresses all frequencies above the cut-off frequency W_c.

b. Ideal High-Pass Filter

An ideal high-pass filter transfer function (Fig 4.1b) is defined by

$$H(U,V) = \begin{cases} 1 \text{ for } W \geq W_c \\ 0 \text{ for } W < W_c \end{cases} \qquad (4.4)$$

It suppresses all frequencies below the cut-off frequency.

c. Ideal Band-Pass Filter

A band-pass filter transfer function suppresses all

frequencies outside its pass band (Fig. 4.1c), i.e.

$$H(U,V) = H(W) = \begin{cases} 1 & \text{for } W_1 \leq W \leq W_2 \\ 0 & \text{elsewhere} \end{cases} \qquad (4.5)$$

It passes the frequencies only between W_1 and W_2.

d. Notch Filter

A notch filter transfer function (Fig. 4.1d) is the inverse of a band-pass filter transfer function; it suppresses all frequencies in a specified band, i.e.,

$$H(U,V) = H(W) = \begin{cases} 0 & \text{for } W_1 \leq W \leq W_2 \\ 1 & \text{elsewhere} \end{cases} \qquad (4.6)$$

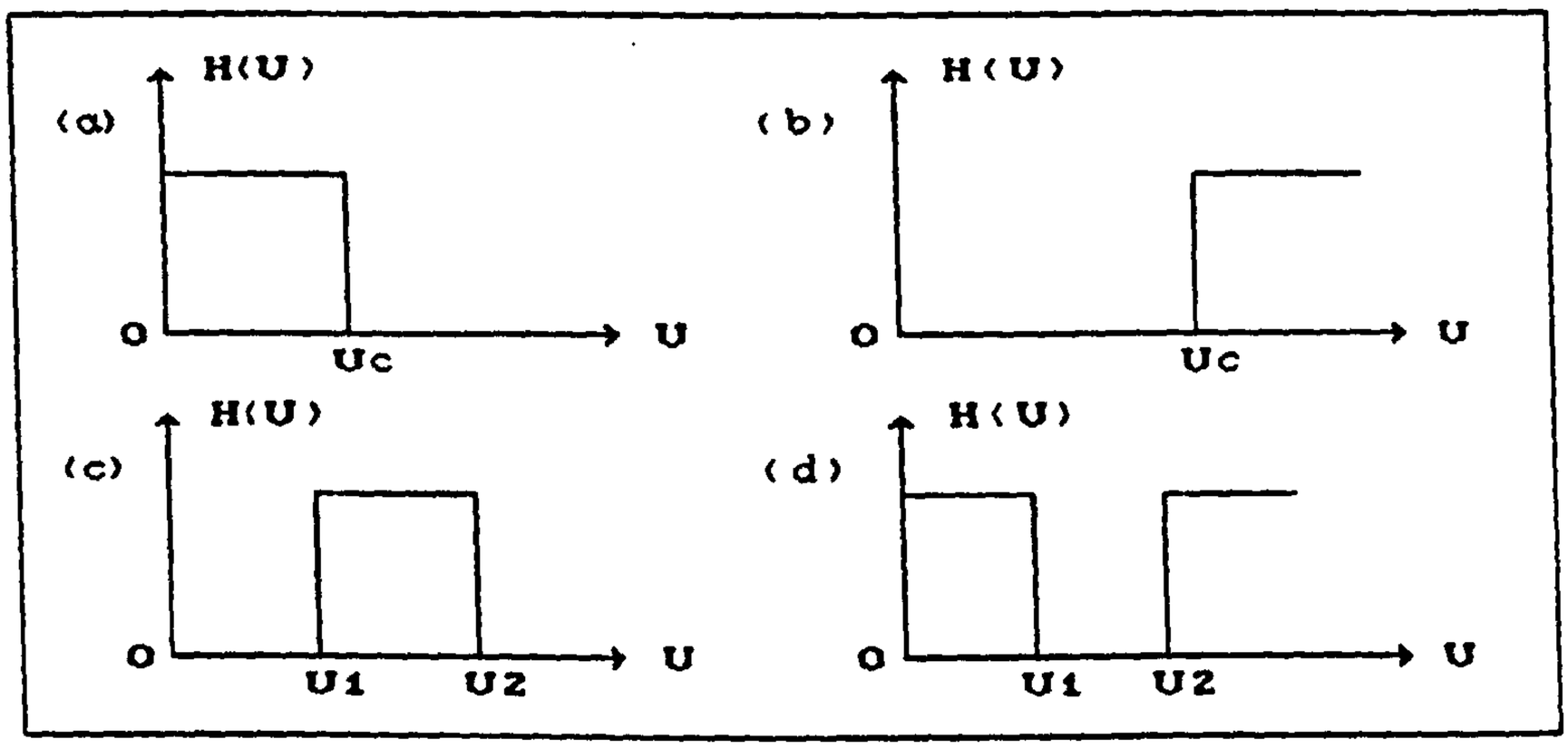


Fig. 4.1 a. IDEAL LOW-PASS FILTER TRANSFER FUNCTION
b. IDEAL HIGH-PASS FILTER TRANSFER FUNCTION
c. IDEAL BAND-PASS FILTER TRANSFER FUNCTION
d. IDEAL NOTCH FILTER TRANSFER FUNCTION

(THE FILTER TRANSFER FUNCTIONS ARE PRESENTED IN ONE DIMENSION)

4.6 Preprocessing Operation

As mentioned before, the necessity to carry out preprocessing operations on digitized aerial photos is often not as urgent as it is for remotely sensed image data acquired by solid state sensors, e.g. as used in MSS scanners. For example, different lighting conditions due to variable solar elevation cycles, which are a significant source

of pictorial degradation for remotely sensed images, are usually insignificant for aerial photos where the time interval between exposures is relatively short. The atmospheric effect is minimized for aerial photos by acquiring the image in reasonable weather conditions. The effect of periodic noise is reduced to its lowest level due to the relatively uniform response of the aerial film emulsion and the microdensitometer's photomultiplier tube. However, although these degradations for digitized aerial photos are small, they do exist. They prevent the same object appearing with the same density value in the left and right images and hence, in some applications such as automatic image correlation (Chapters 8 and 9), their removal may improve the correlation results. Moreover, there are certain situations where the pictorial degradations mentioned above do become significant. These are mainly:

- where the image is acquired under poor illumination and weather conditions;
- since the resolution of the camera lens falls off from the centre of the photograph outwards, the same points appearing in different positions on the photograph may have different density values;
- where there is illumination variation due to the different viewing angles; and
- where a low quality image digitizer is used.

4.6.1 Equalization of Different Density Values for the Same Objects

To reduce the effect of the two different responses for the same object points, different methods may be applied. One method is to calibrate the effect of contributing factors (e.g. the atmospheric effect; lens resolution fall-off; viewing angle; etc.) for each position on the image and to eliminate the effect mathematically. An alternative approach for removing the degradations caused by the above mentioned factors, which is easier to implement and is perhaps more effective than the first approach, is to normalize the left image patch and the corresponding right one. This may be carried out by adding half the difference of the mean values of the two corresponding image patches to the elements of the lighter image patch. In the

present project, the normalization was carried out during the image correlation stage and will be discussed in Chapter 9 where the correlation algorithm which has been implemented will be discussed.

4.6.2 Periodic Noise Removal

The essence of coherent noise removal is to isolate and remove the identifiable noise components in a manner that involves the minimum change and does the minimum amount of damage to the actual image data.

Coherent noise removal may be performed in either the space or frequency domains. For example, a technique to correct a periodic noise pattern in the space domain is to change the grey level of each noisy pixel to the average of the grey levels of neighbouring pixels. Although this method may provide global improvement, it may also introduce artifacts into certain details. However, those methods which perform noise removal in the frequency domain seem to offer a more elegant approach. The principle of the frequency domain noise removal approach is described below.

Let g(x,y) be an image with coherent additive noise n and G(u,v) be the Fourier transform of the noisy image g(x,y), i.e:

$$\begin{aligned} g(x,y) &= f(x,y) + n \\ G(U,V) &= \mathbb{F}\{g(x,y)\} = \mathbb{F}\{f(x,y)+n\} \\ &= F+N \end{aligned} \qquad (4.7)$$

where f(x,y) is the original image without noise; F and N are the Fourier transforms of f and n respectively. Now N has all its energy concentrated at a set of small spots at specific frequency positions. These positions depend on the frequencies at which the noise appears on the image (see the Fourier transform, Section 2.5.3). For a periodic line pattern noise, the small spots will be located along a line perpendicular to the direction of the noise lines (Rosenfeld and Kak, 1976). These small spots may be suppressed by removing or setting their amplitudes to zero using an appropriate two-dimensional filter H(u,v) i.e.,

$$F(U,V) = G(U,V)H(U,V)$$

By taking the inverse Fourier transform, a smoothed picture is obtained in which n has been deleted, while f is relatively unaffected, i.e.

$$f(x,y) = \mathbb{F}^{-1}\left\{F(U,V)\right\}$$

4.7 Deconvolution Operations

The problem of deconvolution is the determination of an estimate f' of the original image f, given the preprocessed image g_n and knowledge about the imaging system and microdensitometer's psf. The problem is, therefore, to find the inverse operator $\mathbb{U}_p^{-1}$ such that

$$f'(x,y) = \mathbb{U}_p^{-1}\left\{g_n(x,y)\right\} \qquad (4.8)$$

The techniques of deconvolution have been well-known and well developed in electrical filter design and time series analysis. However, the detailed treatment of the subject does not lie within the scope of this thesis. Since in digital photogrammetry, deconvolution operations may be essential in some situations (e.g. in the case of a blurred image), a brief introductory description of the subject seems appropriate.

To determine the inverse operator $\mathbb{U}_p^{-1}$ (Equation 4.8) for a deconvolution operation, the convolution operation (Eq. 2.14) may be rewritten here in the presence of noise as,

$$g_n(x,y) = \int_{-\infty}^{\infty}\int f(u,v)h(x-u,y-v)dudv+n(x,y) \qquad (4.9a)$$

where h is the total system psf for the imaging and densitometer systems; and n is the random noise which has not been removed during the preprocessing operations. Discretization of the above linear model leads to the matrix vector equation (the details are not given here, see Pratt, 1978; Moik, 1980):

$$g_n = \mathbb{B}\ f + n \qquad (4.9b)$$

where g_n , f, and n are the vectors created from the digitized image, original image, and noise fields respectively; and B is a matrix

resulting from the system psf. As mentioned above, the solution of the deconvolution problem is to find an estimation f′ which is as close as possible to the original image f. The estimation f′ may be found using two different approaches -- the so-called **deterministic** and **statistical (stochastic)** image models. (Rosenfeld and Kak, 1976; Castleman, 1979; Moik 1980; Fallvik, 1986):

The deterministic model is based on the assumption that the fundamental nature of the image is known (i.e the image is assumed to be free of noise). This leads to the solution of the system of linear Eq. 4.9b while the noise variable n is cancelled out from the equation. The statistical model, on the other hand, implies the estimation of an image subject to random disturbances. Each of these approaches leads to different solutions, namely the **Inverse filter** and the **Wiener filter** respectively. These approaches will be discussed briefly in the following sections.

4.7.1 Inverse Filter

Assuming that the degraded image g_n and the original image f satisfy the Eq. 4.9b, in the absence of noise, Eq. 4.9b may be written as,

$$f' = \mathbb{B}^{-1} g_n \tag{4.10}$$

where f′ is an estimate of the original image f. The numerical solution of Eq. 4.10 is in general impractical or impossible. Suppose the image has dimensions of 512 by 512 pixels, then inversion of a 262,144 by 262,144 matrix is required (Moik, 1980). However, in the absence of noise, the Fourier transform of Eq. 4.9a leads to,

$$G(U,V) \simeq F(U,V)H(U,V)$$

or,

$$F'(U,V) = G(U,V)/H(U,V) \tag{4.11}$$

and

$$f'(x,y) = \mathbb{F}^{-1}\left\{F'(U,V)\right\}$$

where F′ is the Fourier transform of the estimated image; and H is the Inverse filter transfer function.

Eq. 4.11 implies that, if H(U,V) is known, f(x,y) can be restored by multiplying the Fourier Transform G(U,V) of the degraded picture by 1/H(U,V) and then deriving the inverse Fourier transform.

The deconvolution approach formulated above has its own shortcomings, namely:

1. There may be points in the UV-plane where H(U,V)=0. causing the inverse filter to approach infinity.
2. Eq. 4.10 is ill-conditioned (i.e. the presence of noise is neglected). This means that small perturbations in g_n can produce undesirable effects in the image restored by the Inverse filter. Therefore, the Inverse filter should be used mainly to restore images with a high signal-to-noise ratio.

To reduce the noise amplification effect of the Inverse filter, a possible solution is to prevent the Inverse filter from going above a certain limiting value (Castleman, 1979; Moik, 1980)), i.e,

$$H_i(U,V) = \begin{cases} 1/H(U,V) & \text{for} \quad U,V \leq S \\ H_m & \text{for} \quad U,V > S \end{cases} \qquad (4.12)$$

where $H_i(U,V)$ is the modified Inverse filter (Fig. 4.2). The limitation S in Eq. 4.12 depends on the signal-to-noise ratio of the image to be processed and is determined empirically.

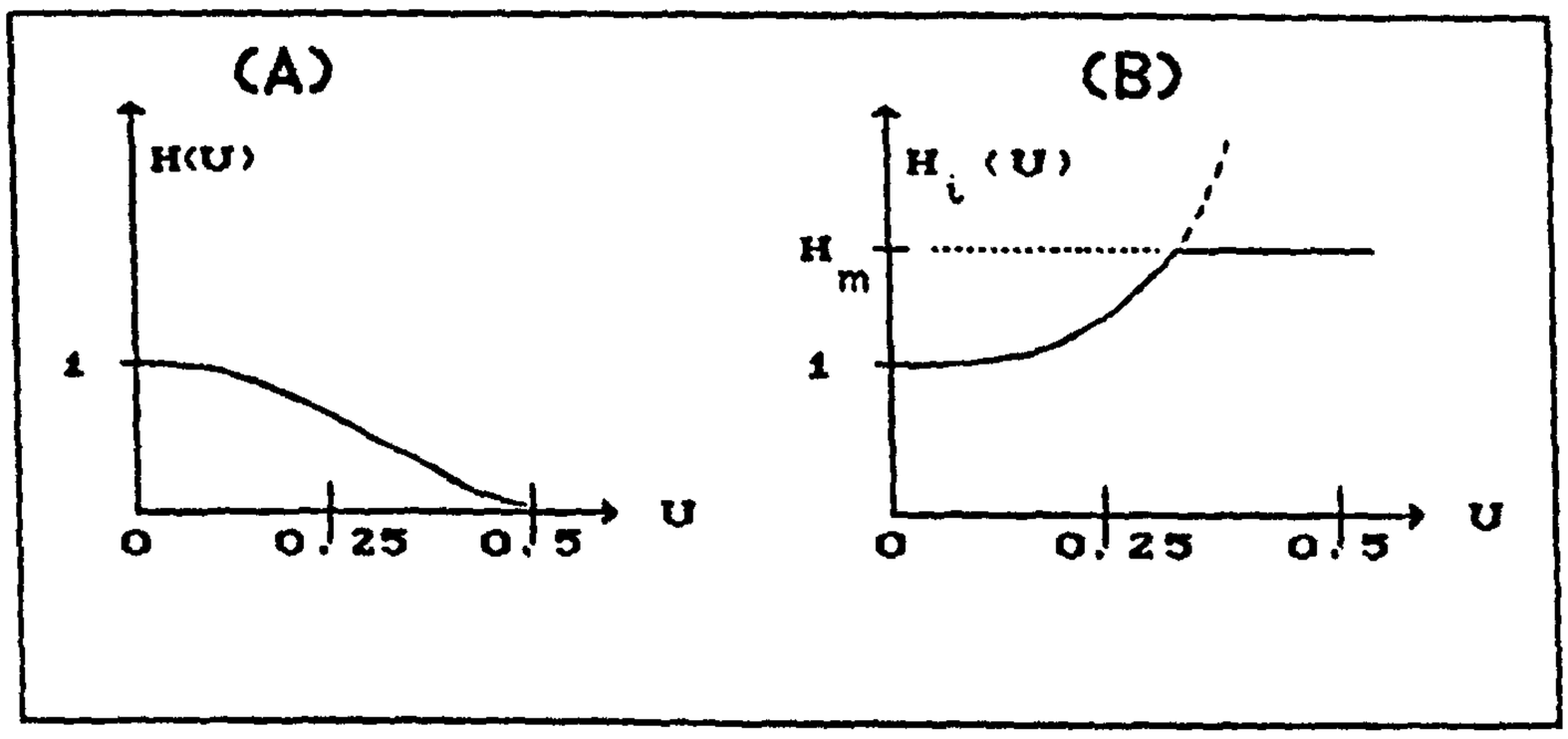


Fig. 4.2 (A)- INVERSE FILTER MTF
(B)- MODIFIED INVERSE FILTER MTF
(ONE-DIMENSIONAL REPRESENTATION)

4.7.2 Wiener Filter

To overcome the noise amplification problem of the Inverse filter discussed above, the stochastic approach for image restoration may be implemented by employing the Wiener filter (sometimes referred to as the optimal filter), as follows:

If a digital image g(x,y) has been generated by digitizing the original hard copy image f(x,y) and if it has been contaminated by an additive random noise function n(x,y) (see Fig. 4.3), then the intention is to design a linear filter y(x,y) which reduces the contaminating noise as much as possible and generates an image f'(x,y) which resembles as closely as possible the original form, f(x,y).

As it will be shown later, the Wiener filter requires some prior information regarding the power spectra[1] of the image g and the noise n.

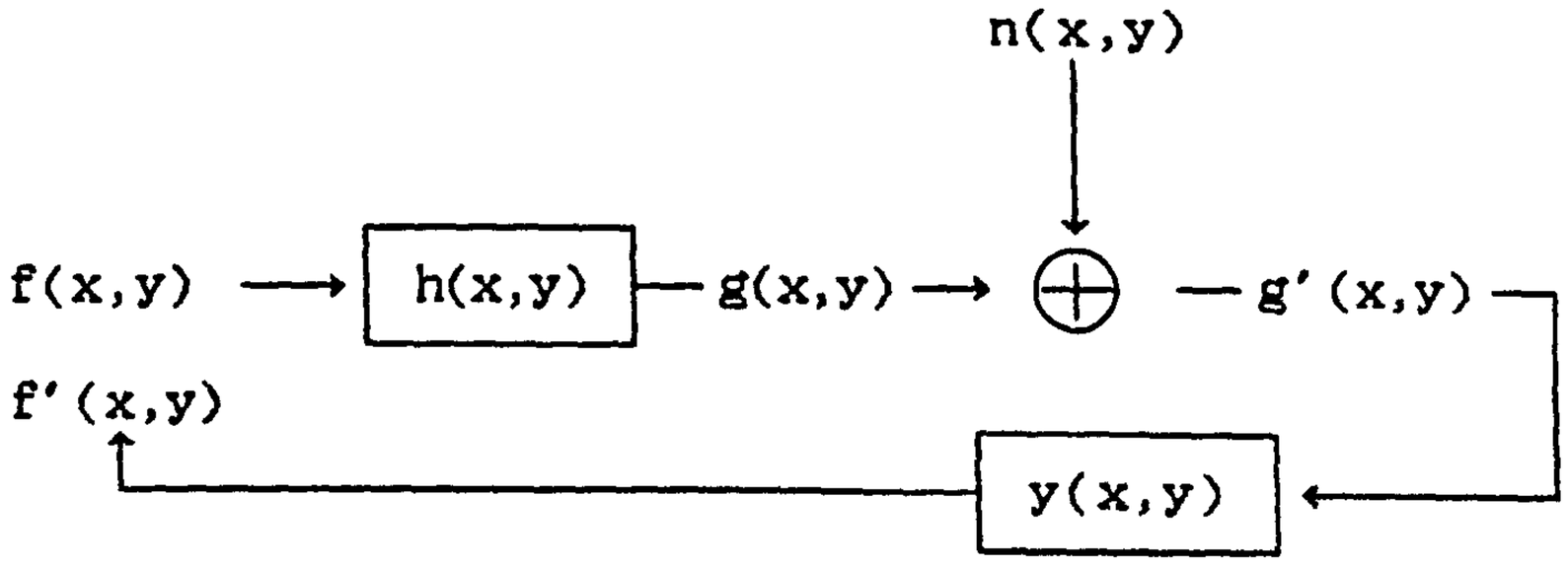


Fig. 4.3

To find out the optimal solution, the criterion of optimality is defined as follows:

The error at the output of the filter y(x,y) can be expressed as the difference between the actual degraded and the desired output, i.e,

1 *Given a function f(x,y), its autocorrelation function* $\mathbb{R}_{ff}(u,v)$ *is defined as,* $\int\int_{-\infty}^{\infty} f(x+u,y+v)f(x,y)dxdy$. *The Fourier transform of the autocorrelation function* $\mathbb{R}_{ff}$ *is called its power spectrum,* $\mathbb{P}_{ff}$, *and is given by,* $\mathbb{P}_{ff}(U,V) = \mathbb{F}\{\mathbb{R}_{ff}(U,V)\} = F(U,V)F^{*}(U,V) = |F(U,V)|^{2}$ *where F(U,V) is the Fourier transform of f(x,y); and * denotes the complex conjugate.*

$$e(x,y) = g(x,y)-f'(x,y)$$

or as a measure of the average error, the MSE may be used,

$$MSE = \mathbb{E}\left\{e^2(x,y)\right\} \qquad (4.13)$$

where $\mathbb{E}$ denotes the expectation operator. The optimality criterion is hence to minimize the MSE.

The problem can now be tackled as follows: Given the power spectra of g and n, a filter w must be determined so that the MSE is minimized.

It can be shown that the MSE is minimized if the following condition is satisfied (Rosenfeld and Kak, 1976; Castleman, 1979),

$$\mathbb{C}_{g'g} = w * \mathbb{R}_{g'g'} \qquad (4.14)$$

where $\mathbb{C}_{g'g}$ is the cross correlation function[2] of g' and g; and $\mathbb{R}_{g'g'}$ is the autocorrelation function of g' (see the footnote on the previous page).

By taking the Fourier transform of both sides, the following equation results:-

$$\mathbb{Q}_{g'g} = W\,\mathbb{P}_{g'g'}$$

or

$$W = \frac{\mathbb{Q}_{g'g}}{\mathbb{P}_{g'g'}} \qquad (4.15)$$

where $\mathbb{Q}_{g'g}$ is the cross power spectral density[3] of g' and g; $\mathbb{P}_{g'g'}$ denotes the power spectrum of g' ; and W is the Fourier transform of w. Eq. 4.15 is the filter which minimizes the MSE and thus accounts for random noise. However, it is also required to account for the undesired system psf h(x,y). In other words, it is desired to design a

---

2 *The cross correlation $\mathbb{C}_{g'g}$ of two functions g' and g is defined as,*

$$\mathbb{C}_{g'g}(u,v) = \int\int_{-\infty}^{\infty} g'(x,y)g(x+u,y+v)dxdy \quad \textit{(see Chapter 8)}$$

3 *The Fourier transform of the cross-correlation function $\mathbb{C}_{g'g}$ is called the cross power spectral density function $\mathbb{Q}_{g'g}$.*

linear filter Y(see Fig.s 4.3 and 4.4) that simultaneously removes the undesired psf and takes the noise into consideration.

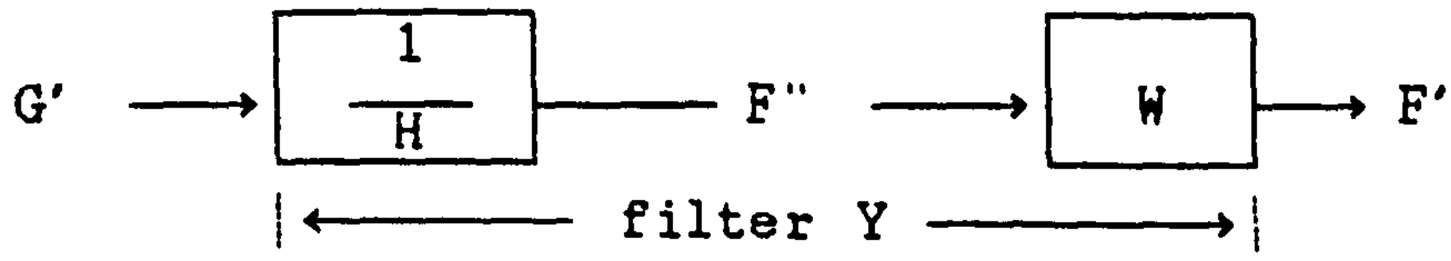


Fig. 4.4 WIENER DECONVOLUTION FILTER

In Fig. 4.4, (1/H) is the Inverse filter; and W is the filter derived by Eq. 4.15. The following paragraphs give a brief description as to how the filter Y can be generated.

It can be shown (see Castleman, 1979) that if the image and the noise are uncorrelated and if either the image or the noise have zero mean, then Eq. 4.15 becomes

$$W = \frac{\mathbb{P}_{gg}}{\mathbb{P}_{gg} + \mathbb{P}_{nn}} \qquad (4.16)$$

where $\mathbb{P}_{nn}$ is the noise power spectrum.

Figs. 4.3 and 4.4 imply that the spectrum of the observed image g'(x,y) (recalling the convolution operation and convolution theorem) is

$$G' = F\,H + N \qquad (4.17a)$$

where G' and F denote the Fourier transform of g'(x,y) and f(x,y) (original image) respectively; and N is the Fourier transform of the noise. Assuming that H has no zeros, the input F" (the Fourier transform of f"(x,y)) to the filter W is (see Eq. 4.11 and Fig. 4.4),

$$F" = G' .(1/H)$$

Replacing Eq. 4.17a in the above equation yields,

$$F" = F+(N/H) \qquad (4.17b)$$

Having F" or f"(x,y), its equivalent in the space domain, as the input

to the filter W, Eq. 4.16 becomes

$$W = \frac{\mathbb{P}_{ff}}{\mathbb{P}_{ff} + \mathbb{P}_{kk}} \tag{4.18}$$

where k = (N/H). Since, $\mathbb{P}_{kk} = |N/H|^2$ (see the footnote of the page 72), then

$$W = \frac{\mathbb{P}_{ff}}{\mathbb{P}_{ff} + |N/H|^2} \tag{4.19}$$

Finally, combining the Inverse filter H and the filter W (see Fig. 4.4) and rearranging the terms, yields

$$Y = W/H = \frac{1}{H} \frac{|H|^2}{|H|^2 + \mathbb{P}_{nn}/\mathbb{P}_{ff}} \tag{4.20}$$

Eq. 4.20 is the final formulation of the Wiener restoration filter. There is no ill-conditioned behaviour associated with the Wiener filter (such as that of the Inverse filter). Thus, a restored image can be generated even if the matrix $\mathbb{B}$ (Eq. 4.10) is singular and noise is present. However, the Wiener filter requires some prior information about f and n.

There is another restoration technique, refered to as **constrained filtering**, which requires information about the variance of noise only and allows the designer additional control over the restoration process (Andrews and Hunt, 1977). Constrained filtering was not implemented in the present project and hence is not discussed here.

4.8 Determination of the PSF

It is clear that, for both the Inverse and Wiener restoration filters, some prior knowledge of the system psf is required. The total psf or the corresponding transfer function of the camera or microdensitometer systems may be measured directly using a sine wave or target bars (Castleman, 1979; Manual of Photogrammetry, 1980). In some cases, however, it is impractical or impossible to calibrate the imaging

system and the image digitizer under the same conditions in which a particular degraded image was recorded. This is true for motion blur, for example, and also in those cases where a digital image is presented for restoration and the original densitometer is unavailable for calibration. In such instances, the psf may be determined from the image itself (Castleman,1979; Rosenfeld and Kak, 1976). For instance, if there is any reason to believe that the original image contains a sharp point, then the image of the point in the degraded image is the psf (see Section 2.4.2).

There are some other cases in which the physical phenomenon underlying the degradation can be modelled to determine the psf. As an example, one can consider the degradation caused by the relative motion between the camera and the scene. The final formulation of the psf h(x,y) of the image motion is given by Rosenfeld and Kak (1976) as the following:

$$h(x,y) = (1/v^2T)\prod (x/vT)$$

where T is the duration of the exposure; and v is the motion velocity in the x-direction (the displacement along the y-direction is assumed to be zero).

4.9 Digital Image Enhancement

So far, various methods for correcting the pictorial degradation (i.e. methods of reconstructing the original scene) have been considered. However, there are some other approaches by which the pictorial information of the image may be enhanced without, necessarily, trying to restore (i.e. reconstruct) the original scene. The goal of these image enhancement techniques is to aid the human analyst in the extraction and interpretation of pictorial information.

For digitized aerial photographs, the enhancement methods may be divided into: (i) contrast enhancement and (ii) edge enhancement. For digitized aerial photos, unlike the remotely sensed data acquired by scanners, image enhancement does not seem to be a necessity in all circumstances. However, since these enhancement techniques are sometimes essential for displaying the digitized aerial photos and for

linear feature extraction, a brief introduction to each is given in the following sections.

4.9.1 Contrast Enhancement

The goal of contrast enhancement is to produce a picture that uses the full dynamic range of a display device. The contrast characteristics of an image are influenced by factors such as camera exposure settings, solar lighting effects, the digitized image quantization levels, etc.

The grey scale transformations can be applied to the image to stretch the contrast to the full dynamic range of the display device, using the different approaches of linear contrast stretch; non-linear contrast stretch; histogram equalization; etc. In the linear contrast stretching approach, the range of the maximum to the minimum grey levels of the image is mapped linearly into the full range of the screen density levels. For a non-linear stretch, the same range of densities are mapped non-linearly. The histogram equalization approach is based on the principle that each histogram grey level in the displayed image must contain an approximately equal number of pixel values. If this is done, the entropy of the image (a measure of the information content of the image) will be increased (Mather, 1987). This results in a better representation of the displayed image.

Since the test image used in this research work was digitized into 14-bit transmission values (see Section 3.5), the density range of the image was not in conformity with the dynamic range of the display device. The functions RANGE and RRANGE (robust grey-value reranging) provided by the IAX image processing system (see Chapter 6) were used to rescale the grey values of the image. The RANGE function linearly re-scales the values of the elements of a vector or image to lie between the user's specified minimum and maximum values. The default range for the minimum and maximum values is 0-255. The RRANGE returns a version of the input image in which the pixel values have been altered to maximize the contrast (IAX manual). The result of applying the RRANGE function to the original image is given in Fig. 4.5.

(a) (b)

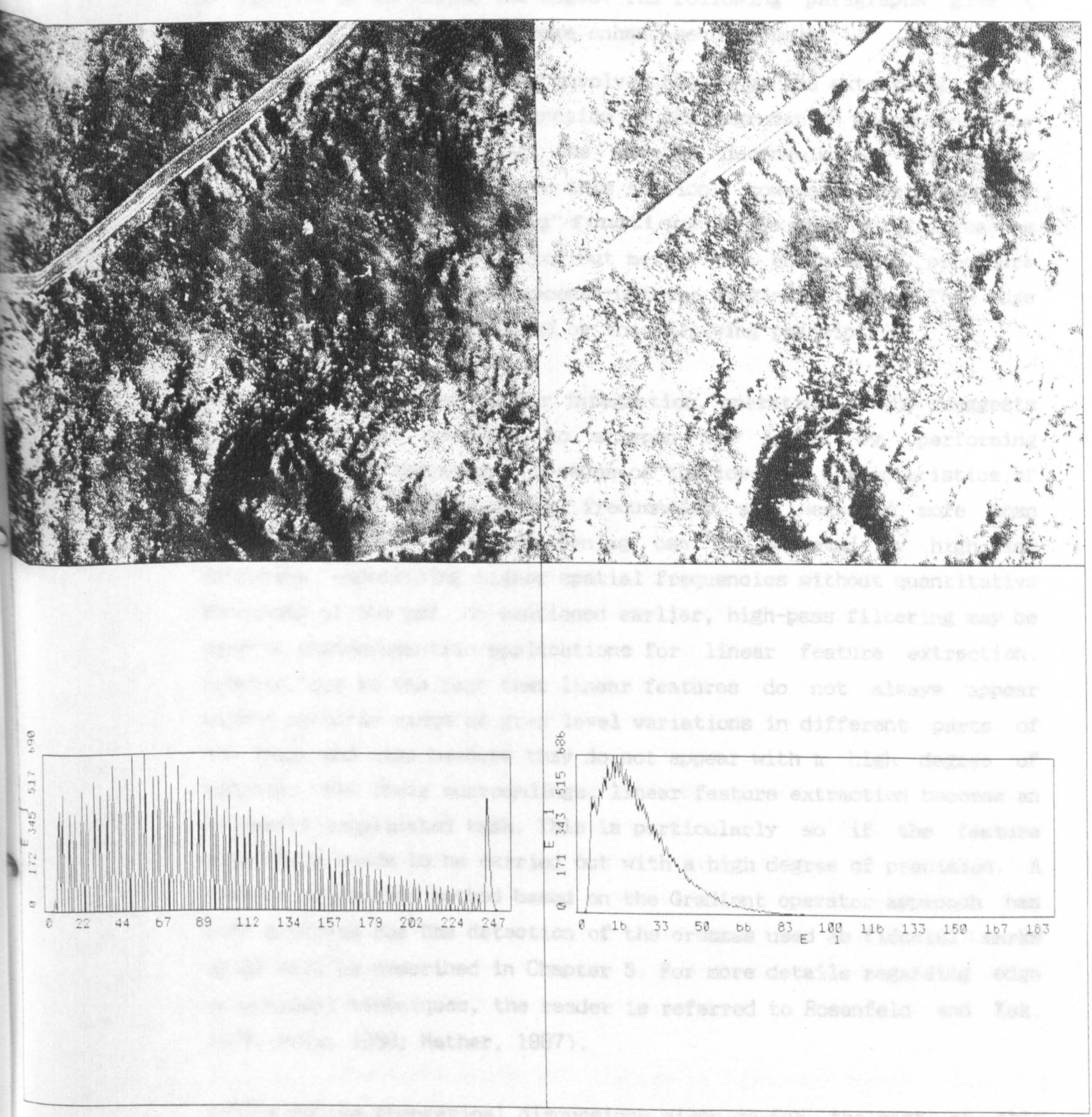


Fig.4.5 a. The rescaled grey level image and its histogram.
b. The original image and its histogram.

4.9.2 Edge Enhancement Techniques

The pictorial analysis and interpretation of a digital image may also be improved by enhancing the edges. The following paragraphs give a brief review of some of the edge enhancement methods.

Pictorial degradation usually involves blurring, the extent of which is described by the spread function of the degradation operation. The deblurring operation using the inverse deconvolution filter was discussed in Section 4.7.1. In this section, however, deblurring is considered using **"sharpening" functions**. These deblurring approaches are not a restoration operation but merely an enhancement procedure which does not necessarily reconstruct the original image. The edge enhancement operation is based on the following principle.

Blurring is an averaging, or integration, operation; this suggests that it may be possible to sharpen the image by performing differentiation operations. Because of the low-pass characteristics of imaging systems, higher spatial frequencies are weakened more than lower frequencies. Thus sharpening can be achieved by high-pass filtering, emphasizing higher spatial frequencies without quantitative knowledge of the psf. As mentioned earlier, high-pass filtering may be used in photogrammetric applications for linear feature extraction. However, due to the fact that linear features do not always appear with a specific range of grey level variations in different parts of the image and also because they do not appear with a high degree of contrast with their surroundings, linear feature extraction becomes an extremely complicated task. This is particularly so if the feature extraction needs to be carried out with a high degree of precision. A feature extraction method based on the Gradient operator approach has been employed for the detection of the crosses used as fiducial marks which will be described in Chapter 5. For more details regarding edge enhancement techniques, the reader is referred to Rosenfeld and Kak, 1976; Moik, 1980; Mather, 1987).

Following the theoretical discussions given so far, the rest of this chapter is devoted to the experiments carried out in the present project for the removal of pictorial degradation.

4.10 Experiments to Remove Pictorial Degradation

As mentioned above, the pictorial degradations were not a serious problem in the present project. Nevertheless, the digital images acquired for photogrammetric purposes are sometimes digitized with low quality microdensitometers. Also since other types of imagery (e.g. those produced by CCD cameras) might be used for photogrammetric purposes, and also due to the fact that image restoration may improve the results of the automatic image correlation techniques (Chapters 8 and 9), various experiments concerned with the removal of pictorial degradation were carried out.

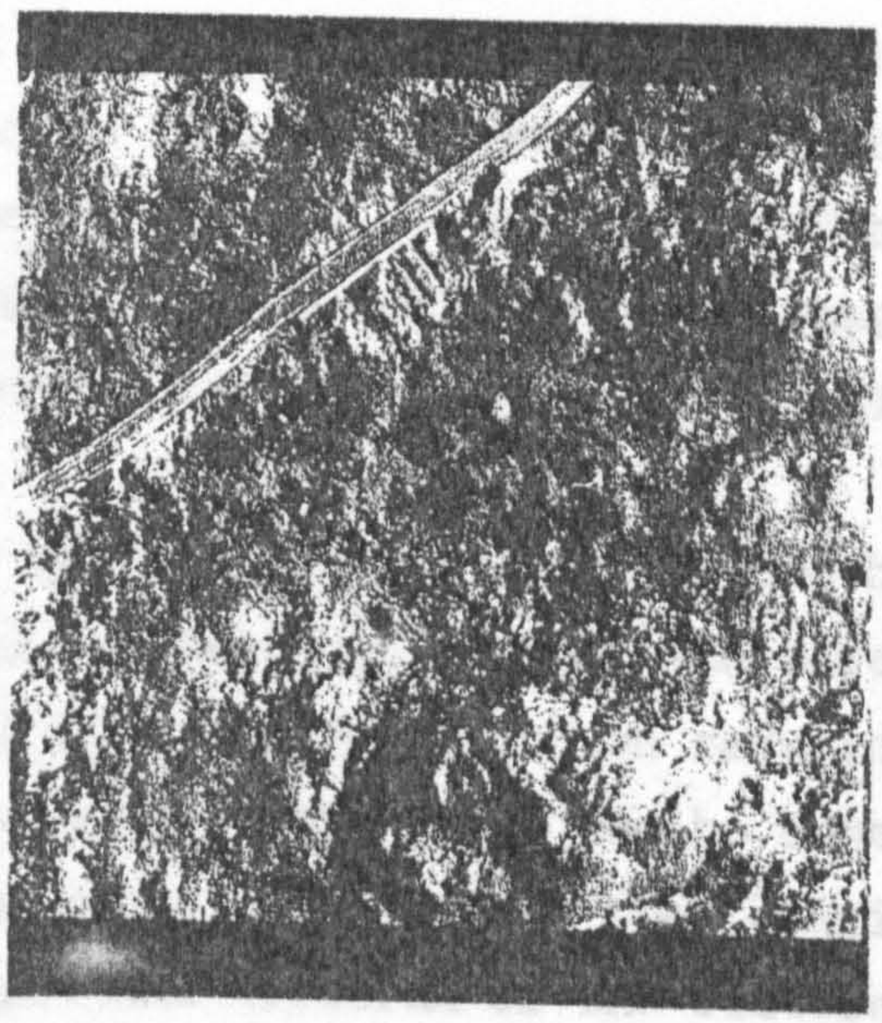
Throughout the tests, the frequency domain manipulation of the data was carried out using the Fourier transform routine supplied by the IAX image processing system (a more detailed description regarding the IAX system is given in Chapter 6).

4.10.1 Experiments Concerned with the Removal of Periodic Noise

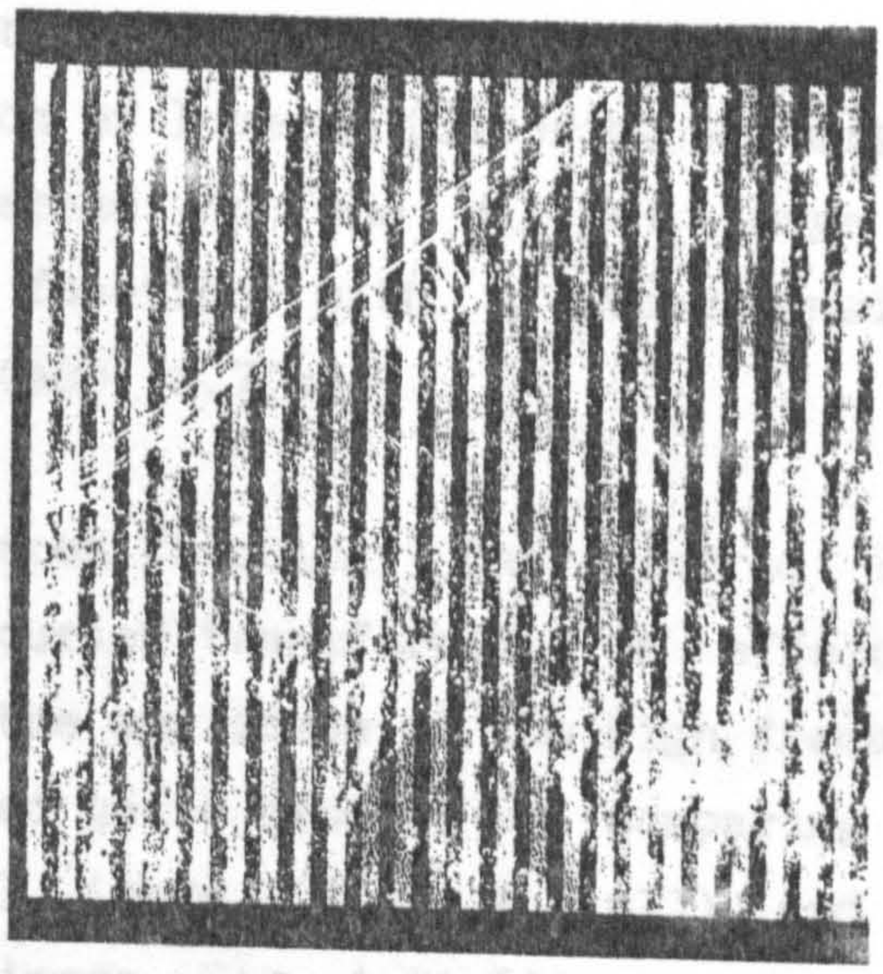
To examine empirically the removal of periodic noise in the Fourier domain, a portion of the digital image was selected and a simulated periodic strip noise was added to the image (Figs. 4.6(a1) and 4.6(a2)). As expected, the Fourier transform of the periodic noise showed a set of spots (impulses) perpendicular to the direction of the noise. To remove the noise, the noisy image was Fourier transformed and a notch filter (see Section 4.5d) was applied subsequently to set the amplitudes of the noise spots to zero (Fig. 4.6(b1)). The inverse Fourier transform was then carried out which produced an image without noise (Fig. 4.6(b2)). There is, however, some loss of contrast in the reconstructed image. This is due to the removal of the amplitude of some low frequency contents of the frequency spectrum of the digital image by the notch filter.

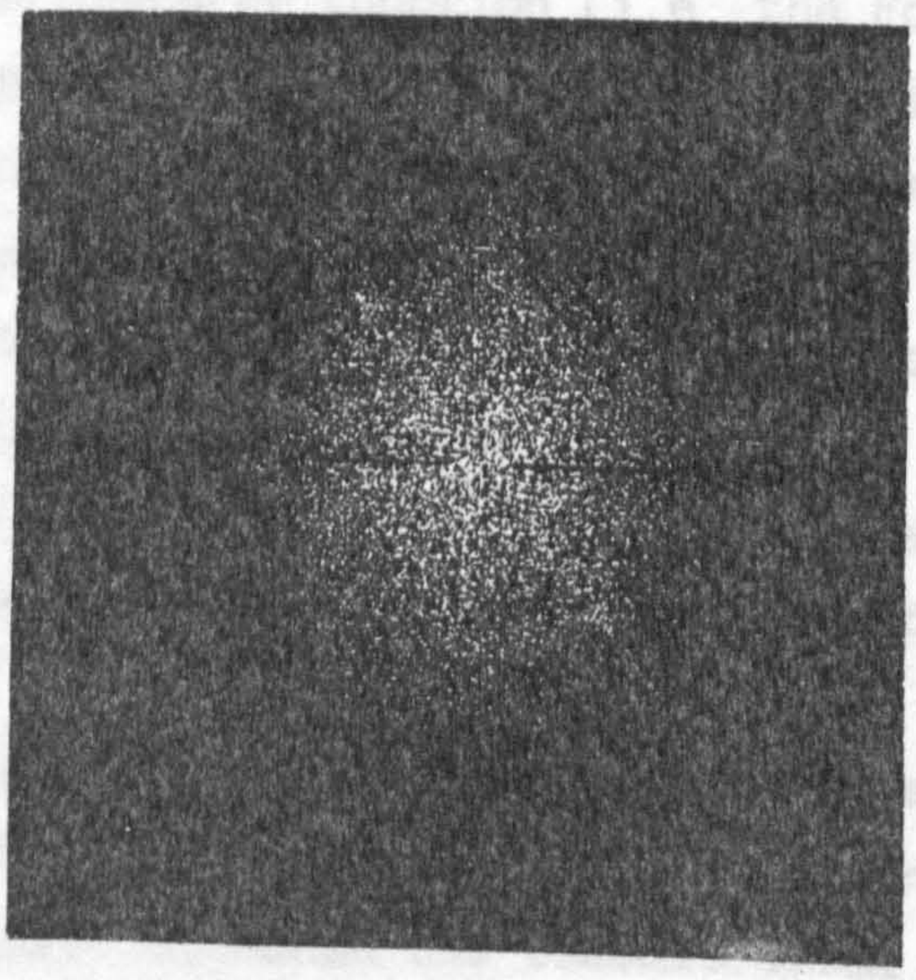
The above experiment shows that the use of Frequency domain filtering is a strong and reliable approach to implementing the removal of periodic noise with a small amount of information loss.

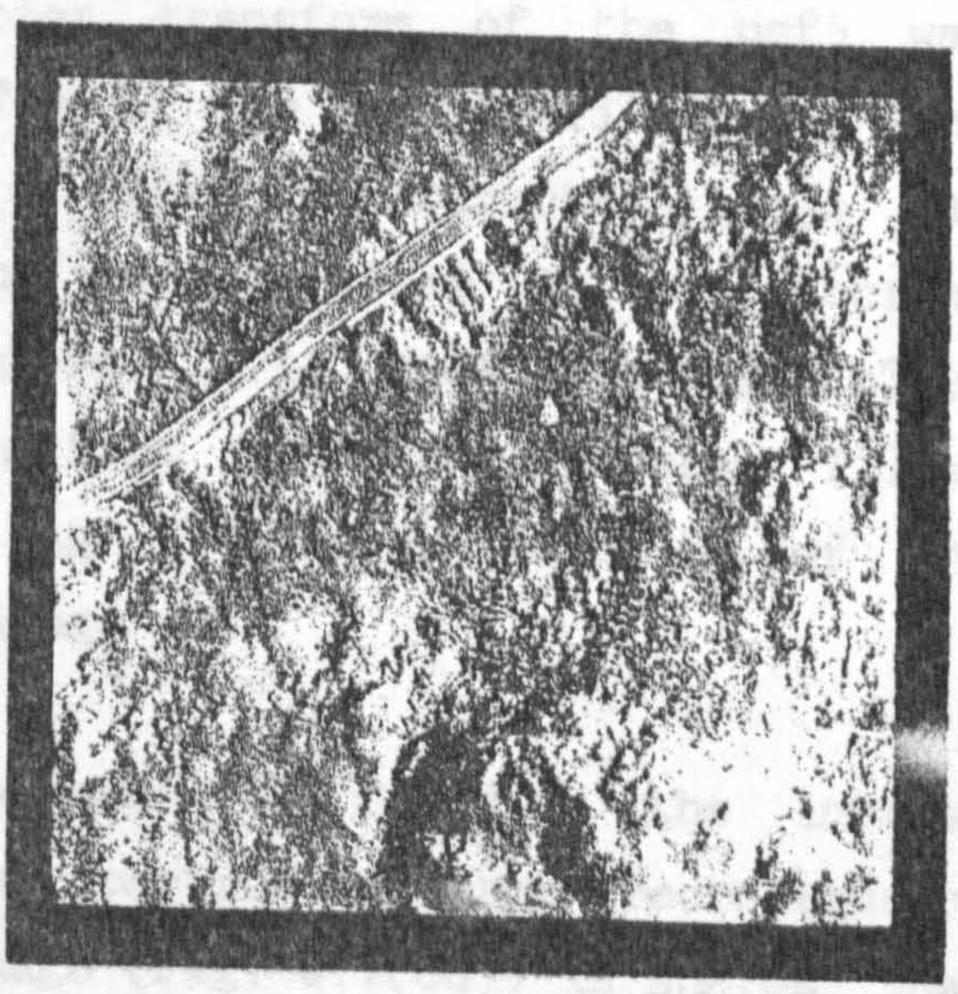
a1



a2







b1

b2

Fig.4.6 Periodic noise removal.
a1. Original image.
a2. Image plus noise.
b1. Amplitude spectrum of the noisy image after Applying a notch filter.
b2. Reconstructed image.

4.10.2 Experiments with Inverse and Modified Inverse Restoration Filters

a- Inverse Filter

To validate experimentally the theoretical discussion in Section 4.7, and to find the potential application of Inverse filtering in digital photogrammetry, a test area was selected on the digital image and blurred by convolving the image function with a two-dimensional rectangular psf (Figs. 4.7(a1), 4.7(a2), 4.7(a3)).

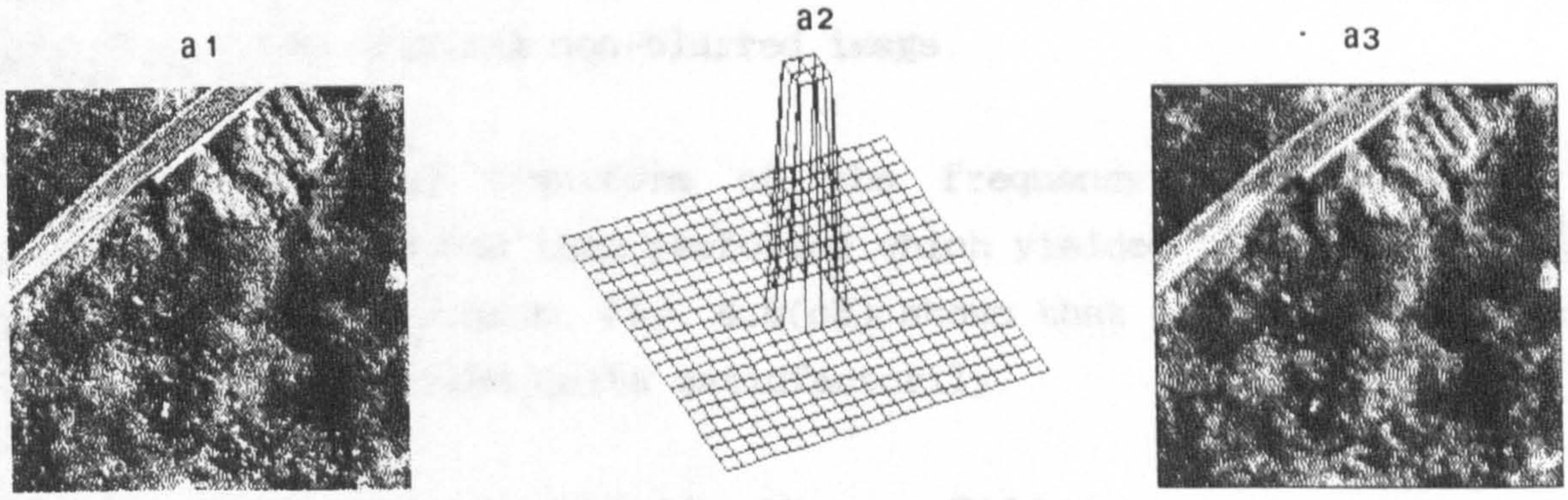
To reconstruct the original non-blurred function, the blurred image and the rectangular function were Fourier transformed. The effect of the blurring operation on reducing the amplitude of the high frequency contents of the image can be observed quite clearly by comparing the amplitude spectrum of the original image (Fig. 4.8(a1)) and the amplitude spectrum of the filtered image (Fig. 4.8(a2)).

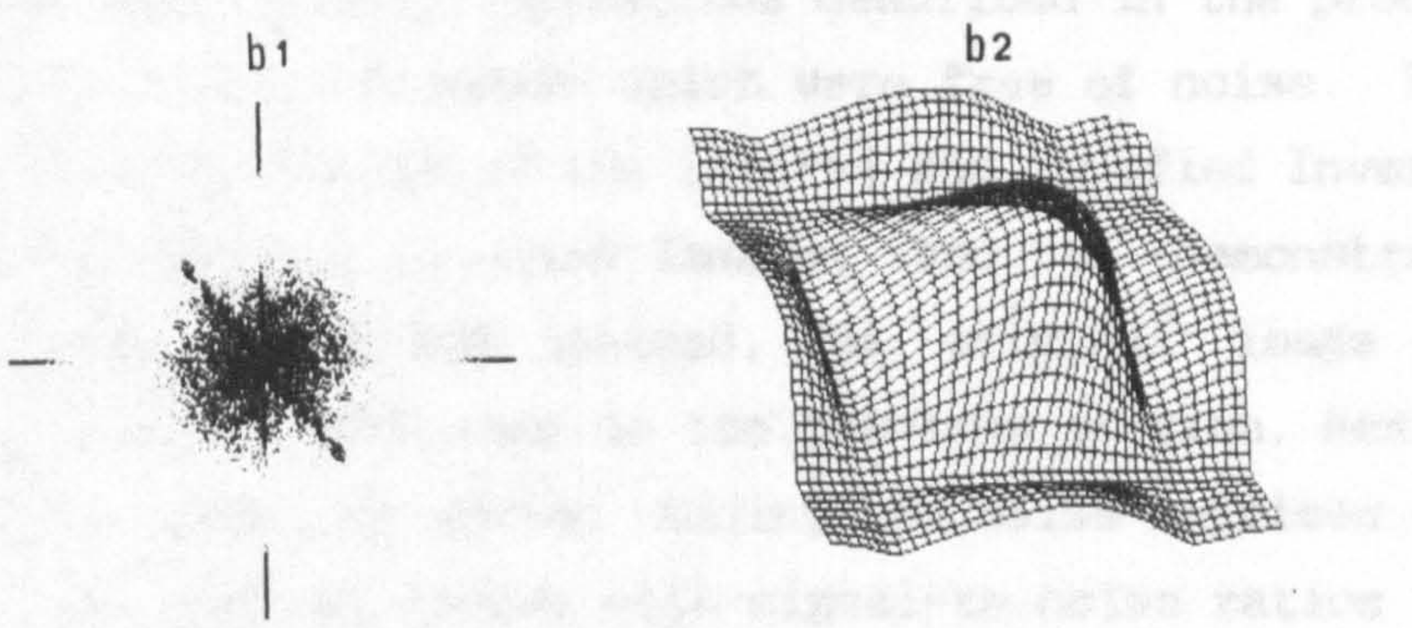
The transfer function (i.e. the Fourier transform of the psf) was normalized to give the MTF (Fig. 4.7(b2)). The Inverse filter was then calculated. The Fourier transform of the reconstructed image was obtained subsequently by applying the Inverse filter to the Fourier transform of the degraded image using Eq. 4.11 (see Fig. 4.7(b3)). Fig. 4.7(b3) clearly demonstrates the fact that the Inverse filter increases the amplitudes of the high frequency contents of the frequency spectrum of the degraded image.

The reconstructed image in the space domain was found by the inverse Fourier transform of the frequency spectrum of the reconstructed image. Although the reconstructed image (Fig. 4.7(c3)) is sharper than the blurred one, the increase in the amplitude of the high frequency contents of the image has obscured to a considerable extent the information content of the image.

b- Modified Inverse Filter

To suppress the noise amplification problem of the Inverse filter, a modified filter was designed by selecting a threshold value (which was found empirically (see Section 4.7.1)), beyond which the MTF amplitude remained constant (Figs. 4.8(b2) and 4.8(b3)). The modified filter was





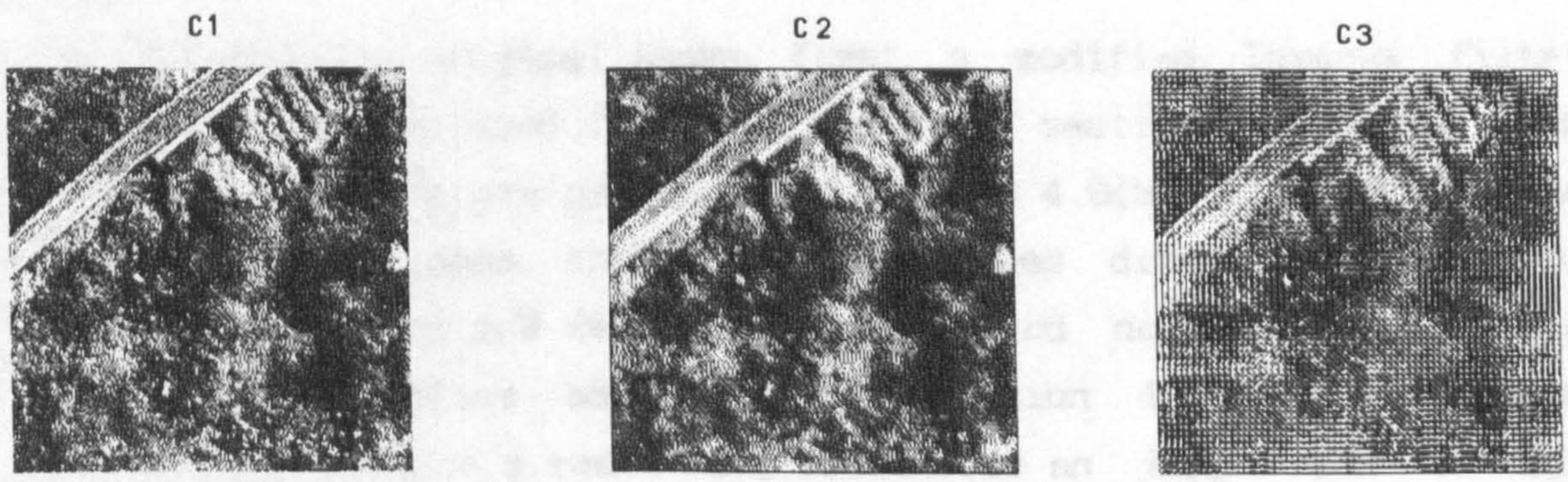


Fig.4.7 a1. Original image.
a2. Rectangular psf.
a3. Blurred image.
b1. Amplitude spectrum of the blurred image.
b2. Transfer function (Fourier transform of a2).
b3. Amplitude spectrum of the reconstructed image.
c1. Original image.
c2. Blurred image.
c3. Reconstructed image.

then applied to the Fourier transform of the blurred image. The result is given in Fig. 4.8(a3). The comparison between the amplitude spectra of the original non-blurred, blurred and the reconstructed images (Figs. 4.8(a1), 4.8(a2), 4.8(a3)) shows the degree of closeness of the amplitude spectrum of the reconstructed image to the amplitude spectrum of the original non-blurred image.

The inverse Fourier transform of the frequency spectrum of the reconstructed image was then performed which yielded the reconstructed image in the space domain. Fig. 4.8(c3) shows that the reconstruction task has been performed quite satisfactorily.

4.10.3 Experiments with the Wiener Filter

The deconvolution operations described in the preceding section were carried out on images which were free of noise. To test empirically the deficiencies of the Inverse and modified Inverse filters for the deconvolution of noisy images, and to demonstrate how the Wiener filter can be used instead, the original image (Fig. 4.9(a1)) was blurred as described in the previous section. Next two sets of random noises were generated. Adding the noise matrices to the image without noise yielded images with signal-to-noise ratios of about 7 and 1 respectively (Figs. 4.9(a2), 4.9(a3)).

To reconstruct the original image, first a modified Inverse filter similar to that described in the preceding section was used. The reconstructed images are presented in Figures 4.9(b2) and 4.9(b3). As expected, in both cases, the noise is amplified drastically and for the image with lower S/N ratio, the amplified noise dominates the entire scene. Therefore, as concluded in Section 4.7.1, the Inverse filter cannot produce a reasonable result for an image with a high degree of noise and blurring.

To find the optimal solution to the above problem, the Wiener restoration filter was designed and implemented as follows.

The power spectra of the noise and the image without noise were determined using samples of the noise and the image without noise and

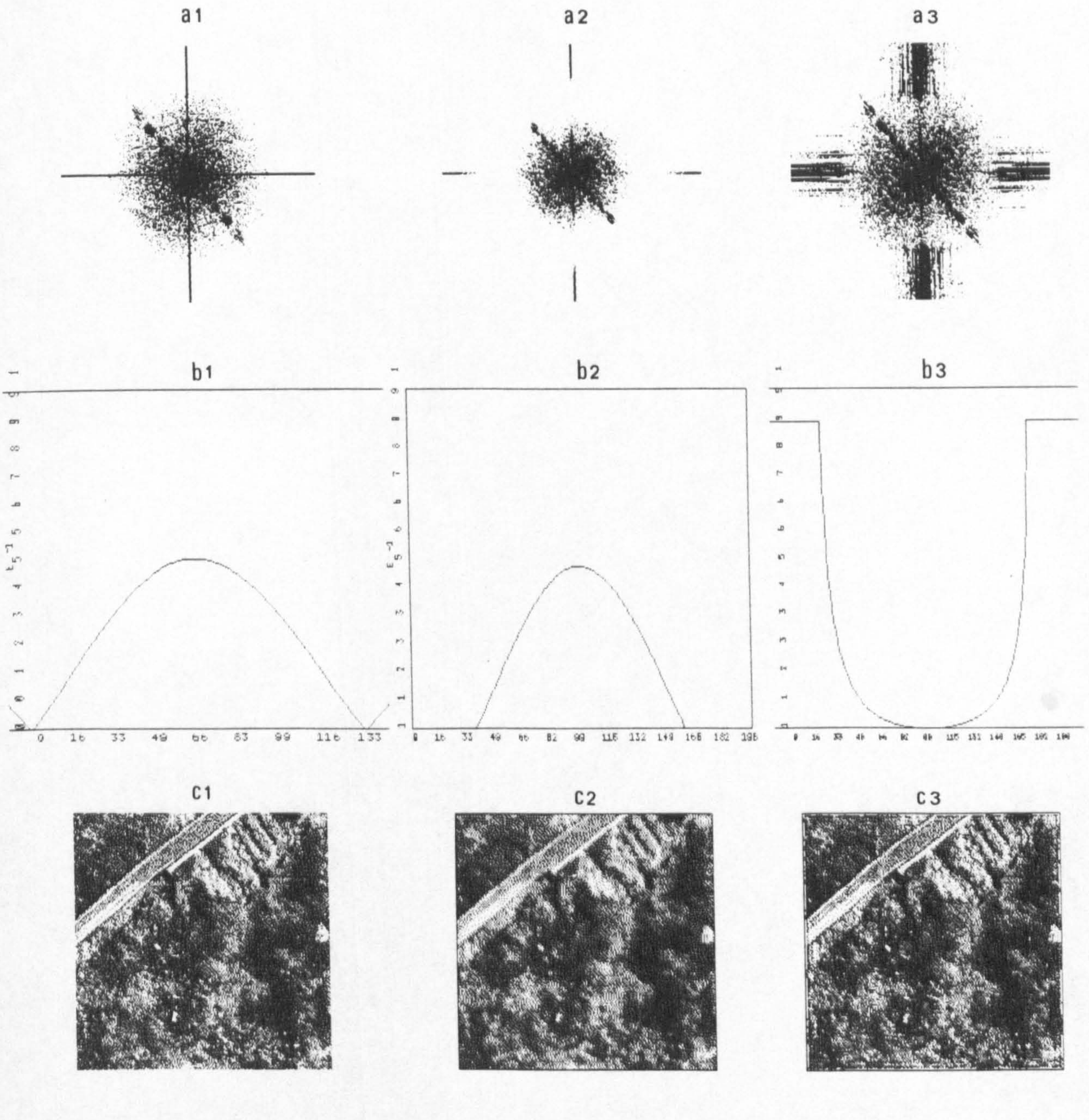


Fig.4.8 a1. Amplitude spectrum of the original image.
a2. Amplitude spectrum of the blurred image.
a3. Amplitude spectrum of the reconstructed image by MIF.
b1. Modulation transfer function (MTF).
b2. Modified MTF.
b3. Inverse modified MTF.
c1. Original image.
c2. Blurred image.
c3. Reconstructed image.

subsequently the Wiener filter was calculated using the relations given in Eq. 4.20. The results of applying the Wiener restoration filter on both noisy images are given (in the space domain) in Figures 4.9(c2) and 4.9(c3). As the results show, the reconstructed images have much better quality than those images reconstructed by the Inverse filter.

4.11 Conclusion

In this chapter, some theoretical aspects of pictorial rectification have been discussed together with the results of some related experiments. It has been demonstrated that, if pictorial degradations are present in the digital images, Fourier domain deconvolution filtering is a very powerful tool for pictorial rectification. In particular, for photogrammetric applications, the Inverse filter seems to be able to satisfy the requirements for image quality improvement. However, in those situations where the image has been digitized with a pixel size smaller or equal to the digital image, the Wiener filter seems to be a good solution for image restoration.

In the next chapter, the operator $\mathbf{U}_g^{-1}$ of Eq. 4.1 for geometric rectification will be discussed.

CHAPTER FIVE

Digital Rectification and Model Formation

5.1 Introduction

In the preceding chapter, some methods of correcting for the pictorial degradations occurring in digitized aerial photographs (operator U_p^{-1}, Eq. 4.1) were considered. This chapter is devoted to the corrections for the geometric displacements/distortions which are also found in such photographs (operator U_g^{-1}, Eq. 4.1). As explained in Section 2.8.1, **geometric distortions** are caused by the imaging and the scanning microdensitometer systems. Typical factors causing the geometric distortions are: lens distortion; film-base instability; non-orthogonality of the microdensitometer's axes and non-linear movement of its flying spot, etc. **Geometric displacements**, on the other hand, originate from sources such as the deviation of the camera axis from the vertical; terrain relief; Earth curvature, etc. In normal photogrammetric practice, an image which is corrected for geometric distortions and displacements (excluding relief displacement), is referred to as a **rectified image**. If the geometric corrections also include the relief displacement, the final product will be referred to as an **orthophotograph**. The digital production of an orthophoto is not the main concern here. Many experiments have already been carried out in this area (see for example, Keating and Boston, 1979; Konecny, 1979; Wiesel, 1985). Instead, the main concern here is to rectify the digital images for only the tilt displacements and the geometric distortions and therefore, to retain the relief displacements. If this height information can be retained during the image rectification stage, it can then be used for the reconstruction of a three-dimensional model of the terrain, including the stereo-viewing of the left and right digitally rectified images and the subsequent measurement of the model using the human operator or image matching techniques which will be discussed more fully in Chapters 6, 8 and 9.

Digital geometrical rectification usually comprises two stages:

(i) **analytical rectification**, in which each pixel position in the digital image is corrected for the geometric displacements/-distortions mentioned above; and

(ii) the **assignment of density values** to each corrected pixel

location, based on methods which will be explained in Section 5.3.

Analytical rectification and density value assignment, which constitute the basic principle of digital image rectification, will be covered in the first part of this chapter. This will then be followed by a description of the actual implementation of the digital image rectification with a view to removing the y-parallaxes and their associated x-parallaxes to form the left and right images of the stereo-pair.

5.2 Analytical Rectification

The analytical rectification of digitized aerial photos is in principle similar to the well known conventional procedures applied in non-digital photogrammetry, i.e. inner orientation to establish the position of the projection centre, followed by relative and absolute orientation to reconstruct the position of the camera at exposure time. To perform these three orientation procedures, fiducial marks, image points and image control points must be measured with the required accuracy. The manner of carrying out these measurements on digital images is quite different to those used with non-digital images. In the conventional case, the human operator performs the measurement task, while in digital image photogrammetry, there are two possible methods for target measurement:

(i) **manual measurement** of the individual pixels displayed on a display device using a cursor or cursors; and

(ii) using **mathematical algorithms** which are employed to locate the centres of the required targets. Recent developments in this area have explored the high potential of digital image processing techniques for target measurement with sub-pixel accuracy (Thurgood and Mikhail, 1982; Mikhail et al., 1984; Zhou, 1986). In the next sections, a general description and discussion of the requirements for the measurement of targets in inner, relative and absolute orientation is given.

5.2.1 Automatic Measurement of Digital Fiducial Marks for Inner Orientation

Since the fiducial marks employed in aerial photographs are normally in the form of a cross, the methods of measurement of cross targets are considered here specifically. However, these methods may also be applied to other types of target, such as pre-marked control points.

Automatic target measurement normally comprises three different stages - **detection, recognition** and **location.**

1. **Cross detection** may be accomplished by different methods, such as measuring the local grey level maxima which correspond to a possible cross target.

2. **Cross recognition** procedures are necessary when pre-marked cross-type control points are used. These targets need to be discriminated from buildings, road intersections and other similar physical objects, since all these objects may exhibit two-dimensional local maxima. There are a variety of approaches to the problem of object recognition in different scientific areas and an enormous amount of literature has been produced in this subject. Recently a method based on the calculation and assessment of the Fourier descriptors of cross targets has been described for a photogrammetric application (Mikhail et al., 1984). In this method, the first stage involves the grey value thresholding of the image to yield a binary image (see Section 7.3). If the threshold is chosen correctly, the object will be segmented from the background data. A Fourier transform is applied to the boundary of the segmented object to produce the object's Fourier coefficient. If the coefficients (descriptors) match those of a "true" cross target within a specified accuracy, the object will be classified as a cross target.

3. **Cross location.** The existing methods for accurately determining the location of a target may be classified into two broad approaches:

    a. **Matching methods**: These will be considered in detail later in Chapters 8 and 9, where the techniques of image correlation are discussed. In this chapter, only a short introduction to

the subject is given.

The matching method is based on the assessment of the extent of the **correlation** between a real target and a simulated ideal target (the simulated target may be referred to as a template). The most commonly used measure of the match between a template f(x,y) and a target g(x,y), is the normalized cross correlation function ρ_{fg} given by Eq. 8.2 (Chapter 8). The template f is shifted into all possible positions relative to g, and the value of ρ_{fg} is computed for each position. The normalized cross correlation function has the maximum value when f and g are identical. The difficulty which may arise when using this method is that, when searching for targets which may be affected by scale change and orientation differences, a large number of different templates are needed.

Another possibility is to use a so-called **image modelling approach** which minimizes the differences between the ideal and actual target by applying a least squares solution to a series of observation equations which express the measured densities of a digitized target as a function of the densities of an ideal target convolved with certain types of point spread functions. (In this way, the characteristics of the simulated target are altered to resemble those of the real target). The least squares solution then yields values for the unknown parameters defining the target. The method has been implemented successfully by Thurgood and Mikhail (1982). However, due to the high degree of non-linearity in the functional model, convergence does not occur in all cases (Thurgood and Mikhail, 1982). This approach has been implemented in this project and will be discussed fully in Chapter 9.

b. **Edge detection method**: In this method, different kinds of digital edge operators (see Section 5.5.1(iii)) are used to detect the target position. Most recently, a method based on **boundary tracing** has been suggested by Zhou (1986). The

method involves the segmentation of the object, boundary tracing and decomposition, and the location of the MATs (a set of midpoints in a pair of target legs with the same orientation), followed by line fitting through the MATs, and finally the line intersection which gives the location of the cross centre. The solution to the problem is made easier by some assumptions, e.g. regarding the thickness of the cross, which is assumed to be at least four pixels wide except for those at the end of the leg; also there must be no hole in the cross (Zhou, 1986).

5.2.2 Automatic Image Point Measurement for Relative and Absolute Orientations

Automatic image point measurement for **relative orientation** may be performed using the image correlation technique described in Section 5.2.1a. In this case, however, the template and the target are two small patches of the image, referred to as a window array and a search array respectively, comprising the left and the right image points (see Chapter 8). Clearly if the points in the six standard positions of the model area are artificial pricked points or signalized ground control points, maximization of the normalized cross correlation function becomes a relatively easy task. If such artificial points are not available, then to guard against poorly matched natural pass-points, some algorithms needs to be applied prior to the correlation process, to determine those areas with high signal content which exist in the six standard positions used for relative orientation.

An automatic digital solution to image control point measurement for **absolute orientation** is only possible if either the control points are pre-marked targets set out on the ground before the aerial photos are taken, or the control points which have been selected are geometrically simple natural targets with a high degree of grey level contrast with the surroundings (e.g. road intersections). Otherwise, to measure a natural point on the image using correlation techniques, a template needs to be formed (simulated) with the same shape and grey level characteristics as the natural target. In practice, this becomes

quite impractical due to the geometrical and grey level complexities of natural targets.

5.3 Assigning The Grey Levels

After the solution of the analytical rectification, the elements of exterior orientation (i.e. φ, ω, $\varkappa$, X_o, Y_o, Z_o) of each camera are determined and recorded. These exterior orientation parameters, plus those known or calculated parameters describing the geometric distortions(-- if they are significant) are used subsequently to establish the new position of each pixel of the original image using geometrical transformations. A density value will then be assigned to each geometrically corrected pixel. The geometric transformation required for correcting the positions of pixels may be carried out using either of two methods -- the direct method or the indirect method.

1. **The direct method**: In this approach, the positions of the pixels on the original image (referred to as the input image) are corrected by applying the geometrical transformation, while each pixel retains its grey value. This means, the geometrically corrected pixels (referred to as output image) will be irregularly distributed. Hence to obtain a regularly distributed output image (needed for example if a hardcopy image is to be generated), a rectangular grid must be constructed for which the grey levels must be interpolated in an off-line solution (Fig. 5.1).

2. **The indirect method**: Alternatively, the output image may be considered to have a regular pixel pattern. The grey value for each output pixel can be computed by geometrically transforming the output pixel into the plane of the input image and assigning the grey value of the corresponding pixel on the input plane to the output pixel. However, the location of the output pixel after transformation to the input plane will not necessarily coincide with the pixels on the input image (Fig. 5.2).

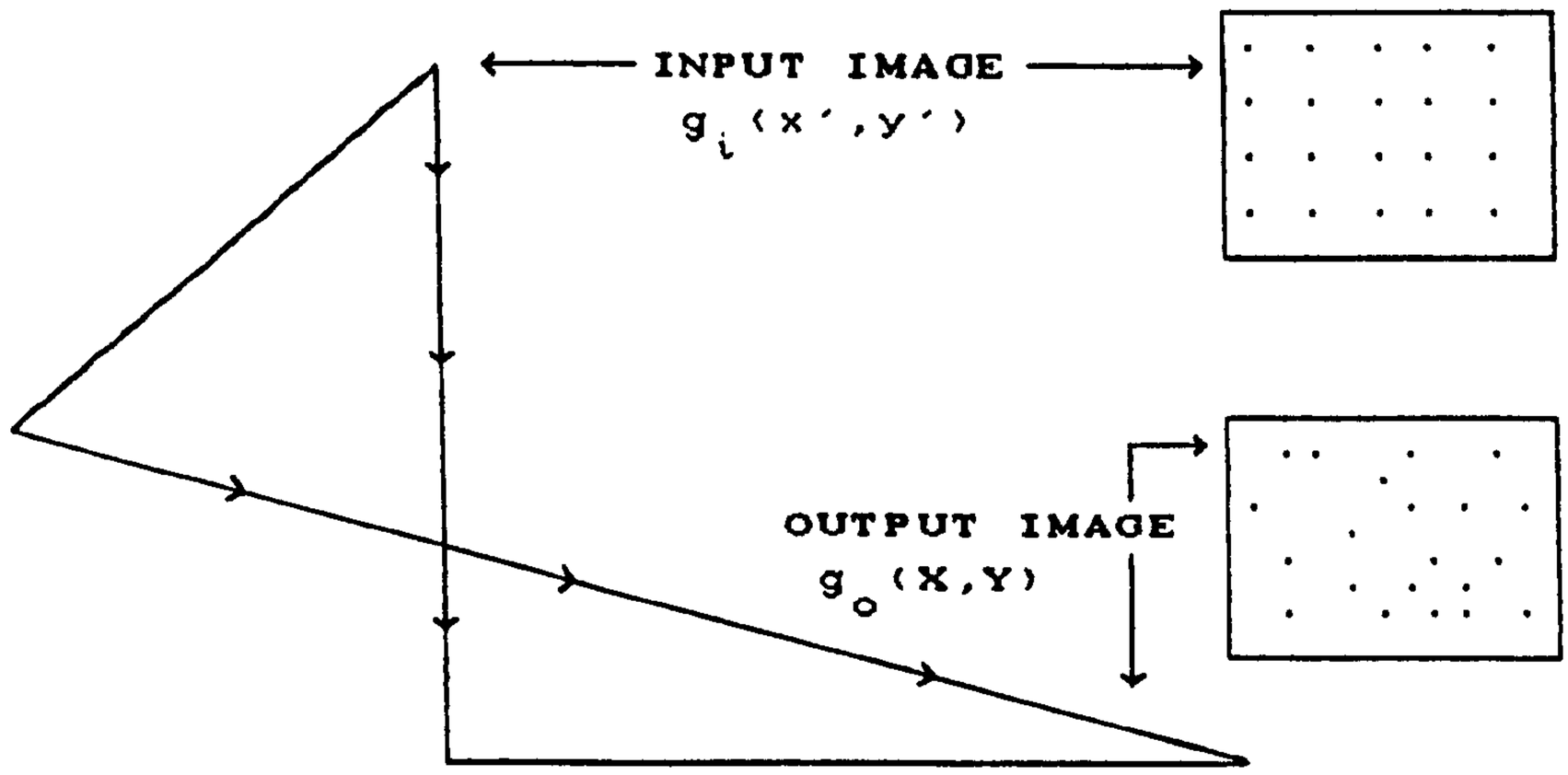


Fig. 5.1 DIRECT GEOMETRIC TRANSFORMATION

The intensity values to be assigned to each output pixel must, therefore, be determined by interpolation in the input image which is referred to as **resampling** and will be discussed in the next section.

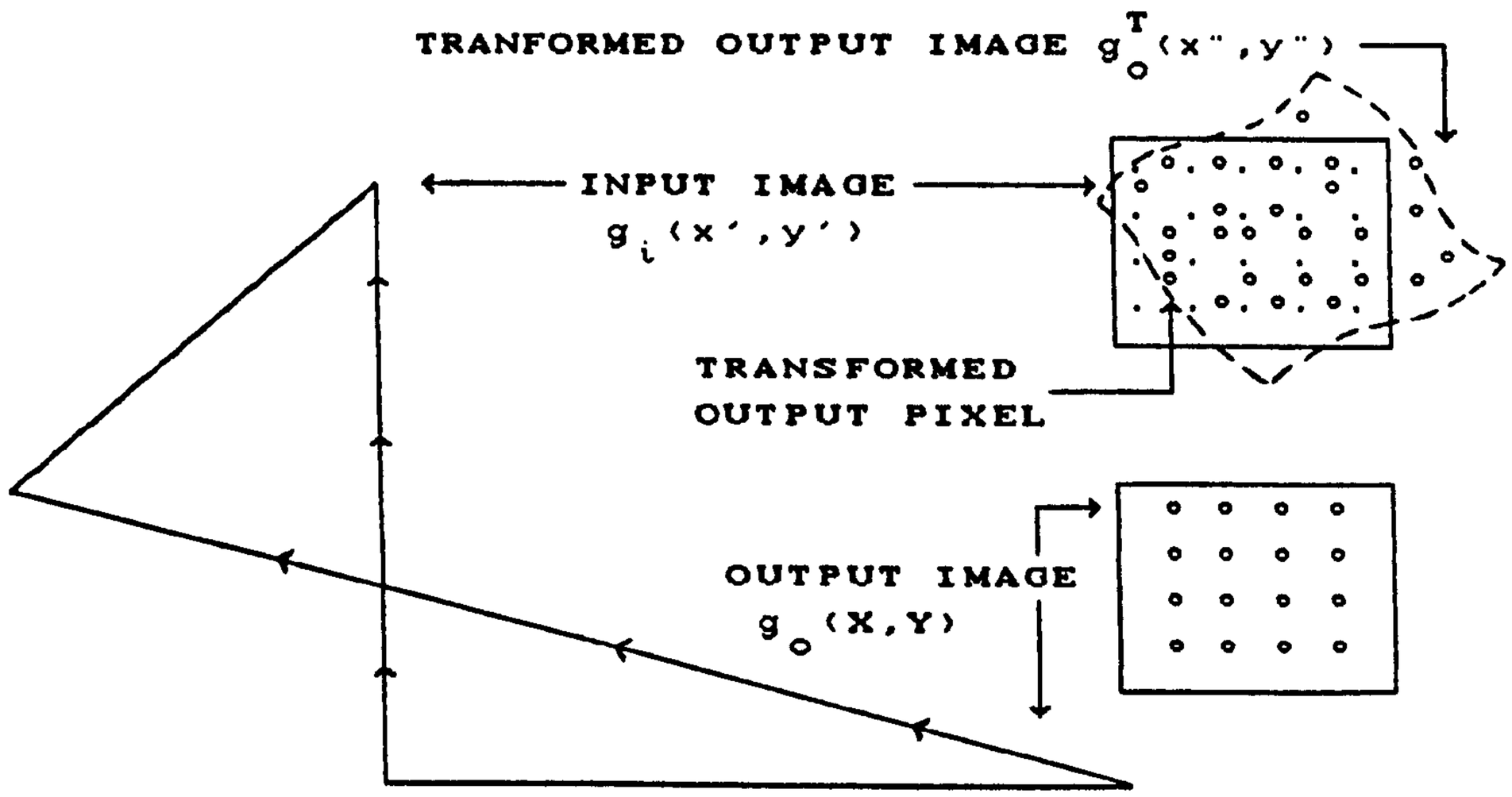


Fig. 5.2 INDIRECT GEOMETRIC TRANSFORMATION

5.3.1 Resampling Methods

Consider the input image represented by a function $g_i(x',y')$ of a

regularly spaced grid; and the geometrically corrected output image represented by function $g_o(X,Y)$. The two functions are geometrically related to each other by the collinearity equations. However, the relationship between the density values of the input and output pixels may be established using the familiar convolution operation (Eq. 2.14),

$$g_o^T(x'',y'') = \int_{-\infty}^{\infty}\int g_i(x',y')h(x''-x',y''-y')dx'dy' \qquad (5.1)$$

where $g_o^T(x'',y'')$ is the output image geometrically transformed to the plane of the input image (Fig. 5.2); and $h(x',y')$ denotes the impulse response of the interpolation function. Depending on the choice of $h(x,y)$, various interpolation schemes can be implemented, with different speed and accuracy. These interpolation methods are: Zero-order Interpolation, First-order Interpolation and Higher-order Interpolation.

a. Zero-order Interpolation

The simplest interpolation scheme is the so-called zero-order or **nearest neighbour interpolation**. Fig. 5.3 shows the nearest-neighbour interpolation function for one dimension.

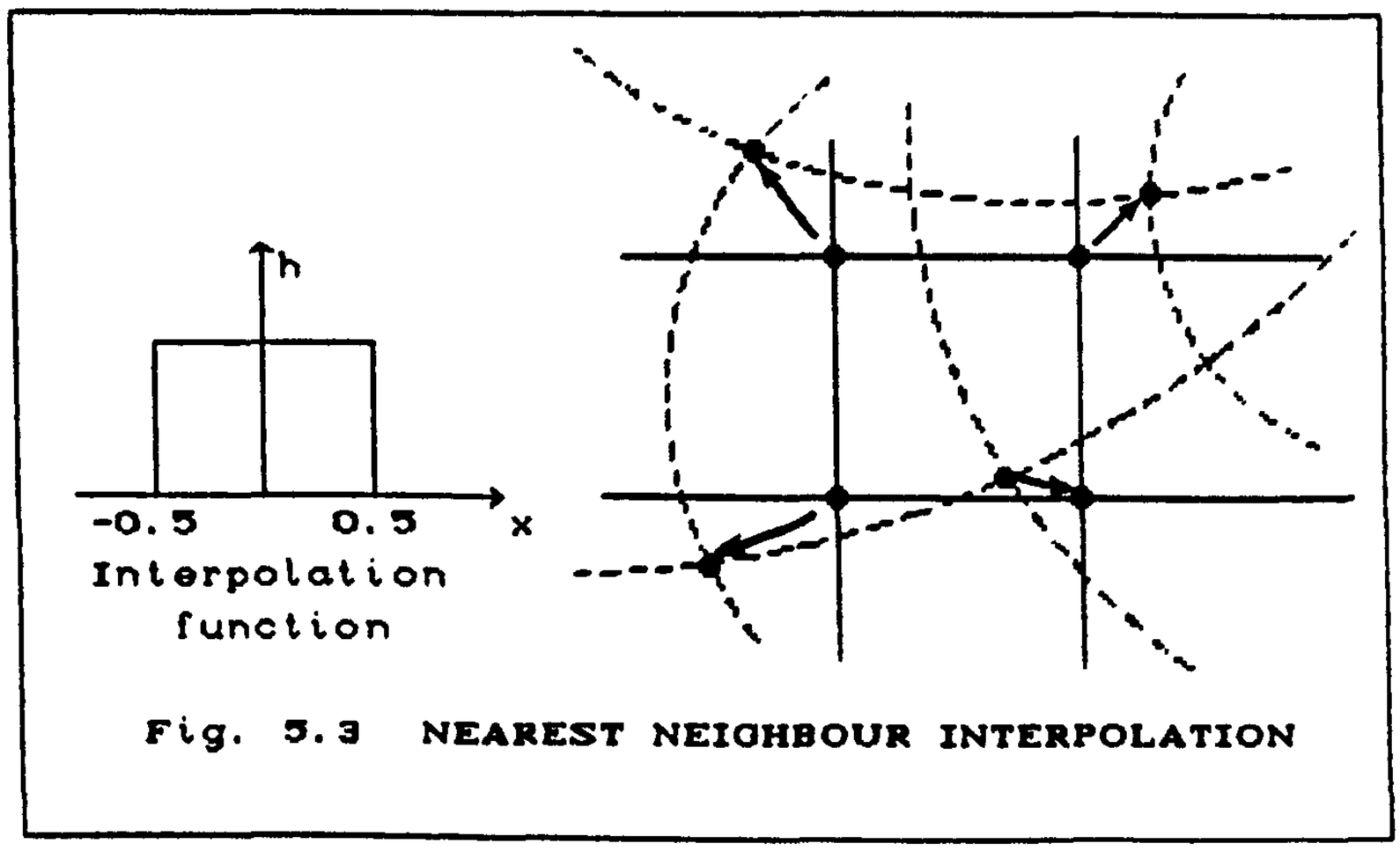


Fig. 5.3 NEAREST NEIGHBOUR INTERPOLATION

This approach takes the value of the nearest pixel to the transformed output pixel $g_o^T(i,j)$ and assigns it to the output

pixel $g_o(k,l)$, where i,j are the row and column numbers of the transformed output pixel with respect to the input image coordinate system; and k,l are the row and column of the output pixel. This means that the resulting grey levels correspond to true input pixel values, but the geometric location of a pixel may be inaccurate by as much as ±0.5 pixel spacing. The sudden shift of the true pixel values also gives rise to a blocky appearance of the linear features.

b. First-order Interpolation

To find a grey value for the transformed output pixel at the position i+m,j+n on the input grid, using first order (or so-called **bilinear interpolation**), a hyperbolic paraboloid is fitted to the four neighbouring pixels of the transformed output pixel (Figs. 5.4). This may be done by the bilinear equation:

$$g_o^T(i+m,j+n) = a(m) + b(n) + c(mn) + d \qquad (5.2)$$

where $g_o^T(i+m,j+n)$ is the interpolated density at location i+m, j+n for the transformed output pixel; and the coefficients a ... d are to be determined so that a hyperbolic paraboloid is fitted to the known values of the four neighbouring input pixels. (see Fig. 5.4).

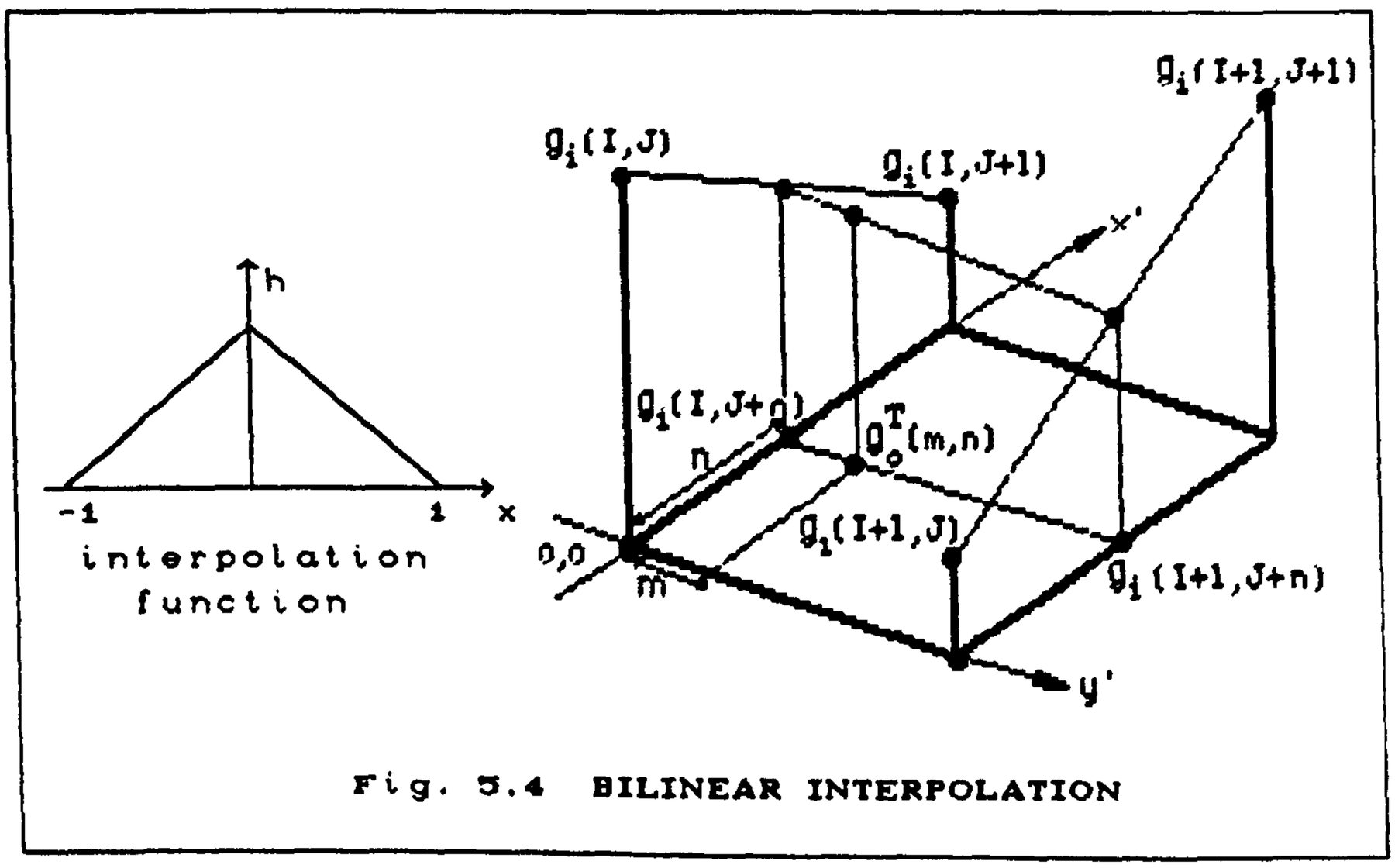


Fig. 5.4 BILINEAR INTERPOLATION

To determine the coefficients, first a linear interpolation is carried out between the two input pixels at locations $g_i(i,j)$ and $g_i(i,j+1)$ to establish the value at position i,j+n (see Fig. 5.4):

$$g_i(i,j+n) = g_i(i,j)+(n)\left[g_i(i,j+1)-g_i(i,j)\right] \qquad (5.3)$$

and similarly, for the two other pixels at locations $g_i(i+1,j)$ and $g_i(i+1,j+1)$ to establish the value at position i+1, j+n,

$$g_i(i+1,j+n) = g_i(i+1,j)+(n)\left[g_i(i+1,j+1)-g_i(i+1,j)\right] \qquad (5.4)$$

Finally, the two values derived above can be interpolated linearly in the column direction to give the final interpolated value at position i+m,j+n,

$$g_o^T(i+m,j+n) = g_i(i,j+n)+(m)\left[g_i(i+1,j+n)-g_i(i,j+n)\right] \qquad (5.5)$$

Substituting Equations 5.3 and 5.4 into Eq. 5.5, expanding, and collecting terms yields,

$$\begin{aligned} g_o^T(i+m,j+n) = {} & \left[g_i(i+1,j)-g_i(i,j)\right](m)+ \\ & \left[g_i(i,j+1)-g_i(i,j)\right](n)+ \\ & \left[g_i(i+1,j+1)+g_i(i,j)-g_i(i+1,j)-g_i(i,j+1)\right](mn) \\ & +g_i(i,j) \end{aligned} \qquad (5.6)$$

which is in the form of Eq. 5.2 and thus bilinear.

Bilinear interpolation may cause a small loss of image resolution due to the smoothing or blurring nature of linear interpolation. On the other hand, the blocky appearance of linear features associated with zero-order interpolation is reduced at the cost of additional computation.

c. Higher-Order Interpolation

The image blurring of bilinear interpolation may be overcome by applying extra computational effort, using higher-order

interpolation algorithms. Examples of higher-order interpolating functions are cubic splines and the sinc function (Castleman, 1979). The cubic approximation of the latter has been implemented in the present project and hence will be discussed briefly in the following paragraphs:

If the input image $g_i(x',y')$ is band-limited (recalling the discussion given in Section 2.6.2), the output $g_o(X,Y)$ can be reconstructed exactly by applying the sampling theorem. The reconstruction filter is, therefore, a sinc function. To implement this function in one dimension, its cubic approximation with the 4 nearest neighbours (Fig. 5.5) can be employed. This is given as (Moik, 1980):

$$h(x) = \begin{cases} 1-2|x|^2+|x|^3 & 0 \leq |x| < 1 \\ 4-8|x|+5|x|^2-|x|^3 & 1 \leq |x| < 2 \\ 0 & |x| \geq 2 \end{cases} \qquad (5.7)$$

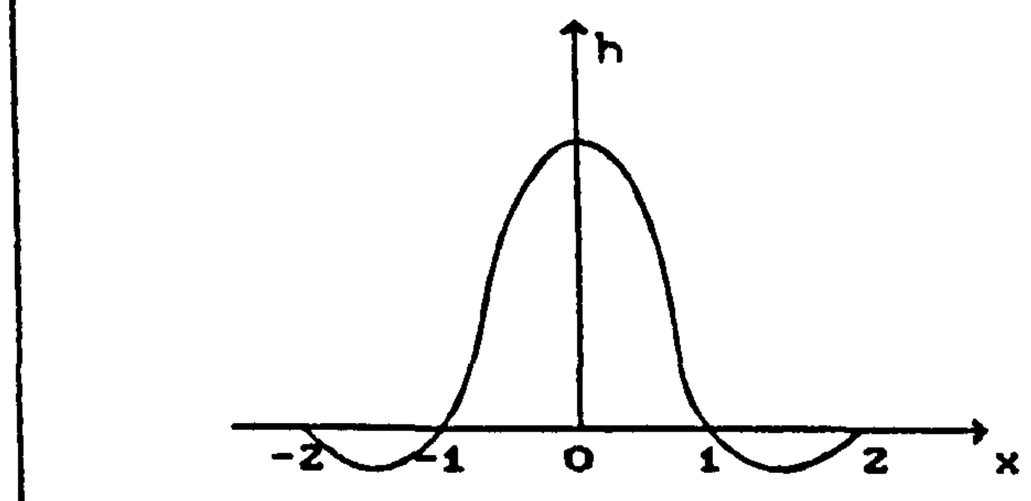


Fig. 5.5 BICUBIC INTERPOLATION FUNCTION

A two-dimensional implementation of Eq. 5.7 with 16 nearest neighbours may be obtained by first interpolating in the scan line direction (i.e. for the pixels (i,j-1), (i,j), (i,j+1), (i,j+2) and then interpolating in the column direction of the input image (i.e. (i-1,j), (i,j), (i+1,j), (i+2,j)), (see Fig. 5.6).

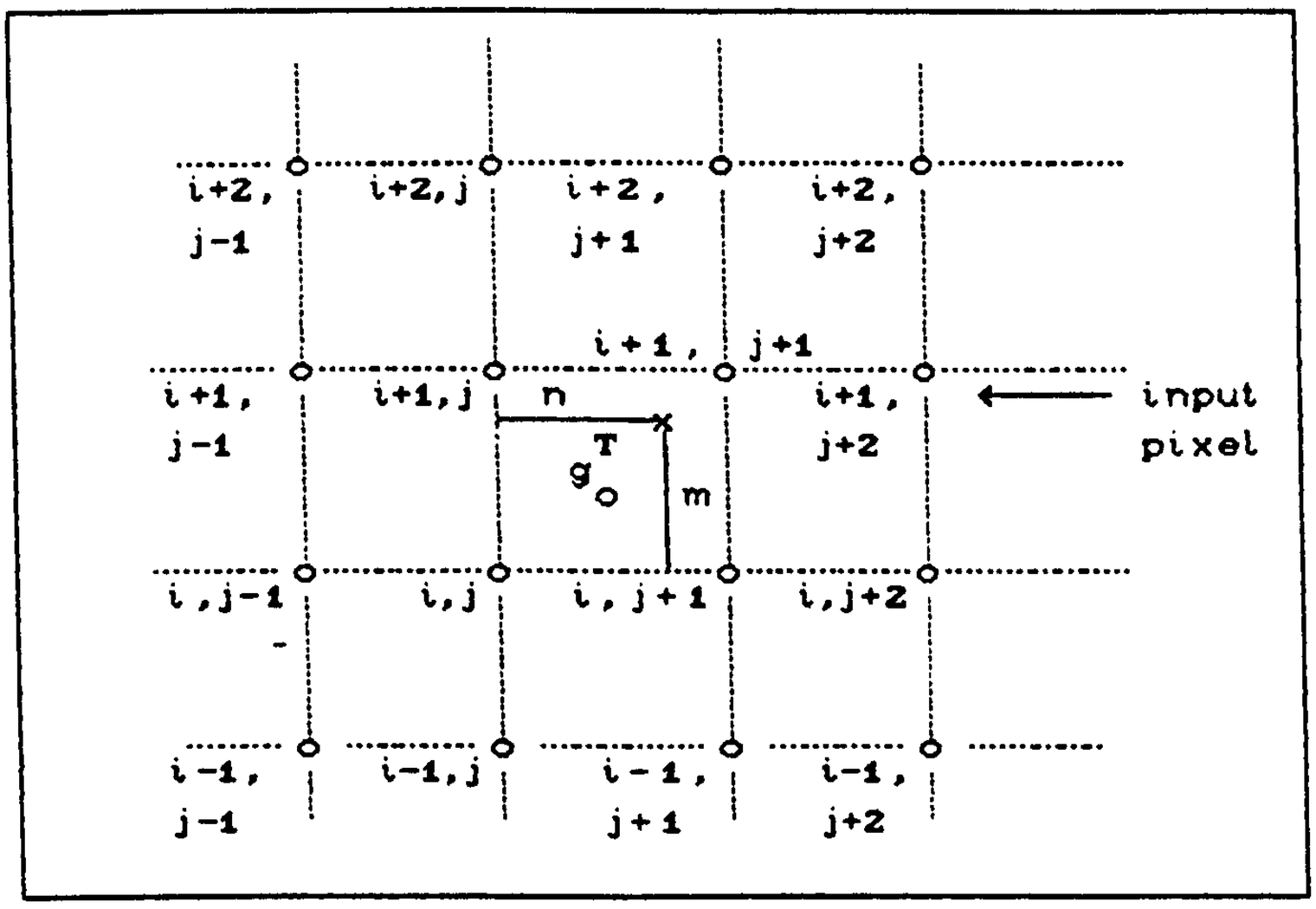


Fig. 5.6

i.e.

$$g_i(k,j+n) = -n(1-n)^2\left[g_i(k,j-1)\right]+(1-2n^2+n^3)\left[g_i(k,j)\right]+ n(1+n-n^2)\left[g_i(k,j+1)\right]-n^2(1-n)\left[g_i(k,j+2)\right]$$

$$k = i-1,\ i,\ i+1,\ i+2$$

The final interpolation value is obtained by:

$$g_o^T(i+m,j+n) = -m(1-m)^2\left[g_i(i-1,j+n)\right]+(1-2m^2+m^3)\left[g_i(i,j+n)\right]+ m(1+m-m^2)\left[g_i(i+1,j+n)\right]-m^2(1-m)\left[g_i(i+2,j+n)\right] \quad (5.8)$$

where, $g_o^T(i+m,j+n)$ is the final interpolated value for the transformed output pixel at location i+m,j+n.

Although applying the above resampling scheme (which is also referred to as a bicubic convolution) improves the accuracy of the output image, the computing time increases dramatically as compared with the previously discussed interpolation approaches.

Having the theoretical analysis presented in the preceding sections in mind, the rest of this chapter describes the implementation of the

digital rectification procedures used in the project.

5.4 Overall Strategy and Procedures for Digital Geometric Rectification and Model Formation

As mentioned before, in this project, rectification was performed with the main objective of removing only the y-parallax and its associated x-parallax throughout the model in order to allow the subsequent reconstruction of the three-dimensional model of the terrain and the stereo-viewing of this model. Hence, in the rectification stage, only the relative positions of the cameras have been reconstructed (i.e. the right image has been rectified with respect to the left image), and not their absolute position during the exposure time. This means that absolute orientation is not performed during this stage. Therefore, the term "rectification" as applied in this thesis should not be confused with the normal case in which each image is rectified for its absolute deviation from the vertical axis.

In this project, the rectification of the images resulting in the removal of y-parallaxes and associated x-parallaxes, is carried out based on the following two stages:

a. The **analytical rectification stage** consists of:

(i) **Inner orientation** which is carried out in two steps, namely, fiducial mark measurement, and the solution of a two-dimensional affine transformation to derive the transformation coefficients.

(ii) **Relative orientation** which is also carried out in two stages of image point measurement and solution of the coplanarity equations.

(iii) The subsequent indirect **geometric transformation** (see Section 5.3(2)) by which the output pixels are projected into the plane of the input image using the elements of the relative and inner orientations calculated in the previous stages. This backward projection requires three different geometric transformations of: 1. transformation into the epipolar plane (more detail is given in Section 5.8), 2. transformation into the principal point

coordinate system (input image plane), 3. and finally transformation into the microdensitometer coordinate system.

b. The **Resampling stage** which is implemented to assign a density value to the output pixels, may follow any one of three approaches of: nearest-neighbour interpolation, bilinear interpolation or bicubic convolution which were discussed in Section 5.3.1.

In the following sections, a detailed description of each of these stages or operations is given.

5.5 Fiducial Mark Measurement

To implement the inner orientation phase, the coordinates of the fiducial cross marks were measured using both automatic and manual methods, both being carried out on the digital images. The manual measurement of the fiducial marks will be discussed in the next chapter.

For automatic measurement of the fiducial cross position, a specific approach which is a combination of two methods - the edge detection and correlation methods described in Section 5.2.1- was adopted. This seemed to be advantageous, both from the point of view of improved efficiency and accuracy. The methods were combined in a single program which comprises two stages.

1. The first stage involves the measurement of the cross position using an **edge detection method** based on the Gradient operator approach. The method was developed by the author and has provided a satisfactory sub-pixel accuracy for the measured coordinates of the cross centres. In addition to the cross coordinates, the program also measures the following parameters of the cross (Fig. 5.7):

   - the **cross mean grey level**, i.e the mean value of the pixels forming the cross;
   - the **cross mean background grey level**, i.e. the mean value of the surrounding pixels of the cross; and

- the **cross orientation angle** with respect to the scan lines.

Although these additional parameters were not required for the inner orientation stage, they were calculated in order to be used in the second stage of the program which requires those parameters for the improvement of the calculated coordinates of the cross centre derived by the first stage.

2. The second stage achieves a further improvement in the measured coordinates of the cross centre. This approach, as mentioned in Section 5.2.1a, employs an **image correlation technique** based on the image modelling and least squares approaches. The method requires close initial values for the cross parameters. A detailed treatment of this method will be presented in Chapter 9, where the implementation of the image correlation techniques is considered.

The Gradient approach for measurement of the coordinates of cross centre is covered in the section which follows.

5.5.1 Measurement of the Cross Position Using the Gradient Method

Since, in the present project, automatic target measurement was carried out specifically for each individual fiducial cross, the cross detection and recognition stages were not a difficulty due to the homogeneous grey values of the cross background. Hence the three stages of detection, recognition, and location, mentioned in Section 5.2.1, were carried out in a single stage.

To find the cross parameters mentioned above, the following steps are implemented (see Fig. 5.7):

(i) calculation of the mean grey level (MGL) of the cross and its mean background grey level (MBGL);

(ii) calculation of a threshold value (using MGL and MBGL), to differentiate the cross edges from the noise present in the image;

(iii) detection of the positions of the left and right edges of the

cross legs (using the Gradient operator);

(iv) calculation of the positions of the midpoints of the cross legs for each scan line using the detected positions of the edges; and

(v) fitting a line to each set of midpoints of the legs having the same orientation, and intersecting these lines to give the coordinates of the cross centre and its orientation.

These are described in more detail in the following paragraphs:

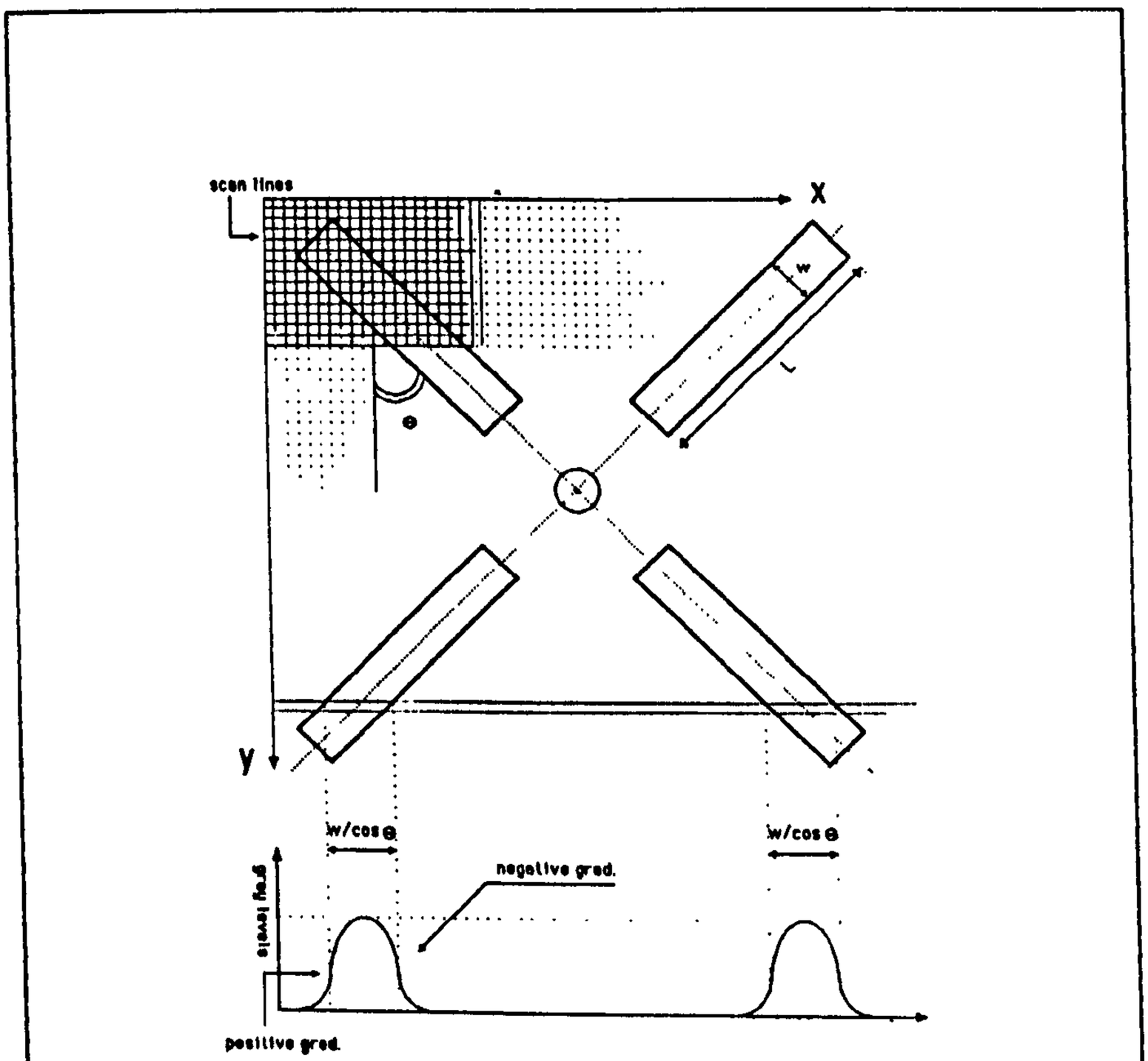


Fig. 5.7 DETAILS OF CROSS AND DENSITY LEVELS

(i) The cross and its mean background grey levels calculation

The mean grey level (MGL) of the cross and the mean grey level (MBGL) of its background, are each a single valued estimation of the overall density of the cross and its background respectively. These values were calculated to serve two different purposes. Firstly, they will be

used for the calculation of a threshold value which serves as a criterion for the identification of the cross edges. Secondly, they will be fed together with the other calculated cross parameters as initial unknown approximations to the cross measurement program, which is based on the image correlation and image modelling approach (see Chapter 9).

To calculate the values of MGL and MBGL, the cross target needs to be extracted roughly from its background. Each of the fiducial crosses exhibits an abrupt change in grey levels as compared with its background. Therefore, to segment the cross, an initial threshold value T was calculated as follows:

$$T = (MXL - MNL)/2 \tag{5.9}$$

where MXL and MNL denote the maximum and minimum grey levels of the image patch containing the cross. The MGL and MBGL values were then calculated using the so-called area function (AF) and integrated optical density (IOD). The **area function** of an image is the number of points having grey levels greater than or equal to a specific value. i.e.,

$$AF = \int_{T}^{\infty} H(p)dp \tag{5.10}$$

where H(p) represents the image histogram; and T (in this experiment) is the initial threshold value calculated in the previous stage. For a discrete function (digital image), Eq. 5.10 may be written as

$$AF = \sum_{P=T}^{max} H(P) \tag{5.11}$$

where max is the maximum range of grey values present in the image.

The **integrated optical density** (IOD) is defined as total grey values of pixels of any specific image feature (e.g. a fiducial cross). For an image g(x,y), the IOD is given by:

$$IOD = \iint_{a}^{b} g(x,y)dxdy \tag{5.12}$$

where a and b delimit the region of the interested object in the image. For a discrete function, Eq. 5.12 can be rewritten as:

$$IOD = \sum_{i=1}^{b} \sum_{j=1}^{a} g(i,j) \qquad (5.13)$$

where g(i,j) is the density value of pixel at row i, column j. For the cross target used in this experiment, the area limited by a and b can be considered as equal to the AF calculated by Eq. 5.11.

The MGL and MBGL can now be calculated as,

$$MGL = IOD_c / AF_c \qquad (5.14)$$
$$MBGL = IOD_b / AF_b$$

where subscripts c and b denote the cross and its background respectively.

(ii) Threshold value determination

As mentioned before, to differentiate, somehow, between the cross edges and the emulsion scratches and the noise, a better estimation of the threshold value was calculated which was used as a criterion to reject the mis-identified edge points.

The threshold value was calculated based on the previously determined values for MGL and MBGL using the following relation:

$$THR = (MGL^2 + MBGL^2)^{1/2} \qquad (5.15)$$

(iii) Edge detection by Gradient operator

One of the common approaches to object extraction is based on the detection of a discontinuity (edge), i.e., of those places where there is a more or less abrupt change in grey level, indicating the end of one region and the beginning of another. Thus, the detection of the cross edges may be carried out based on employing the derivative operators which are used normally to enhance the discontinuities of the grey levels on the digital images. Given a function f(x,y), the Gradient operator, which indicates the rate of the change of the function in any direction, is given as a vector function defined by:

$$\nabla f(x,y) = \mathbf{i}\,\frac{\partial f(x,y)}{\partial x} + \mathbf{j}\,\frac{\partial f(x,y)}{\partial y} \qquad (5.16)$$

where ∇ indicates the vector Gradient operator; and $\mathbf{i}$ and $\mathbf{j}$ are unit vectors in the x and y directions respectively. The vector ∇ points in the direction of maximum upward slope.

The Gradient magnitude is, therefore, given by:

$$|\nabla f(x,y)| = \left\{ \left[\frac{\partial f(x,y)}{\partial x}\right]^2 + \left[\frac{\partial f(x,y)}{\partial y}\right]^2 \right\}^{1/2} \qquad (5.17)$$

This function gives a high value at points where the grey level of the image is changing rapidly.

For digital images (discrete functions), Equation 5.17 may be written as:

$$\text{grad}(i,j) = \left\{ \left[f(i,j+1)-f(i,j)\right]^2 + \left[f(i+1,j)-f(i,j)\right]^2 \right\}^{1/2} \qquad (5.18)$$

where grad(i,j) is the Gradient magnitude at the location i,j. In the present project, a smoothed Gradient magnitude, which is less sensitive to the noise, has also been implemented as an option which is given by:

$$\text{grad}(i,j) = \left[\Delta x f(i,j)^2 + \Delta y f(i,j)^2\right]^{1/2} \qquad (5.19)$$

where

$$\Delta x f(i,j) = 1/2\left\{\left[f(i,j+1)+f(i,j+2)\right]-\left[f(i,j-1)+f(i,j-2)\right]\right\}$$

and

$$\Delta y f(i,j) = 1/2\left\{\left[f(i+1,j)+f(i+2,j)\right]-\left[f(i-1,j)+f(i-2,j)\right]\right\}$$

The value of the Gradient magnitude, given by Eqs. 5.18 or 5.19, above the calculated threshold THR (given by Eq. 5.15) in a given scan line indicates the presence of the cross edge.

(iv) Determination of the midpoints of the cross legs

The calculated value of grad(i,j) (from Eqs. 5.18 or 5.19) for a given

cross leg in a scan line, will have positive and negative signs (+grad and -grad) which indicate the starting and ending position of a cross leg respectively. From these values, the cross legs midpoints for each scan line can be calculated as (see Fig. 5.7),

$$\begin{cases} W = j_{+grad} - j'_{-grad} \\ xmid = j_{+grad} + W/2 \\ ymid = n \end{cases} \qquad (5.20)$$

where W, xmid, and ymid denote the width of the cross leg and the x and y coordinates of the midpoints respectively; j_{+grad} ,j'_{-grad} represent the pixel number in the column direction for which the value of grad(i,j) has positive and negative signs respectively; and n is the current number of the scan lines.

(v) Identification of the cross legs orientation

The midpoints of the legs with the same orientation are determined using the following relation:

$$sign(xmid_n) = xmid_n - xmid_{n-1}$$

where $xmid_n$ and $xmid_{n-1}$ are the two successive detected midpoints in the left or right cross legs; and $sign(xmid_n)$ is a value corresponding to the midpoint $xmid_n$ which may be positive or negative. The midpoints having negative or positive signs are arranged into separate arrays so that each array comprises only those midpoints with the same sign.

5.5.2 General Outline of the Program "GRAD"

The program "GRAD" was developed to detect and locate the position of the centre of the cross and other cross parameters, using the method described in the preceding section. The features discussed in the previous sections are implemented in the program according to the following sequences:

(i) The first step involves loading a rectangular search array, which includes the target, into the computer's RAM. Obviously the search array should be as large as possible to ensure

that it includes the target. In the search array, the MGL and MBGL are calculated as described above.

(ii) Based on the MGL and MBGL values, a **threshold value** is calculated using the Eq. 5.15.

(iii) The image is scanned systematically along both the row and column directions and a **two-dimensional Gradient magnitude** (Eqs. 5.18 or 5.19) is calculated for each pixel.

(iv) The location of **midpoints of the cross legs** in each scan line are determined using the Eq. 5.20. If the calculated value of cross width (Eq. 5.20) is outside a specified range, it will be rejected. Therefore, the contribution of the pixels at the extreme corners of the legs, which affect the accuracy of the coordinates of the cross centre, can be eliminated.

(v) The calculated midpoints on the legs with the same orientation are identified and arranged into different **arrays** using the approach described in 5.5.1(v).

(vi) If fewer than two midpoints are determined on the two cross legs with the same orientation, then the **threshold value** is reduced by a certain factor and the procedures (i) to (v) are repeated until either more than one midpoint is determined, or the threshold reaches its minimum specified level which means no cross is detectable.

(vii) A line is then fitted to each series of midpoints and the intersection of the lines and their slopes gives the position of the centre of the cross and its **orientation** respectively. The calculated parameters are input to the subroutine "PTSUB" which, as mentioned before, employs the image modelling and least squares approach and will be discussed in Chapter 9.

The program was tested by a series of simulated crosses as well as the real digitized fiducial marks. The results and the assessment of accuracy are given in Chapter 9.

A sample of the program's output for one of the artificially generated fiducial marks is given in Table 5.1. The listing of the program is presented in Appendix A1.

Detected cross midpoints with the same orientation:

unit: pixels

xmid (left)	ymid (left)	xmid (right)	ymid (righ)
22.50	17.00	52.50	17.00
23.50	19.00	49.00	19.00
25.50	22.00	43.50	22.00
26.50	24.00	40.50	24.00
27.50	25.00	38.50	25.00
27.50	26.00	37.00	26.00
33.50	36.00	19.50	36.00
34.50	38.00	16.00	38.00
35.50	39.00	14.50	39.00
36.50	41.00	11.00	41.00
37.50	43.00	8.00	43.00

CALCULATED CROSS PARAMETERS:

X (PIXELS)	Y (PIXELS)	ORIENTATION ANGLE (DEGREE)
90.85	98.89	59.92

MGL	MBGL
130	25

Table 5.1 A SAMPLE OF THE OUTPUT OF PROGRAM GRAD

5.6 Coordinate Transformation for Inner Orientation

After the coordinates of the centre of the fiducial marks have been determined using the procedures developed above, the second stage of the inner orientation procedure, i.e to perform a two-dimensional transformation to establish the relationship between principal point and microdensitometer coordinate systems, is carried out using the known calibrated coordinates of the fiducial marks. This transformation was performed using a two-dimensional affine transformation given by:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x_d \\ y_d \end{bmatrix} + \begin{bmatrix} x_o \\ y_o \end{bmatrix} \qquad (5.21)$$

where x' and y' are the calibrated coordinates of the fiducial marks with respect to the principal point coordinate system; x_d, y_d are the coordinates of the same fiducial marks with respect to the microdensitometer coordinate system and measured on the digital image;

a,b,c,d are the transformation coefficients; and x_o ,y_o are the two translations in x and y directions respectively.

Due to the high degree of geometrical accuracy and stability of the COSMOS machine (Section 3.4.1), the two-dimensional affine transformation was used only to compensate for the affinities inherent in the aerial photo itself. To test and determine the actual positional accuracy of the COSMOS machine, it was decided to scan an accurate reseau plate, which was used to calibrate the Department's mechanical projection stereo-plotters. The plates were sent to the Royal Observatory in Edinburgh to be scanned with the finest COSMOS spot size. Unfortunately, due to the thickness of the glass plate, it could not be accommodated by the machine. However, although it was not possible to test the geometrical accuracy of the machine, the measurements of the fiducial marks and other sets of measurements carried out during the course of the project showed that the machine can scan images with a high geometrical accuracy (see Chapters 6 and 9).

After the completion of the inner orientation stage, the calculated coefficients of the affine transformation are recorded to be used for the transformation of the pixels from the microdensitometer coordinate system to the principal point coordinate system and vice-versa.

5.7 Relative Orientation

The analytical relative orientation stage starts with the measurement of the image points in the six standard positions in the model area. These points may be measured either automatically using digital image correlation techniques, as described in Section 5.2.1, or manually using a cursor, when the image is displayed on the screen. In the present project, manual pointing to each of the image points was carried out on the IBM 5080 graphics system. The detailed description of the manual pointing procedures is given in Chapter 6.

After the coordinates of the image points in the standard positions of the model had been determined, the usual procedure as applied in analytical photogrammetry was followed, i.e., the coordinates of the measured image points were transformed to the principal point

coordinate system using the coefficients calculated in the inner orientation stage; and the relative orientation was solved using the familiar and well established coplanarity condition equations given by Methley (1986) as follows:-

$$\begin{vmatrix} X_1 & Y_1 & Z_1 & 1 \\ X_2 & Y_2 & Z_2 & 1 \\ X_1+x & Y_1+y & Z_1+z & 1 \\ X_2+x'' & Y_2+y'' & Z_2+z'' & 1 \end{vmatrix} = 0 \qquad (5.22)$$

where X_1 ... Z_2 are the coordinates of the left and right projection centres respectively; x, y, z are the coordinates of the left image points; and x'',y'',z'' are the coordinates of the corresponding points in the second photo rotated into a system parallel to the left image plane.

Eq. 5.22 is non-linear, its first order approximation being given by:

$$C_1(bY)+C_2(bZ)+C_3(d\varphi)+C_4(d\omega)+C_5(d\varkappa)+C_6 = 0 \qquad (5.23)$$

where $bY=(Y_2-Y_1)$, $bZ=(Z_2-Z_1)$, $d\varphi$, $d\omega$, $d\varkappa$ are the five unknown parameters of relative orientation of the right hand camera with respect to the left hand camera, and the values of C are known constants evaluated using approximate values for the unknowns. They are given by:

$$C_6 = bX\begin{vmatrix} y & z \\ y''_o & z''_o \end{vmatrix}, \; C_1 = -\begin{vmatrix} x & z \\ x''_o & z''_o \end{vmatrix}, \; C_2 = \begin{vmatrix} x & y \\ x''_o & y''_o \end{vmatrix}$$

$$C_3 = \begin{vmatrix} bX & bY_o & bZ_o \\ x & y & z \\ z''_o & 0 & -x''_o \end{vmatrix}, C_4 = \begin{vmatrix} bX & bY_o & bZ_o \\ x & y & z \\ 0 & -z_o & y''_o \end{vmatrix}, C_5 = \begin{vmatrix} bX & bY_o & bZ_o \\ x & y & z \\ -y''_o & x''_o & 0 \end{vmatrix}$$

where bX is an assumed value; and subscript o denotes the approximate values determined from the previous iteration.

The solution to the coplanarity equations yields the values for the five unknown parameters of φ, ω, $\varkappa$, bY and bZ for the right hand camera, with respect to the exterior orientation parameters of the left hand camera.

The **model coordinates** and the **residual y-parallaxes** are calculated as follows:

$$\begin{cases} X = X_1 + \lambda x \\ Y' = Y_1 + \lambda y \;,\; Y'' = Y_2 + \lambda y'' \;\Rightarrow Y = (Y' + Y'')/2 \\ Z = Z_1 + \lambda z \\ pY = Y' - Y'' \end{cases} \tag{5.24}$$

where Y' , Y'' denote the Y model coordinates calculated using the values of y and y'' ; pY represents the residual y-parallaxes; and

$$\lambda = z'' \frac{(X_2 - X_1) - x''(Z_2 - Z_1)}{(xz'' - x''z)}$$

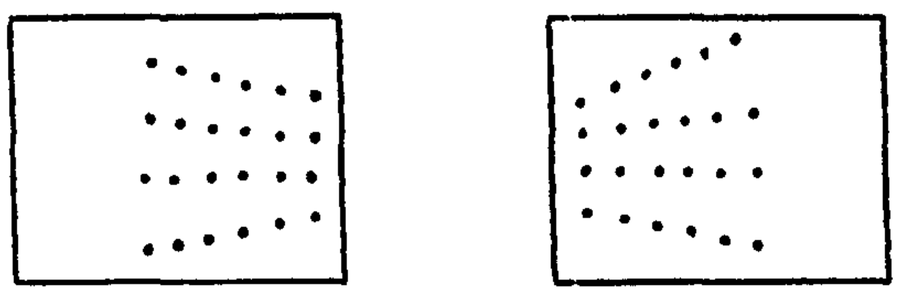
The program "COPLAN" was developed to carry out the task mentioned above. The output of the program including the orientation parameters, the calculated model points and the residual y-parallaxes for different sets of image points data, are given in Chapter 6. The listing of the program is given in Appendix A2.

The geometrical corrections such as lens distortion, atmospheric refraction, etc. may also be implemented in the program. However, in the present project, those corrections were not applied, because they were much smaller than the size of the actual digital image pixels.

5.8 Indirect Geometric Transformation and Resampling

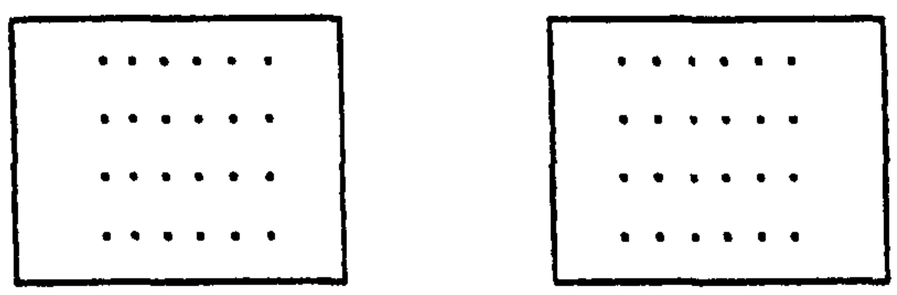
As mentioned earlier, the indirect method of geometric transformation (i.e. the transformation from the output image to the input image plane) has been implemented in this project. Since, the main objective of this stage of the work is to remove y-parallaxes and associated x-parallaxes, the left and right images were resampled into the corresponding epipolar lines. The epipolar geometry is a basic and familiar concept in photogrammetry and is illustrated in Fig. 5.9. This figure depicts the condition of coplanarity and shows the line of intersection of the **epipolar plane** with the left and right image

planes. These lines of intersection are the so-called **epipolar lines**. They are important because, given the left image point p, its corresponding point on the right image p' is known to lie along the corresponding right epipolar line. In a tilted photograph, the corresponding epipolar lines will not be parallel to each other (Fig. 5.8a). If the corresponding epipolar lines are made parallel to the image scan lines, i.e. the output images are resampled along the epipolar lines, then there will be no y-disparity for the corresponding image points (Fig. 5.8b).



ORIGINAL IMAGES

Fig. 5.8a



RESAMPLED IMAGES
INTO EPIPOLAR LINES

Fig. 5.8b

However, if the line AB on the output matrix (i.e. the output pixels along the row directions, see Fig 5.9) is projected into the planes of the left and right input images, the corresponding lines on the left and right image planes will not necessarily be the corresponding epipolar lines. This is due to the fact that the line AB may not lie on the epipolar plane. Hence, to resample the left and right output images along the epipolar lines, the output pixels should first be transformed to the epipolar plane. This transformation to the epipolar plane may be calculated as follows:-

$$\begin{cases} X_{Po} = X_P \quad bX/base \\ Y_{Po} = Y_P + X_P \quad bY/base \end{cases} \tag{5.25}$$

where X_P , Y_P are the coordinates of the output pixel P; X_{Po} , Y_{Po} are the coordinates of the output pixel transformed onto a line which lies in the epipolar plane (pixel Po in Fig. 5.9); $bY=(Y_2 - Y_1)$; $bX=(X_2 - X_1)$; and base is the model base value.

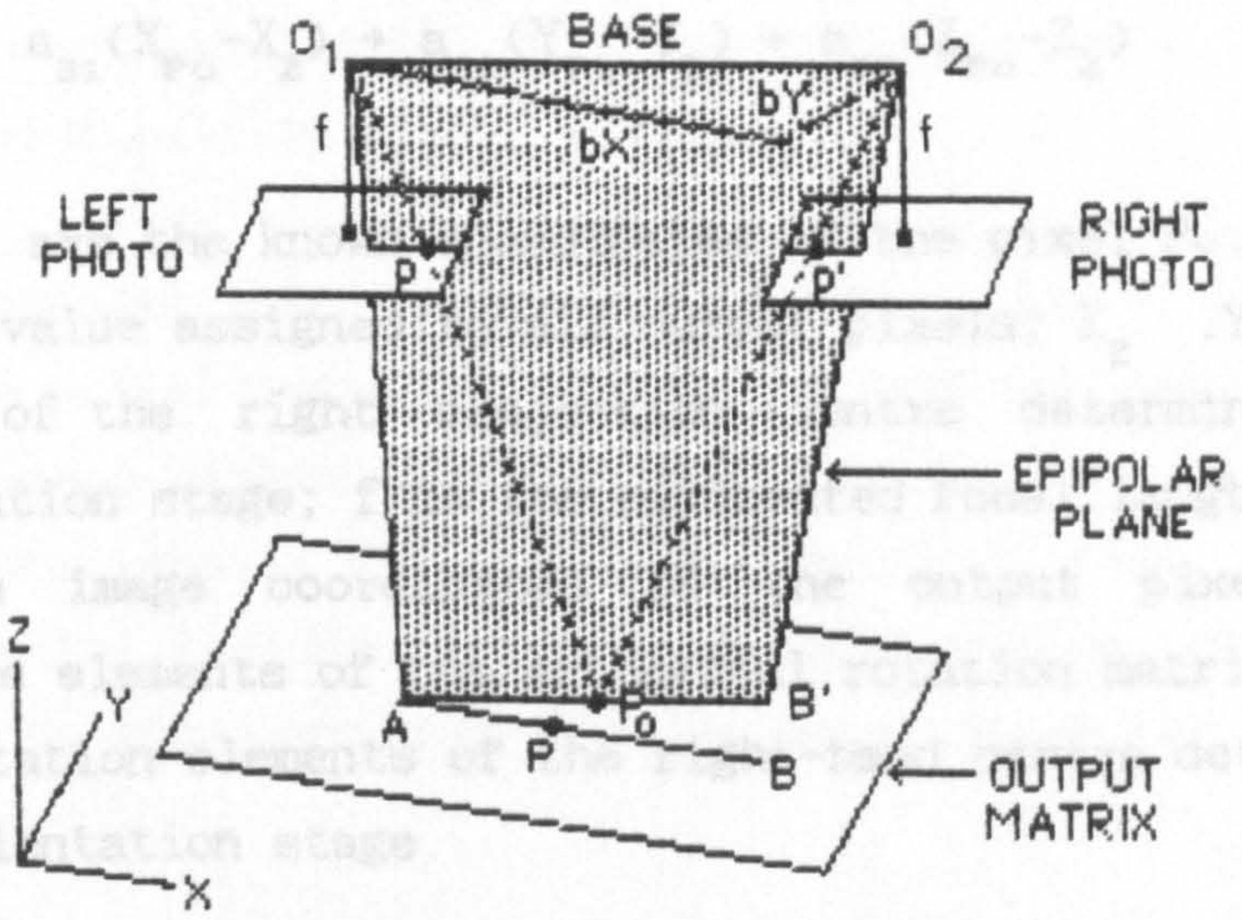


Fig. 5.9 THE EPIPOLAR PLANE.

Regarding what was mentioned in the preceding paragraphs, the indirect geometric transformation of the output pixel was carried out using three different transformations of:

(i) transformation into the epipolar plane;
(ii) transformation into the principal point coordinate system; and
(iii) transformation into the microdensitometer coordinate system;

The first transformation (i.e. transformation into the epipolar plane) has been performed according to the procedures described in the previous paragraphs. For the second transformation, the pixel Po (Eq. 5.25), which has already been transformed into the epipolar plane (see Fig. 5.9), is projected up into the principal point coordinate system using the well known collinearity equations which for the right hand image is given by :

$$x_p' = -f \frac{a_{11}(X_{Po}-X_2) + a_{12}(Y_{Po}-Y_2) + a_{13}(Z_{Po}-Z_2)}{a_{31}(X_{Po}-X_2) + a_{32}(Y_{Po}-Y_2) + a_{33}(Z_{Po}-Z_2)}$$

$$y_p' = -f \frac{a_{21}(X_{Po}-X_2) + a_{22}(Y_{Po}-Y_2) + a_{23}(Z_{Po}-Z_2)}{a_{31}(X_{Po}-X_2) + a_{32}(Y_{Po}-Y_2) + a_{33}(Z_{Po}-Z_2)} \qquad (5.26)$$

where X_{Po} , Y_{Po} are the known coordinates of the pixel Po; Z_{Po} is an arbitrary fixed value assigned to all output pixels; X_2 ,Y_2, Z_2 are the coordinates of the right projection centre determined in the relative orientation stage; f is the calibrated focal length; x_p', y_p' are the unknown image coordinates of the output pixel Po; and $a_{21}...a_{33}$ are the elements of the orthogonal rotation matrix evaluated using the orientation elements of the right-hand camera determined in the relative orientation stage.

The pixel p' , which has already been transformed into the principal point coordinate system, is then transformed into the microdensitometer coordinate system using the calculated coefficients obtained during the inner orientation stage (i.e., the inverse solution of Eq. 5.21).

As mentioned in Section 5.3, the output pixel transformed into the plane of the input image, does not necessarily coincide with the centre of a pixel in the input image and the grey level interpolation becomes necessary.

The three interpolation algorithms -- nearest neighbour, bilinear and bicubic -- have all been implemented. To apply the **bilinear interpolation**, Eq. 5.6 has been rearranged into the following form :-

$$g_o^T(i+m,j+n) = (1-m)(1-n)g_i(i,j)+m(1-n)g_i(i+1,j)+ n(1-m)g_i(i,j+1)+(mn)g_i(i+1,j+1) \qquad (5.27)$$

For the **bicubic convolution**, Eq. 5.7 was implemented in two dimensions using the same procedures described in Section 5.3.1.

The geometric transformations and density assignment carried out for the pixel P which were described in the preceding paragraphs have been repeated for all pixels in the image.

The same procedures have also been carried out for the left output image. However, the values of a_{21} ... a_{33} (Eq. 5.26) are evaluated by substituting zero values for the φ, ω and $\varkappa$; which means the rotation matrix is reduced simply to a unit matrix.

It should be mentioned that through the rectification of both the left and right images by following the procedures described above, not only are the y-parallaxes removed but the process will also eliminate the associated x-parallaxes. This is due to the fact that, the differential digital rectification procedure simulates digitally the same operations as those applied in analogue plotters for rotating the projectors.

5.9 Test Results

The program "DIGREC" was developed to perform both the geometric transformation and the density assignment (resampling) on a pixel by pixel basis. Although not strictly necessary, this was carried out mainly to provide intensive numerical computations for the University/IBM Kelvin Project. To investigate the potential of the program when utilizing the vector hardware of the IBM 3090 machine, a modification was applied to the program. The vectorization procedures and the speed-up factors achieved when running the program will be presented and discussed in Chapter 10. The listing of the program is given in the Appendix A3. In the following paragraphs, the test results are presented.

(i) Simulated data

To carry out the preliminary tests on the rectification program and to demonstrate the effect of the different interpolation strategies, an ideal simulated cross was generated. The simulated cross was rotated 45 degrees and three different interpolation algorithms were applied. The results are given in Fig. 5.10.

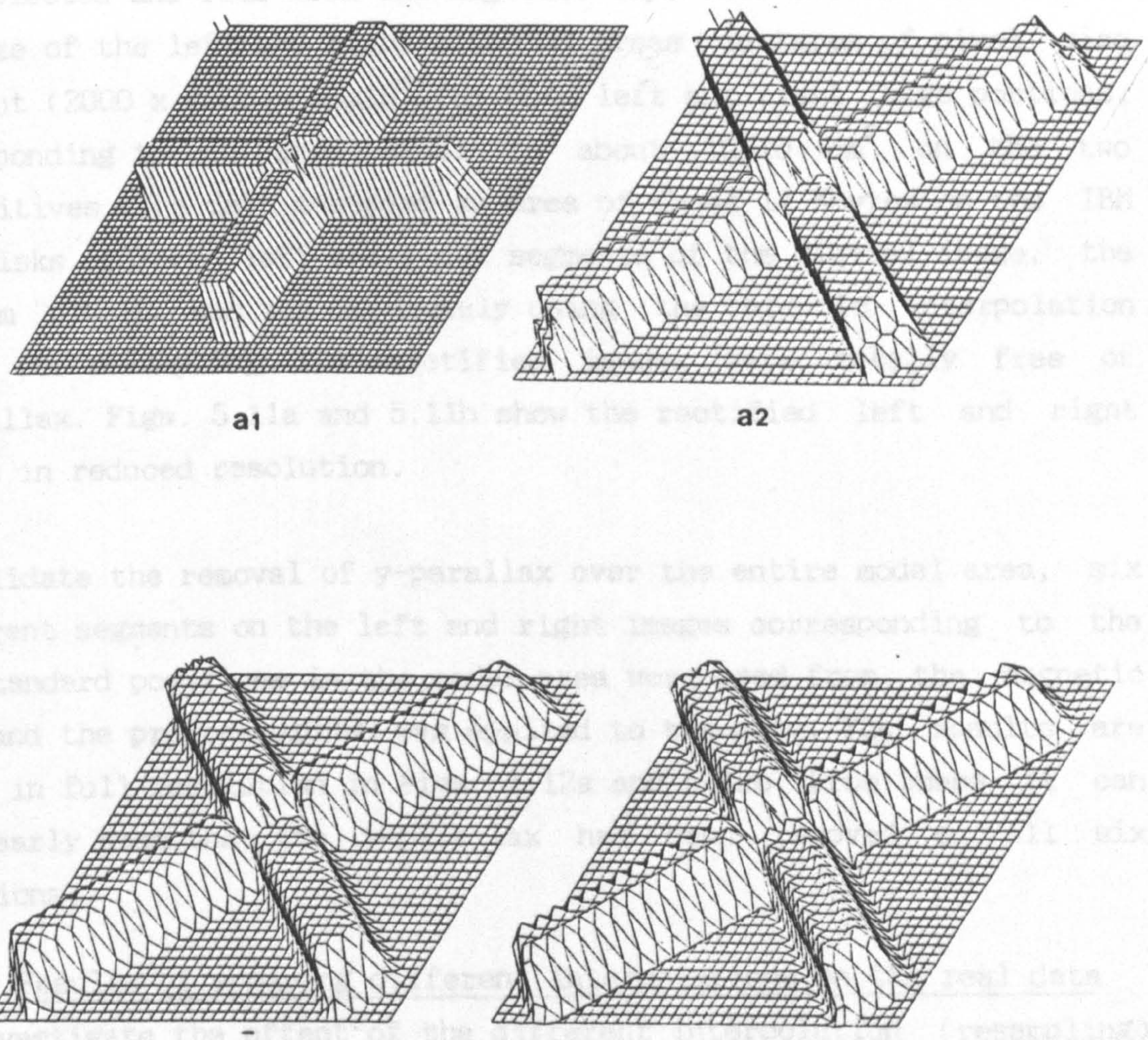


Fig. 5.10 DIGITAL RECTIFICATION
OF SIMULATED DATA.
a1 - ORIGINAL IMAGE
a2 - NEAREST NEIGHBOUR INTERPOLATION
a3 - BILINEAR INTERPOLATION
a4 - BICUBIC CONVOLUTION

(ii) Real data

In the next step, two portions of the left and right digital images were selected and read from the magnetic tape onto the IBM 3090 disks. The size of the left and right selected areas in terms of pixel size is about (2000 x 2000)x2=$8x10^6$ for both left and right image portions, corresponding to an area equal to about 10x10 cm^2 on the two diapositives. The data occupied an area of about 14 Mbytes on the IBM 3090 disks. For the left and right segments of the digital image, the program "DIGREC" was run separately using the bicubic interpolation method for resampling. The rectified images were totally free of y-parallax. Figs. 5.11a and 5.11b show the rectified left and right images in reduced resolution.

To validate the removal of y-parallax over the entire model area, six different segments on the left and right images corresponding to the six standard positions in the model area were read from the magnetic tape and the program DIGREC was applied to the data. The results are given in full resolution in Figs. 5.12a and 5.12b. From these it can be clearly seen that the y-parallax has been removed on all six positions.

(iii) Results of applying different interpolations on the real data

To investigate the effect of the different interpolation (resampling) functions on the test data, three different runs of the program were carried out, each with a different interpolation algorithm. The results are given in Fig. 5.13. As the results show, there are very small differences between the grey level images produced by interpolation with the different algorithms. The nearest neighbour interpolation, for example, gives rise to a very slight blocky appearance on the linear features, but this is quite insignificant and can be seen only when the image is enlarged (Fig. 5.14b). The same holds true for the image produced when using the bilinear interpolation routine (compare the images in Fig. 5.13 and their corresponding enlarged images in Fig. 5.14). This may suggest that in so far as the rectification of a vertical photo with small amounts of tilts and relief displacements is concerned, the choice of the interpolation function has a very small effect on the output quality.

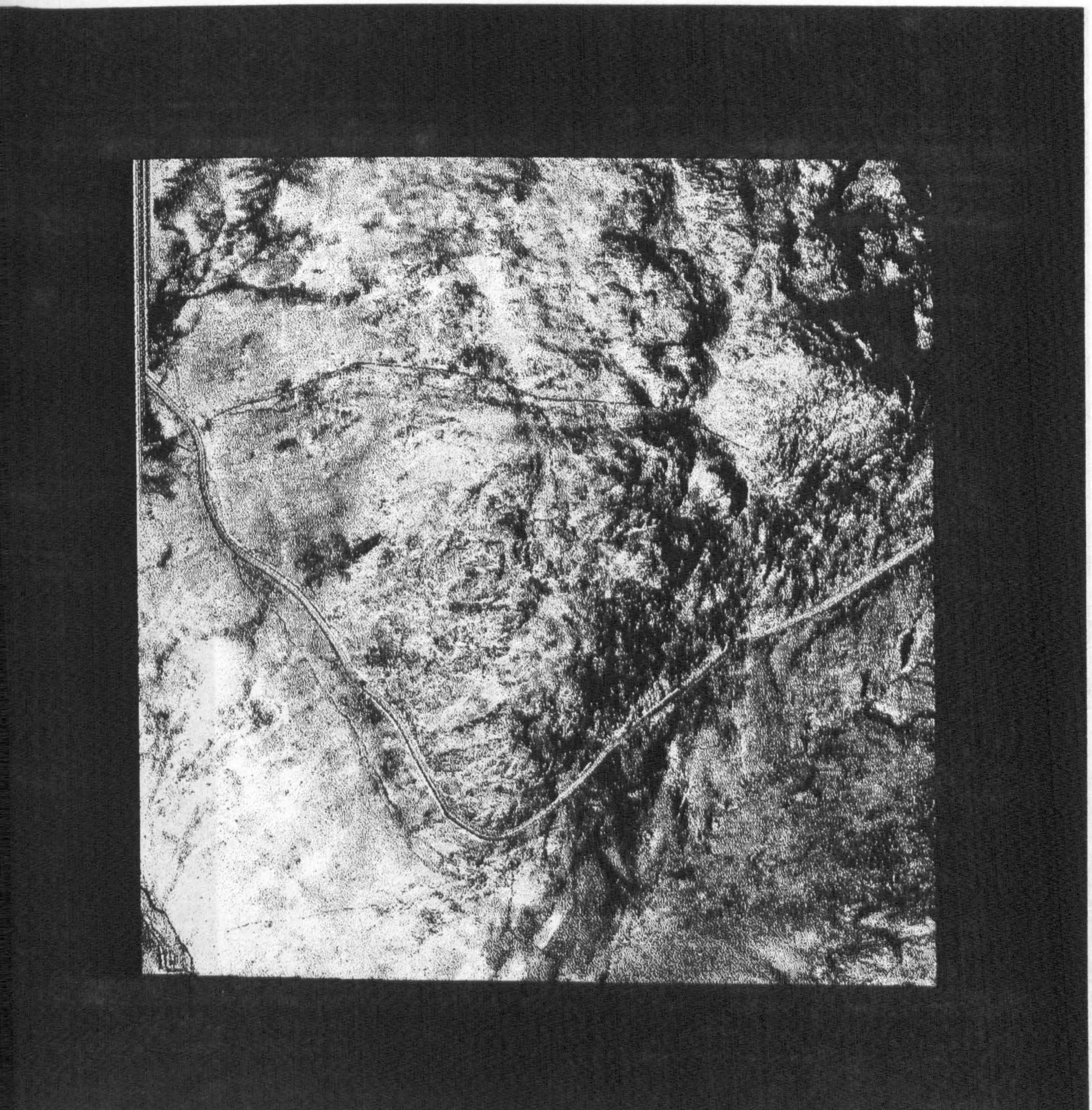


Fig.5.11a Rectified right image (reduced resolution).

Fig.5.11b Rectified left image (reduced resolution).

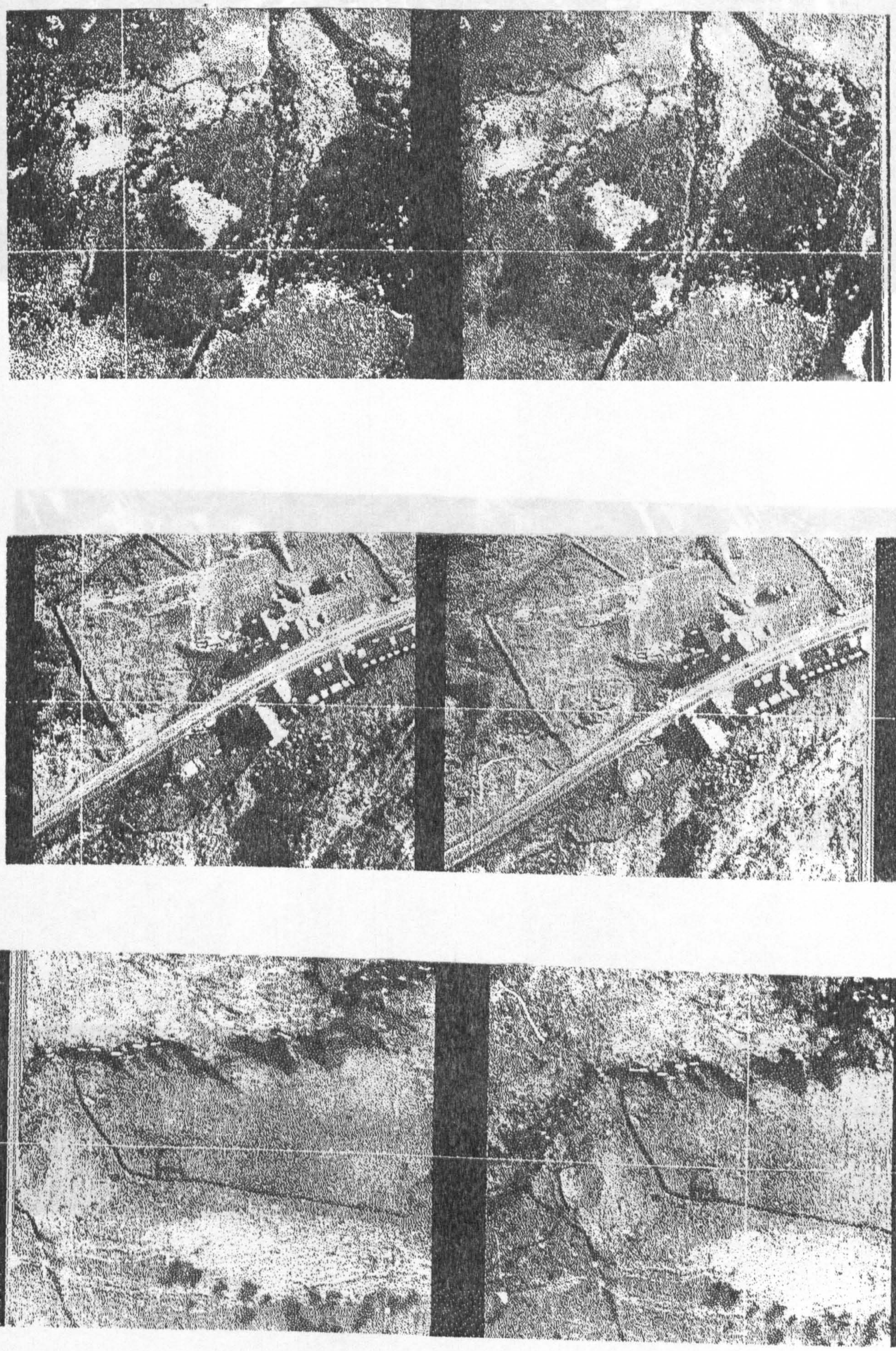


Fig. 5.12a

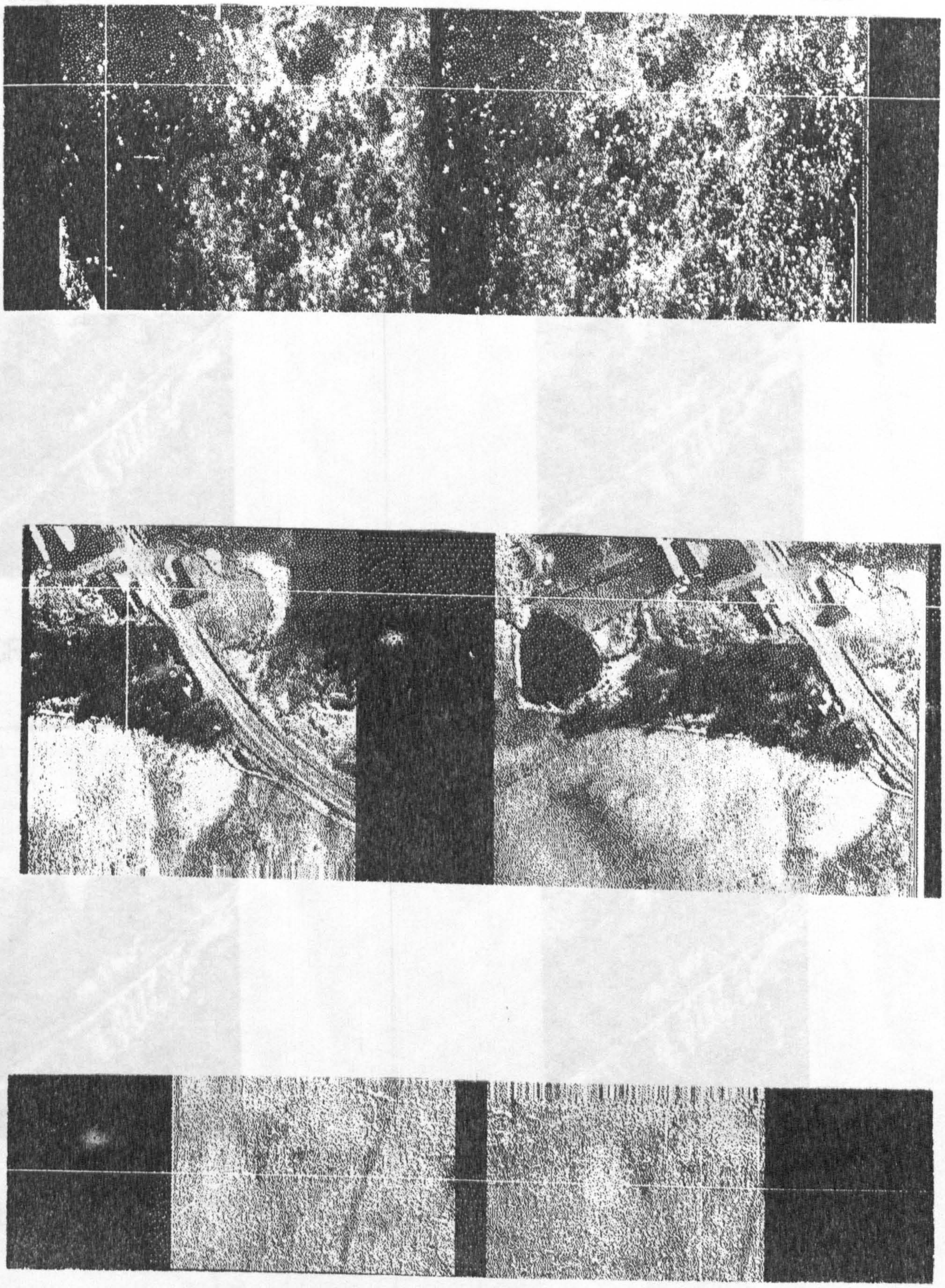
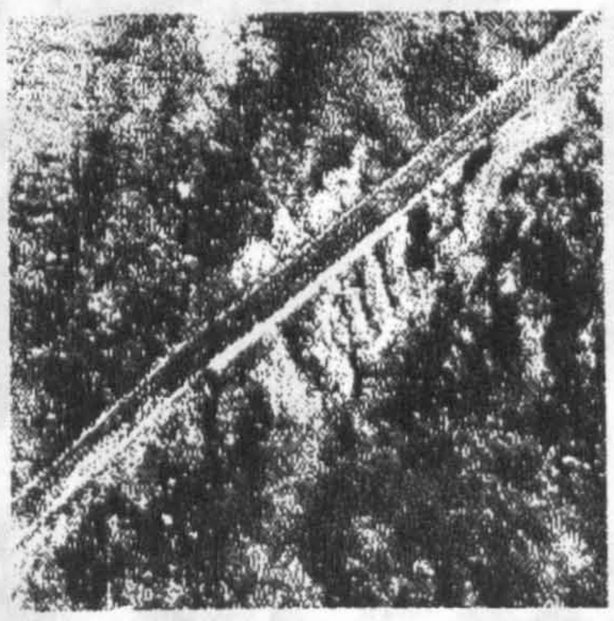
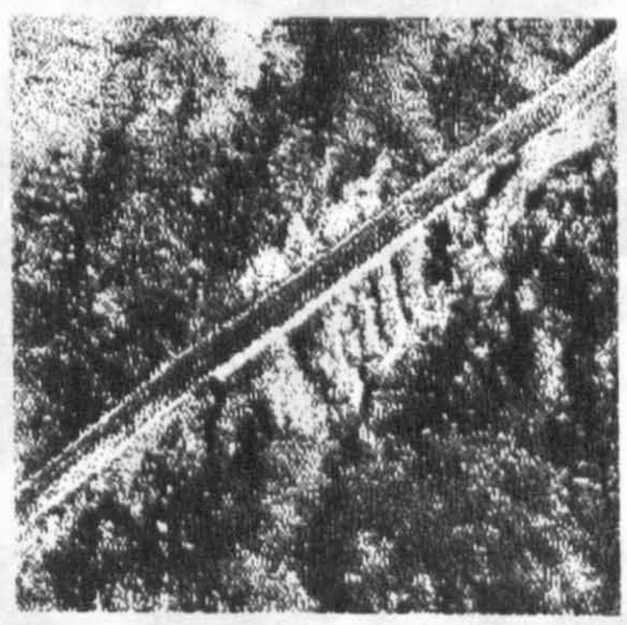


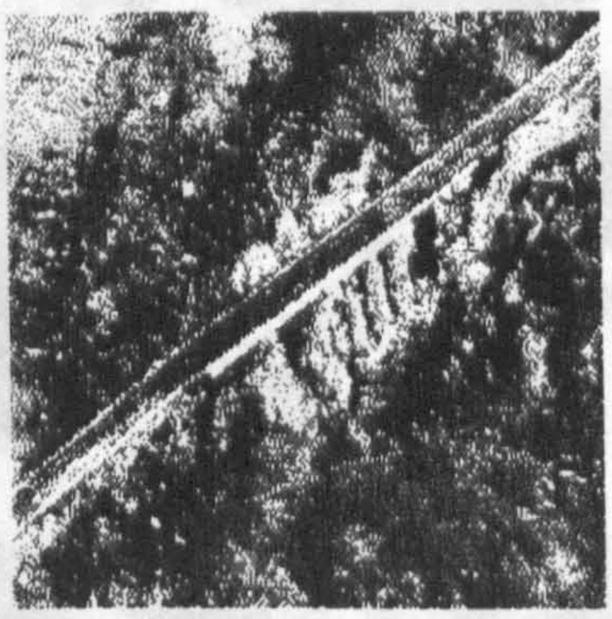
Fig. 5.12b



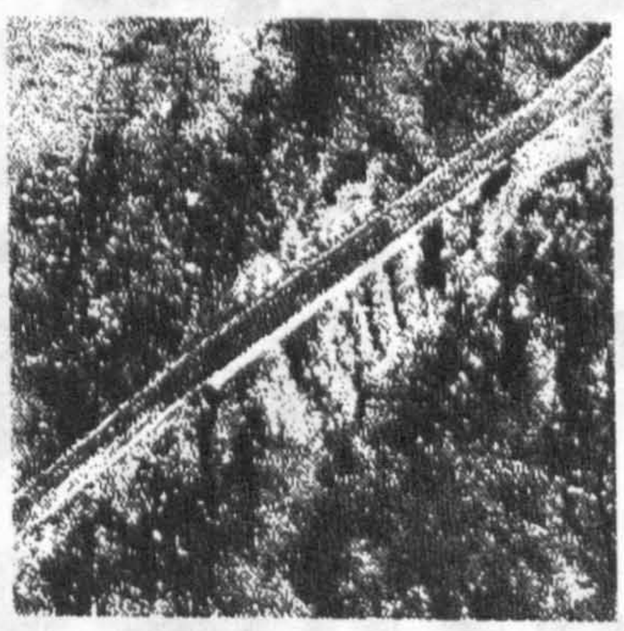
ORIGINAL IMAGE



NEAREST NEIGHBOUR INT.



BILINEAR INT.



BICUBIC INT.

Fig. 5.13

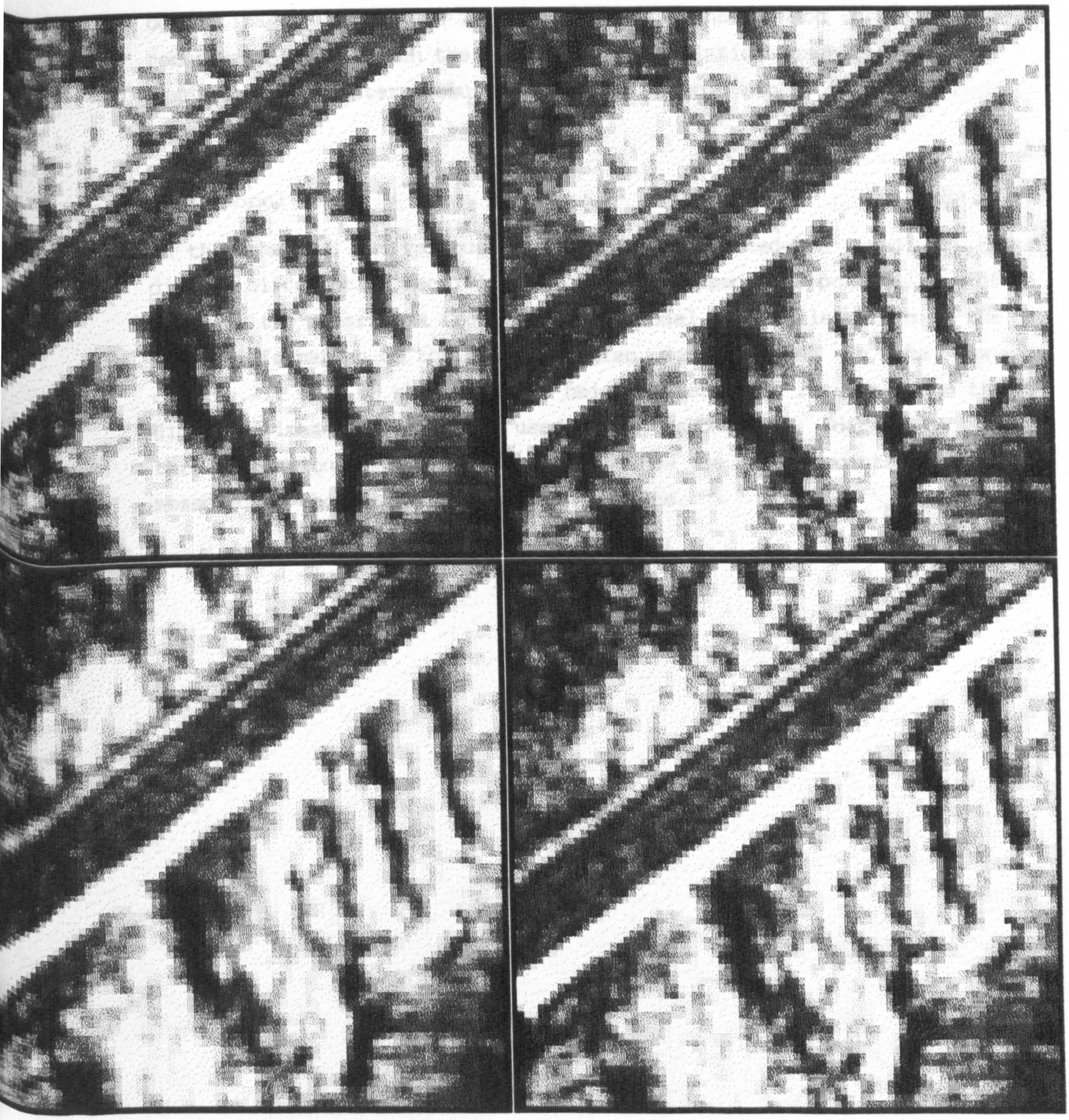


Fig.5.14
a. Original image (enlarged).
b. Nearest neighbour interpolation.
c. Bilinear interpolation.
d. Bicubic convolution.

Hence, in many cases, a simple nearest neighbour interpolation, which is computationally much more efficient than either of the two other interpolations (see Chapter 10), can be used with a good quality output image. This is not in agreement with the common practice in remote sensing in which the bicubic interpolation usually gives a significantly better result than the two other interpolation algorithms.

5.10 Conclusion

In this chapter, the various aspects and the procedures involved in digital differential rectification were discussed. The possibility of removing the y-parallax from the entire model area while retaining the height information on the image was also demonstrated. In the next chapter, it will be shown how the height information retained on the digital images can be used to reconstruct digitally a three-dimensional view of the model and how the model points may be measured to reconstruct the terrain surface.

CHAPTER SIX

Design and Implementation of a Purely Digital Stereo-Photogrammetric System on the IBM 3090 Computer

6.1 Introduction

In the previous chapters, it was shown that the digitized images contain two sets of grey level degradations and (more importantly) geometric displacements/distortions and it was demonstrated that these result from the degradations and displacements present on the original analogue images plus those degradations imposed by the digitizing microdensitometer itself. In the last two chapters, it was shown how these pictorial degradations and geometric displacements can be removed and the digital image restored and rectified. In the present chapter, an attempt is made to reconstruct a three-dimensional model surface using the restored and rectified left and right digital images, and subsequently to sample and measure the model points manually to generate a DTM. In order to generate the model surface and measure the model points, a purely digital stereo-photogrammetric system, which digitally simulates features of both analogue and analytical plotters has been designed and implemented. This chapter reviews the principle, configuration and state-of-the-art of digital stereo-photogrammetric systems, and describes the design and implementation of the digital stereo-photogrammetric system implemented in this project. The output and the accuracy estimation for the output (i.e. the generated DTM) are also presented and discussed. The automatic generation of a DTM will be discussed later in Chapters 8 and 9.

To realize the concept of a digital stereo-photogrammetric system, it is necessary to have a brief review of the principle of analytical photogrammetric instruments.

6.2 The Basic Principle of Analytical Photogrammetric Instruments

Analogue plotters use an **optical or mechanical projection** of the image to solve the collinearity equations. Instruments based on the analytical procedures, on the other hand, establish the image/object point relationship by a **digital solution** of the collinearity equations. From the instrumental point of view, there are several alternative ways in which these solutions may be implemented (Petrie, 1990). These are mainly:

(i) The use of a **comparator** as a measuring device and the execution of an analytical photogrammetric solution as an off-line process carried out later in a computer. Mono and stereo comparators are typical examples of this group of analytical photogrammetric instruments. They are analytically-based instruments which have no oriented stereo-model. Thus measurement is carried out on a point by point basis with off-line computation of the measured image coordinate data.

(ii) The use of a comparator but with the numerical computational solution executed as an on-line process. Examples are **digital monoplotters** and **image space plotters**. They have no oriented stereo-model, and measurement is carried out on a point by point basis, but unlike the first group, the computational processing of the measured image coordinate data is carried out in an on-line mode.

(iii) The construction and operation of an **analytical plotter** in which the computer is totally integrated into the design of the instrument and the numerical solution is always an on-line process, executed in real time. In other words, the analytical plotter features a closed-loop system in which the computer provides a real-time solution of the analytical photogrammetric equations, and issues control signals to motors and driving elements which also move the plates to the required position in real time, resulting in a continuously oriented stereo-model.

From an algorithmic point of view, there are two basic approaches to implementing analytical photogrammetric solutions (Helava, 1980):

(i) **Image space primary**. This approach involves the measurement and input of image coordinates, and the calculation of the object coordinates of these points (Fig. 6.1a). The mathematical basis of this approach is the analytical solution of inner, relative and absolute orientation to establish the camera attitude and altitude at exposure time. This is followed by space intersection to calculate the object space coordinates for the image coordinates of the corresponding points.

(ii) **Object space primary**. This approach is the inverse of the first, in which the input is the model coordinates and the output is the corresponding image coordinates of these points (Fig. 6.1b). The mathematical basis of this approach is again the analytical solution of inner, relative and absolute orientation. This is followed by the solution of collinearity equations with the terms rearranged so that the input and output values are the object and the corresponding image coordinates respectively.

From the brief review presented above, it can be seen that, in analytical plotters, some photogrammetric operations are performed digitally and hence, compared with analogue plotters, the mechanical

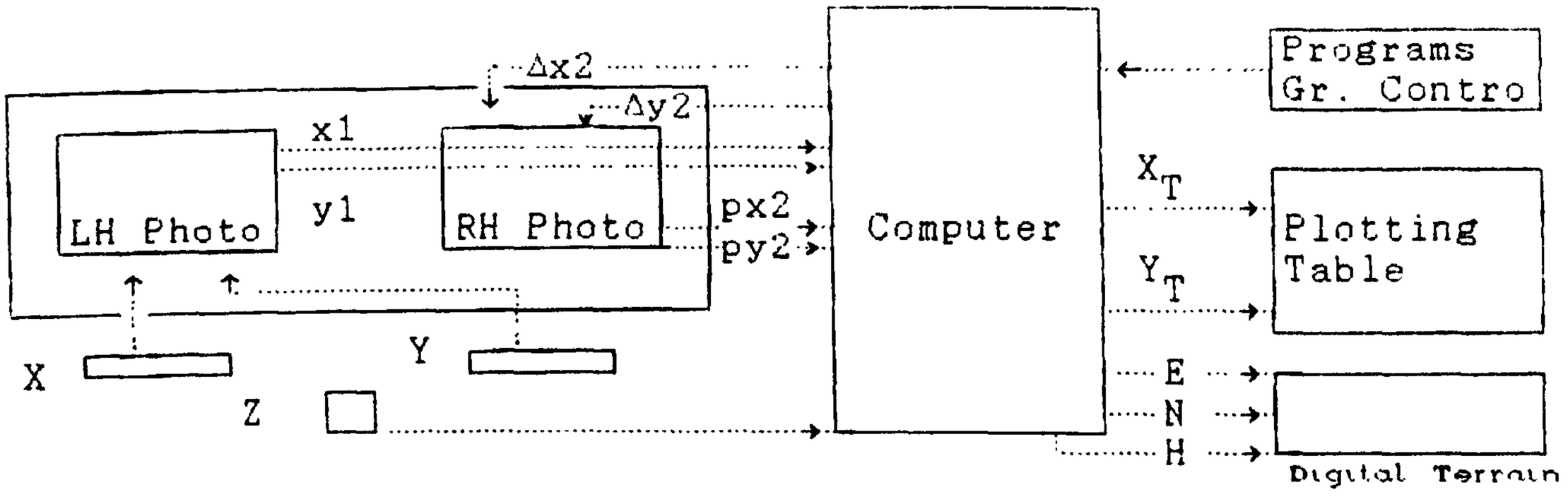


Fig. 6.1a Analytical plotter with image coordinates primary. (After Petrie, 1990)

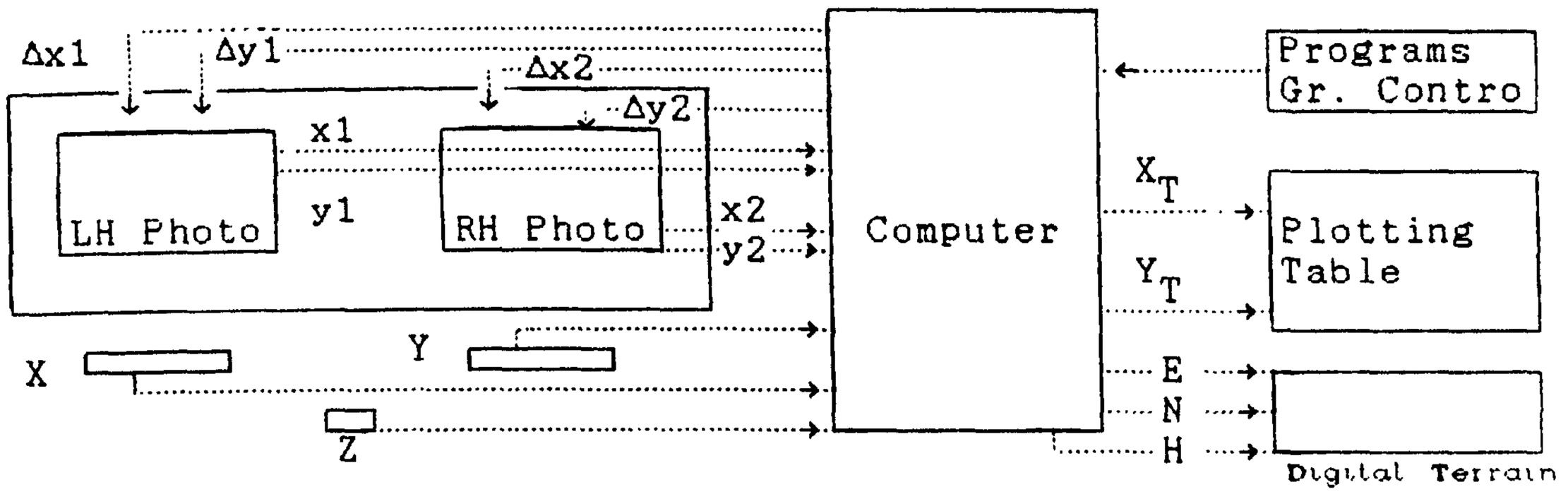


Fig. 6.1b Analytical plotter with object coordinates primary. (After Petrie, 1990)

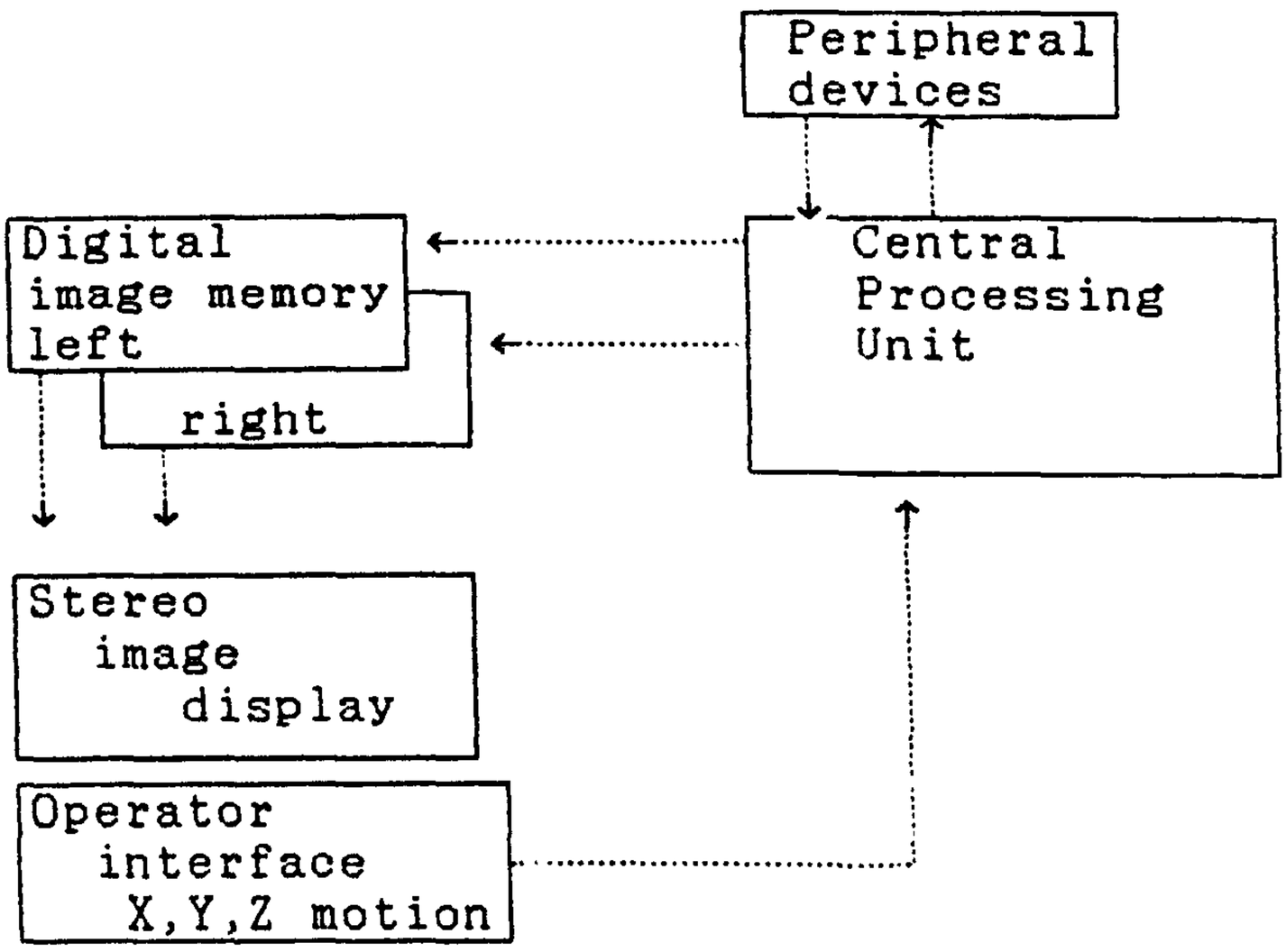


Fig. 6.2 General configuration of purely digital stereo-photogrammetric system

parts of the machine are greatly simplified. Nevertheless, in analytical plotters, the mechanical parts for shifting the hard copy images are still required.

One further step towards a fully-digital solution is the possibility of implementing a **"hybrid analytical system"**. In a hybrid system, the images are digitized "on demand", as needed for processing. Nevertheless, the images remain stored on film. Analytical plotters equipped with CCD cameras (e.g. the Kern DSR 11) are typical examples. This approach provides the possibility of implementing an automatic digital solution to target measurement and DTM generation using correlation techniques (see Chapters 8 and 9). However, this approach still needs servo motors and mechanical photo stages for the image displacements and high quality optical trains to allow stereoscopic viewing of the overlapping photographs.

In the next section, the possibility of realizing a fully-digital stereo-photogrammetric system is presented.

6.3 The Concept of an All Digital Stereo-Photogrammetric System (DSP)

The third generation of plotter, the so-called **Digital Stereo Photogrammetric plotter (DSP)** gives a purely digital solution to the problems mentioned above by replacing the analogue images stored on film by their digital equivalents stored in computer-compatible form. Hence the possibility arises to replace the mechanical parts of the plotter by the computer's capability to carry out data manipulation, processing and scrolling using image processing techniques. The analytical solution of the photogrammetric equations and the algorithmic approaches are the same as for the analytical plotter mentioned in the preceding section -- though the digital image processing required is a considerable task. The reason is that it is necessary to maintain the continuous orientation and formation of stereo-model, and to execute the image movement in real time with the ability to roam over the entire stereo-model as required.

The concept of an ideal DSP working according to the object space

primary solution is shown in Fig. 6.2. The two images forming the stereo-model are stored in the computer's memory. The object points are sent to the computer in a manner similar to that of the analytical plotters, and the basic photogrammetric problems are solved and the image coordinate increments required to maintain an oriented model are calculated. These increments are used directly to shift the images, which are stored in the memory in real time. As discussed in Chapter 5, these image shifts require the implementation of a sampling method. Also, in a similar manner to that of the analytical plotter, the y-parallaxes are removed differentially when the operator moves from one model point to another. However, unlike analytical plotters, no mechanical parts are required.

For **stereo-viewing**, the images are displayed on a display device (e.g. a CRT). The operator interface for inputting the object point coordinates is implemented via a tracker ball, mouse, etc., and the floating marks are simulated graphically by cursors. Hence, the two main requirements for implementing a DSP, in addition to a digital computer, are:

(i) that the data of the two images are displayed in such a way that stereoscopic vision is made possible;

(ii) that for stereo-photogrammetric measurements, a real time three-dimensional control of the measuring marks (cursors) is provided.

These are discussed in more detail in the following sections.

6.3.1 Digital Image Display

To display an image with a continuous grey level variation, a suitable raster-driven display unit is required. The raster-driven display involves the scanning of every pixel on a rectangular matrix which covers the entire screen.

Since the display units are one of the vital components of the DSPs, a brief review of the principle and construction of these display units is given.

Display units have basically two items of hardware: (i) display controller, (ii) display device.

(i) **display controller**. The display controller sits between the computer and the display device, receiving information from the computer and converting it into signals acceptable to the display device. Most refresh displays incorporate a large digital memory or **frame buffer** to store the information received from the computer and to continually refresh the image displayed on the screen. Different kinds of memory have been used as frame buffers, such as drums, disks, integrated circuits, shift registers, and core stores. Nowadays, most frame buffers comprise Random Access Memories (RAM) implemented in integrated circuit technology.

The frame buffer accepts and stores data from the computer in a digital form while the CRT is an analogue device which accepts only analogue signals. Hence conversion of the signals from the digital representation into the analogue representation must take place so that the image can be displayed on the CRT. This task is accomplished by a **digital-to-analogue converter (D/A)** which is fitted to the output end of the frame buffer.

The basic black and white frame buffer has one memory bit allocated for each location (i.e. pixel) on the raster display device. The overall memory dedicated to the display of the image or graphic is called a **bit plane** (Fig. 6.3a). A single bit memory of this type can only control the spot at a specific location on the screen to be either on or off. The variation in the intensity levels can be accomplished by using more bit planes (Fig. 6.3b). With 8 bit plane, $2^8 = 256$ intensity levels can be achieved.

For colour displays, a group of bit planes can be implemented (Fig. 6.3c) for each primary colour (Red, Green, Blue) to give a density range for each primary colour. For example, 8 bit planes for each primary colour can produce a density variation of 256 for each primary colour yielding 2^{24} =16,777,216 possible colour shades. This is termed a full colour frame buffer.

(ii) **display device**. The purpose of the display device is to convert the signals received by the display controller into a visible image. A Cathode Ray Tube (CRT) is the device mainly used for this purpose. The basic arrangement of a monochrome CRT is shown in Figs. 6.3a and 6.3b. At the narrow end of a sealed conical glass tube is an electron gun that emits a high velocity, finely focused beam of electrons. The other end, the face of the CRT, is more or less flat and is coated on the inside with phosphor, which glows when the electron beam strikes it. The energy of the beam can be controlled so as to vary the intensity of the light output. A yoke, or system of electromagnetic coils, is mounted on the outside of the tube at the base of the neck; it deflects the electron

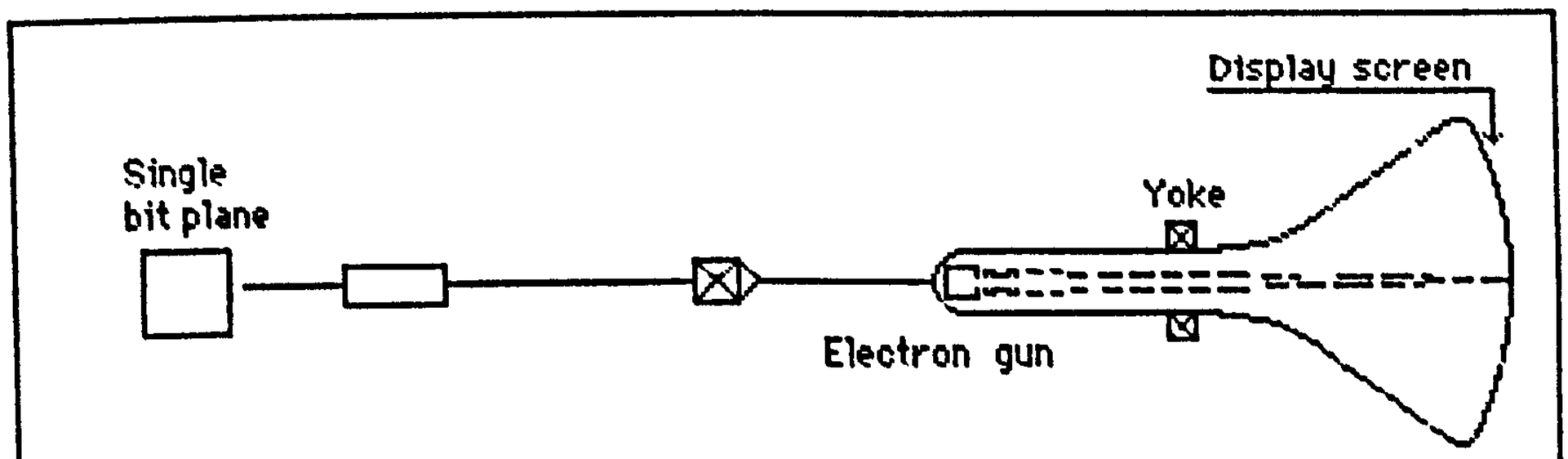


Fig. 6.3a Single bit-plane monochrome frame buffer display unit.

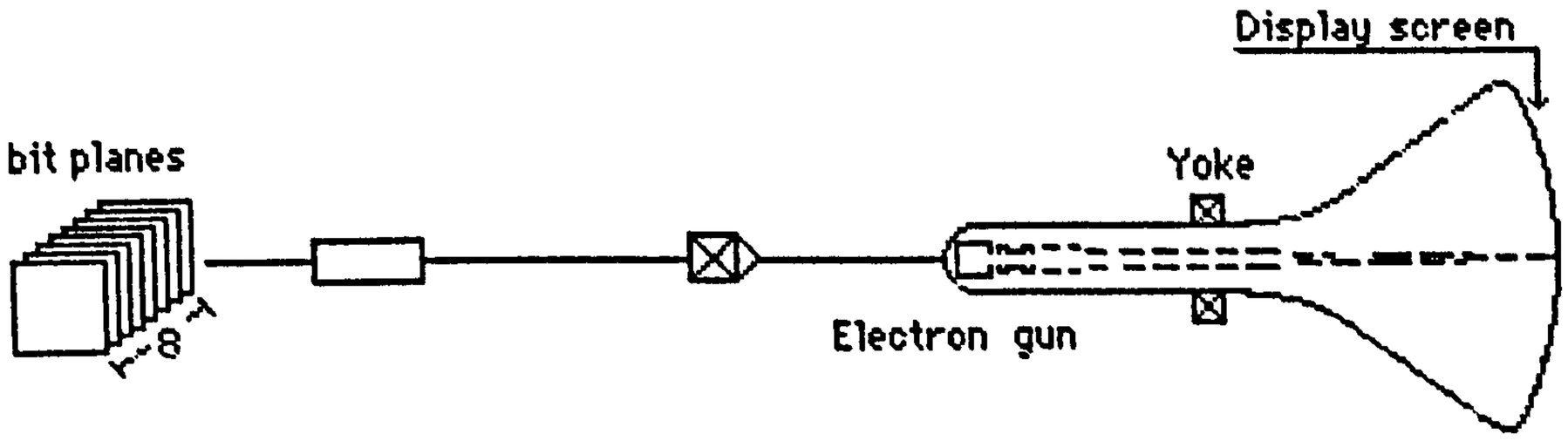


Fig. 6.3b An 8-bit plane monochrome frame buffer display unit.

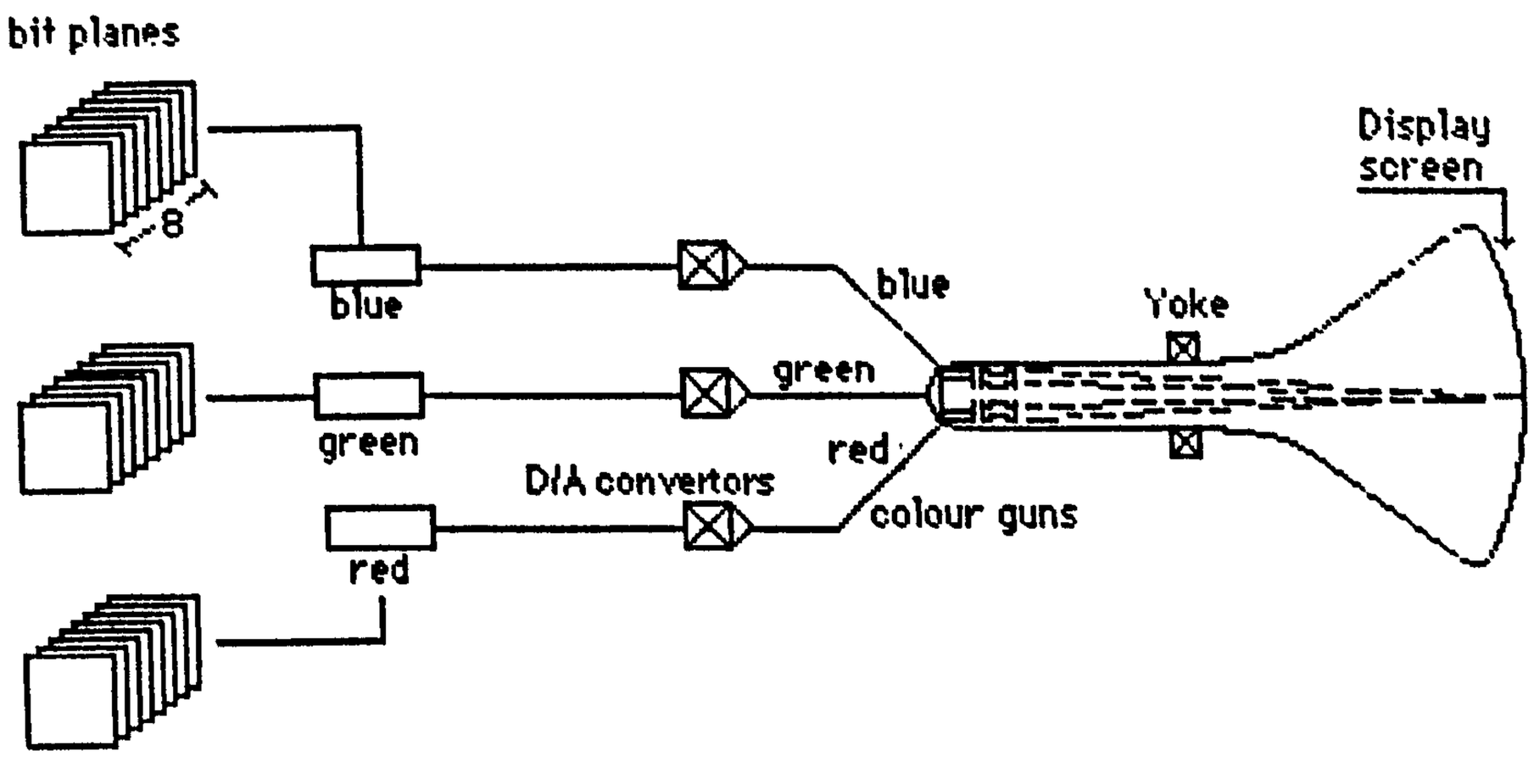


Fig. 6.3c A 24-bit plane colour frame buffer display unit.

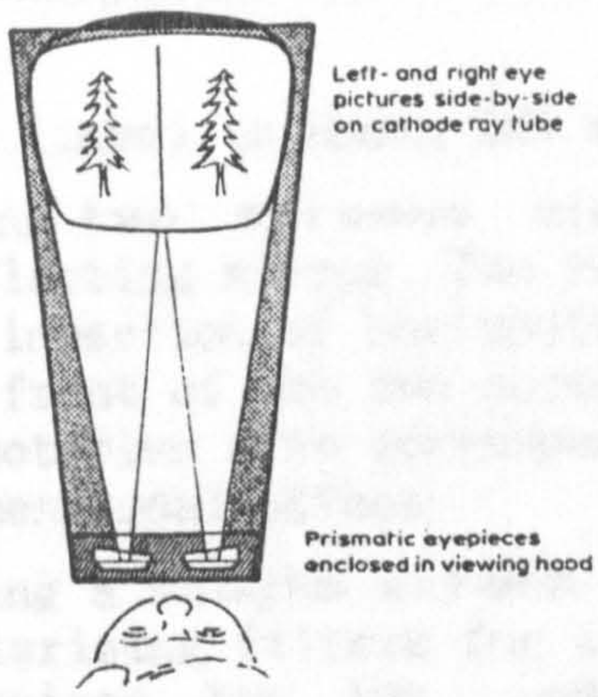
beam to different parts of the tube face when currents pass through the coils. The light output of the CRT's phosphor falls off rapidly after the electron beam has passed by; and a steady picture is maintained by tracing it out rapidly and repeatedly. This is achieved by systematically scanning the whole of the screen in a parallel series of horizontal scan lines, i.e. in a raster scan pattern. When the electron beam has reached the end of the screen it flies back to the top corner of the screen. A display device which employs these systematic line by line scanning is usually referred to as **non-interlaced** display device. Generally this refresh process is performed with rate of 25 to 60 Hz. If the refresh rate of the device is low, then only every second line, e.g. the even-numbered lines, of the display image may be written every 1/50th second. During the next period of 1/50th second, the odd-numbered lines, will be written. Each of the two sets of interlaced lines persists long enough on the screen to give the impression of a single continuously displayed image. A display device which employs this approach is referred to as **interlaced** display device.

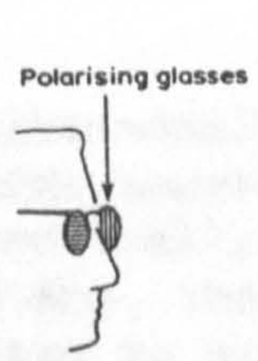
Unlike a monochrome display, a **colour display** usually utilizes a single tube with three electron guns, one for each of the three basic colours. Instead of the normal monochrome phosphor coating, the whole screen is covered by an array of tiny Red, Green, and Blue phosphor dots which are arranged in groups of three each, each group arranged in a triangular pattern to match the layout of the electron guns. Between the screen and the guns is a steel shadow mask, which contains a pattern of tiny holes and aligned so that each primary colour beam hits only its corresponding phosphor dot. Hence, when an image is displayed on the screen using the three electron guns, the images merge together to form a composite colour image.

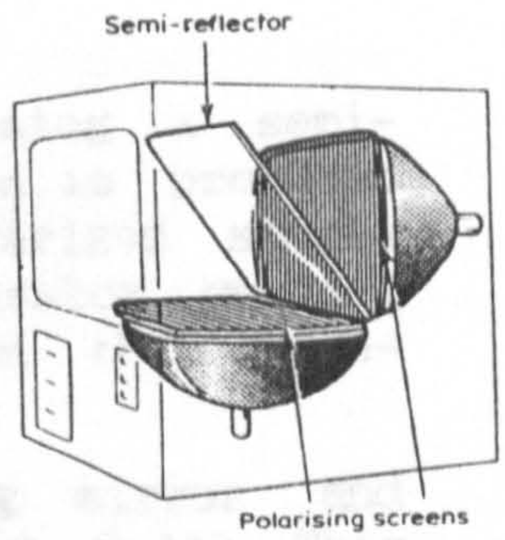
6.3.2 Stereo Viewing on Display Devices

As mentioned earlier, one of the main requirements for implementing a DSP, is to make possible the three-dimensional viewing of the digital left and right overlapping images. A comprehensive survey of the different possibilities to achieve stereo-viewing using display units has been given by Petrie (1983). A brief summary covering some of those possibilities is given in the following paragraphs.

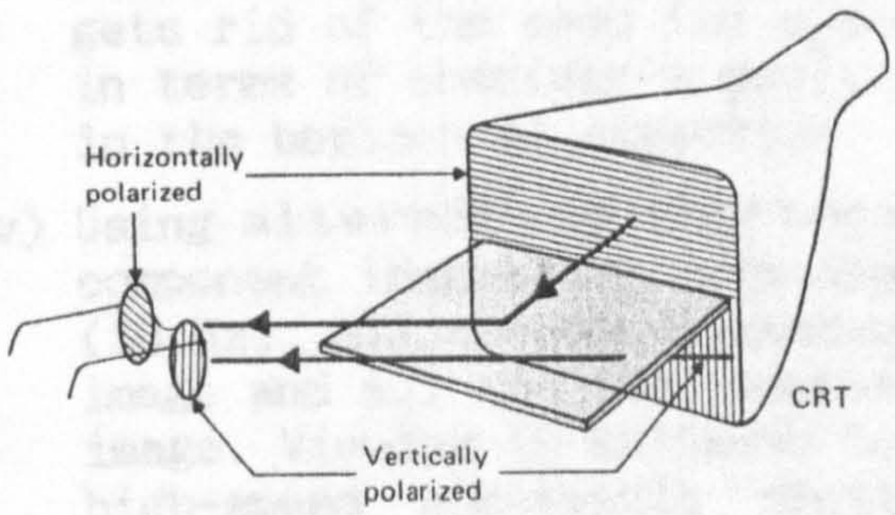
(i) The use of a **single screen** on which the two component images appear side-by-side and are viewed by a hood containing two lenses with prismatic wedges which may be directed towards two component images making up the stereo model (Fig. 6.4a). A disadvantage of the arrangement is that it reduces the coverage to only half the size of the display device. This is particularly significant when the image is digitized with the smallest possible pixel size. The largest portions of the left and right images digitized with a 10 μm pixel size which can be viewed at each time (for a display screen of 512 by



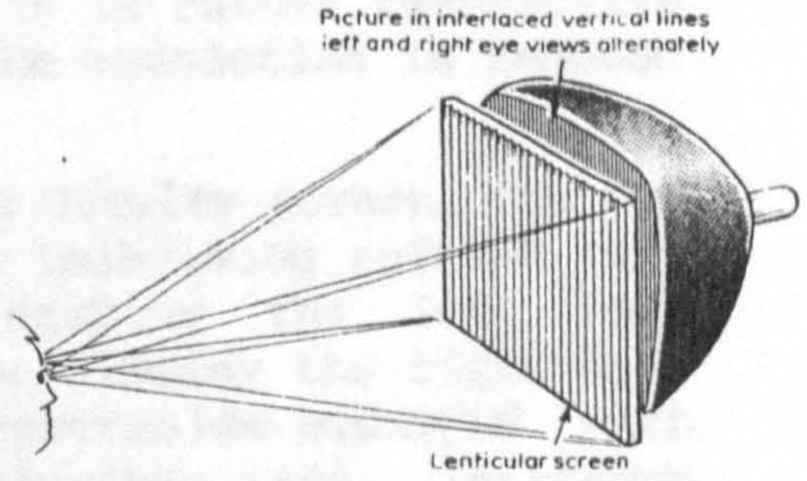




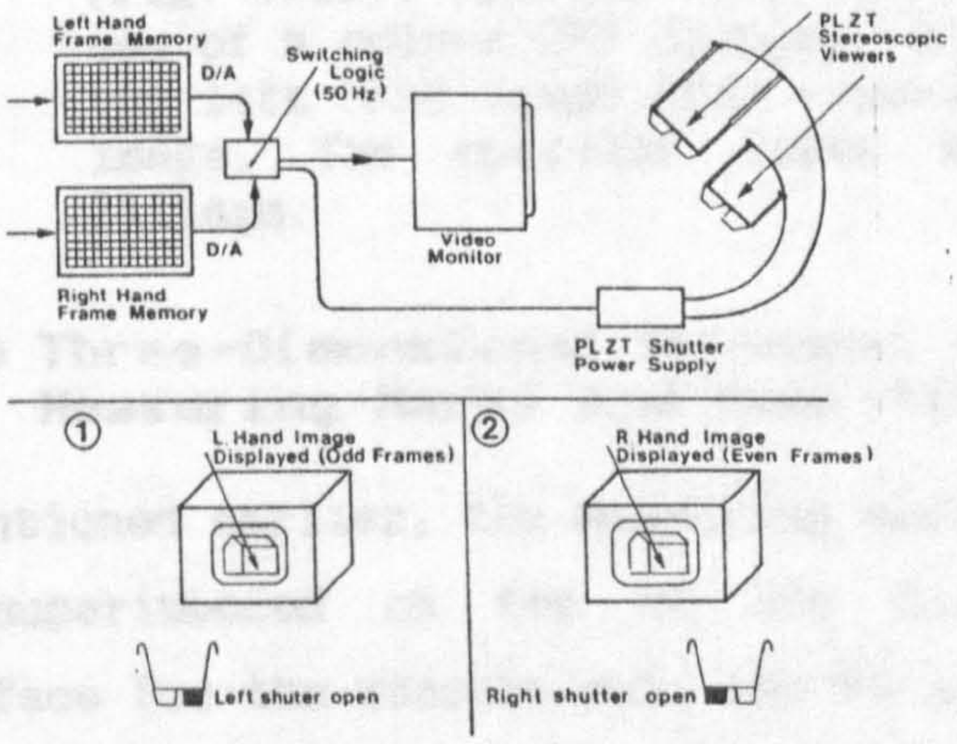
a



b

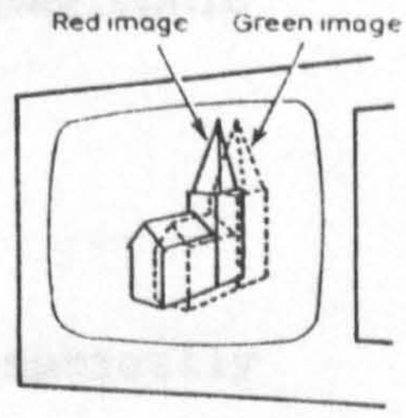


c



e

d



f

Fig. 6.4

512 lines) is about 2X5 mm^2.

(ii) Using **two screens** viewed simultaneously using a semi-reflecting mirror. The required image separation is produced by insertion of horizontally and vertically polarized sheets in front of the two screens (Fig.6.4b), the operator wearing spectacles with corresponding filters to produce the three-dimensional effect.

(iii) Using a **single screen** with a semi-reflecting mirror and polarizing filters for stereoscopic viewing (Fig. 6.4c). This requires the two component images to be displayed simultaneously on the upper and lower halves of the screen.

(iv) Using the so-called "**direction selective**" method. The two component images are displayed as alternating vertical strips of information (image data) on the video monitor tube. A lenticular screen consisting of an array of cylindrical lenses directs left and right hand images to the left and right eyes of the observer (Fig.6.4d). While this solution gets rid of the need for spectacles, it is rather restrictive in terms of observer's position and the resolution is reduced in the horizontal direction.

(v) Using **alternating shutters**. On the display screen, the two component images are displayed at the television refresh rate (50 Hz). All the odd-numbered frames display the left hand image and all the even numbered frames display the right-hand image. Viewing is achieved by using spectacles equipped with high-speed electronic shutters employing lead lanthanum zirconate titanate (PLZT) ferro-electric ceramics which can switch their state (open or closed) in one millisecond (Fig. 6.4.e).

(vi) Finally the solution which is familiar to photogrammetrists is the use of complementary (red/green) **anaglyphic images** (Fig. 6.4f). Applied to stereo television, this requires the use of a colour CRT display, in which the red gun displays the left hand image (for example) and the green, the right image. The operator views with the familiar anaglyphic filters.

6.3.3 Three-Dimensional Movement of the Measuring Marks and Real-Time Image Roaming

As mentioned earlier, the measuring marks can be generated graphically and superimposed on top of the displayed images. The operator interface for the measurements may be accomplished via a device such as a mouse, tracker ball, joy stick, etc. The three-dimensional control of the **measuring marks** may be achieved by using two cursors and employing one of the following **movement schemes**:

(i) using **stationary images** and **movable measuring marks**.
The x-parallax measurement may be achieved by the differential movements of the cursors in the x-direction. The

common x and y movements are carried out by the common shift of the cursors.

(ii) using **fixed measuring marks** and **movable images**.

(iii) a **combination** of the above mentioned approaches may be used. For example, x-parallax measurement may be carried out by the cursors and the common movements in the x and y-directions are performed by shifting the images.

Regarding the **scrolling** and roaming over the **entire model**, the following approaches may be implemented.

(i) Using automatically loading "**reserve**" **refresh memories**. Depending on the direction of movement of the images or the measuring mark, the next model area is loaded and is ready to be displayed. This function has been implemented in the I^2S image processing system with 600 software.

(ii) Using **large memories** to hold the entire model and hence the possibility of roaming over the model using the fast access to the data held on the RAM. For example, the large memory of the GEMSYS 35 system can have up to 256 Mbytes of RAM for image storage. This can easily hold two overlapping images digitized with 25 μm pixel size.

(iii) Employing **distributed array processor** technology or using **transputers** which provide both the executional speed and the fast data input/output rate (see Section 6.5 and Chapter 10).

6.4 DSP: State of The Art

The concept of DSP was initially suggested by Sarjakoski (1981). The main advantages of the system were outlined by the same author as follows:

- There are no **high precision optical-mechanical parts** in the instrument.
- There is no need for **geometrical calibration** as required for the optical and mechanical parts of analogue and analytical plotters.
- Some procedures such as **image correlation** can be automated.
- Optimization of the **image presentation** is possible using digital image processing techniques.

And some of the problems regarding the implementation of the system were also outlined by Sarjakoski:

- **Real-time differential image movement**.

- Sufficient capacity of image memories.
- Big peripheral storage requirements.

Since the first suggestion by above mentioned author, different researchers have been involved in the development of such a system. Case (1982); Petrie (1983); Albertz and Koenig (1984); Kunji (1984); Amin (1986); are some of those researchers who have put some efforts into solving the problems of implementing the DSP. More recently Gugan (Gugan and Dowman, 1986) seems to have fully implemented a DSP using powerful image processing systems (I^2S and GEMS). The characteristics of the system as described by Gugan and Dowman are as follows:

- **Stereo-viewing** accomplished by a split screen and a mirror stereoscope.
- **Fixed digitally produced marks**, which serve as the measuring marks, and moving images.
- X and Y movements in the object space are realized by **tracker ball**. Since there is no independent control for the Z movements, different approaches have been used to overcome this problem such as using a single button to switch the tracker ball to Z control and back to X and Y. Another method is to use real-time correlation for automatic height control in order to reduce the awkwardness resulting from the use of a single button switch.
- The **3-D model** is adjusted by calculating the new image positions relating to a newly defined X, Y, Z ground points. The calculations are reported to be performed in 1/15th second.
- **Sub-pixel image movement** is achieved by magnifying the image with the disadvantage of some time lapse in the I^2S system.
- Full model **area scanning** by automatically reading images from disk and displaying them when the edge of the current model area is approached.

An experimental digital image photogrammetric system, based on a commercially available image processing system, has also been set up at the Technical University of Berlin (Koenig et al,. 1988). The system is reported to be capable of handling any kind of stereoscopic imagery, such as digitized aerial photographs, scanning electron microscope data, close-range imagery, and satellite image data. However, due to the hardware limitations, the system can only handle sub-sections of image in a reasonable time (Koenig et al., 1988).

The first DSP constructed by one of the main photogrammetric system suppliers is the Kern DSP 1 introduced at the Kyoto ISPRS Congress 1988 (Cogan et al., 1988). This prototype system is a joint development with GEMS of Cambridge. The hardware design of the Kern DSP 1 (Fig. 6.5) consists of:

(i) **Processors:**

P1 - the main processor (DEC Micro VAX) running the orientation and application software.
P2 - the dedicated processor (GEMSYS 35 hardware) for the image display hardware with sufficient memory to hold the digital images.
P3 - the dedicated processor for the operator interface via footpedal and keypad input.
P7 - transputer hardware for high speed parallel processing: one of its tasks is to allow the parallel processing of image correlation algorithms. It consists of four INMOS T800 floating point transputers (see Section 10.4).

(ii) **Viewer:**

Monitor - to display two digital images side-by-side using a 1480*1024 pixel resolution screen.
Optics - to view the digital images in stereo using a split screen technique with some possibilities for squint and image rotation correction.

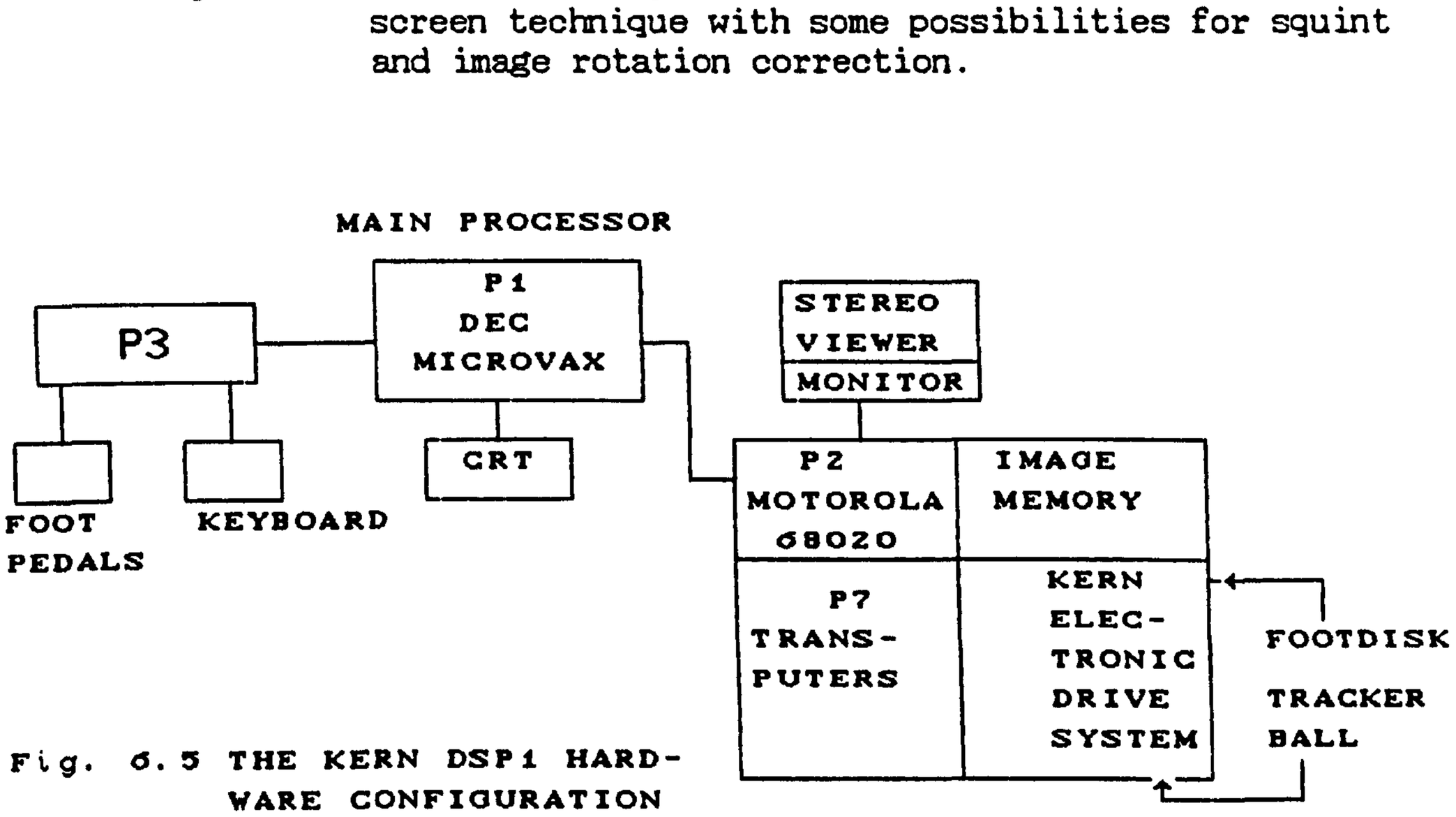


Fig. 6.5 THE KERN DSP1 HARDWARE CONFIGURATION

The DSP 1 software has been selected and modified from the standard Kern DSR analytical plotter software with some additional image processing routines for contrast stretching, edge enhancement, etc. The overall structure of the software is as follows:

- **Model orientation and application programs** which run on the P1 processor.
- **Real-time geometrical computations and image handling routines** to control the digital image displays (mono/stereo, zoom), the image movement (panning, speed control) and processing routines to enhance the digital image running on the P2 processor.
- Real-time execution of the **image correlation programs** is handled by the P7 transputer-based accelerator.

The system overview may be described as follows:

The executable program to control the movement of the digital images according to their image geometry, is sent from the main processor, P1, to the image processor, P2. The digital images which are stored on disk or tape are sent to the on-line, direct access image store comprising a block of random access memory. Fixed digital symbols are generated and overlaid on the images to serve as the measuring marks. The operator has the option of performing the height measurement manually, automatically or using a combination of both these modes.

Another system which has been developed very recently at the University of Hannover in conjunction with the Context Vision of Sweden (Lohmann et al., 1989) is a digital photogrammetric stereo-workstation based on the commercially available GOP 302 image processing system from Context Vision. The system includes a stereo display and measurement and orientation programs as well as a package for digital correlation and orthophoto production.

It is worth mentioning that other photogrammetric systems suppliers have also shown some tendency towards implementing the design of a DSP. In this context, one can mention the Matra Traster T1ON Digital Stereoplotter which seems to be one of the future generation of DSP systems. However, in summary, one can say that, at present, the DSP concept is only in its elementary stages of development with several prototype systems having been constructed and many experiments under way to develop more practical systems.

6.5 The Future Developments of the DSP

The long-term future of photogrammetric mapping and modelling systems

appears to lie with all-digital solution of the types discussed above. This is, as mentioned before, mainly due on the one hand to the advances in computer technology and the continuous reduction in the price of computers, and the availability of directly acquired digital data (e.g. from SPOT) on the other hand. Therefore, further research work needs to be carried out to ensure still more rapid execution of the huge computational tasks inherent in the all-digital photogrammetric solution. In this context, the new advances in **Distributed Array Processor (DAP)** technology appear to offer a breakthrough in the execution speed and the real-time display of the results of many application areas having huge input/output data, such as simulation and modelling, signal and image processing, etc. (Hunt, 1989). The **transputer array** technology also seems to be very promising in the fields of image processing and correlation (Muller et al., 1988) and, as already discussed above, has already been exploited to a limited extent in the Kern DSP 1 prototype. These technologies will be considered in greater detail in Chapter 10.

In the next sections, the design and implementation of the DSP in this research work is presented.

6.6 Implementation of a Digital Stereo-Photogrammetric System on The IBM 3090 Computer

The main intention behind implementing a DSP in this project was to investigate the feasibility of having a purely digital solution for generating a DTM using a standard multi-user mainframe machine without utilizing any purpose-built hardware and software. Therefore, the system should not be compared with those DSP systems which use purpose designed hardware and software and have been developed by a large team over a much longer period of time. The main characteristics of the DSP as designed in this project is as follows:

(i) The **y-parallaxes** are not removed differentially as in typical analytical and the digital stereo-plotters mentioned above. Since the digitally rectified images generated in the previous stage (see Chapter 5) were used, no y-parallax was present on the stereo-model. Regarding this feature, the system can be considered rather as a digital equivalent of the analogue plotters.

(ii) **Stereo-viewing** of the overlapping digital images has been

achieved using overlapping images on a colour display in conjunction with anaglyphic filters.

(iii) The approach of having **stationary images** and **movable measuring marks** has been adopted.

(iv) The **image space primary scheme** has been employed. Although, the object space primary approach was desirable, its implementation made the system extremely slow. This is because, in the object space primary method, for any change in the X, Y and Z object coordinates, a corresponding increment in the image space needs to be calculated. Either the images or the cursors must then be shifted with an amount equal to the calculated increments. Due to the hardware limitations of the 5080 graphics system used in this project, the cursor could not be shifted automatically, nor could the image shifting be performed by the 5080 processor. As a result, the data would have to be sent to the host multi-user machine for processing which makes the operation inefficient. In view of these various considerations, the image space primary scheme was chosen since it avoids most of these difficulties.

(v) Due to the hardware and software limitations of the system, especially regarding the amount of digital data required for the entire overlapping left and right images, **scrolling** and **roaming** over the model area could not be achieved efficiently. Therefore, the images were divided into smaller patches and each patch was given a code number so that each required individual left and corresponding right patches could be loaded into the display memory.

The following sections deal with the hardware and software configurations, and the design and implementation of the DSP system respectively.

6.6.1 Hardware Configuration

The Digital Stereo-Photogrammetric System as implemented in this project comprises the following four hardware components:

(1) A **computer** on which the processing is carried out. As already mentioned, all investigations during the project were carried out on the IBM 3090 Model 150E mainframe computer equipped with the Vector Facility.

(2) A **display unit** for image display. The display unit utilized in this project is the high resolution IBM 5080 colour display system (Fig. 6.6). The IBM 5080 display system is directly connected to the IBM 3090. It can operate in terminal mode (to talk to the mainframe) and in graphics mode simultaneously. The 5080 graphics system comprises two hardware components, namely a display processor (display controller) and a display device. The controller receives the data from the host and stores it on its internal memory for subsequent feed to the 5080 screen for display. The

controller can also carry out some operations on the vector data locally using its own built-in processor. However, it does not perform computations locally on the image data and hence any processing on the image data is performed by the host computer. The 5080 display is a non-interlaced device with a 19-inch monitor equipped with an anti-glare screen (1024*1024 resolution) and hence offers flicker-free viewing of high quality images for long hours of continuous use with minimum fatigue. Indeed, the flicker-free, high image quality and the high-speed image display capabilities which are offered by the 5080 graphics system made the stereo-viewing of the images quite satisfactory.

Fig. 6.6 IBM 5080 COLOUR GRAPHICS SYSTEM

The 5080 graphics system has three electron guns generating three primary colours for colour display. It has a total of 8 bit planes for both monochrome and colour display. Therefore it can produce 256 grey shades for monochrome and the same range of colours for the colour images. Hence, the system cannot utilize the full 256 grey levels for each primary colour. More detail considering this latter factor is given in Chapter 7.

(3) **An operator interface** which consists of a keyboard and a cursor control pad. This allows the operator to interface with the system interactively for image coordinate measurements.

(4) A **plotter** to generate the contour lines from the DTM data. This is the plotter connected to the ICL 3980 mainframe computer available in the University. Altough the Winsom graphics package was available on the IBM 3090, due to the familiarity of the author with the use of the PANACEA DTM

package (see Section 6.7) which was already installed on the ICL machine, the generated DTM data were transferred to the ICL.

The overall configuration of the hardware components of the DSP is presented in Fig. 6.7.

6.6.2 Software Configuration

To implement the DSP on the hardware configuration described above, the **IAX general purpose image processing package** was used. The availability and flexibility of IAX allowed the different photogrammetric tasks to be programmed and performed much more easily than otherwise. As mentioned above, the PANACEA contouring package was also used to generate the contour lines from the measured DTMs.

A brief introduction to the IAX image processing system is given in the following sections.

6.6.3 IAX Image Processing System

IBM's, IAX (Image Application eXecutor) is a fully featured, interactive image processing system. The system is designed to operate under the IBM Virtual Machine /System Product Conversational Monitor System (VM/SP CMS). It can be used with terminals which support image display with the Graphical Data Display Manager (GDDM). It can also be used with the IBM 5080 Graphics Display System. The latter was used in the present project as explained in the preceding sections.

IAX provides a wide range of image processing routines. These include, for example, Fast Fourier Transform, statistical calculations on the image, histogram manipulations, convolution operations, resampling by nearest neighbour and bilinear interpolation, image rotation and translation, Laplacian and Sobel operators, arithmetic (sin, log, abs, phase), etc. However, those operations which are not part of the basic system can be programmed and added to the system and can be used in the same way as those which are standard components. This feature of IAX makes it flexible so that different operations may be programmed and performed. The extensions to IAX can be written in the System

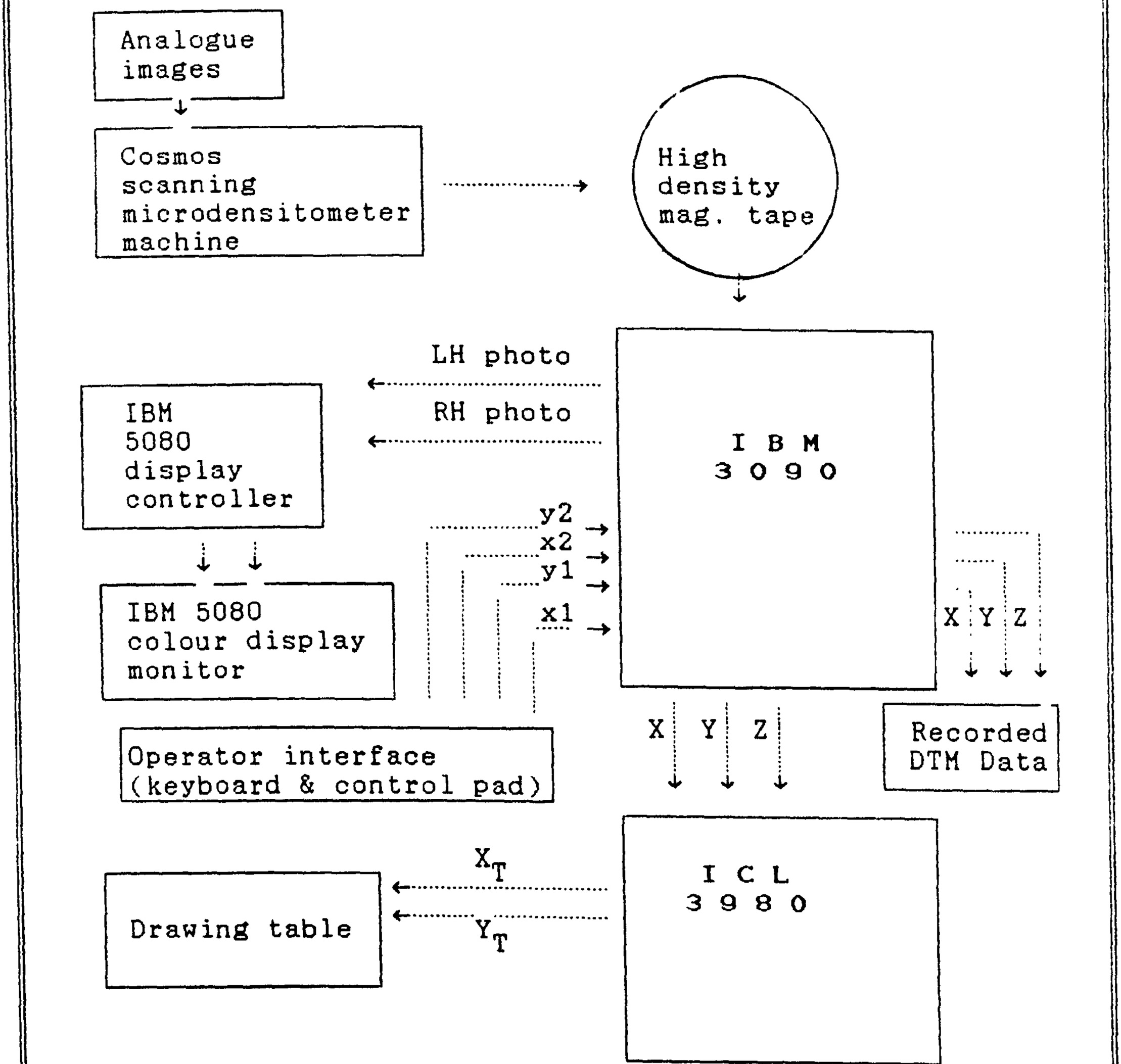


Fig. 6.7 The overall hardware configuration of the DSP

Product Interpreter (REXX) language or in PL/I. The former was used in the present project and a brief description of it will be given in Section 6.6.4.

6.6.3.1 Interacting With IAX

Interacting with IAX takes place by means of its special image processing language. The language allows the use of IAX routines, which are called as functions in a similar manner to that used to call the maths functions available in FORTRAN and PASCAL. The functions may be called by typing the function's name directly at the terminal. The arithmetic operators and variables can be used in the same manner as in other high level languages, for example:

C = A + B causes the value of A to be added to the value of B and the result stored in C. In IAX, the values of A or B could be either a single floating point number or a complete picture. Thus the value of A which is the entire image or a portion of a picture can be added to the values of another picture B on a one-by-one basis. Therefore it is possible to perform different arithmetic operations on the images by regarding each image as a single variable. A simple program in IAX language, which is directly and interactively typed at the terminal, can be written as follows:

```
F = FILT
A = IMAG
A = A * 0.5
D = CONV(A, F)
DISP D
```

In the above program, FILT can be either a variable already loaded on the display memory, or the name of a file whose contents are stored on the host disk. In the latter case, the file type will be IAXn, where n takes the values of 1 to 4 depending on the data type (see below). The first statement of the program reads the contents of the variable FILT or the file 'FILT IAXn' into the variable F --the file type need not be specified. The file "FILT" can be a one- or a two-dimensional matrix of the size n by m. In a similar way, the contents of the variable IMAG or the file IMAG IAXn are read into the variable A. Each element of variable A is then multiplied by 0.5 and the result is

stored in the same variable. The variables A and F are then convolved using the IAX routine "CONV" and finally the IAX command "DISP" causes the image to be displayed on the screen.

Commands are those statements which behave like a function which returns no result, for example IAX DISP and PUT commands which are used for displaying the data on the screen and storing the data on the disk respectively, take parameters and, in this respect, are identical to functions but they do not return any result.

IAX variables can have any of the following data types:

IAX1	8 bits per pixel unsigned integer
IAX2	16 bits per pixel signed integer
IAX3	32 bits per pixel floating point number
IAX4	64 bits per pixel floating point complex

where IAX1 through IAX4 are the file types.

In addition to the direct interactive use described above, IAX statements can be combined into the so-called IAX "macros". These, as mentioned in the previous section, are System Product Interpreter (REXX) programs. They can be used to extend the IAX commands and built-in functions tailored to specific needs. Any newly developed macro of file type IAXFN may be called as a function; any of type IAXCMD may be called as a command.

Since the REXX language was used in this project to implement the DSP, a brief introduction to the language is given in the next section.

6.6.4 REXX Programming Language

REXX (Restructured EXtended eXecutor) is a powerful high level language. REXX programs consist of a combination of REXX instructions such as DO LOOPS; IF statements, etc.; and CP (Control Program) and CMS commands. Parameters may be passed to and returned from a REXX program. It is possible for a REXX program to call itself recursively. Some of the REXX programming language characteristics are given below:

- The first line of a REXX program must be a comment. This is used to enable the IBM CMS to distinguish a REXX program. A comment can consist solely of /* */, but obviously a brief

description of the program would be more meaningful.

- REXX allows more than one **instruction** on a line or a single instruction may occupy more than one line. Instructions may be separated by a semi-colon.

- There is no explicit way to declare **variables** or **arrays**, so these are generated as they are required. Data assigned to a variable may be:

  (a) numeric - real, or integer;
  (b) Boolean;
  (c) string.

- Numeric/arithmetic operators are as follows:

+	add
-	subtract
*	multiply
/	divide
%	divide and return the integer part of the result
//	divide and return the remainder
**	raise a number to a whole-number power

- An **IF statement** takes the following format:

```
IF expr [;] THEN instruction
[ELSE [;] instruction]
where [] denotes an optional statement.
```

- A DO LOOP can take the following forms:

(a)
```
IF A=5 then DO
. . .
. . .
END
```

(b)
```
DO n/FOREVER [UNTIL expr]/[WHILE expr]
. . .
. . .
END
```

where n is the number of times the loop is to be performed; UNTIL/WHILE are the conditions under which the LOOP should be terminated.

(c)
```
DO I = 3 to -2 by -1
DO J = 2 to 10 by 2
. . .
END J
. . .
END I
```

- To read data from the terminal into a REXX program, the **PULL** instruction is used and to display a text string on the terminal, the **SAY** instruction is used. These are equivalent to the FORTRAN read and print statements respectively.

An example of the REXX programming language as implemented in this

thesis is given in Section 6.6.7.

6.6.5 The Design and Implementation of the System

As in other photogrammetric plotters, the DSP system was designed to work in three different phases, namely:

- **model formation** which is performed by the inner and relative orientations followed by the digital rectification.
- sampling and measuring the positions and elevations of the model points for the **generation of a DTM.**
- **absolute orientation** to transform the model points to the terrain coordinate system.

A set of programs were developed in FORTRAN77 and REXX to carry out different phases of the DSP procedures. The **FORTRAN programs** perform the automatic fiducial mark measurement and coordinate transformation for the inner orientation; relative orientation; digital rectification; etc. The mathematical basis of these operations were covered in Chapter 5.

The **REXX programs** perform those operations which directly use IAX functions and commands, for example, image loading and displaying; image point measurement; etc. The REXX programs were linked together using the calling routines facilities available in REXX. However, although a new version of IAX allows a FORTRAN program in Vector or Scalar modes to be called from within IAX, a bridging PL/I routine which causes such a task to take place was not available during the time that this project was carried out.

The flow diagram of the procedures involved in the DSP is presented in Fig. 6.8. These procedures will be described in the sections which follow.

6.6.6 Model Formation Phase

(i) Inner orientation

As mentioned in Chapter 5, inner orientation procedures start with the measurements of the fiducial marks. The measurements of the fiducial marks were carried out using two options of

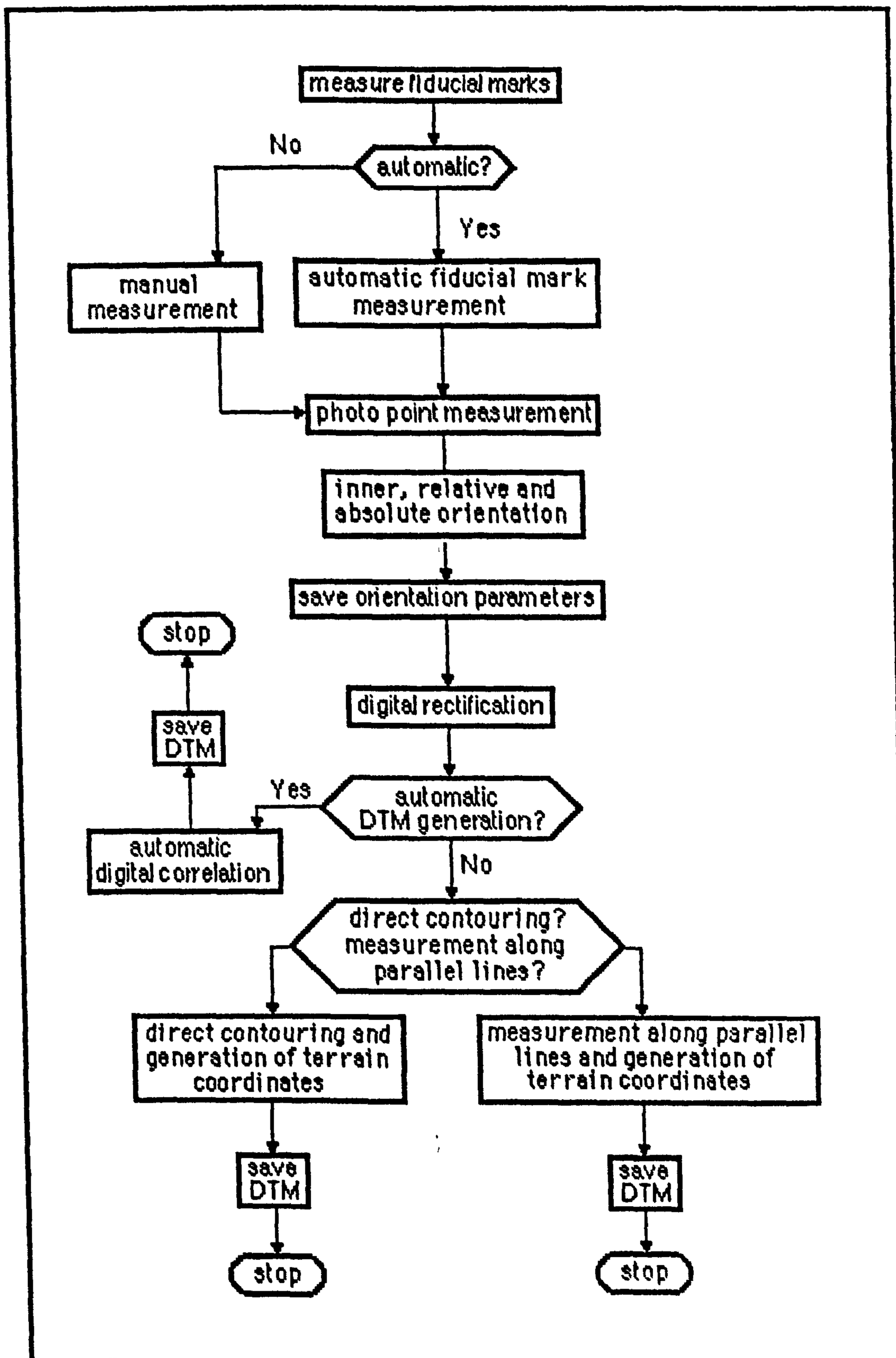


Fig. 6.8 The overall scheme of the working procedures of the DSP.

automatic and manual measurements. The automatic fiducial mark measurements and coordinate transformation for inner orientation are described in Chapters 5 and 9. In this chapter, the manual pointing to the fiducial marks are described. The subroutine "FIDPT" written in REXX was developed for manual measurement of the fiducial marks using a single cursor. The program consists of two sections. In the first section, the file containing the fiducial mark is read from the storage disks of the host computer and is displayed on the screen. The cursor function available in the IAX is called to measure the centre of the fiducial mark. The number of observations is specified by the user. The mean and standard deviation values for the measurements are then calculated and displayed.

The second part of the program was developed with a view to improving the fiducial mark measurement by enlarging the size of the fiducial marks. The enlargement is performed by constructing a window around the centre point which has been measured by the first stage. The window is then enlarged to a specified size. The measurement is carried out several times and the mean and standard deviation values are again calculated. The flow diagram of the program FIDPT is given in Fig. 6.9. The program listing is given in the Appendix A4.

The measured coordinates of the fiducial marks are fed to the program "TRANS" written in FORTRAN77 to calculate the transformation coefficients. These coefficients are used in the later stages to transform the measured points in the microdensitometer coordinate system to the principal point coordinate system and vice-versa.

(ii) Relative orientation

The actual procedures involved in the analytical relative orientation and the image resampling have been described in Chapter 5. In this section, the measurement of the image points, which is the first stage of the relative orientation and has not been covered in Chapter 5, is described:

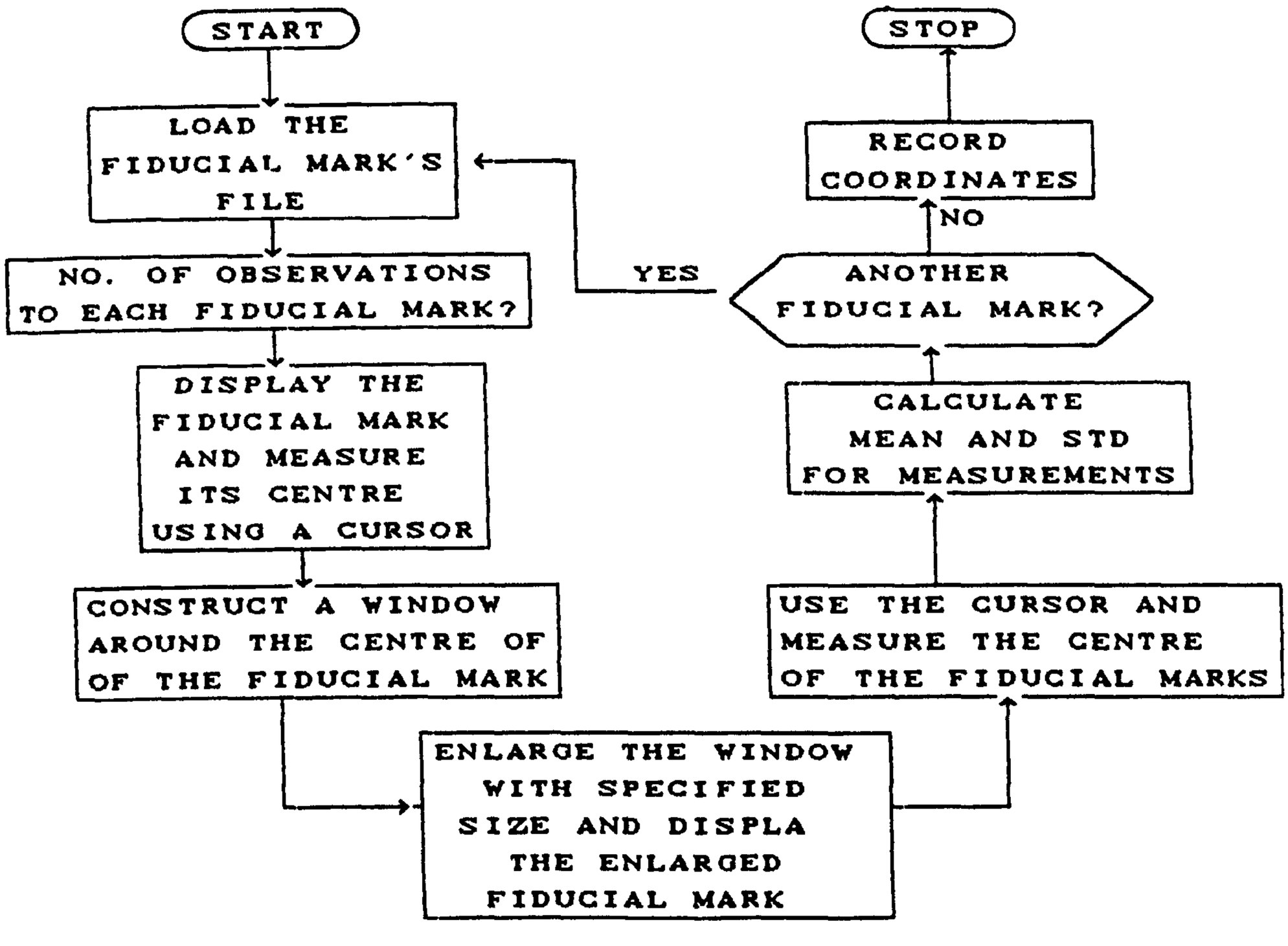


Fig. 6.9 FLOW DIAGRAM OF THE PROGRAM "FIDPT"

Program "IMPT" written in REXX was developed to perform the measurements in the six standard positions in the left and right images. The program comprises two phases. In the first phase of the program, the left and right sub-images containing the image points are loaded from the host computer's disk to the 5080 buffers. The selected sub-images are shifted so that the left and right image points to be measured have zero y-parallax. The left and right sub-images are then displayed superimposed in green and red for stereo-viewing (-the implementation of the three-dimensional viewing will be described in the following sections). The image points can then be measured stereoscopically using two cursors. The number of image points and the number of measurements of each image point are specified by the user. The program calculates the mean and standard deviation values for each group of measurements. This approach (i.e. stereoscopic image pointing) which is ,in

principle, similar to image point measurement by a stereocomparator, was found to be a time consuming method and difficult to arrange. The reason was the fact that these measurements are carried out on the original non-rectified images, and therefore, to view the two image patches stereoscopically, the y-parallax should be removed differentially (i.e. similar to a stereocomparator). This image shifting, due to the hardware limitation of the system were very inefficient.

In view of these difficulties, the second part of the program was developed to allow monoscopic measurement. In the second section of the program, the left and right images are displayed side by side and the operator uses a single cursor to measure the positions of the corresponding left and right image points. A window is then constructed around any measured point and the window is enlarged with the enlargement factor of about 3. This enlargement factor has been found to be the most suitable value in practice, since enlargements greater than this obscure the clarity of the details. Finally the mean and standard deviation values are again calculated and displayed for each set of measurements. The flow diagram of the program IMPT is given in Fig. 6.10. The program listing is given in Appendix A5.

The measured image points are fed to the program TRANS to be transformed to the projection centre coordinate system, using the coefficients calculated in the inner orientation phase. The transformed points referred to the projection centre are fed to the relative orientation program "COPLAN" (see Chapter 5 for more details) by which the relative orientation is carried out and the elements of the exterior orientation for the right hand camera with respect to the left one are calculated and recorded.

(iii) Digital rectification

As mentioned in Chapter 5, the program "DIGREC" has been developed to perform the digital rectification. The program

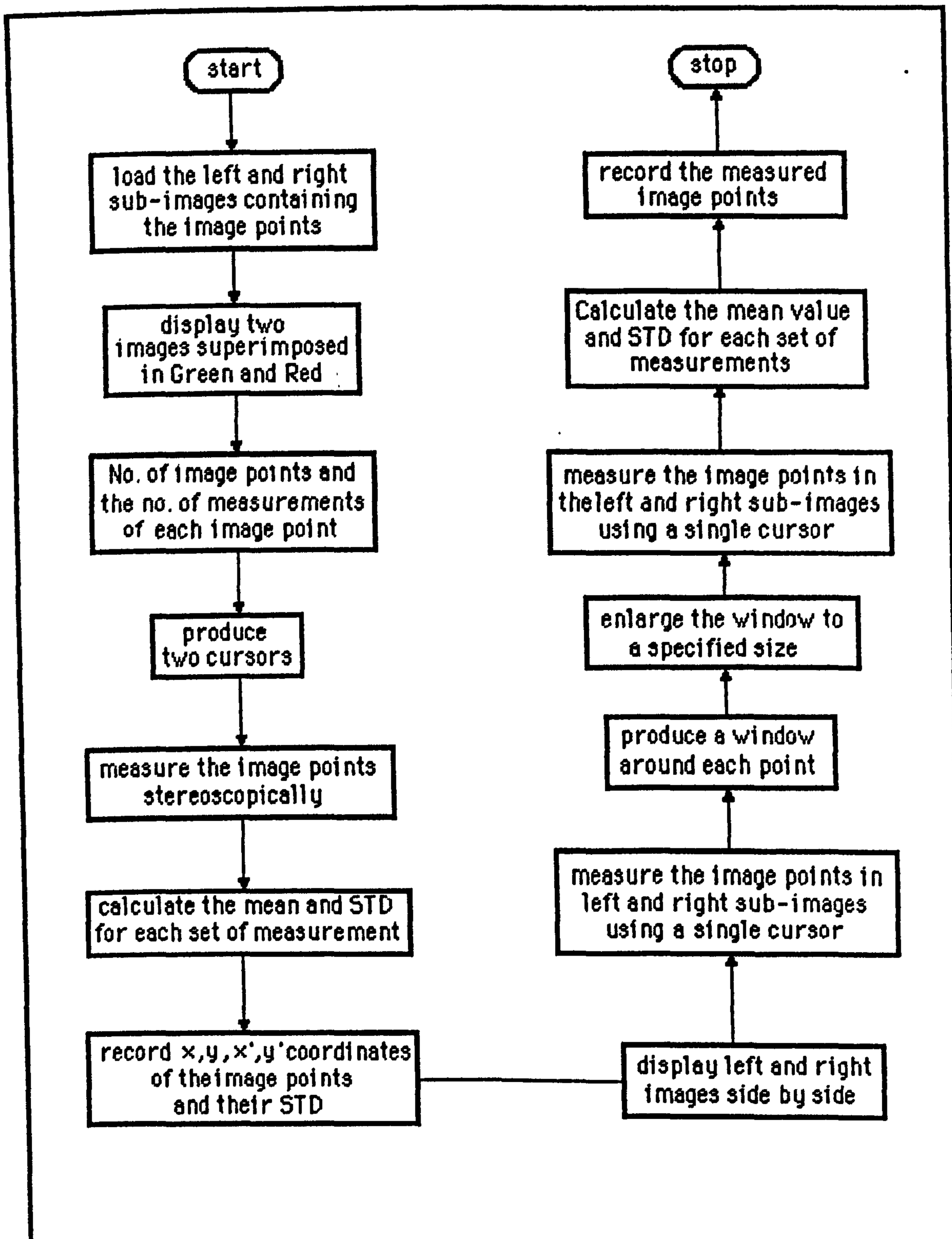


Fig. 6.10 Flow diagram of the program "IMPT" (image point measurment program).

uses the camera parameters calculated in the previous stage. Details about the rectification phase have already been given in Chapter 5.

(iv) Stereo-viewing of the model

Since the left and right rectified images are free from y-parallaxes, they satisfy the requirement for stereo-viewing. For stereo-viewing of the left and right images, two methods were used -those of a split screen with a mirror stereoscope and of complementary anaglyphic images in conjunction with suitable red/green spectacles. The latter was found to give a more satisfactory result. The anaglyphic approach enables the observer to have a reasonably large area of the model available to be viewed instantly. It also produced a very good three-dimensional viewing perception and allowed several observers to view the stereo-model simultaneously. Due to the excellent performance of the anaglyphic approach, there was no hesitation in using it predominantly for stereo-viewing throughout the project. The anaglyphic method was implemented as follows:
The IAX routine SRGB was used to produce Green and Red superimposition of the left and right images. The routine superimposes three equal-sized arrays and activates the red, green and blue electron guns for the first, second and third arrays respectively. To suppress the effect of the blue electron gun, an array with the same size as the left and right images was constructed but with zero values and superimposed on the left and right images. The best three-dimensional perception was achieved (for the test images) when the images were separated with an amount equal to:

{[MAX x-parallax]-[MIN x-parallax]} / 2

Fig. 6.11a shows the stereo-model for the test area displayed on the display screen. The image is taken by a 35mm camera from the screen. Fig. 6.11b shows the same image produced by the IBM electro-erosion printer.

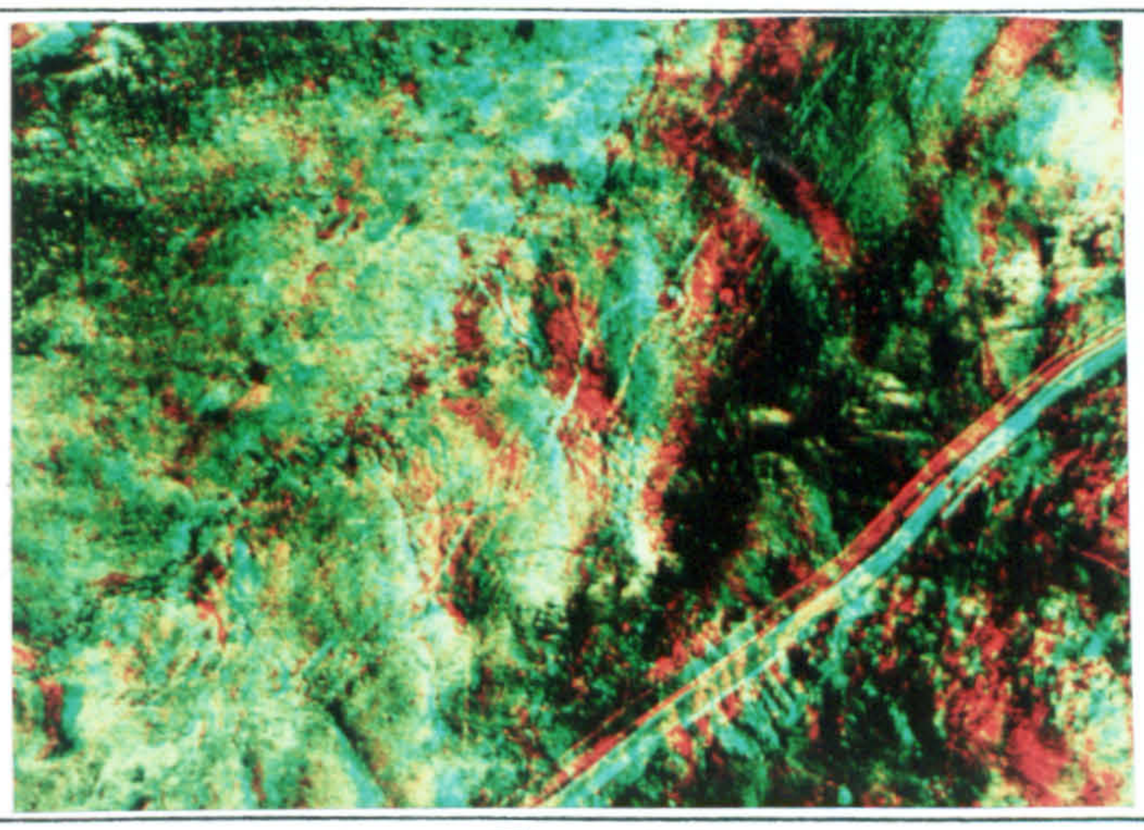


Fig. 6.11a

6.6.7 Model Point Sampling Phase

(i) Generating two cursors

Having formed the three-dimensional viewing of the model, it is now possible to sample the model points to measure their elevation values and so generate a DTM. To accomplish this, two cursors are required. However, the 5080 graphics system can only generate one single cursor. The movement of this cursor is controlled via a control pad which is similar to that of a tablet digitizer. The possibility of generating a second cursor to be controlled via the same control pad was found to be impractical because of the hardware limitations of the 5080 graphics system (personal communication with the IBM representative Mr. Rick Turner). It was finally found that a second cursor could be generated by using the IAX graphics command "GTEXT". This command takes the following form,

GTEXT(<text> <x> <y>)

The GTEXT command adds the text in the <text> parameter at the coordinate position x,y on the screen. The size and colour of the text may be controlled via another command called "THEIGHT and TCOLOUR". It is, therefore, possible to read the symbol "+" as a text value and draw it at any position on the screen. By controlling the size of this symbol, a second cursor similar to

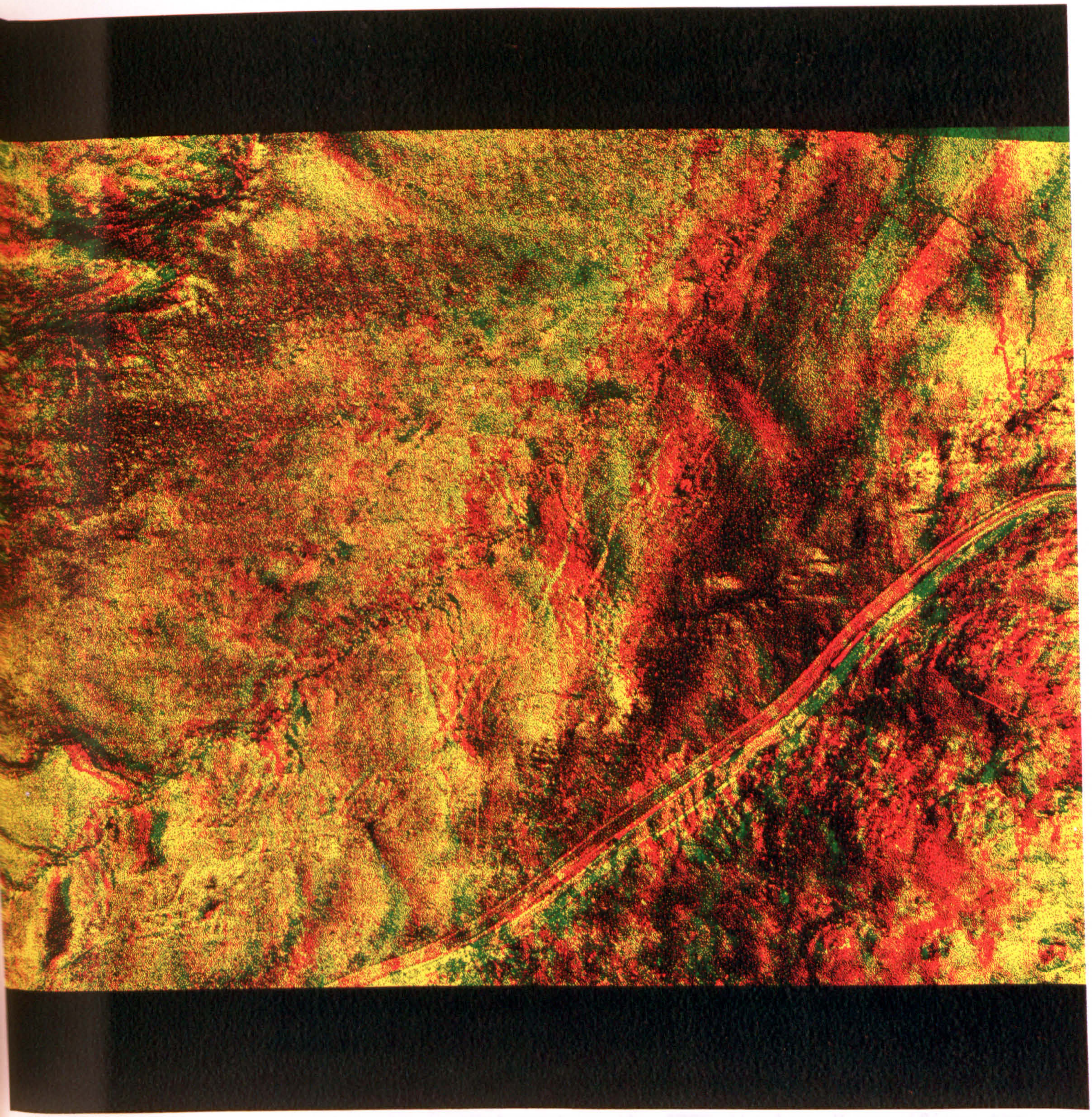


Fig. 6.11b

the first one was generated. However, this cursor could still not be controlled via the control pad. This problem was eventually overcome using the following scheme.

Suppose the second cursor is required to be shifted to a specific position. To do this, the first cursor is transferred to the same position via the control pad and its position is input to the computer. The x and y coordinates giving the position of the cursor are then read into two variables using the IAX function and the REXX statement "TRANSFER CURSOR" and "PULL" respectively. In turn, the IAX command "SCREEN MOVE" drives the second cursor to the position of the first cursor. These procedures are realized by the following macro developed in REXX,

```
/* a routine to transfer the symbol "+" to */
/* any desired position on the display screen */
/* using the cursor control pad */
/*-------------------------------*/
'SCREEN DEF 2 0 100 0 100'
'GTEXT(+ 50 50 IN 2)'
DO FOREVER
     'TRANSFER CURSOR'
         PULL XR YR BUTTON
          IF XR>800 THEN LEAVE
        'FF='XR'-68';'HH='YR'-23'
   'SCREEN MOVE 2 TO 'FF' 'HH''
  END
```

The routine generates a cursor and the symbol "+" and moves the symbol to any position on the screen defined by the first cursor. The process will be terminated if the operator shifts the cursor to any position on the screen which has an x value greater than 800 (i.e. any position on the extreme right side of the screen) and inputs the coordinates of the cursor.

The following IAX commands and functions are used in the above subroutine:

The command "SCREEN DEF N XMIN XMAX YMIN YMAX" is used to set the logical screen layout on the 5080 display. N specifies the screen number; and the values XMIN,YMIN,XMAX,YMAX define the coordinates of the top left and the bottom right of the screen

respectively. The IAX function "TRANSFER CURSOR" generates an integer vector of length 3. The elements of this vector are:

element 0 cursor X coordinate
element 1 cursor Y coordinate
element 2 cursor button number

This vector is generated via one of the buttons of the control pad. The values of 68 and 23 are constant shifts determined empirically for the precise positioning of the second cursor. The REXX command LEAVE terminates the LOOP.

To sample the model points for generating a DTM, the two alternative approaches of **measurement along parallel lines in image space** and **direct contouring** have been implemented. Each will be described in the following sections.

(i) Sampling the model points along parallel lines in image space

The program "PROF" written in REXX was developed to sample the model points along parallel lines and generate the X, Y and Z model coordinates for the measured points. As mentioned earlier, the image space primary scheme has been adopted in this project. That is, the operator is measuring and generating the x, y, x', y' corresponding image points. The object coordinates are determined by space intersection. The program stages are as follows:

The first stage of the program involves loading the left and right sub-images with user specified sub-image code and generating two cursors as described above. The left cursor is then shifted systematically in the y-direction. The shift interval is determined by the user. The operator controls the movement of the right cursor via the control pad. By shifting the right cursor in the x-direction, at each position of the left cursor, it is possible to put the measuring marks (i.e. two cursors) in contact with the ground. After the completion of each y-scan, the left cursor is shifted one step (specified by the user) to the right hand side and another line is measured. In this way, the whole of the determined area is scanned. The flow diagram of the program PROF is given in Fig.

6.12. The program listing is presented in the Appendix A6.

During the measurement phase, two problems arose. The first one was related to the cursor movement which was rather slow and, in particular, during those times when work loads on the IBM 3090 were very high, the measurement operation became extremely slow. The second problem was related to the stereo-viewing of the model and the simultaneous three-dimensional movement of the cursors. Since the two cursors are viewed by both eyes simultaneously, when the model is viewed stereoscopically, the two cursors appear as if they are four. This was, of course, very confusing for the eyes in deciding which two cursors should be fused. After some practice, this problem was solved and it became easy two fuse the corresponding left and right cursors.

(ii) Model points sampling along contour lines

Since the measurement along parallel lines using two cursors, as described in the preceding paragraphs, is rather slow due to the awkwardness of shifting the two cursors, it was decided to search for an alternative solution which allows the contour tracing be performed by a single cursor. To achieve this, an effort was made to simulate the method of contour tracing used in optical projection stereoplotters (e.g. the Multiplex). In these optical projection stereoplotters, a single measuring mark is used. For contouring, the operator sets the measuring mark at a specific height by shifting the measuring mark of the tracing stand in the vertical direction.

To trace the contour lines at that specific height, the operator then moves the tracing stand in different directions in order to keep the measuring mark in contact with the ground. In our case, however, the measuring mark is superimposed on top of the image and when the image is viewed stereoscopically, the cursor (N.B. a single cursor) appears on the surface of the screen and cannot be moved in the vertical dimension. This problem was overcome by a method which can be called the "zero x-separation" method. The method is based on the following well

known principle:

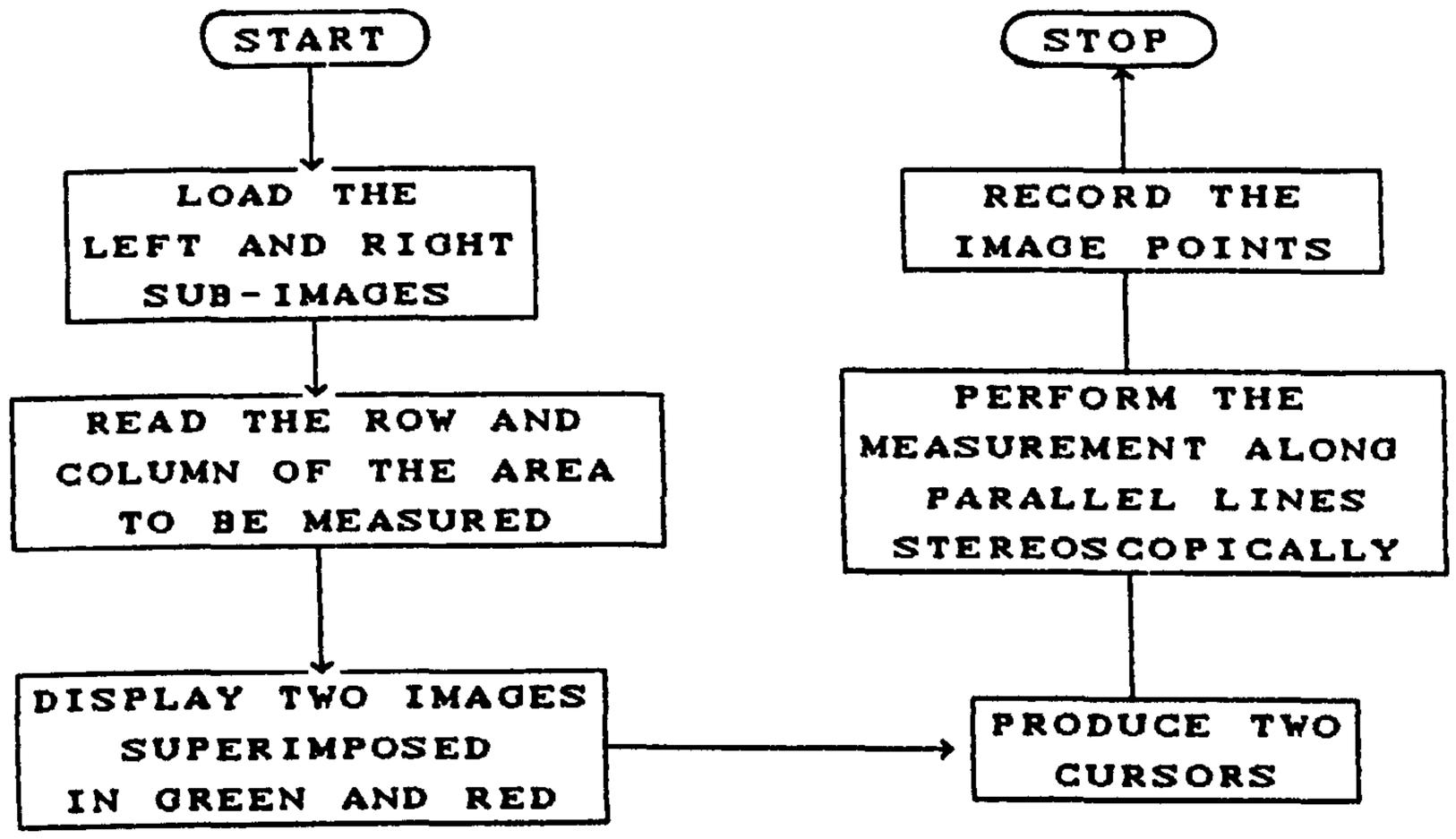


Fig. 6.12 FLOW DIAGRAM OF THE PROGRAM "PROF"

The x-separations of the points on a stereo-image displayed on a CRT screen may be divided into three elevation zones as shown in Fig. 6.13.

It can be shown that, for any point appearing on the zones (A), (B) and the screen, the x-separations between the left and corresponding right image points are,

$$\begin{cases} xl' - xl < 0 \\ xk' - xk > 0 \\ xm' - xm = 0 \end{cases}$$

where xl', xl, xk', xk, xm' and xm are the x-coordinates of the left and the corresponding right image points appearing in the zones of (A), (B) and the screen respectively (Fig. 6.13). Therefore, the superimposed single cursor on the screen corresponds virtually to two cursors with zero x-separation. In other words, a single cursor may be used to trace all points having zero x-separation (i.e., the points which appear on the screen). As a result, to trace any specific contour line, it is enough to shift the left and right images so that for the points having the required height value, the x-separation is reduced to zero and hence appears on the display screen.

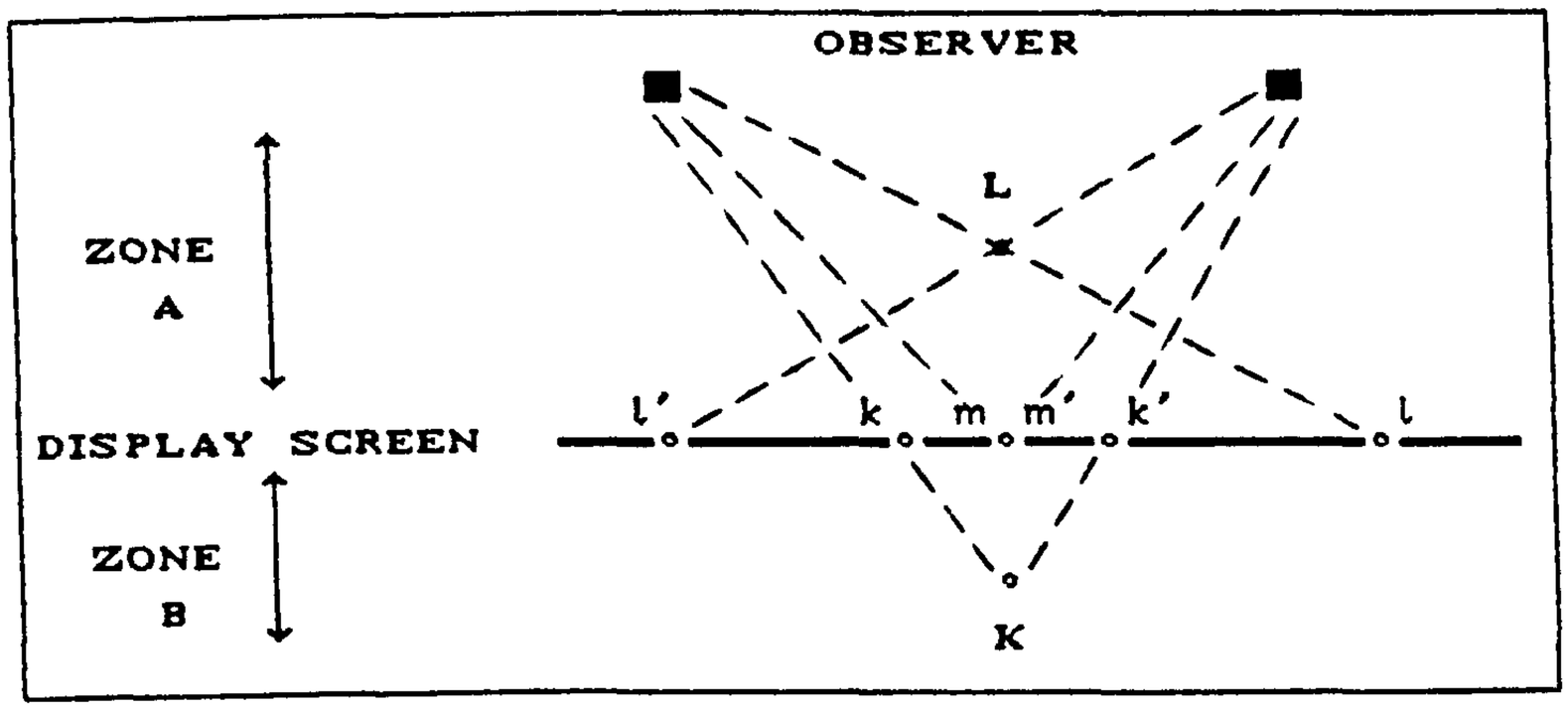


Fig. 6.13

The above approach was devised and implemented in this project. The program "CONT" in REXX was developed for this task. The different stages of the program are as follows:

The first stage involves, as before, the loading of the desired sub-images and the display of both images using the anaglyphic method.

To trace any specific contour line, e.g. that for the height h, the operator searches for any point on the model having the required height value. This is achieved by using two digitally generated cursors as before. As the operator moves from one point to the other in searching for the required height, the height values of the points are calculated and displayed on the screen successively so that the operator can read the height of the points in real time. This enables him to find a point having the height value of h. As soon as points with the required height value are found --for example, points a and a' -- their coordinates are recorded and the program will be transferred to the direct contouring mode. The shift value of the left and right images is then calculated so that the x-separation of the points a and a' is reduced to zero. Clearly, as discussed earlier, this also reduces the x-separation of all points having the same x-parallax, to zero.

The next stage involves the contour tracing by a single cursor which is controlled via the control pad. As before, the left and right image points are measured by the operator and the model coordinates of the points are calculated in real time. After the contour tracing is terminated for the first specified height, the operator shifts the cursor to any position at the extreme right corner of the screen and presses the cursor's button which signals the program to go back to the first stage of the height searching mode. The operator then repeats all procedures mentioned above for the next contour line and subsequently for all desired contours.

Regarding the generated model points using the above mentioned approach, it should mentioned that, if the absolute orientation has not been carried out prior to the measurement stage, the word "contouring" cannot be applied in its real sense and the measured data are merely the points with same "model height" values.

The flow diagram of the program "CONT" is given in Fig. 6.14. The program listing is presented in Appendix A7.

6.6.8 Absolute Orientation Phase

The absolute orientation was carried out on the model points generated by the DSP. This is due to the fact that the digital images were only rectified to remove y-parallaxes and associated x-parallaxes (see Chapter 5). The absolute orientation was performed using a three-dimensional conformal transformation. The absolute orientation could also be applied prior to the point measurement, i.e., during the digital rectification stage, by introducing the elements of the absolute orientation to the transformation equations. However, the former approach requires a smaller number of computations, since the transformation is carried out only for the generated model points.

6.7 Test Results

A sub-image was selected on the left and right digitized aerial photos

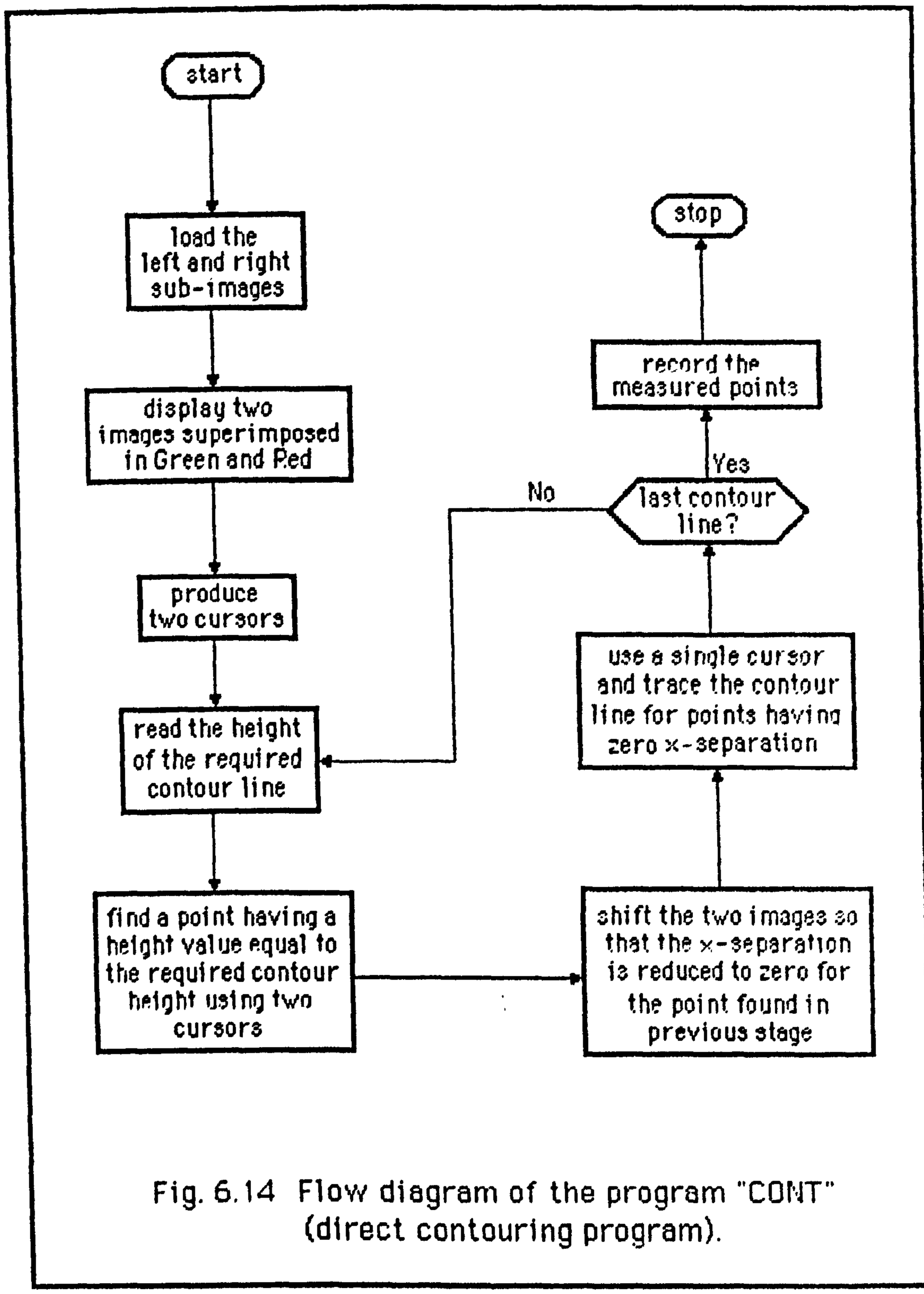


Fig. 6.14 Flow diagram of the program "CONT" (direct contouring program).

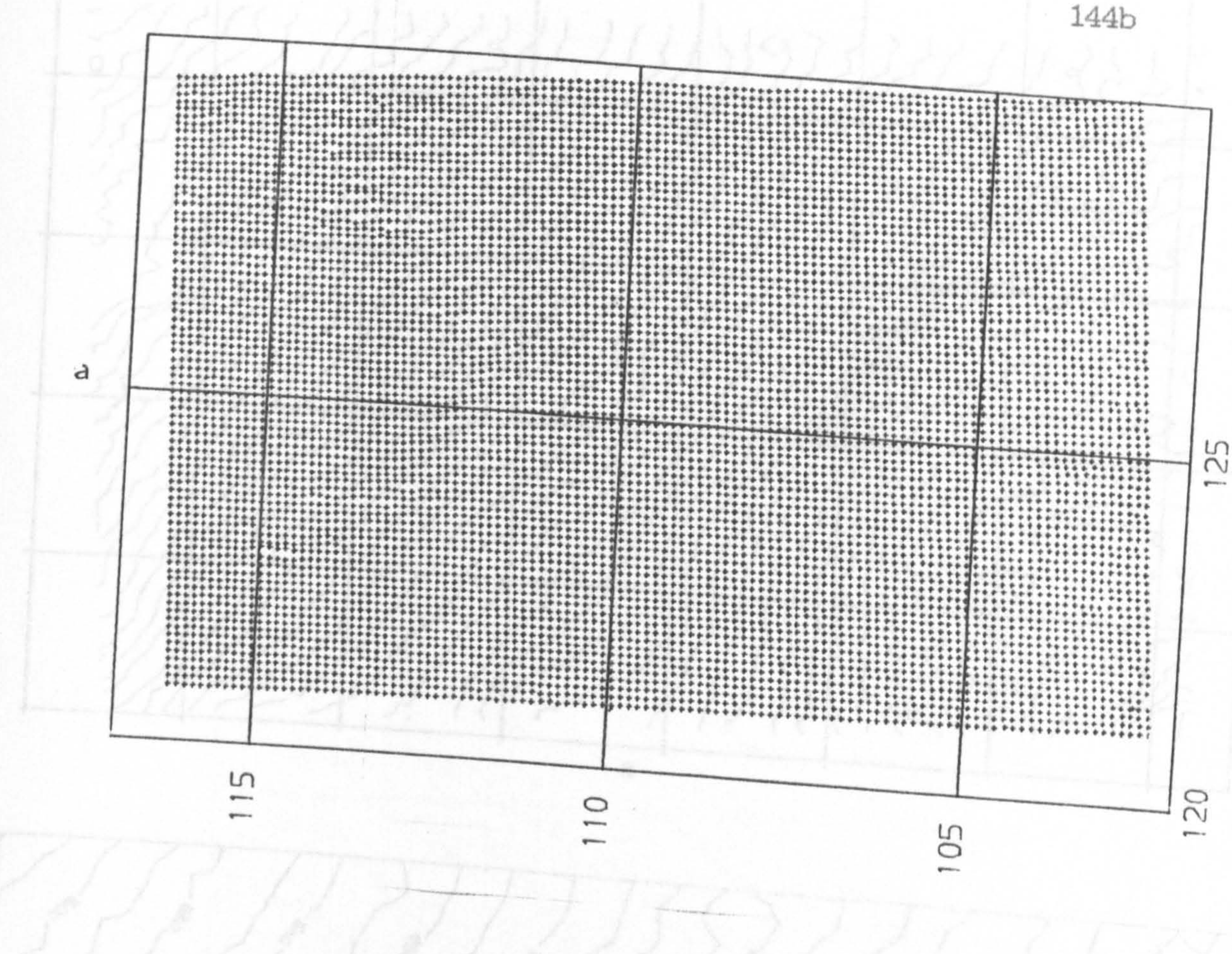
to test the programs described above. First, measurements along parallel lines in image space were carried out over the test site. Nearly 9,000 model points were generated (Table 6.1). The contour lines were then derived for the same points using the PANACEA DTM package developed by Dr. McCullagh of the University of Nothingham (McCullagh, 1983) and mounted on the ICL 3980 mainframe computer. The PANACEA package consists of different modules from which three modules were selected in this case. These comprised the modules PANIC for the Delaunay triangulation of the model points; PANDER for the derivative calculation of the triangulated points; and PANCON for generating the contour lines. PANCON uses the outputs generated by the PANIC and PANDER modules. The distribution of the DSP data, the resulting triangulation network and the contours generated for the test area are presented in Figs. 6.15a, 6.15b and 6.16a respectively. A smoothed version of the contour lines was generated by convolving the contour points by a smoothing function (Fig. 6.16b). The smoothing operations were carried out by my research colleague Mr. Zhilin Li (1990). The fluctuations of the contour lines are partly to the noise introduced by the operator's hand shaking during the measurement stage. This problem may be overcome by enlarging the pixel sizes (i.e. enlarging the digital image) so that any small shake of the hand during the pointing stage does not deviate the cursor from the pixels being measured.

To provide an accuracy estimation for the generated contour lines, for the same test site, another set of contour lines were plotted on the Wild B8 stereo-plotting machine using hard-copy (analogue) images. The two sets of the generated contour lines were then superimposed. The result is given in Fig. 6.17. As the result shows, there is a rather good agreement between the two sets of the generated contour lines.

The measurement along the lines with the same model height values was also carried out for the same area. The generated points were sent directly to the plotter. The result is given in Fig. 6.18a. The generated isometric view of the same area is presented in Fig. 6.18b.

X m	Y m	Z m
265224.7795	355327.1923	311.2470
265224.7474	355329.1959	312.2112
265224.7153	355331.1977	313.1739
265224.6833	355333.1982	314.1362
265224.7006	355335.0400	313.1750
265224.6440	355337.1161	315.0972
265224.6613	355338.9590	314.1378
265224.5799	355341.1068	317.0185
265224.5481	355343.0997	317.9770
265224.4916	355345.1646	319.8918
265224.4598	355347.1532	320.8485
265224.4277	355349.1400	321.8031
265224.3960	355351.1255	322.7579
265224.3887	355353.0384	322.7585
265224.3324	355355.0910	324.6654
265224.3251	355357.0022	324.6659
265224.3668	355358.7775	322.7601
265224.2130	355361.0940	328.4701
265224.2057	355363.0023	328.4707
265224.1739	355364.9757	329.4196
265224.1181	355367.0125	331.3150
265224.0865	355368.9821	332.2619
265224.0793	355370.8870	332.2625
265224.0236	355372.9162	334.1540
265224.0163	355374.8197	334.1546
265224.0333	355376.6629	333.2106
265223.9777	355378.6864	335.1003
265223.9219	355380.7065	336.9860
265223.9388	355382.5496	336.0439
265223.8832	355384.5658	337.9296
265223.8279	355386.5783	339.8095
265223.8207	355388.4772	339.8100
265223.7894	355390.4306	340.7512
265223.7101	355392.4896	343.5639

Table 6.1 A sample of the DTM data generated by the DSP.



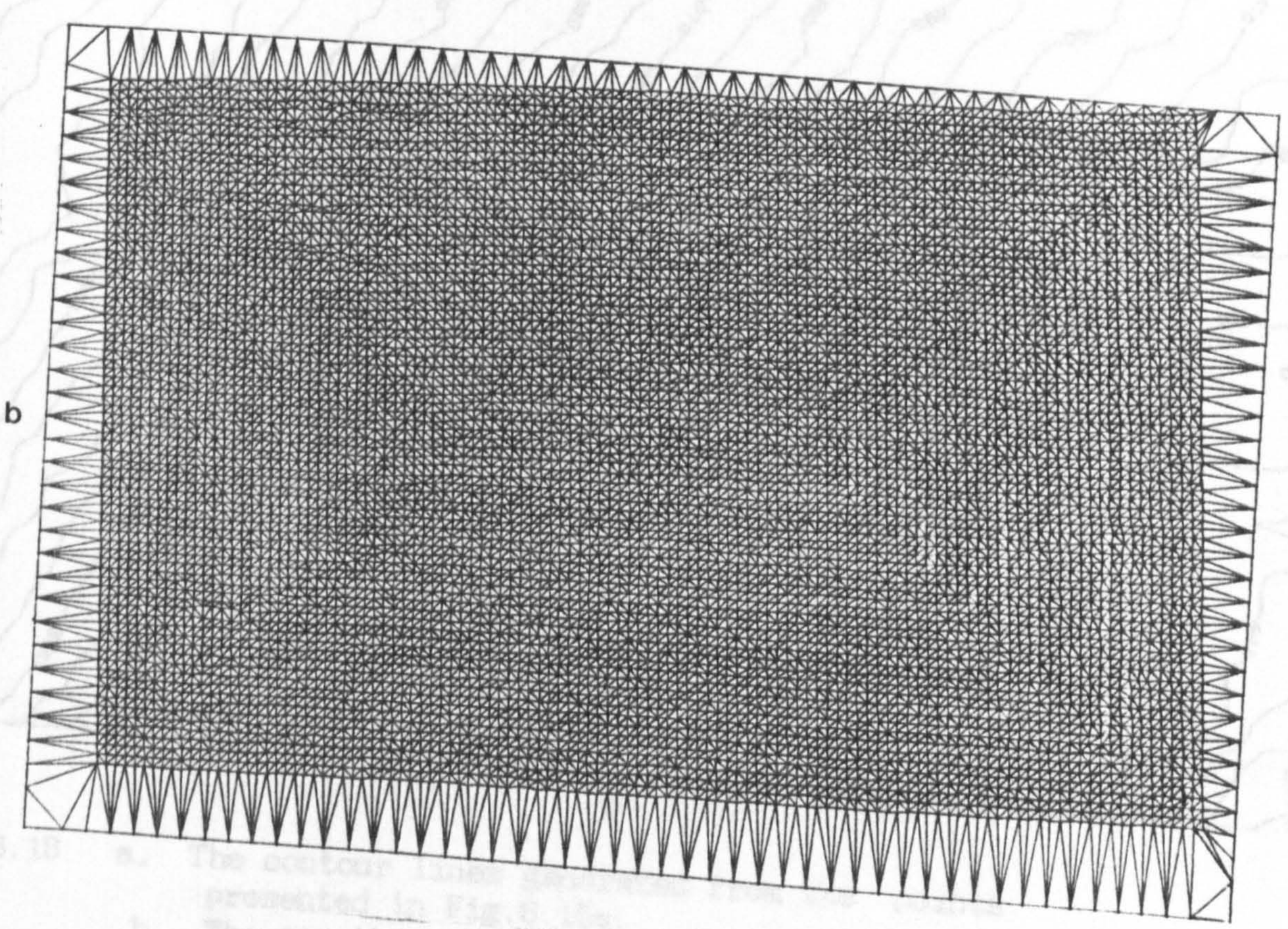
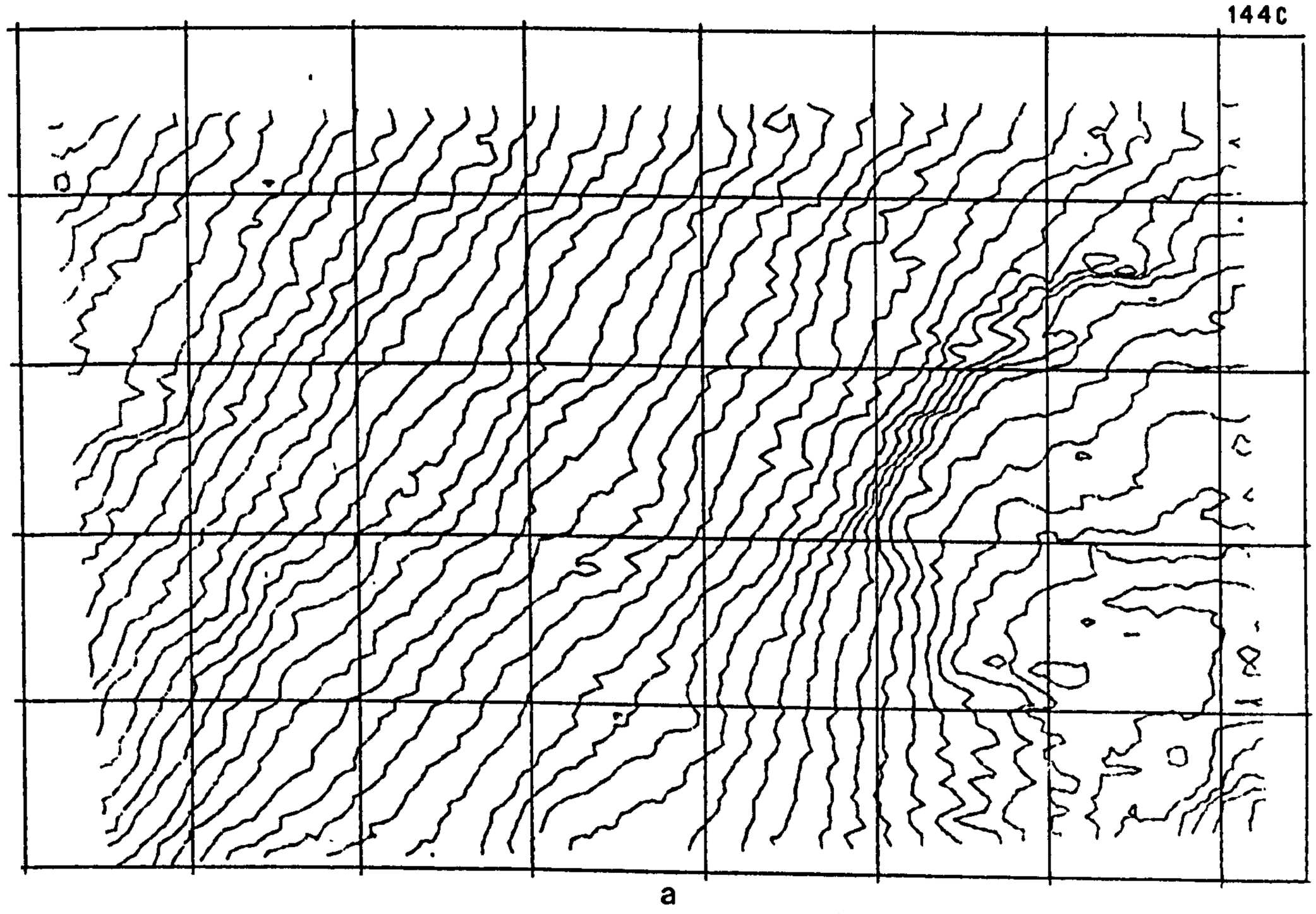
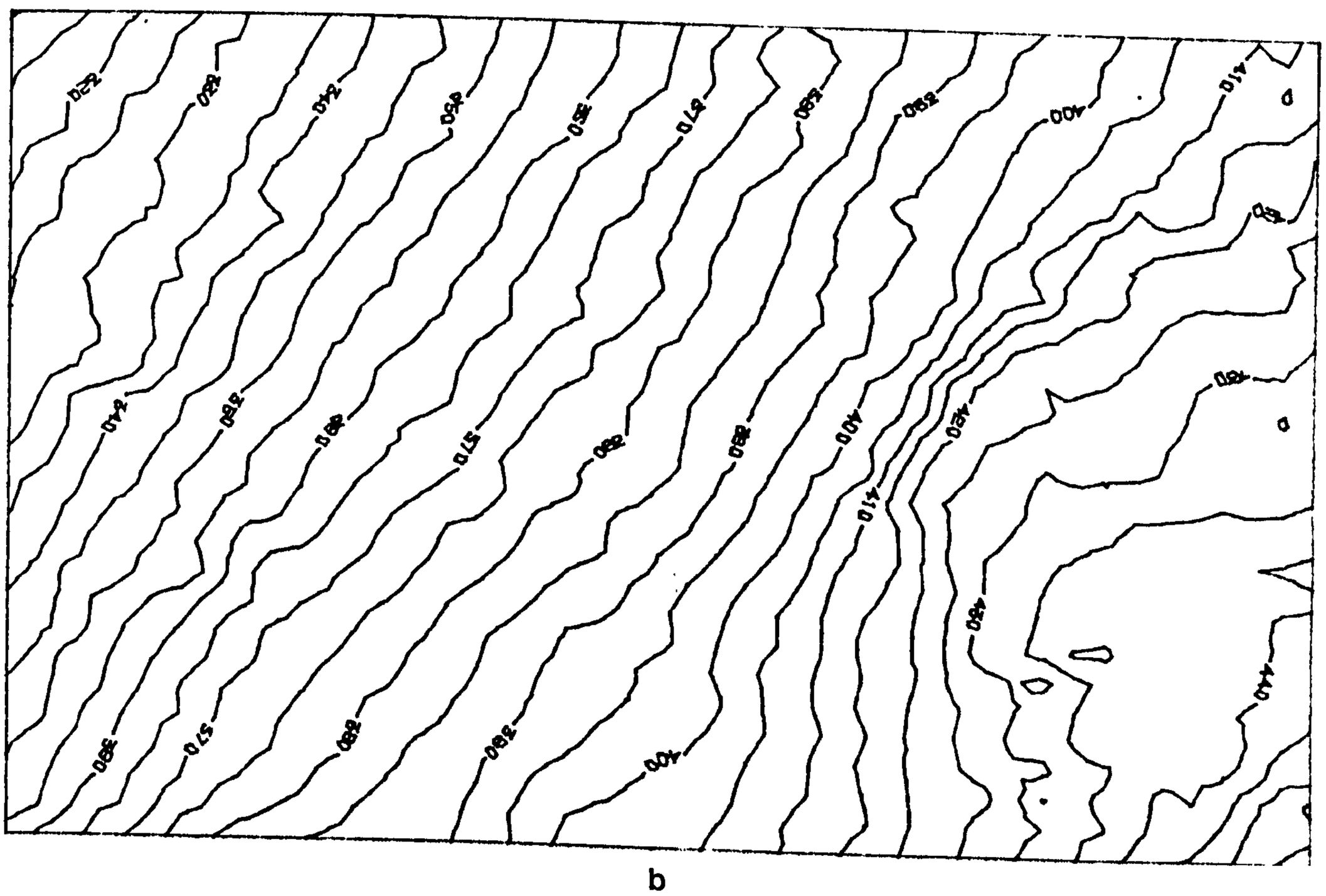


Fig.6.15 a. Distribution of the object points measured by the DSP.
b. The triangulation network for the points presented in Fig.6.15a.



a



b

Fig.6.16 a. The contour lines generated from the points presented in Fig.6.15a.
b. The smoothed contour lines generated from the DSP data.

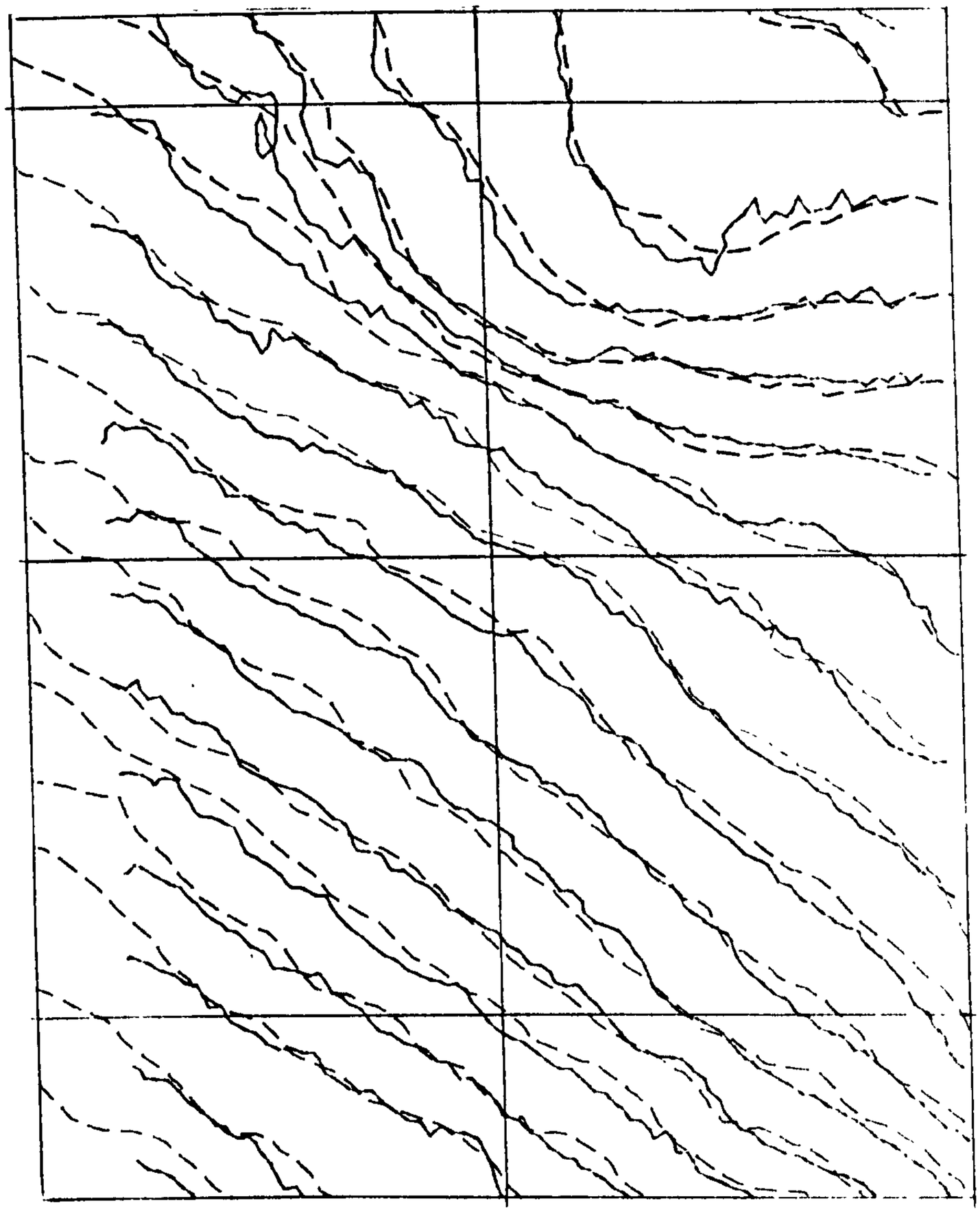
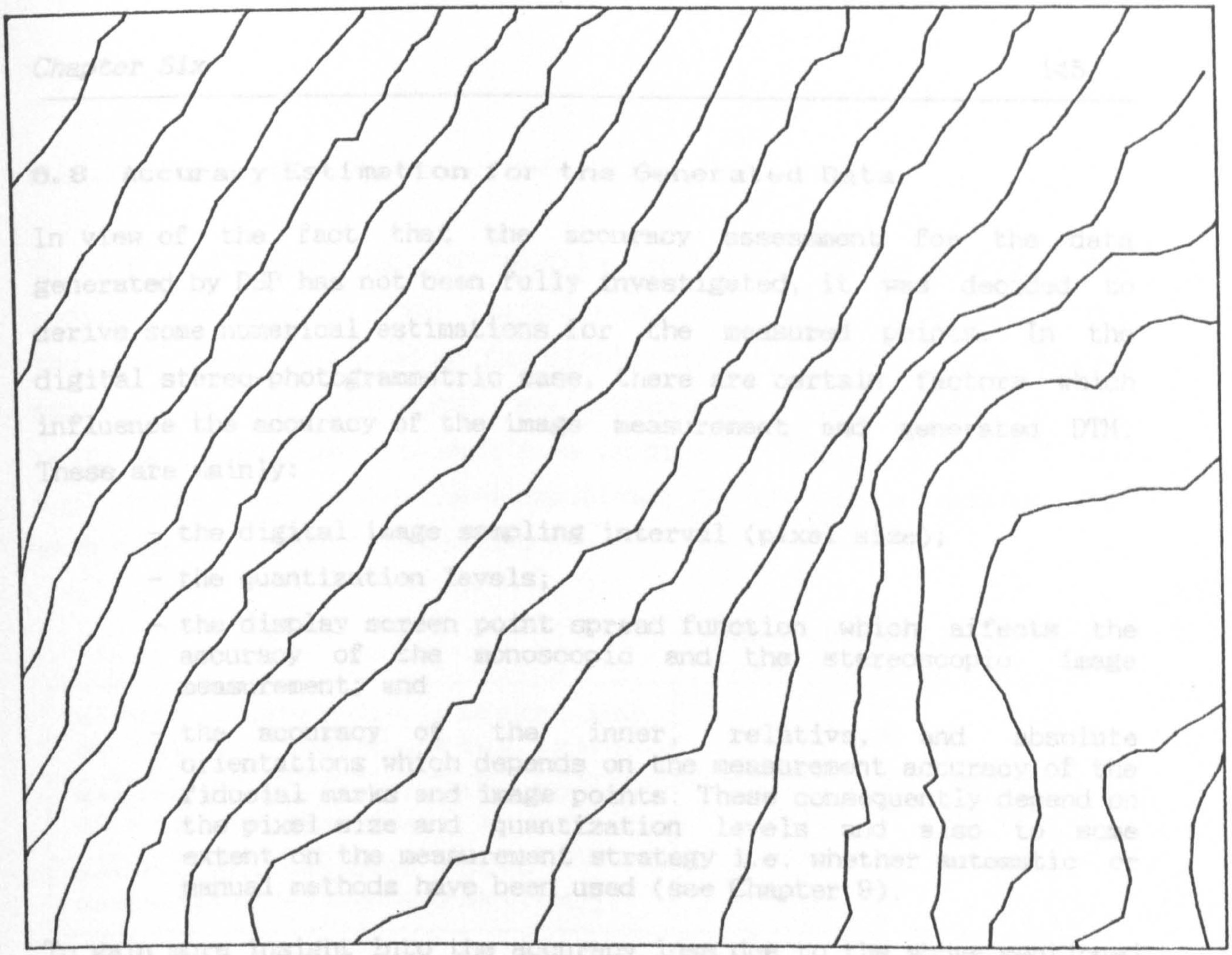
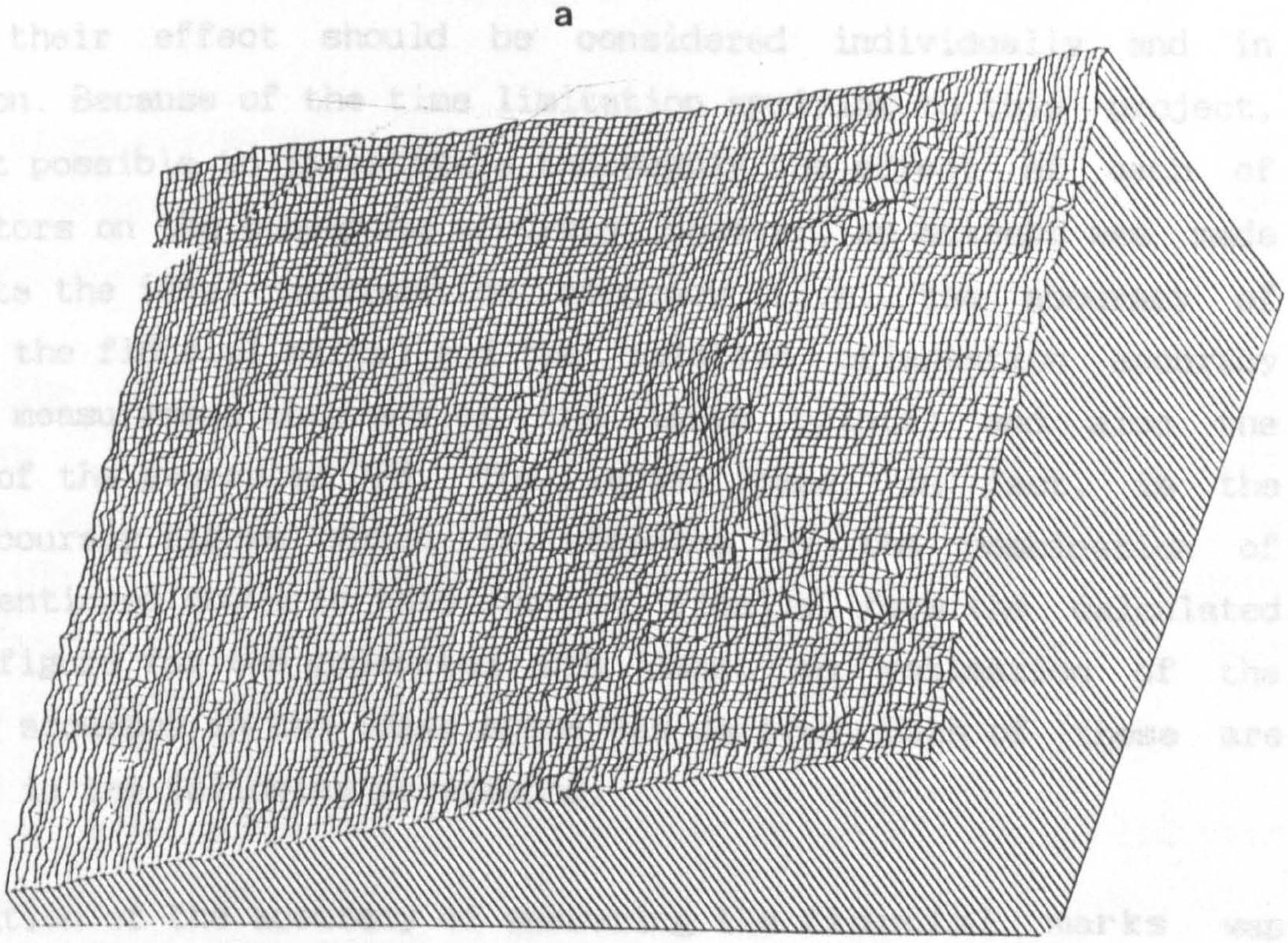


Fig.8.17 Superimposition of the contour lines generated by the DSP and B8 stereo-plotting machine respectively.



a



b

Fig.6.18 a. Contour lines generated by the DSP (direct contouring mode).
b. The isometric view produced from the generated data by the DSP.

6.8 Accuracy Estimation for the Generated Data

In view of the fact that the accuracy assessment for the data generated by DSP has not been fully investigated, it was decided to derive some numerical estimations for the measured points. In the digital stereo-photogrammetric case, there are certain factors which influence the accuracy of the image measurement and generated DTM. These are mainly:

- the digital image sampling interval (pixel size);
- the quantization levels;
- the display screen point spread function which affects the accuracy of the monoscopic and the stereoscopic image measurement; and
- the accuracy of the inner, relative, and absolute orientations which depends on the measurement accuracy of the fiducial marks and image points. These consequently depend on the pixel size and quantization levels and also to some extent on the measurement strategy i.e. whether automatic or manual methods have been used (see Chapter 9).

To gain more insight into the accuracy loss due to the above mentioned factors, their effect should be considered individually and in combination. Because of the time limitation applying to this project, it was not possible to investigate thoroughly the effect of each of these factors on the measuring accuracy. However, an attempt was made to estimate the inner orientation accuracy (i.e. the accuracy of measuring the fiducial marks) and the relative orientation accuracy (i.e. the measurement accuracy of the image points) and also the accuracy of the generated DTM. The latter case, in fact, is the overall accuracy figure which is affected by the combination of factors mentioned above in this section. Finally, from the calculated accuracy figure for the generated DTM data, an estimation of the measuring accuracy in the image space was derived. Each of these are discussed in the following paragraphs.

An estimation of the accuracy of measuring the **fiducial marks** was derived by comparing the coordinates of the fiducial marks measured on the digital image with the corresponding set measured by the Zeiss Jena Stecometer stereocomparator available at the National Engineering Laboratories (NEL) at East Kilbride. The results of these comparisons

are given in Chapter 9, Tables 9.5, 9.6, 9.7.

To give an estimation of the **relative orientation accuracy**, the residual y-parallaxes after the solution of relative orientation for the image points measured on the display device and the same points measured by the Stecometer stereocomparator, were compared. The residual y-parallaxes for each case is given in Tables 6.2, 6.3 and 6.4. The RMSE values for the y-parallaxes of the check points, for each case are given in Table 6.5.

	RMSE (RESIDUAL Y-PARALLAXES)
MANUAL MEASUREMENT CARRIED OUT ON THE CRT (NORMAL SIZE IMAGE)	±30 μM
MANUAL MEASUREMENT CARRIED OUT ON THE CRT (ENLARGED IMAGE)	±20 μM
MANUAL MEASUREMENT USING STECOMETER STERECOMPARATOR	± 8 μM

TABLE 6.5

The RMSE values presented in Table 6.5 give some estimation of the accuracy of measurement of the image points carried out for the relative orientation.

To assess the **accuracy of the generated DTM data**, the image coordinates of nearly 150 check points were measured using the Stecometer stereocomparator. To compare these points with the generated DTM data, they were transformed into the terrain coordinate system. These transformed coordinates were compared with the DTM data using the following approach.

A height value is calculated for the points on the DTM having the same x and y values of the check points, using a linear interpolation method. This can be justified by regarding the fact that the measured points are very dense. PANDEMON, one of the PANACEA modules, which is an editor, was used to perform the interpolation. The interpolation is carried out based on the triangulated data generated by the PANIC module. The linearly interpolated heights were then compared with the

---------- FINAL ROTATION MATRIX ----------

Iteration = 3

0.99872	-0.04960	0.01020
0.04946	0.99868	0.01353
-0.01086	-0.01301	0.99986

BY	BZ	PHI	OMEGA	KAPPA
2.88730	0.98138	0.01020	-0.01353	0.04962

Model coordinates (mm) Digital image (normal size):

PT. NO.	X	Y	Z	PY
1.	112.44371	30.46998	-5.49626	0.00855
2.	110.56895	31.48500	-4.88622	0.00801
3.	113.43266	31.18990	-5.89529	-0.00368
4.	112.24636	30.16342	-5.44856	-0.01290
5.	89.20174	122.27828	7.33301	0.02992
6.	89.14270	120.21298	7.14529	0.00122
7.	88.81231	120.42454	7.17760	-0.01400
8.	90.21673	123.71901	7.22831	-0.01831
9.	115.65114	187.20927	8.37285	-0.02344
10.	116.92765	185.31905	8.28478	0.00842
11.	114.70556	190.74810	9.05857	-0.01186
12.	115.51864	191.22696	8.84022	0.02766
13.	195.27261	37.80145	8.37157	-0.01994
14.	196.01736	37.08798	8.50329	-0.02174
15.	194.43153	40.69513	8.23770	0.02503
16.	193.71905	42.43945	8.03646	0.01975
17.	175.59232	136.69011	2.82561	-0.00343
------- CHECK POINTS:				RMSE 17 μm
18.	175.27980	136.02972	2.82514	0.00815
19.	175.95054	137.18718	2.86912	-0.04444
20.	176.83836	138.31226	2.91879	0.00867
21.	200.49838	201.45560	15.39385	0.03504
22.	200.70072	201.51452	15.39239	-0.04139
23.	200.03072	201.93664	15.49614	-0.01754
24.	199.61286	202.86248	15.93482	-0.03582
				RMSE 30 μm

Table 6.2

FINAL ROTATION MATRIX

Iteration = 3

0.99871	-0.04978	0.01029
0.04965	0.99868	0.01313
-0.01093	-0.01260	0:99986

BY	BZ	PHI	OMEGA	KAPPA
2.82699	0.99183	0.01029	-0.01313	0.04981

Model coordinates (mm) Digital image (enlarged):

PT. NO.	X	Y	Z	PY
1.	112.44689	30.47692	-5.47279	-0.00848
2.	110.60312	31.49847	-4.84522	0.00292
3.	113.45441	31.18679	-5.92222	0.00727
5.	89.20662	122.26230	7.34601	0.01079
6.	89.28366	120.16875	7.18025	-0.00390
7.	88.87494	120.39829	7.27109	-0.00876
10.	116.94357	185.26584	8.40943	-0.01238
11.	114.74327	190.72349	9.15750	-0.00400
12.	115.55830	191.21292	8.93782	0.01597
13.	195.30197	37.79296	8.33679	-0.00342
16.	193.71468	42.54126	8.11847	0.00186
17.	175.60525	136.66528	2.87107	-0.00009
18.	175.30045	136.03986	2.88433	0.00429
21.	200.49999	201.44265	15.43343	0.00939
22.	200.68720	201.45997	15.42948	-0.00800
23.	200.01985	201.87940	15.55631	0.00953
24.	199.59612	202.83588	15.94640	-0.01239
------- check points:			RMSE	8 μm
4.	112.29987	30.18671	-5.39008	-0.01345
8.	90.37021	123.68523	7.45875	-0.02527
9.	115.68072	187.13836	8.51465	0.01507
14.	196.07226	37.05945	8.44574	-0.01925
15.	194.45485	40.78100	8.28680	0.03223
19.	175.94365	137.17579	2.89234	-0.01855
20.	176.83628	138.29838	2.92358	0.01639
			RMSE	20 μm

Table 6.3

FINAL ROTATION MATRIX

Iteration = 3

0.99871	-0.04970	0.01041
0.04956	0.99869	0.01301
-0.01104	-0.01248	0.99986

BY	BZ	PHI	OMEGA	KAPPA
2.80312	0.97326	0.01041	-0.01301	0.04972

Model coordinates (mm) for the points measured by Stecometer:

PT. NO.	X	Y	Z	PY
1	112.50416	30.56162	-5.33574	-0.00763
2	110.64025	31.56899	-4.73392	0.01005
3	113.47777	31.22822	-5.88055	-0.00235
5	89.23338	122.29688	7.40245	0.00480
6	89.32620	120.15353	7.24147	-0.00360
7	88.85757	120.40282	7.29526	-0.00559
8	90.40521	123.68016	7.58169	0.00369
9	115.74172	187.17460	8.53110	-0.00030
10	116.96766	185.26100	8.45100	0.00641
11	114.75649	190.72619	9.19177	-0.00476
12	115.57216	191.22653	8.95744	-0.00074
13	195.28364	37.84393	8.41921	-0.00266
14	196.01581	37.12226	8.56587	0.00278
17	175.60045	136.64940	2.94770	0.00229
18	175.42594	136.01292	2.90646	-0.00702
19	175.94373	137.17662	2.93050	0.00673
20	176.81842	138.30392	2.96614	-0.00265
----- check points			RMSE	5 μm
87	216.03968	195.78253	10.79621	-0.00011
88	217.05910	196.04296	10.70820	0.01240
90	121.06015	102.68766	4.67541	0.00197
91	117.35234	93.47696	0.22437	0.01429
92	119.23709	92.98773	0.30977	0.00182
93	110.47171	100.58806	1.73794	-0.00718
			RMSE	8 μm

Table 6.4

height values of the check points and the residuals were calculated to give an estimation of the accuracy of the generated DTM with respect to the check points. The frequency (histogram) of the residual height error occurrences and the numerical values of the residuals are given in Figs. 6.19a and 6.19b respectively. The vector plot of the residual height errors is given in Fig. 6.20.

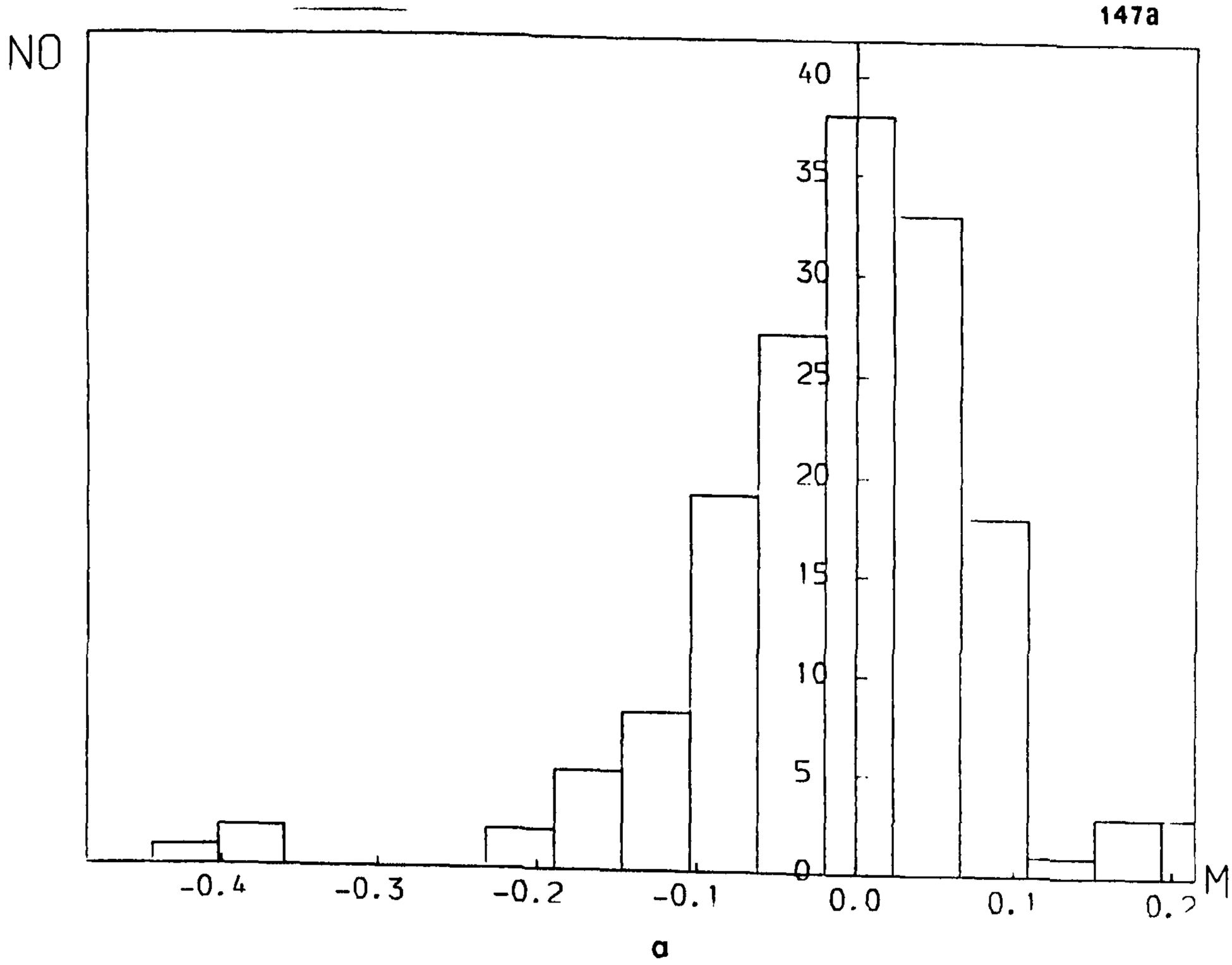
The mean, standard deviation and the root mean square error values for the residual height errors are given in Table 6.6.

mean res. (m)	std. dev. (m)	rmse (m)
-0.118	±1.110	±1.116

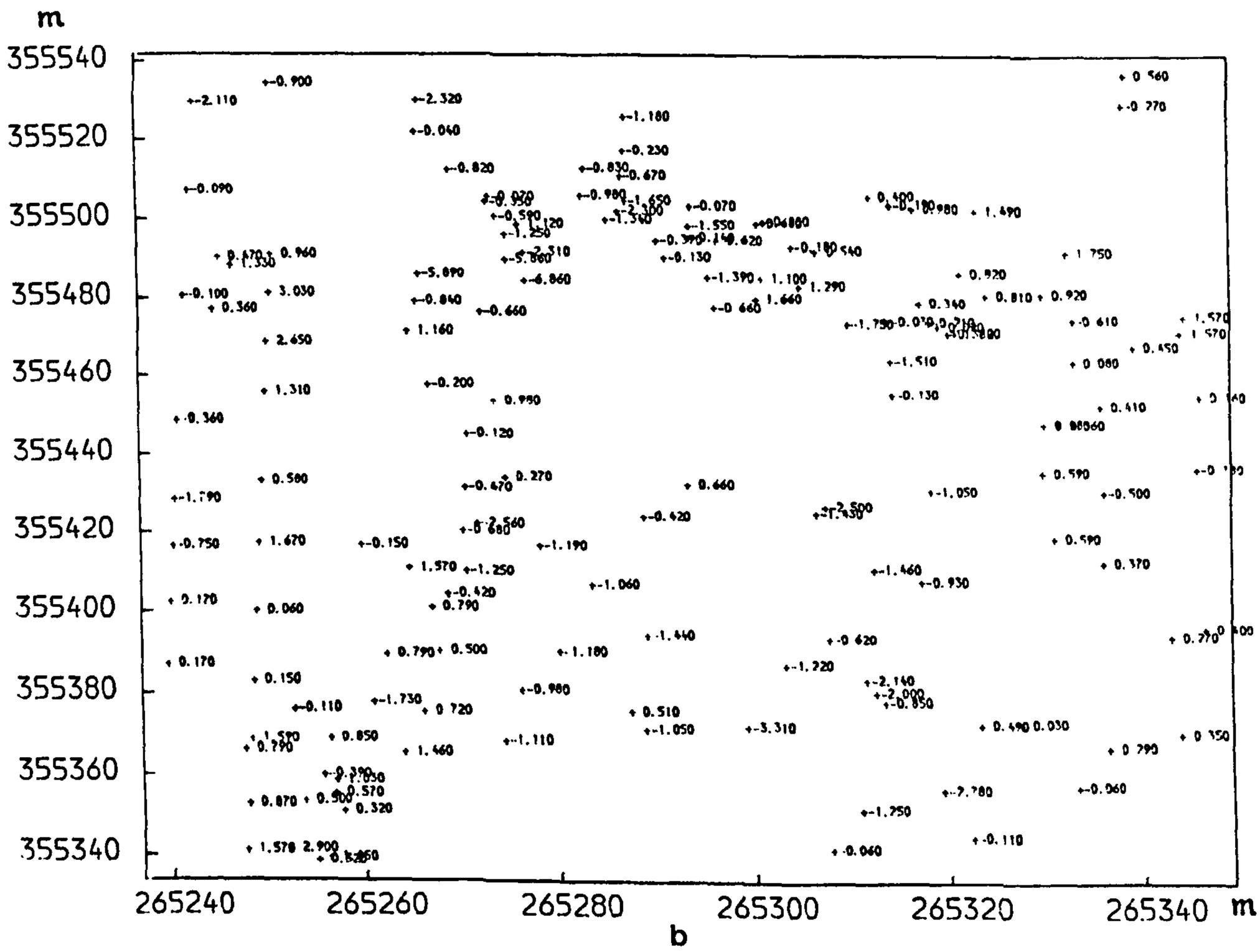
TABLE 6.6

The RMSE value of ±1.11 metre on the terrain (Table 6.6) gives the overall figure for the entire test area. However, the test area has quite different characteristics as far as its texture is concerned. There are some areas of shadow and low signal content for which the measuring accuracy falls to 3 and 2 metres respectively. Excluding these areas from the accuracy figures yields an accuracy of about 1 metre. Regarding the scale of the photography, this yields a relative height error of about 0.35 %.H (flying height). When shadow and low signal content areas are included, the relative height error is increased to about 0.4 %. H (flying height).The mean value of the residuals (Table 6.6) shows a systematic pattern. This is possibly due to the incomplete relative orientation resulting from inaccurate image point measurement (by comparison with the image point measurement by the Stecometer machine which incorporates the full resolution of the image) during the relative orientation stage. This, of course, is directly related to the pixel size.

Regarding the fact that non-digital analogue or analytical stereo-plotters can achieve a relative height error of about ±0.3 %.H (flying height) in a dynamic mode of measurement (under the condition of having no systematic errors), the accuracy achieved for the data



a



b

Fig.6.19 a. The histogram of the residual height errors presented in Fig. 6.19b.
b. The residual height errors at the check points.

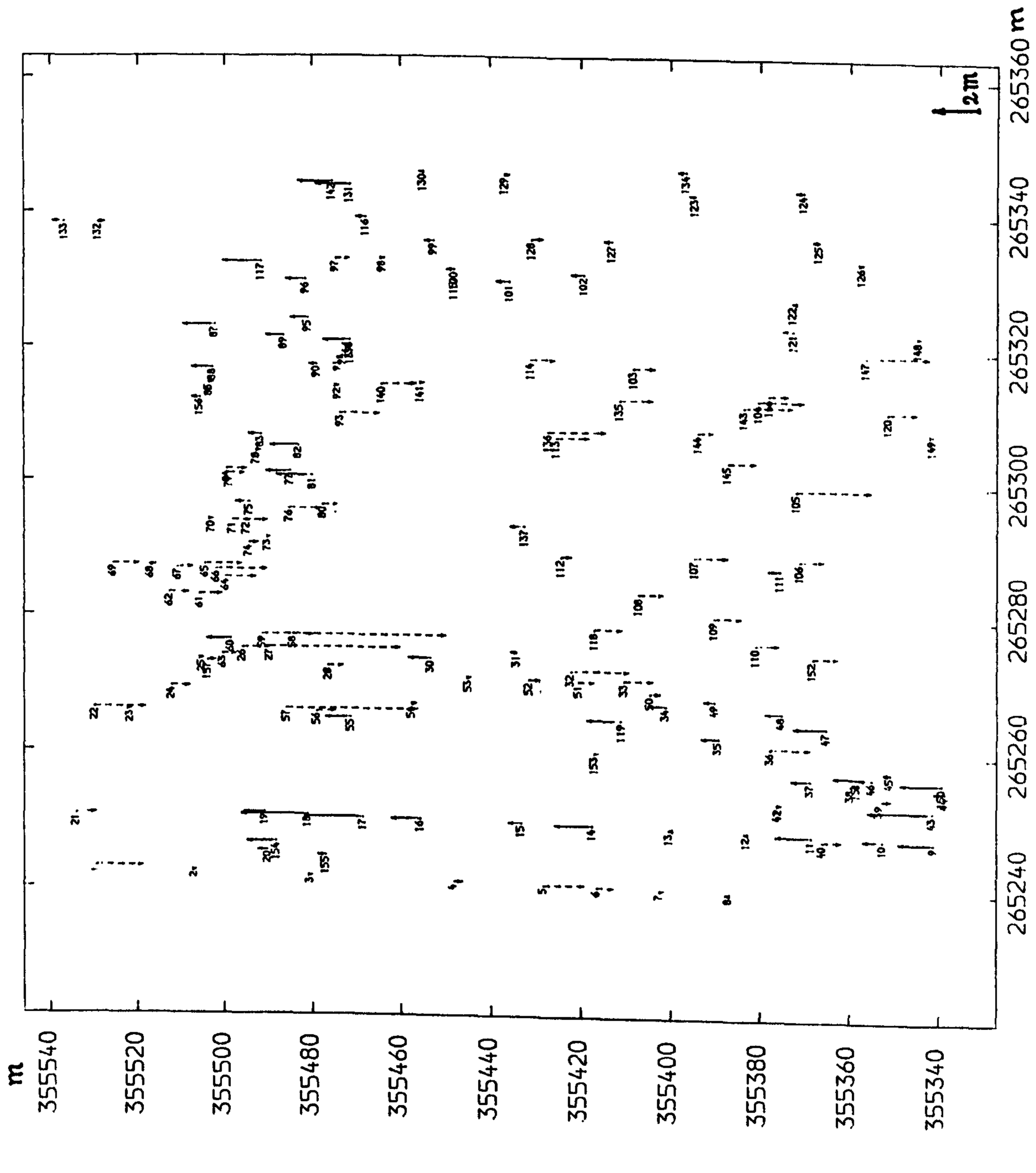


Fig.6.20 The vector plot of the residual height errors of Fig. 6.19b.

generated by the DSP, which were measured in an almost dynamic mode[1], seems to be quite satisfactory. As mentioned before, the accuracy figure for the generated DTM may be further improved by reducing the effects of systematic errors and also by carrying out the measurements on enlarged digital images.

6.8.1 Estimation of The Accuracy of The Pixel Measurement in The Image Space

To give an estimation of the measuring accuracy in the image space in terms of pixel size and therefore to present a rough approximation of the accuracy of measuring the pixels when the images are viewed stereoscopically on a display device, the following procedures were followed.

Initially the accuracy figures were calculated for the model points following the same procedures as those given in the preceding section, but on this occasion, the check points and the generated DTM points were not transformed to the terrain coordinate system. The calculated accuracy figures for the residual height errors in the model space are presented in Table 6.7.

mean res. (mm)	std. dev. (mm)	rmse (mm)
-0.022	±0.070	±0.074

TABLE 6.7

From the calculated discrepancies in the model space, an accuracy estimation for the measurement of the pixel position in the x-direction in image space was derived according to the following.

In the standard case of stereo-photogrammetry, the following relation exists between the Z model coordinate and the measured x, x' image coordinates:

1 *An almost dynamic mode, since, due to the hardware limitations of the system, the measurements were performed at a very low speed and hence a fully dynamic mode of measurement could not be realized.*

$$Z = b \frac{f}{x - x'} \tag{6.1}$$

where f is the camera focal length and b is the model base length. Assuming the orientation parameters to be error free, the application of the law of propagation on Eq. 6.1 yields the variances of the Z model coordinate as follows:-

$$\sigma_z^2 = \frac{1}{b^2}(f^2\sigma_x^2 + f^2\sigma_{x'}^2) \tag{6.2}$$

With the simplification of $\sigma_x = \sigma_{x'}$, the final formulation will be obtained as

$$\sigma_z^2 = \frac{1}{b^2} 2f^2\sigma_x^2 \tag{6.3}$$

or

$$\sigma_x = \pm\left[\frac{b^2\sigma_z^2}{2f^2}\right]^{1/2} \tag{6.4}$$

Regarding the value of the RMSE given in Table 6.7, an estimation of the accuracy of measurement of the pixels in the image space was calculated using the relations derived in Eq. 6.4. This yields the measuring accuracy of about one pixel size. The estimation of the measuring accuracy excluding the shadow and low signal content areas yields a value of around half the pixel size.

6.9 Conclusion

In this chapter, it has been demonstrated that a fully digital solution for DTM generation is feasible on a multi-user mainframe computer and using hardware which is not designed or intended specifically to carry out such a task. As expected, the operation speed for point measurement is slow, and efficient image scrolling is not possible. However, this particular DSP is simply a prototype or demonstrator and these operations would be greatly speeded up and made more practical and more efficient if a dedicated machine with purpose-built hardware features was available for the task.

In this chapter, some accuracy estimations for the generated DTM and

the pixel measurements have also been derived and, with regard to the pixel size, it gave a satisfactory result.

The special feature of the DSP which is designed and implemented in this research work is that the left and right images are first rectified in an off-line solution and the measurements are then carried out in real time after the images are rectified. Hence, the working principle of this DSP is not similar to any other group of analytical photogrammetric instruments presented in Section 6.2.

In the next chapter, an investigation into the feasibility of using binary images for stereo-viewing and DTM generation is presented.

CHAPTER SEVEN

Stereo-Viewing and Measurement on Overlapping Binary Images

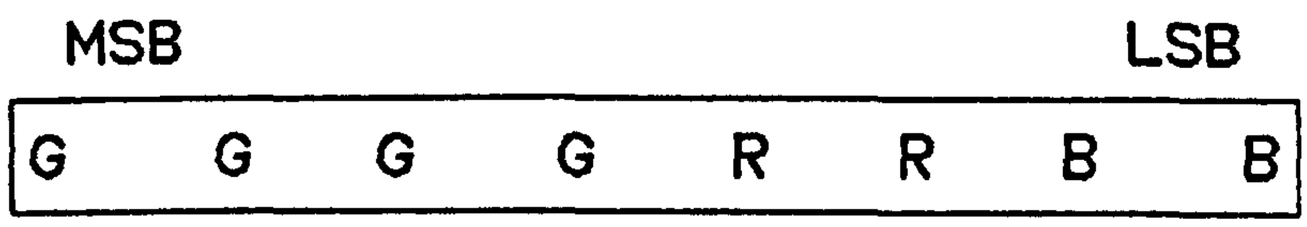
7.1 Introduction

In the preceding chapter, the model reconstruction and measurement using a DSP were presented and discussed. It was noticed during the course of the project that the range of grey levels representing the densities of the pixels may be as low as 2 without affecting the stereo-viewing perception. This was true under the condition of using a special mathematical algorithm which transforms the digital image into binary form (i.e. a two-dimensional image with two possible values representing on or off) without significantly affecting the image quality. This was considered to be an important factor from the digital photogrammetric point of view in that a binary image could be used for stereo-viewing and measurement which leads to the possibility of using cheaper display devices for the DSP. As far as photogrammetry is concerned, since the main objective behind the display of the image and stereo-viewing is to extract the metric information, it was considered to be essential to test the geometric fidelity of the produced binary image experimentally. The test was carried out by measuring a series of points on images with two levels of grey shade and comparing them with the points measured on the images with more grey level variation. The algorithm which was used for reducing the range of the image grey levels and the test results are discussed in the sections which follow.

7.2 The Range of the Grey Shade Requirement

Regarding the fact that the usual consumer of the displayed image - as far as the DSP is concerned - are human eyes, it can be demonstrated that the use of display devices with full 8-bit monochrome frame buffers and full colour frame buffers are not a necessity in all circumstances. Based on the human visual system (HVS) and the display resolutions commonly in use, Cowlishaw (1985) suggests the use of 4 bits per pixel and 8 bits per pixel for the display of monochrome and colour images respectively. This suggestion is reported to be based both on the theory of the visual system in which the detectors in the eye are modelled as simple photon detectors, and on experimental results carried out by Cowlishaw.

The use of the colour graphics frame buffers available on the IBM 5080 graphics display system is considered based on the suggestion presented above. As mentioned in the preceding chapter, the 5080 graphics system has a total of 8 bit planes available for both monochrome and colour display. As a result, it can produce 256 grey shades in monochrome and the same number of the colours for the coloured images. Therefore, for displaying colour images, the full range of the density variation for each primary colour cannot be utilized. The allocation of the 8 available bit planes of the 5080 display device to each primary colour is based on the colour response of human eye and is arranged according to the following scheme.



According to the above configuration, 4 bit planes are allocated to green and 2 bit planes each to the red and blue colours respectively. This yields 2^4=16, for green and 2^2=4 grey shades for red and blue respectively which produces the total number of 256 (16X4X4) colours. The larger number of green levels gives increased resolution and reflects the eye's increased sensitivity in this part of the spectrum (IAX manual). Regarding the bit allocation described above, to display a colour image which has a continuous range of grey shades, IAX incorporates an algorithm, the so-called **Error Diffusion half-toning function (ERD)**, to convert the image from 256 grey levels to n grey levels where n can take any number between 2 and 128.

Since for stereo-viewing of the left and right digital images, the anaglyphic method was used, the green and red images were reduced to a range of 16 and 4 grey shades respectively. This was carried out by the Error Diffusion function algorithm. All the tests which were reported in the previous chapter, were carried out on these reduced grey level images. However, this grey level reduction was found to be insignificant in terms of both stereo-viewing perception and measuring

accuracy. In particular, as mentioned earlier, it was noticed that even images with two levels of grey shade could produce a satisfactory stereo-viewing perception.

To gain more insight into the nature of the algorithms which produce binary images, the next sections give some details about such algorithms and, in particular, the ERD function is presented in more detail and discussed.

7.3 Algorithms for Producing Binary Images

Various techniques for rapidly generating low-cost hard copies of continuous tone digital imageries have been developed in recent years (Floyd and Steinberg, 1976; Roetling, 1977). All these approaches involve some kind of binary representation of the image. The simplest approach towards producing a binary image, is a "**Globally fixed thresholding technique**" by which each pixel in the input image is compared with the specified threshold and is turned on in the output image if it equals or exceeds the threshold value, i.e.

$$g(i,j) = \begin{cases} 0 & \text{if} \quad f(i,j) < \text{threshold} \\ 1 & \text{if} \quad f(i,j) \geq \text{threshold} \end{cases} \qquad (7.1)$$

where f and g denote the continuous tone and binary images respectively. This approach was applied to the test image used in the previous chapter and the result is given in Fig. 7.1a. Clearly, this method causes a dramatic deterioration in the image quality, due to the removal of some information content. It also introduces artifacts and distorts the image geometry (i.e. it causes the aliasing problem), and hence cannot satisfy the photogrammetric requirements.

However, there are some more sophisticated approaches which use more complex mathematical algorithms and hence produce more satisfactory results. Examples are the use of the constraint average threshold; the ordered-dither approach; employment of the error diffusion function; etc. (Best et al., 1987). The last of these approaches seemed to be the most promising and was therefore examined in this research work.

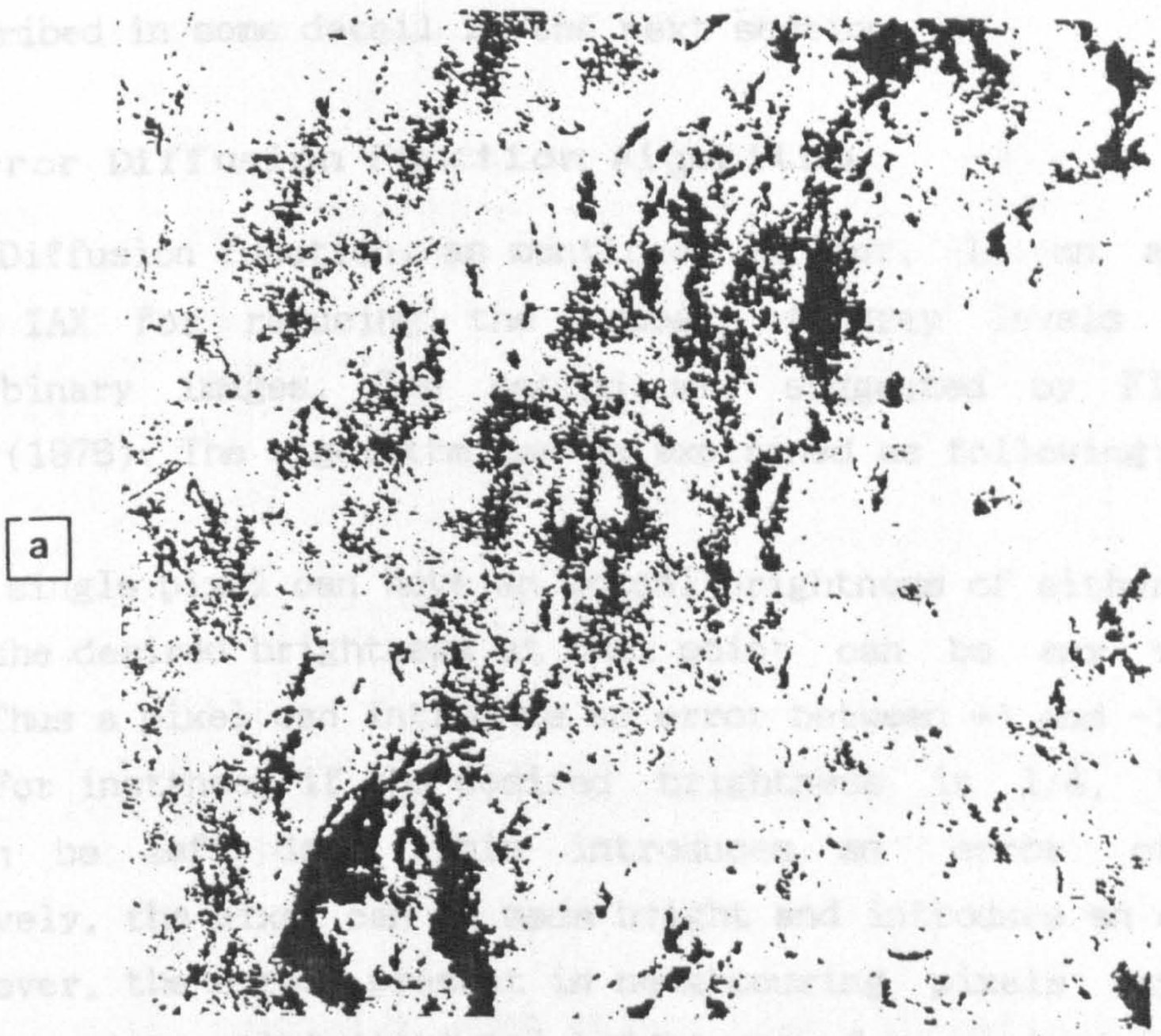


Fig.7.1 a. Binary image produced by a globally fixed thresholding technique.
b. Binary image produced by the error diffusion function algorithm.

It is described in some detail in the next section.

7.3.1 Error Diffusion Function Algorithm

The Error Diffusion function, as mentioned earlier, is an algorithm adopted in IAX for reducing the number of grey levels and for producing binary images. The method was suggested by Floyd and Steinberg (1976). The algorithm can be explained as following:

Suppose a single pixel can have an actual brightness of either 0 or 1, although the desired brightness at that point can be any value in between. Thus a pixel can introduce an error between +1 and -1 to the picture. For instance, if the desired brightness is 1/4, then the pixel can be left dark. This introduces an error of -1/4. Alternatively, the pixel can be made bright and introduce an error of +3/4. However, the errors present in neighbouring pixels may cancel one another out, so that the total brightness of several pixels taken together is close to that of the total desired brightness for that group. Thus if it has been decided that the pixel with the desired brightness of 1/4 should be dark, then there is an error of -1/4. This means that the neighbours of that pixel should have a compensating error of +1/4, to cancel out this first error. Suppose one decides to have all of that error cancelled by one of the neighbours, then that neighbour should be 1/4 brighter than the original desired brightness. Now the process is repeated with its neighbour. If its modified desired brightness is above 1/2, then the neighbour, is made light, otherwise it is left dark. In either case, the desired brightness of a neighbour of the neighbour is modified to cancel the error introduced. This may be shown mathematically as,

$$g(i,j) = \begin{cases} 0 & \text{if} \quad f(i,j) + \text{weighted error} < \text{threshold} \\ 1 & \text{otherwise} \end{cases} \qquad (7.2)$$

This process can be thought of as diffusing the error from one pixel among the neighbouring pixels, thus suggesting the name for the algorithm. The following block diagram shows the information processing required for the conversion of a continuous tone image into

a binary image using the ERD function.

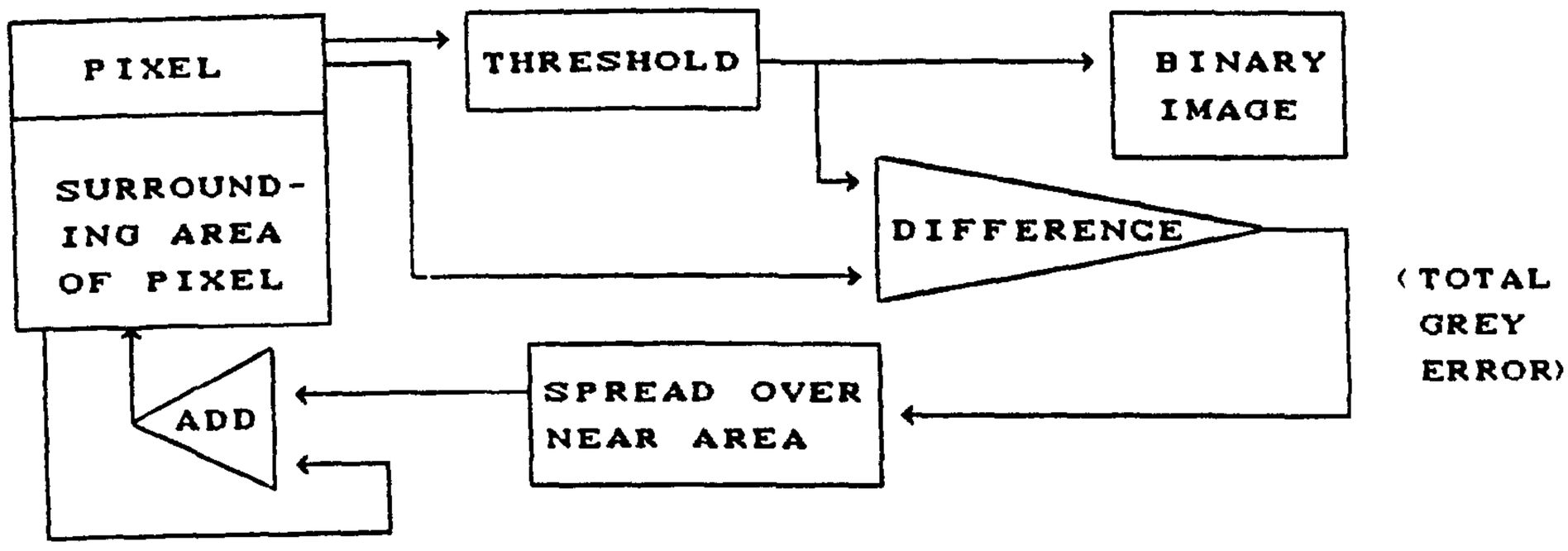


Fig. 7.2 BLOCK DIAGRAM SHOWING INFORMATION PROCESSING FOR CONVERSION OF CONTINUOUS TONE TO BINARY IMAGE USING "ERD" (FLOYD AND STEINBERG, 1976).

This algorithm is reported by Best et al. (1987) to have produced the best results for generating binary images compared with the other algorithms mentioned above. However, the ERD algorithm is more complex and computationally more expensive than the other algorithms such as the ordered-dither and constraint average threshold methods.

7.4 Test Results

The ERD function has been tested by different researchers with the main view of producing satisfactory output on line printers and binary display monitors (Floyd and Steinberg, 1976; Roetling, 1977; Best et al., 1987). However, as mentioned earlier, in this project, the method has been examined here specifically with a view to assessing its suitability for photogrammetric applications.

The ERD function was applied to the test images used in the preceding chapter to produce an image with two grey level variations. The resulting binary image is presented in Fig. 7.1b, this hardcopy having been produced by the IBM 4250 electro-erosion printer. Indeed all of the hardcopy digital images included throughout this thesis have been produced using the ERD algorithm and the IBM 4250 printer. As Fig 7.1b shows, small objects can easily be detected on the image and only very insignificant artifacts have been introduced into the image. The

amount and the quality of the information content of Fig. 7.1b can be easily realized by comparing it with Fig. 7.1a which presents a binary image of the same area generated using a globally fixed thresholding technique. Because the images which are generated by the ERD technique almost maintain their quality, stereo-viewing was achieved with a very satisfactory result. The images were used to generate contour lines (Fig.7.3) by employing the DSP described in the preceding chapter. The generated contour lines were superimposed on the contour lines derived using non-binary images (see Chapter 6, Fig. 6.18a). The result of the superimposition shows a good agreement between the two sets of the data (Fig. 7.4).

7.5 Conclusion

This small experiment does not show the scale of the aliasing effect and the geometric distortion imposed on the image by the Error Diffusion function, and it does not show the effect of the presence of small artifacts when pointing to small image points as required during relative and absolute orientation. But nevertheless the test does demonstrate the possibility of using such images on the binary display devices which might be utilized on DSP systems. In the author's view, more investigations should be carried out to test the ERD algorithm regarding the extent of the aliasing effect which may be introduced on the binary image by this function. Moreover, other algorithms should also be tested and compared with the ERD algorithm.

So far, mainly, the manual method of measuring point targets and the elevations needed for DTM generation have been considered. The next chapter deals with automatic methods of measuring such targets and producing elevations for DTM, by employing image correlation techniques.

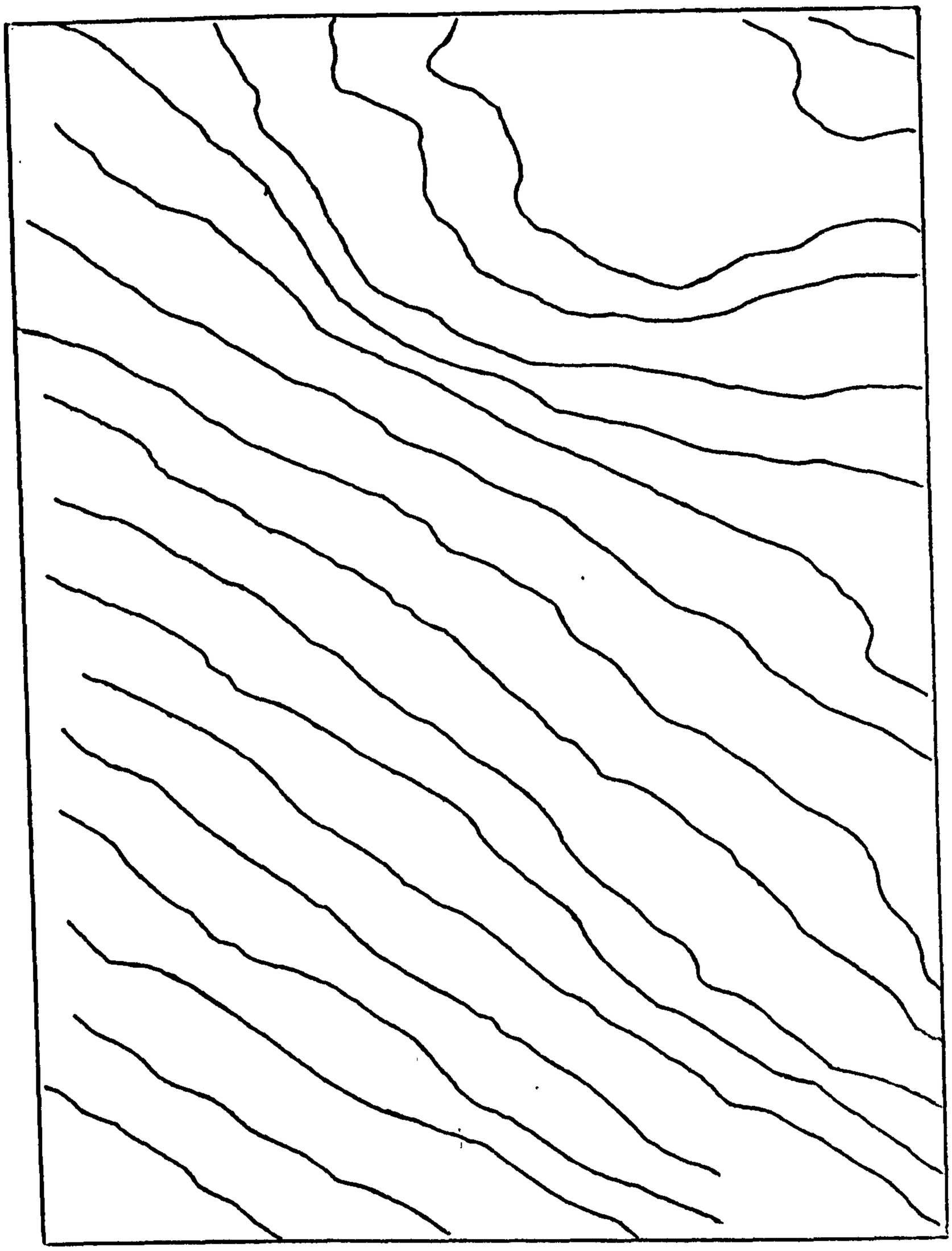


Fig.7.3 Contour lines generated using binary images produced by ERD function.

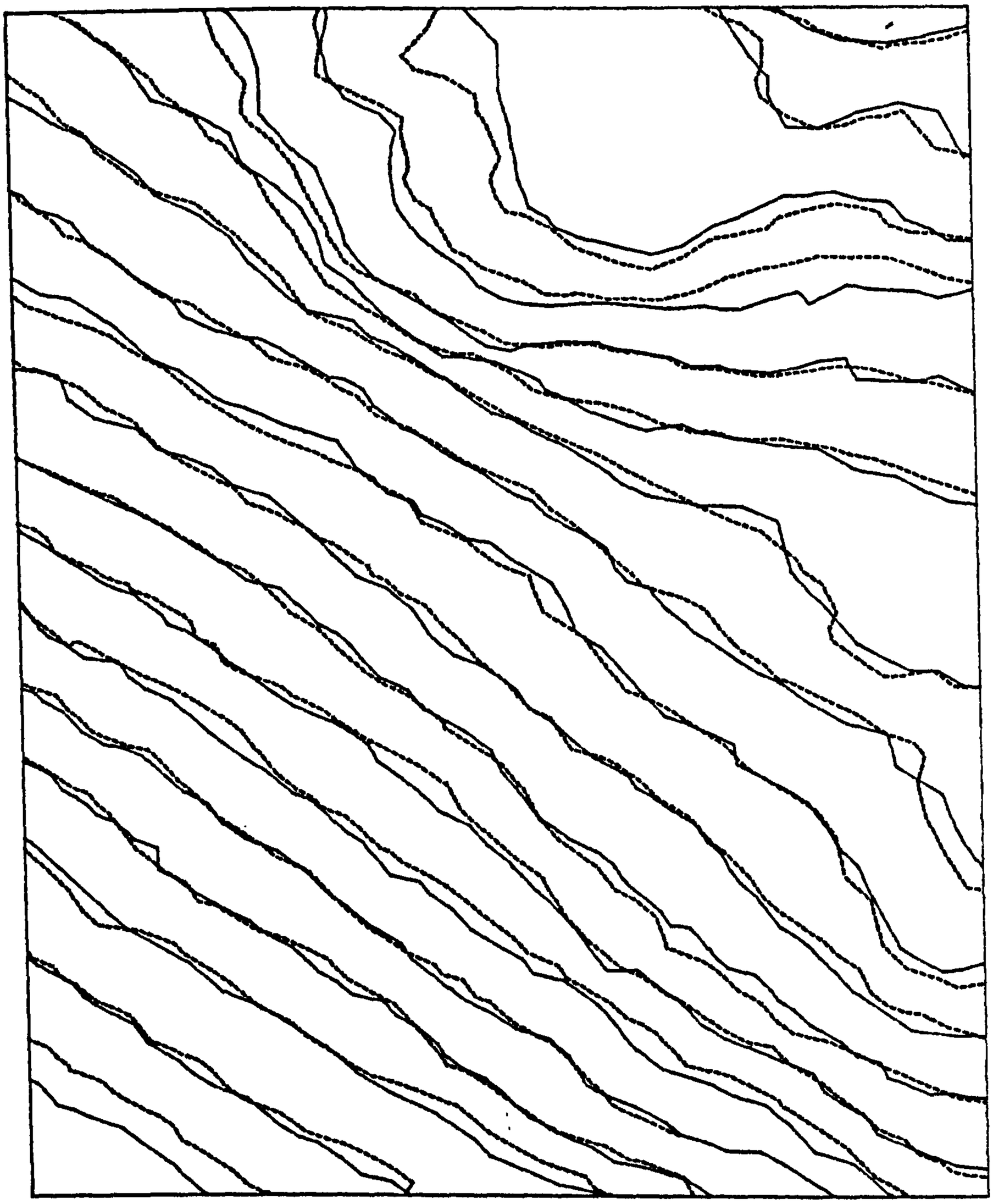


Fig.7.4 Superimposed contour lines:
dotted lines: contours generated using binary images.
solid lines: contours generated using non-binary images.

CHAPTER EIGHT

General Principle and State-of-the-Art of Digital Image Correlation

8.1 Introduction

One of the most fundamental tasks to be undertaken by a photogrammetric plotter operator is stereo-viewing in which he matches corresponding images in overlapping photographs. An alternative to this is automatic image matching (image correlation) which is a technique that recognizes conjugate image elements from multiple images of the same object by analogue (instrumental) or mathematical (computational) approaches. Image matching techniques may also be extended to cover the techniques by which targets (e.g. points, lines, or signalized targets) are detected and measured using a single image and a simulated target. Such techniques have been used widely in different fields such as computer vision, radar and communication, computer aided design (CAD), computer aided manufacturing (CAM), quality control, robot guidance, etc. In photogrammetry, image correlation has been devised with the main purpose of producing orthophotos or of extracting the three-dimensional information (x-parallaxes) from the object or objects (mainly the Earth's surface) which have been photographed with a view to forming a digital surface model or a digital terrain model (DTM). It has also been used for the automatic measurement of the positions of point targets e.g. automatic measurement of fiducial marks, signalized points and other kinds of image points.

The development of automatic image matching in photogrammetry can be traced back to the late fifties by the pioneering work of Hobrough (1959) who first automated the orientation movements and the height measurements of a Kelsh Plotter by incorporating an analogue image correlator. He followed this initial success by designing and building correlators for the Wild/Raytheon partnership which resulted in the Wild B8-Stereomat instrument which was first shown at the ISP Congress in Lisbon in 1964. This resulted in a production series of these instruments being built which were used very successfully by several national mapping agencies, e.g. the Division of National Mapping, Australia (Lambert, 1971), the French I.G.N. (Ducher, 1971), the American D.M.A., during the late 1960s and the 1970s. A similar but slightly later development was undertaken by the Zeiss Oberkochen/Itek partnership which resulted in the Planimat-EC5 instrument (Hardy and

Johnson, 1969). This, like the B8-Stereomat, saw its main application in orthophoto production, though height profiles were also measured and contours could be derived from these. All of these instruments combined an electronic hardware correlator with existing analogue stereo-plotting machines.

However, in the mid-1960s, a purpose-built combination of such a correlator and an analytical plotter, called the UNAMACE, was designed by Dr. Bertram (1963, 1965) and built by the Bunker-Ramo Corporation. The several instruments built to this design were used continuously over a 20 year period by the American D.M.A., again mainly to produce orthophotographs and derive contour plots. In the early 1970s, hardware correlators were also attached to existing analytical plotters as in the case of the OMI/Bendix AS-11-B-X. and at about the same time, Hobrough designed and constructed a series of Gestalt Photo Mapper (GPM) instruments, again purpose-built for orthophoto and contour production, which are still in production use today.

The relative success which was achieved by the above mentioned developments have encouraged photogrammetrists to examine other alternatives for image correlation, mainly software-based digital image correlation which would be much more flexible and less hardware dependent than the analogue correlation approach which had been used in the 1960s and 1970s. This has led to extensive investigations being carried out on both the theoretical and the experimental aspects of digital image correlation by photogrammetrists and computer scientists (Barnard and Thompson, 1980; Förstner, 1982; Ackermann, 1984; Rosenholm, 1986; Gruen and Baltsavias, 1986; Helava, 1988a; Li, 1989). In recent years, many new digital image correlation algorithms have been developed and are reported to have given sub-pixel accuracy for parallax (disparity) measurements (e.g. see Gruen and Baltsavias, 1986). On the hardware side, with the advances in computer technology in particular, image matching techniques are becoming computationally faster and therefore more applicable to a large number of problems in photogrammetry (Muller et al., 1988).

In spite of all these efforts and advances, up to the present time,

there has been no image correlation technique which can satisfy, in a real sense, the requirements for accuracy, efficiency, flexibility and reliability in photogrammetry as compared with a human operator. The human operator does not rely solely on the geometric dissimilarities between the two images (e.g. those due to x-parallax) but mainly on his ability to interpret the patterns in the areas to be matched as well as in the surrounding regions. Due to the high degree of complexity associated with the capability of the operator's eye-brain combination in pattern recognition -- in spite of some attempts made in recent years -- the present algorithms have by no means fully realized and duplicated the speed and pattern recognition strategies for similarity assessment associated with the human operator's capability to carry out image matching. Nevertheless, as mentioned above, in many circumstances, image correlation is an appropriate tool for the automatic measurement of point targets and DTM generation. Indeed in some cases, it can produce more accurate results than a human operator, e.g. in measuring signalized points (see Ackermann and Schneider, 1986). As will be demonstrated in Chapter 9, without using much computational sophistication, even a reasonably simple correlation method can, in some situations, generate a DTM comparable to those which use more sophisticated approaches.

The motivation behind implementing a correlation approach in this research work, is to investigate the potential power of the IBM 3090 computer with its Vector processor in handling correlation problems. This will be addressed in Chapter 10. In this chapter, the state-of-the-art in image correlation techniques is reviewed. The implementation of correlation techniques for the measurement of cross targets and DTM generation in the present project and the results of the accuracy estimation for both the measurement of cross targets and the generated DTM will all be presented and discussed in Chapter 9.

8.2 Image Correlation Classification

Image correlation may be classified according to different criteria. Makarovič (1980b), for example, classified image correlation techniques according to the basic ingredients of,

(i) **the input image data** (analogue, digital, hybrid),

(ii) **the processing equipment** which is available (electronic and optical correlators, digital computers), and

(iii) **the kind of operation** (on-line, off-line) being implemented.

In general, however, image matching may be classified conveniently into the two broad approaches of analogue and digital correlation techniques. Before considering digital image correlation, which is the main theme of this chapter, it seems pertinent to have a brief glance at analogue image correlation.

8.2.1 Analogue Image Correlation

In analogue correlation, the degree of correlation (similarities) between conjugate points is evaluated by analogue hardware means, i.e. using electronic or optical components. In **electronic image correlation**, the pictorial information from each of the two overlapping transparencies is converted to electrical signals using flying spot scanners. The two sets of electrical signals are then compared and processed in an electronic correlator to obtain image matching information. Fig. 8.1 shows a general schematic of an electronic correlation system. The flying spots of light are simultaneously generated on the face-plates of two CRTs. The transmitted signals through the left and right diapositives are collected by two photo multiplier tubes and converted to electrical signals. These electrical signals are then amplified, filtered and fed into the electronic correlator for the similarity assessment. As the spots scan the diapositives from left to right synchronously, the output signal from each of the two photo multiplier tubes may be displayed in a diagram as shown in Fig 8.2.

Because of the existence of x-parallaxes between the conjugate images there will be a time delay τ between two similar signals representing corresponding images. The degree of the similarity between the two signals is evaluated using the **cross-correlation coefficient** realized in analogue hardware form. The cross-correlation coefficient $\mathbb{C}_{fg}$ is given by:

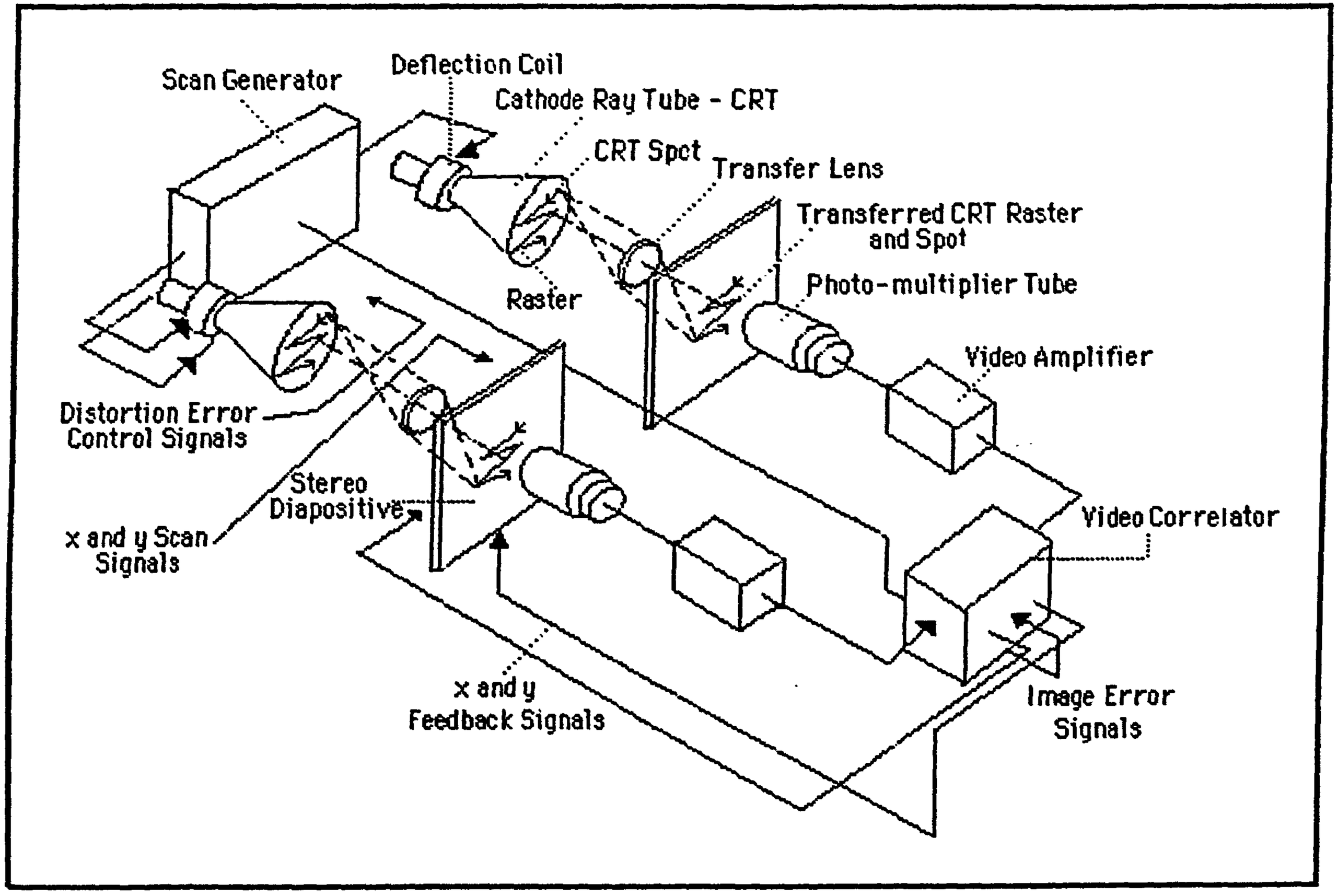


Fig. 8.1 The configuration and principal components of an analogue electrical image correlator used with an overlapping stereo-pair of photographs.

$$\mathbb{C}_{fg}(\tau) = \frac{1}{2T} \int_{-T}^{T} f(x)g(x+\tau)dx \qquad (8.1)$$

where f(x) and g(x) represent the two signals; and 2T corresponds to the area of the signals to be correlated. The presence of any difference in phase between the two signals, which corresponds to the time delay τ, activates the servo motors that introduce shifts to the diapositive carriers to clear it. When $\mathbb{C}_{fg}$ is a maximum, the value of the cleared time delay τ will then correspond to the x-parallax.

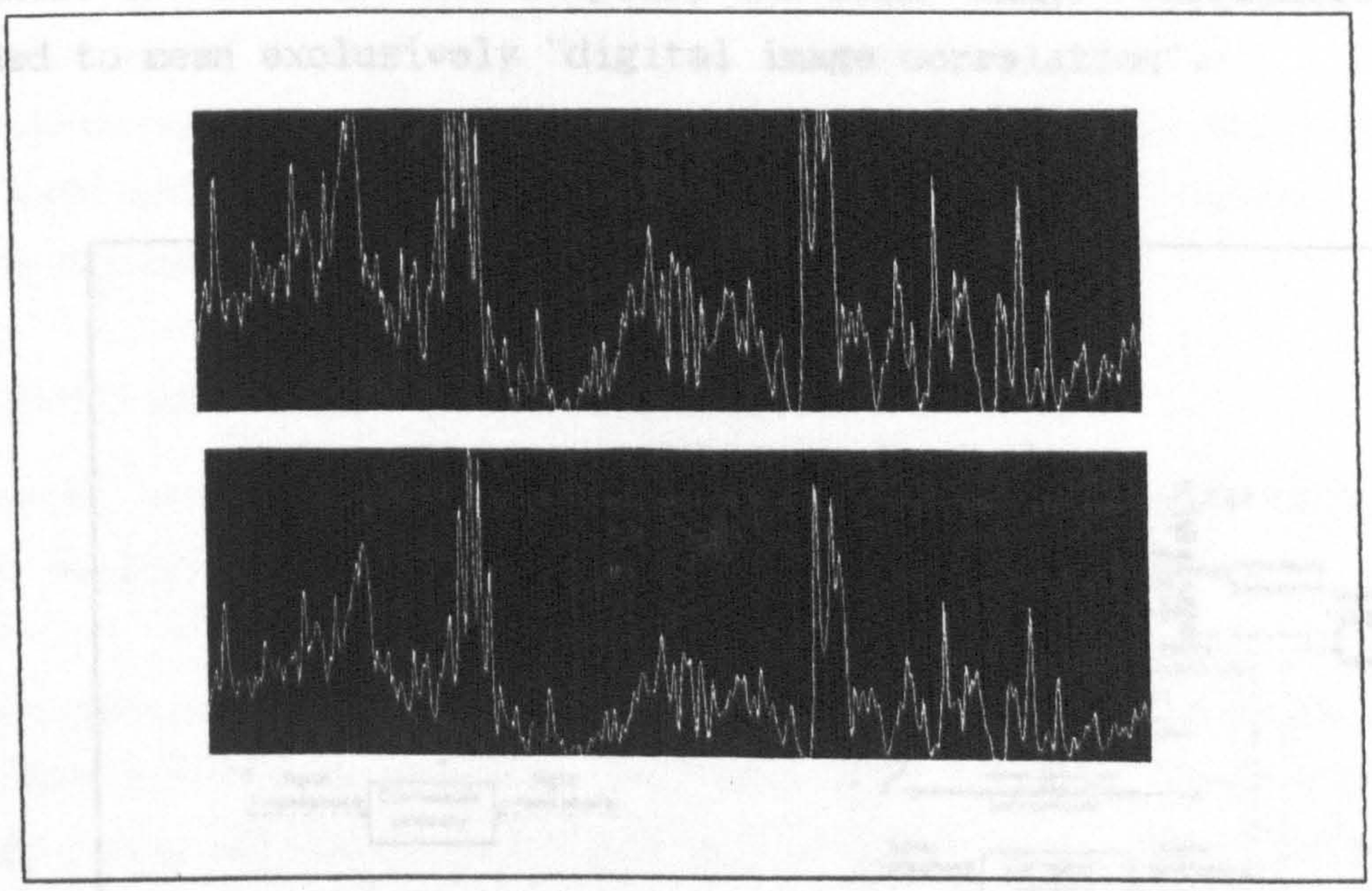


Fig. 8.2

The actual realization of the above mentioned concept in an optical projection instrument and in a mechanical projection instrument is given in Fig. 8.3.

Equation 8.1 may also be realized by **optical correlators**. The implementation of a coherent optical processor to accomplish signal detection for parallax measurement is carried out not by matching the component images directly but by matching the complex Fourier transform representations of each image. The Fourier spectrum is

generated optically using a laser beam and a suitable group of optical elements. Whitehead et al. (1968) demonstrated the first success in generating elevation profiles by optical correlation. However, as outlined by Konecny (1981), one of the draw-backs of optical correlators is the limited control capability which is available during the evaluation process. In particular, rectification within the correlation window is not possible. For that reason, optical correlators do not reach the performance required for height measurements. Due to the current lack of popularity of the analogue correlation approach, resulting from its cost, structural complexity and the lack of flexibility by comparison with the digital correlators, this category is not considered any further and throughout the rest of this chapter, the term "image correlation" is intended to mean exclusively "digital image correlation".

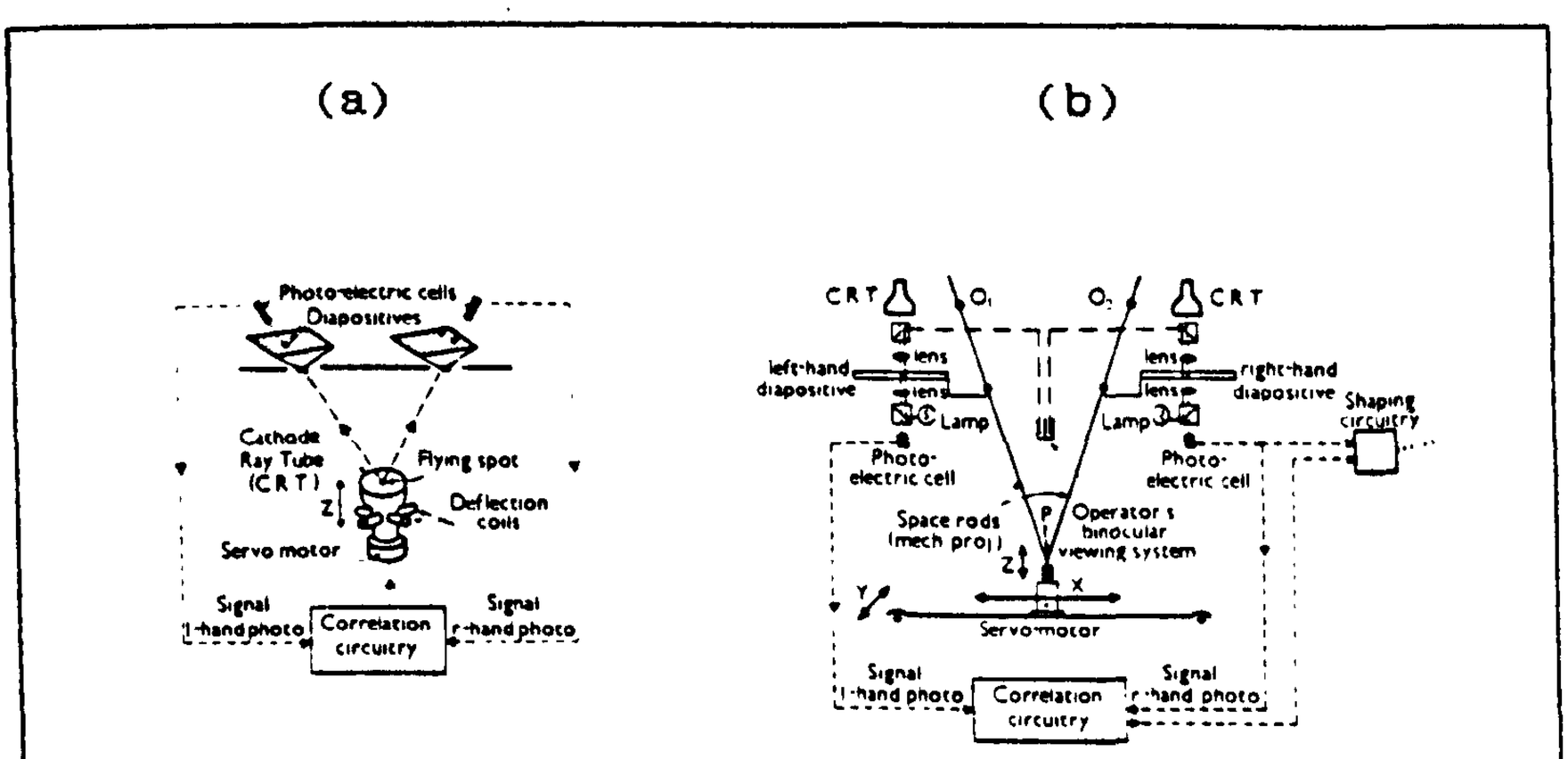


Fig.8.3 a- CORRELATION AS APPLIED IN AN OPTICAL PROJECTION INSTRUMENT.
b- CORRELATION AS APPLIED IN A MECHANICAL PROJECTION INSTRUMENT
(AFTER PETRIE, 1990)

8.2.2 Digital Image Correlation

In digital image correlation, the input data are in digital form and hence the algorithm for similarity assessment (Eq. 8.1) is realized by

a digital computer. The digital data could be either directly acquired (e.g. as in the case of Spot imagery) or analogue photos converted to digital form. Digital correlation can be performed in either of two modes of operation -- on-line or off-line. The former is used predominantly in analogue correlation, whereas in digital correlation, both are equally implemented. Digital **on-line correlation** almost invariably incorporates dedicated processors in an analytical or a digital stereo-plotter in order to have them function as an on-line correlation system, e.g. in the Kern DSR 11 analytical plotter and the Kern DSP1 digital stereo-photogrammetric plotter (see Chapter 6). On the other hand, the increased speed, storage capacity and flexibility of general-purpose digital computers as well as their decreased cost, has made **off-line correlation** quite feasible. Also this approach has the advantage of requiring a minimum of special hardware. In this research work, only the digital off-line correlation technique is being used and hence, as far as on-line correlation is concerned, no further discussion will be given.

8.3 Digital Image Correlation Fundamentals

In general terms, the procedures involved in digital image correlation may be outlined as follows:

A two-dimensional matrix of a certain row and column dimension (called a **window array**) is selected on the left photo. Then a larger corresponding matrix (**search array**) is selected on the right photo so that it will encompass the corresponding image of the window array. The window array is then systematically shifted within the search array and, in each position, the target is compared with that of equal-sized "sub-search" areas within the search array. The sub-search array having the best similarity is the area which best corresponds to the window array. For the next window on the left image, a corresponding search array is again found using the position of the previously matched point and the procedures are repeated until the desired area is covered.

From what has been mentioned above, it can be seen that image

correlation requires two distinct operations:

(i) an **algorithm** or algorithms to assess the similarities between two functions.

(ii) a **method** or strategy for moving from one matched point to the other.

Each of these operations will be considered in the following sections.

Before considering the image correlation algorithms and methods, it seems appropriate to mention briefly the **problems** which are nearly always present during the image matching operation. These are mainly:

(i) **Sub-pixel measurement**. Due to the fact that each pixel has a finite size, sub-pixel matching precision is required in most cases.

(ii) **Geometric complexities**. These may be caused by slopes or discontinuities in surfaces such as those produced by cliffs, breaklines, trees and buildings, causing occlusion in images (Fig. 8.4). There are also geometric displacements and affinities on the left and right images due to the camera axes' deviations from the vertical and various image distortions respectively.

(iii) **Radiometric variations**. As mentioned in Chapter 4, factors such as ground reflection variation with viewing angles; atmospheric refraction; radiometric response variation on different parts of the image; radiometric variations and noise imposed by the scanning microdensitometer; etc. may cause different density levels to be recorded for similar points.

(iv) **Image texture**. Those areas with low signal content, highly periodic texture and shadow can cause ambiguous matching.

The last three cases constitute a significant problem in a correlation-derived DTM by introducing mismatching and gross errors into the generated data. Later in this chapter, different ways of coping with these complexities will be discussed.

8.4 Image Correlation Algorithms

The similarity between two images f (the window array) and g (the

search array) can be measured in several ways (see Rosenfeld and Kak, 1976; Moik 1980, Dowman and Haggag, 1977; Lin, 1986). The most common approach towards similarity assessment is the calculation of the familiar **normalized cross-correlation function** ρ_{fg} which is given in two dimensions as:

$$\rho_{fg}(\tau,\eta) = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N} f(i,j)g(i+\tau,j+\eta)}{\left[\sum_{i=1}^{M}\sum_{j=1}^{N} f(i,j)^2 \sum_{i=1}^{M}\sum_{j=1}^{N} g(i+\tau,j+\eta)^2\right]^{1/2}} \qquad (8.2)$$

where τ and η are the shifting increments of the window array inside the search array in the y and x directions respectively; M,N represent the dimensions of the window array. The maximum value of ρ_{fg} corresponds to the conjugate left and right image points.

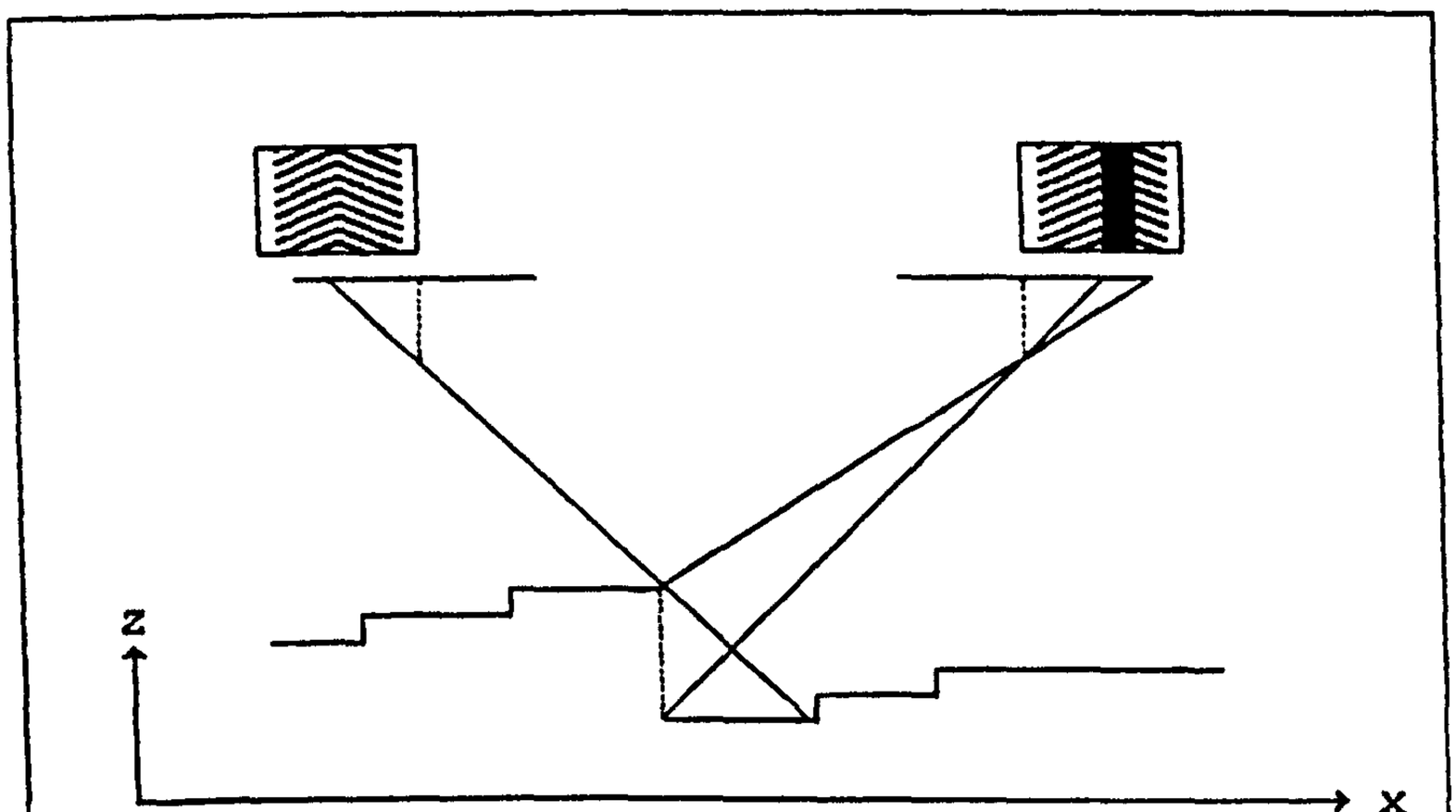


Fig. 8.4 OCCLUDED IMAGES

The number of computations in Eq. 8.2 for large values of f and g may be reduced by performing the operation in the frequency domain.

For similarity measurement, Eq. 8.2 may be replaced by some other functions such as the **absolute difference** A given by,

$$A(\tau,\eta)=\sum_{i=1}^{M}\sum_{j=1}^{N}\left|f(i,j)-g(i+\tau,j+\eta)\right| \qquad (8.3)$$

Eq. 8.3 gives minimum value at position of best match. Clearly Eq. 8.3 is more sensitive to noise and radiometric differences which are nearly always associated with the images. Nevertheless it is computationally much less expensive than the normalized cross-correlation function. For a detailed survey about the efficiency and reliability of the different correlation algorithms and the possibility of implementing multi-criterion algorithms, the reader is referred to Dowman and Haggag (1977) and Lin (1986).

8.5 Image Correlation Methods

During the past two decades, a large number of computational methods have been developed for digital image matching. The intention here is not to cover all of those methods but to classify them according to their main characteristics.

Broadly speaking, image correlation methods may be divided into two approaches of:

(i) systematic window shifting; and
(ii) least squares image matching.

8.5.1 Systematic Window Shifting

This approach computes a similarity assessment algorithm for each point by systematically shifting the window array within the search array and selects the best value as the correct point of correspondence. This method may be implemented according to one or other of two alternative schemes --those of image or object coordinates being the primary input.

(a) **Image coordinates primary**. According to this approach, a regular or irregular grid pattern is constructed on the

left image and the corresponding right image points are identified using systematic searching operations as discussed above. The object points are calculated subsequently. Using this approach, the search procedures may be carried out in one dimension, usually along epipolar lines, or in two dimensions. The former has been very popular since it was devised by Helava and by Masry in the early 1970, (Helava and Chapelle, 1972; Masry, 1974). This is due to the fact that the number of computations are reduced dramatically. The obvious disadvantage of the image space primary approach is the possible need for interpolation if regularly spaced object points need to be generated.

(b) **Object coordinates primary**. According to this approach, either a regular grid or or an irregular network of points is constructed in the object space and the required height for a given planimetric location is computed by upward projection of the object point using an estimation of the height value at the point. At this position, a number of small windows are defined mathematically at different elevation levels above and below the estimated height (Fig. 8.5). The windows at each level are projected back via the projection centres of the two photographs onto the image planes. The left and right windows from these two images for different height levels are then compared using a similarity assessment algorithm. The best match determines which height level corresponds to the known planimetric position. This approach also restricts the searching to one dimension, since the projected windows are displaced radially from the nadir point. This approach is usually referred to as the **Vertical Line Locus (VLL)** method. For more detail regarding this approach see Bethel (1986) and Benard et al. (1986).

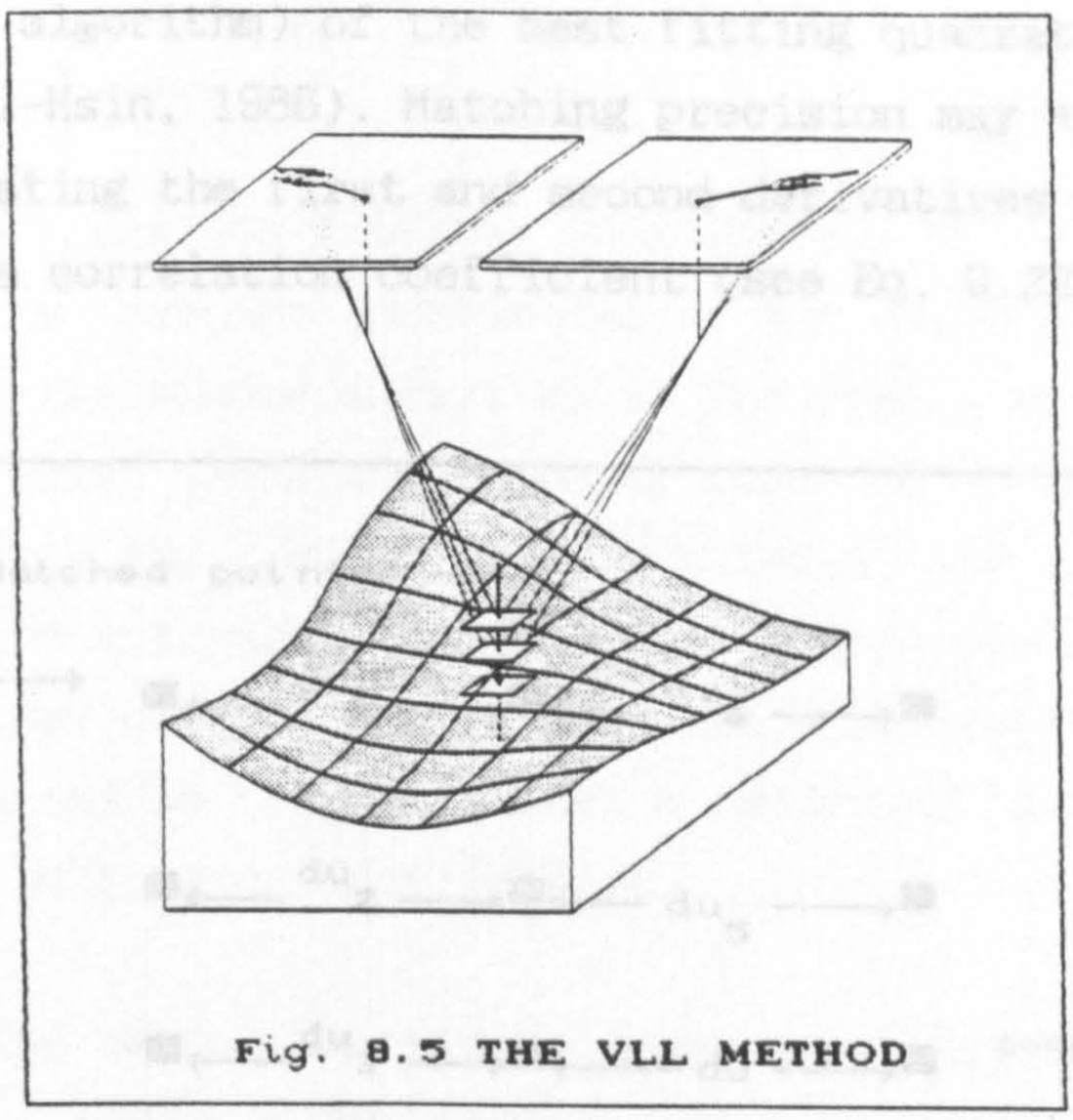


Fig. 8.5 THE VLL METHOD

The initial approximations for the corresponding points (in both the approaches mentioned above) are usually derived using the calculated values of the previously matched points. For example, Wong and Wei-Hsin (1986) use the following weighting scheme for predicting the position of the next match point along the epipolar line.

$$du = \frac{du_1+2du_2+3du_3+4du_4+5du_5}{15} \qquad (8.4)$$

where du is the predicted distance of the next match point from the previous matched point; and du_1 to du_5 are separations between neighbouring matched points (Fig. 8.6).

The initial value for the first point in the image is usually derived using a manual approach.

8.5.1.1 Matching Precision and Reliability Improvement

(i) Matching precision improvement. To improve the matching precision, a quadratic function may be fitted to the best value of the correlation algorithm being used and its neighbouring values. The final coordinate of the matched point may then be computed as the

point of the minimum or maximum value (depending on the type of correlation algorithm) of the best fitting quadratic function (see Wong and Wei-Hsin, 1986). Matching precision may also be improved by incorporating the first and second derivatives at the maximum value of the correlation coefficient (see Eq. 9.22).

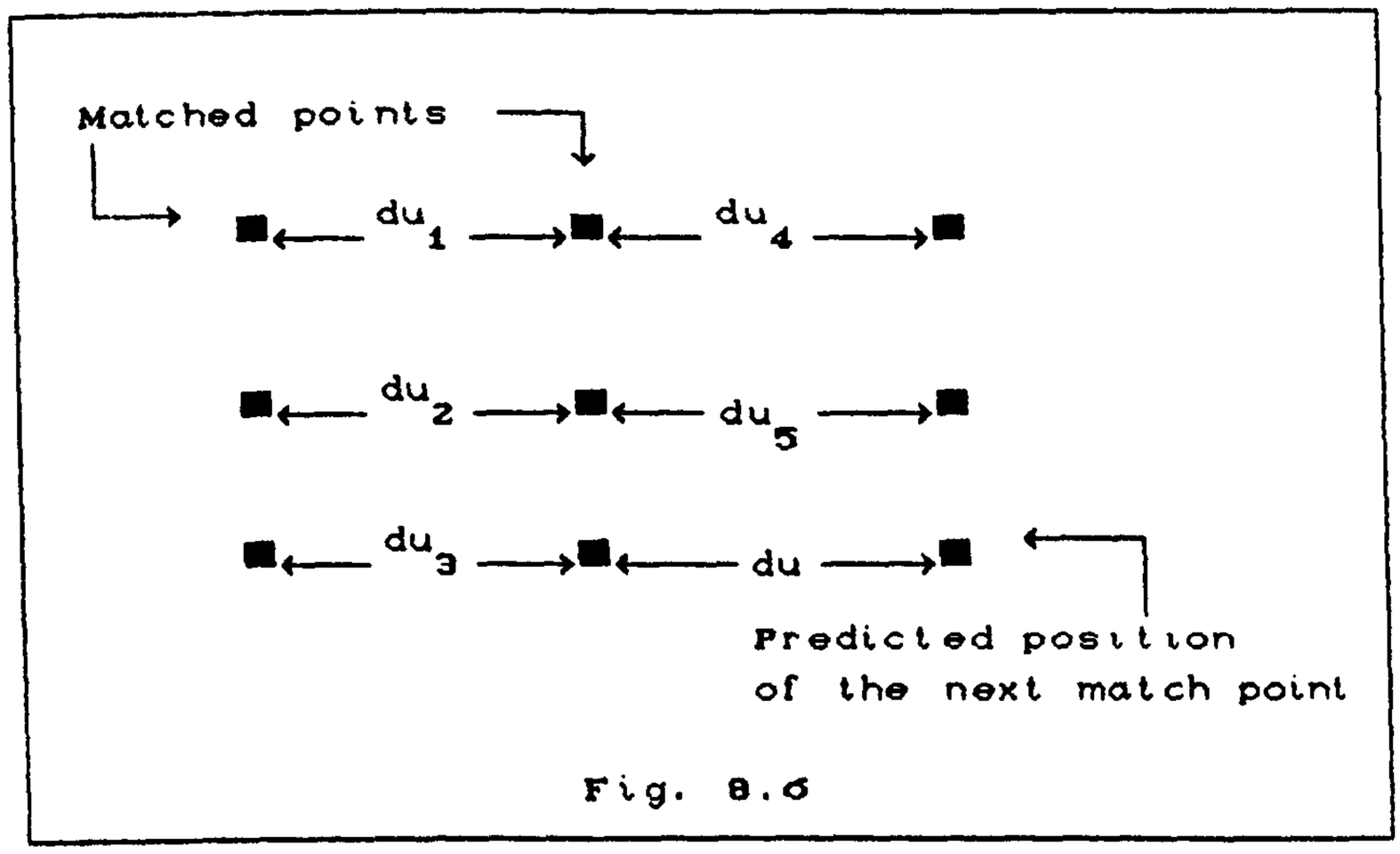


Fig. 8.6

(ii) Reliability improvement. As stated earlier in this chapter, geometric occlusion phenomena as well as the differences in scale and shape between the left window and the corresponding right subsearch array (due to factors such as relief and tilt displacements and other geometric distortions), makes the matching operation quite difficult, if not impossible, in some areas. In order to cope with these perspective differences, some corrective methods have been developed. One possible approach is to reshape (resample) one of the two images with respect to the other during the correlation process so that both the window and search arrays represent the same surface in the object space. The following simple pair of shaping equations has been suggested by Norvelle (1981) and used by Wong and Wei-Hsin (1986) to shape the window with respect to the subsearch array along an epipolar line (see also Panton, 1978):

$$\begin{cases} x_w = \dfrac{du}{dx}\ x \\ y_w = y \end{cases} \qquad (8.5)$$

where x and y are the original coordinates of a pixel in the window; x_w, y_w are the coordinates of the corresponding new pixel created for the window array; du is the predicted distance to the next match point along the epipolar line as computed from Eq. 8.4; and dx is the regular spacing between the centres of the adjacent window arrays. However, this method can only compensate for the linear part of the geometrical differences. The use of polynomials may be regarded as a solution for dealing with non-linear geometrical differences.

As far as the methods mentioned above are concerned, the matching problem for areas with shadow, or low and periodic signal contents is usually solved using different **window sizes**. The use of a small window size increases the mismatching possibility and hence reduces the reliability, while the use of a large window size affects the accuracy. Further discussion regarding the window optimization will be given in Chapter 9.

Low pass **filtering** may also be applied to the image prior to correlation operations. This blocks high frequency signals above certain levels and hence smooths out the noise. However, in practice, filtering does not always improve the correlation; indeed it may even make the correlation worse by removing important information. Filtering can also produce systematic error (Torlegård, 1986). Other restoration techniques (restoration filters), such as those discussed in Chapter 4, may also be used to improve the image quality and hence the correlation. In the author's view, further investigations should be carried out to determine more precisely the extent of geometric deterioration which may be imposed on the images by using filtering and restoration techniques.

8.5.2 Least Squares Image Matching Method

Image matching based on a least squares approach derives a parallax function and/or a shaping function for homologous image points by minimizing the grey value differences between the matching windows using the least squares technique. This requires the solution of a single set of equations to give the position of best fit rather than the computation of a similarity assessment algorithm at each point and the selection of the best value. The idea of the least squares matching technique was taken up at the Institute of Photogrammetry of the University of Stuttgart (see Förstner, 1982; Pertl, 1985; Ackermann, 1984). The basic idea (in the epipolar direction) may be described as follows:

Suppose that two homologous image windows f(x) and g(x) of a pair of overlapping photographs represent the same object. The assumption that only a linear geometric transformation needs to be carried out between the two windows results in the following equation:-

$$g(x) = f(x+x_p)+n(x) \qquad (8.6)$$

where x_p is an unknown shift and n(x) is the additive noise. Eq. 8.6 is non-linear with respect to the unknown shift x_p. A first order approximation of Eq. 8.6 yields:

$$g(x)+v(x) = f(x)+f'(x)dx_p \qquad (8.7)$$

where $f'(x)=\frac{df(x)}{dx}$ is the gradient function of f(x); and v(x)= -n(x). By rearranging Eq. 8.7,

$$\Delta g(x)+v(x) = f'(x)dx_p \qquad (8.8)$$

where $\Delta g(x)=g(x)-f(x)$. Eq. 8.8 may be written for all pixels of a window array. With N pixels in the window array, the normal equation -- which is highly redundant -- will be

$$B\ dx_p = C \qquad (8.9)$$

where

$$B = \sum_{i=1}^{N} f'^{2}(x_i) \quad \text{and} \quad C = \sum_{i=1}^{N} f'(x_i).\Delta g(x_i)$$

The solution of the normal equation 8.9 leads to

$$dx_p = \frac{\sum_{i=1}^{N} f'(x_i).\Delta g(x_i)}{\sum_{i=1}^{N} f'^2(x_i)} \qquad (8.10)$$

Eq 8.8 may be extended to include the modelling of the geometric differences (scale variations and shearing) and the radiometric differences. The final formulation which covers these additional parameters can be given as,

$$\Delta g(x_i)+v(x_i) = f'(x_i)dx_p + x_i f'(x_i)da + dh_o + f(x_i)dh_1 \qquad (8.11)$$

where dx_p and da represent a geometric shift and a geometric scale factor; and dh_o and dh_1 denote a radiometric shift and a radiometric scale factor respectively.

Since Eq. 8.11 is linearized, the solution has to be iterated. This means that, after each iteration, the grey level values have to be interpolated onto the transformed coordinate system using one of the resampling techniques described in Chapter 5.

The least squares approach is reported in several papers to be capable of calculating x-parallaxes for arbitrary image areas with sufficient image texture (as well as being able to carry out coordinate measurement for signalized targets) with a very high precision —better than 1/20th of a pixel (Ackermann, 1984; Ackermann and Schneider, 1986). In particular, Gruen and Baltsavias (1986, 1987) have reported a further improvement in reliability and accuracy by incorporating certain geometrical constraints (the collinearity condition) into the matching and also by utilizing a multi-photo matching approach which matches multiple images of the same object taken from different positions. Clearly the latter case is mainly applicable to non-topographic close-range photogrammetry.

8.5.2.1 The Problems Associated with The Least Squares Correlation Method

Since the least squares approach formulated above operates with linearized equations, two problems arise -- those concerning the initial values for the unknowns and the number of iterations to be employed in the computation.

The problem of the **initial value** is a serious bottleneck for all least squares based correlation techniques (see Thurgood and Mikhail 1982, Ackermann, 1984; Gruen and Baltsavias, 1986; Day and Muller, 1988). It has been reported (Ackermann, 1984) that for windows of 64*64 pixels, the initial overlap of homologous windows, when using the least squares approach formulated above, is about 75 percent in linear terms or more than 50 percent in area. This corresponds to a linear pull-in range (convergence radius) of about 1/4 window size. This means that a very close initial approximation is needed for convergency.

The second problem, which is related to the first one, is the number **of iterations**. In some cases, there may be up to ten iterations required for convergency (Ackermann, 1984). There are two possible remedies for this problem. One is to apply an initial low pass filtering (since low pass filtering smooths out the noise and hence speeds up the convergency[1]). However, in the author's view, this approach does not seem to be appropriate because of the unpredictable metric distortions which may be introduced by filters. The second remedy, which is preferable to the first one, is to use another correlation method to calculate the initial value for the unknown parameters. This may be a phase correlation approach (Section 8.5.4, also see Ackermann, 1984) or a feature-based correlation method (see Section 8.5.4, and also Day and Muller, 1988).

Regarding the problem of **geometric and radiometric differences**, the least squares approach seems to be more flexible than the window shifting approach, since it can incorporate the geometric and

1 *On the other hand, high-pass filtering may increase the matching accuracy by enhancing some information content (e.g. linear features) of the image.*

radiometric variations directly into the observation equations. Nevertheless, image matching in areas with low and periodic signal content or with numerous terrain complexities are problems which still remain unsolved. However, there is a new development in the least squares image matching method which seems to be capable of solving some of the problems quoted above and will be discussed in the section which follows.

8.5.3 Least Squares Multi-Point Matching Method

The multi-point matching method is one of the most recent developments in image correlation methodology and was initially suggested by Rosenholm (1986). The method is an extension of the single point least squares approach and uses basically the same principle formulated by Eq. 8.6, with the difference that the multi-point matching technique matches a group of points from the one image to the other image of the stereo pair simultaneously. This group of points usually forms a regular grid in the one image (the left image) and very likely will have an irregularly distributed set of corresponding points in the other image (Fig. 8.7).

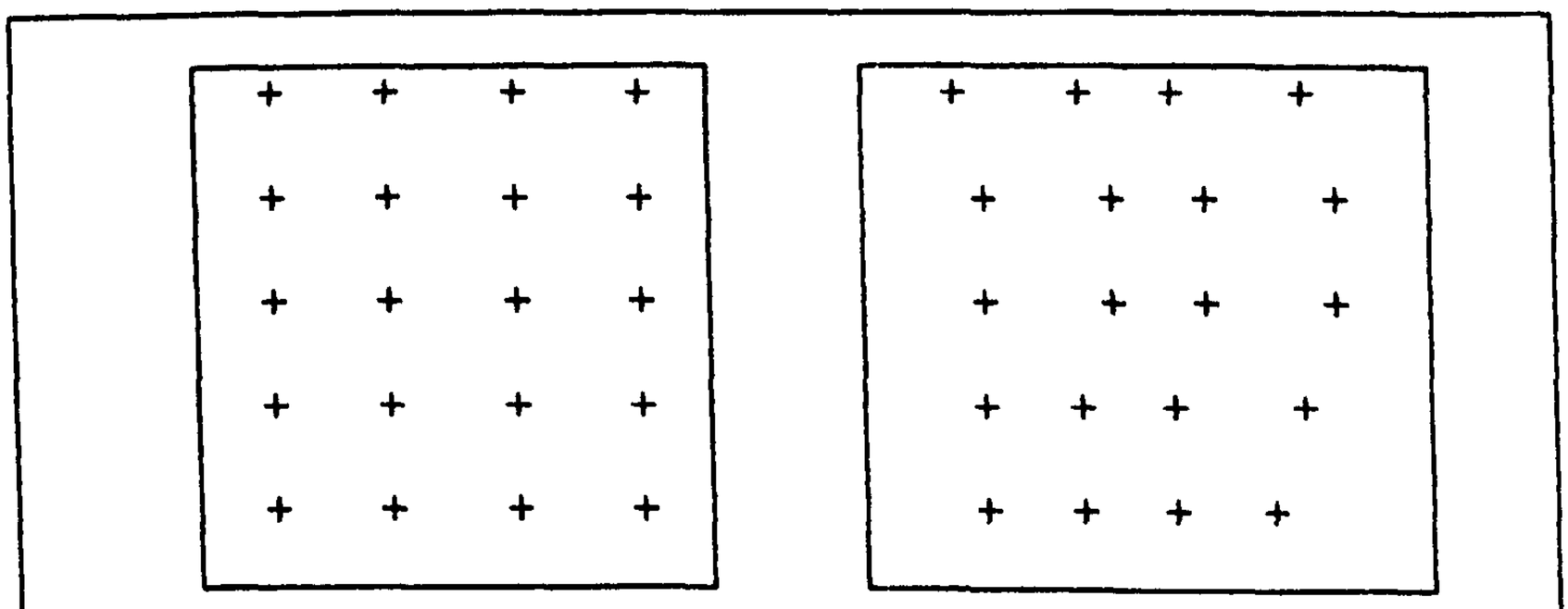


Fig. 8.7. THE LEFT IMAGE COMPRISES REGULAR GRID NODES AND THE RIGHT IS THE SEARCH IMAGE FOR WHICH THE CORRESPONDING POINTS ARE CALCULATED.

A brief description of the method may be given as follows:

Eq. 8.6 is written for each grid point and the unknown x-parallaxes, which are different for each grid node, are related to each other

using a bilinear interpolation of the parallaxes of four surrounding grid points. The correlation matches all the grid nodes at the same time. Therefore, in Eq. 8.9, instead of having a single unknown parallax, there will be a vector of unknown parallaxes corresponding to the grid points. If there are N by N pixels and M by M grid points, then there will be M by M unknowns and N by N equations. Since as with the single point least squares matching method, N is always much greater than M, a large number of redundancies exist.

The reliability of this method may be further improved by incorporating certain **constraints** in the equations; for example, smoothness constraints such as the second derivative constraints which minimize the curvature of the surface. Moreover, the terrain complexities may be detected and modelled on the object surface simultaneously with the image matching in order to improve the model fidelity and hence the correlation efficiency. (For more detail regarding the detection of the break lines in image matching, see Rosenholm, 1987 and Li, 1989).

The main advantage of the multi-point matching method is that the areas with low and periodic signal content may be matched, since it solves the equations simultaneously and hence bridges between those points located in the low texture area and their surrounding points which are located in higher texture areas. The main disadvantage of this method is the need for a large computer storage since the method involves the processing of large amount of data at the same time. This method is also similar to the least squares approach in that it suffers from the problem of requiring close initial values.

As a final note regarding the correlation methods discussed so far, it should be mentioned that they may also be implemented in a hierarchical approach, i.e. starting with a low image resolution and/or a coarse network of matching points and proceeding to fine image resolution and/or a fine network of matching points. The coarse matching stages produce some initial values for the points to be used in the fine matching phase and hence this may improve the computational efficiency by generating initial approximations for the

matching points.

8.5.4 Other Matching Methods

In addition to those image matching methods which have been presented in the preceding sections, there are some other image matching techniques, which seem to be less common than the previously mentioned methods. Image correlation using a **phase method** which measures x-parallaxes in the frequency domain, may be quoted as an example. This method has not been proved to be appropriate for precision matching and may only be used for coarse image matching, i.e. approximate parallax measurement (see Stokes (1986)).

Another image matching method which has not been discussed so far is **feature-based image matching** which matches some distinct features of one image to the corresponding features from the other image. This approach uses initially some kind of interest operator (see Luhmann and Altrogge , 1986) and edge detection techniques to detect the point-type features and other specific features such as edges, lines, etc. respectively (see Section 5.5.1). In the second step of the feature-based matching technique, a subset of the possible corresponding features from the second image, for each feature detected from the first image, is built up. This subset contains all the matchable features. In the last step, the matching is performed to get a consistent and unique solution. For an example of this approach, see Barnard and Thompson (1980). Clearly this approach cannot be used for generating a DTM since it ignores much of the terrain area for which elevation values are required.

Finally, there is another image matching technique which is based on the **dynamic programming approach**. The dynamic programming method was developed in the United States during the World War II. It was intended as a means of determining the most economical connection between two towns for troop displacements. The algorithm is essentially a search process which analyzes all possible connections between the two end points and calculates a so-called cost matrix for each possible connection. The most economical connection is then found by determining the optimal path using the derived cost matrices.

This method may be applied to image matching along epipolar lines as follows. A so-called **cost matrix** is computed which gives the grey value differences between the points on the left and the corresponding right epipolar lines in all possible combinations. This eventually leads to a two-dimensional cost array which is indexed by the points of the left and corresponding right epipolar lines. To each pair of left and right image points, a cost value is associated. These cost values may be connected to each other using different combinations (different paths). The optimal path, i.e. the one that corresponds to the minimum total cost, shows the corresponding matched points. The method is reported to give less accurate results for local areas, than the methods based on least squares and window shifting with cross correlation computation approaches. However, the dynamic programing image matching technique seems to be appropriate for the solution of global matching problems (Benard et. al, 1986; Kölbl and Silva, 1988).

8.6 Conclusion

In summary, as outlined briefly in the preceding sections, recent trends of research in image correlation show a tendency towards the implementation of the least squares matching method combined with other geometric or radiometric constraints and pattern recognition techniques. However, regarding the correlation problems outlined in Section 8.3, it does seem that only a partial solution has been achieved by those image matching techniques presented in the preceding sections and, as stated earlier in this chapter, at the present time, the reliability of image correlation methods has by no means matched that of a skilled and experienced human operator.

The image matching methods presented throughout this chapter may be used both for DTM generation (with the exception of feature-based matching) as well as for the measurement of targets. However, in the latter case, one of the targets is usually generated artificially. This will be discussed in detail in the next chapter where the implementation of image correlation for target measurement and DTM generation in this research work is presented.

CHAPTER NINE

Implementation of Automatic Image Correlation Techniques for Target Measurement and DTM Generation

9.1 Introduction

The theoretical aspects of image correlation have been covered in the preceding chapter. In this chapter, image correlation as implemented in this project is presented and the results are reported and discussed. In this research work, two different correlation algorithms and methods are used. The first approach utilizes an image modelling technique based on the least squares method which minimizes the grey value differences between the left and corresponding right points. This method has been implemented for the measurement of position of cross-shaped targets. The second approach employs a one-dimensional image correlation technique - that of systematic window shifting (see Section 8.5.1) - along epipolar lines to generate a DTM. The method uses the normalized cross-correlation coefficient algorithm for similarity assessment.

9.2 Image Modelling and Least Squares (IML) Matching Method for Cross-Type Target Measurement

As has been mentioned frequently throughout this report, one of the main problems associated with digital photogrammetry is the massive data storage requirements of digital photographs. In order to reduce the amount of this data, the photographic images should be digitized using as large a pixel size as possible (see Chapter 3), while still allowing sufficient accuracy to be obtained in the end-product. Clearly, a pre-requisite for having an accurate end-product, is to establish the inner, relative and absolute orientations with as high an accuracy as possible. The task of achieving an accurate inner orientation by devising a method for highly accurate measurements of the positions of the fiducial marks has already been covered in Chapter 5, where an edge detection approach for the automatic detection and measurement of the position of the cross-shaped targets was described. However, as is usually the case, a very high accuracy result (comparable to the results obtainable by, for example, a least squares correlation approach) cannot be achieved when a method based on the feature detection approach is employed (see for example Day and Muller, 1988).

As was shown in the preceding chapter, image matching based on least squares methods is one of the possible ways of achieving a very high degree of precision in image matching. On the other hand, as discussed in Section 8.5.2.1, least squares image matching methods suffer from the serious problem of having a small pull-in range. In view of this problem, it was decided to combine the target measurement based on the Gradient approach developed in Chapter 5 with the least squares matching method. This provided a good opportunity for the author to examine closely the least squares matching techniques which have been very popular in recent years. However, in this particular case, the method has been implemented only for target measurement.

As discussed in Chapter 5, the conventional matching methods for target measurement normally use a simulated target (referred to as a template) which "looks like" the actual target, i.e. the template has the same size and shape as the target. The best match between the template and the target is then found by moving the template over all possible positions of the target array (see Section 8.5.1) and noting the degrees of similarity between the template and the target in each position. These similarities are assessed using any of the similarity assessment algorithms given in Section 8.4. Wolf and Dewitt (1982) have discussed the possibility of achieving a further improvement in the measuring accuracy by adding another stage to the correlation process. This stage starts by rotating the template by small value increments or decrements and assessing the best match. The template is then translated in the x and y directions by a small amount and again the best match is found. This approach has been reported by Wolf and Dewitt to give sub-pixel accuracy in target measurement.

However, the main limitation of the above approach, in addition to the computational overhead, is the fact that the template cannot fully "look like" the picture unless the digitizer's point spread function is taken into consideration. The least squares image modelling approach implemented in this research work, seeks to compensate for this problem by using certain image processing techniques which will

be discussed later in this chapter. The method has been adopted and modified from Thurgood and Mikhail (1982) and is basically similar to the least squares approach discussed in Section 8.5.2. with the difference that one of the images is produced artificially. The least squares image modelling approach has also a unique characteristic in that it takes into consideration the imaging system's point spread function by modelling the artificially generated target to resemble more closely the digitized target using convolution operations. Therefore, throughout this chapter, this method is referred to as "**image modelling least squares method (IML)**" to be differentiated from the least squares image matching method discussed in Section 8.5.2.

The detailed treatment of the IML method is presented in the sections which follow. To simplify the problem, first of all, the measurement of a one-dimensional pulse is presented. The mathematical model is, then, extended to two-dimensional line and cross-shape targets respectively.

9.2.1 Measuring a One-Dimensional Rectangular Pulse

Consider the one-dimensional ideal rectangular pulse shown in Fig. 9.1. The problem is to find the position of the centre of the pulse with a high degree of accuracy.

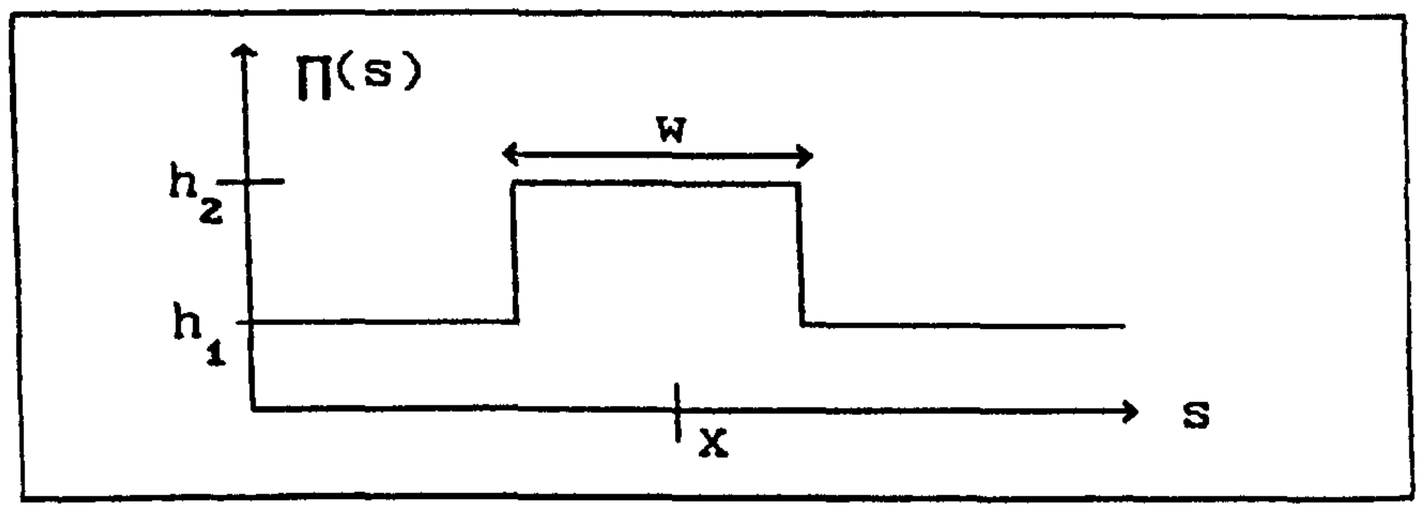


Fig. 9.1 A ONE-DIMENSIONAL PULSE FUNCTION

The ideal rectangular pulse $\Pi(s)$, of Fig. 9.1, can be expressed mathematically as,

$$\prod(s) = \begin{cases} h_2 & \text{for } x-\frac{W}{2} \le s \le x+\frac{W}{2} \\ h_1 & \text{for } s < x-\frac{W}{2} \text{ and } s > x+\frac{W}{2} \end{cases} \tag{9.1}$$

where x denotes the location of the centre of the pulse; w is the width of the pulse; and h_1 and h_2 are the background and pulse signal values respectively. Expression 9.1 may be rewritten as

$$\prod(s) = h_1+(h_2-h_1)\left[\mho(s-x+\frac{W}{2})-\mho(s-x-\frac{W}{2})\right] \tag{9.2}$$

where $\mho$ is the unit step function given by,

$$\begin{cases} \mho(s)= 1 \text{ for } s \ge 0 \\ \mho(s)= 0 \text{ for } s < 0 \end{cases} \tag{9.3}$$

Regarding Eq. 9.2, it can be seen that an ideal pulse function is defined by the four parameters of the background signal, the pulse signal, the signal width and the pulse centre (i.e. h_1, h_2, w, and x respectively). However, as it was demonstrated in Chapter 2, the ideal pulse will be degraded after digitization. Figs. 9.2e to 9.2h show the generation of a degraded pulse resulting from the blurring effect of the image digitizer's flying spot. Using the familiar convolution operation (see Section 2.4.3), it is possible to express mathematically the degraded pulse $\tilde{\tilde{\prod}}(s)$ as a function of the ideal pulse $\prod(s)$ convolved by a Gaussian function p(s) -- shown in Figs. 9.2a to 9.2d, i.e.,

$$\tilde{\tilde{\prod}}(s) = \prod(s) * p(s) \tag{9.4a}$$

or,(recalling Eq. 2.14)

$$\tilde{\tilde{\prod}}(s) = \int_{-\infty}^{\infty} \prod(\Psi)p(s-\Psi)d\Psi \tag{9.4b}$$

Substituting 9.2 in 9.4a yields,

$$\tilde{\tilde{\Pi}}(s) = \left\{ h_1 + (h_2 - h_1) \left[\mathcal{U}(s - x + \frac{W}{2}) - \mathcal{U}(s - x - \frac{W}{2}) \right] \right\} * p(s) \qquad (9.5)$$

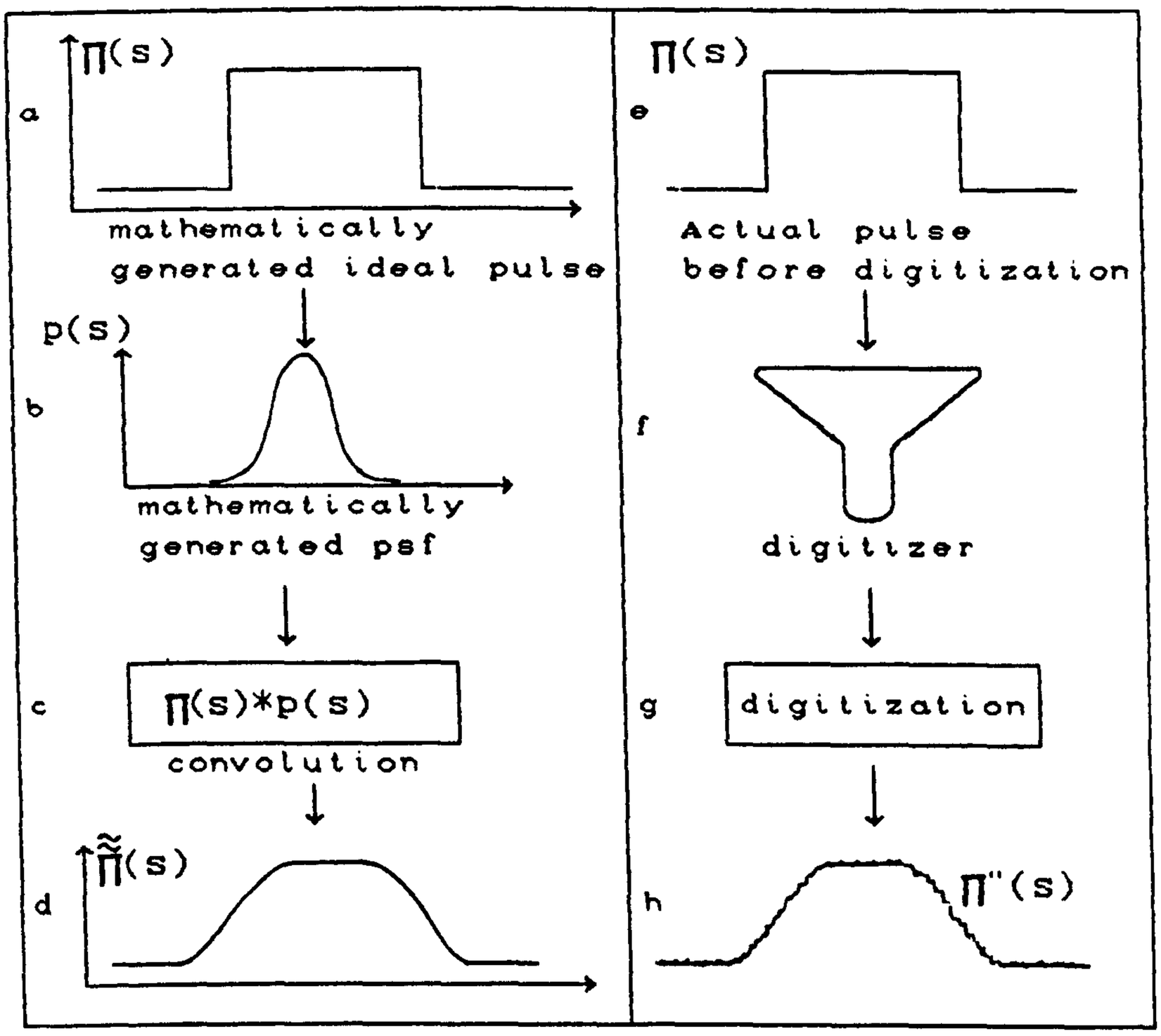


Fig. 9.2 MATHEMATICAL SIMULATION OF THE DIGITIZATION PROCESS.

Theoretically the resulting convolved function $\tilde{\tilde{\Pi}}(s)$ (i.e., the degraded pulse) should be very similar to the actual digitized pulse (compare Figs. 9.2d and 9.2h). In other words, if $\Pi''(s)$ represents the actual digitized pulse, then

$$\Pi''(s) - \tilde{\tilde{\Pi}}(s) = 0$$

However, in reality, due to various factors such as noise and imperfect knowledge about the digitizer's psf, there will be a residual difference v between the theoretically modelled pulse and the real digitized pulse. i.e.,

$$\Pi''(s) - \tilde{\tilde{\Pi}}(s) = v \qquad (9.6)$$

Eq. 9.6 may be thought of as an observation equation which expresses the real digitized pulse as a function of the ideal pulse convolved with the system psf or, using the familiar language of linear system theory, one can say that Eq. 9.6 expresses the relationship between the input and output from a linear shift-invariant (LSI) system (see Chapter 2).

Eq. 9.6 may be written for each pixel over the region of interest. Hence, there will be as many observations as the number of pixels. However, since Eq. 9.6 is non-linear, linearization with respect to the pulse parameters (i.e. h_1, h_2, w and x) is necessary prior to any attempt to ensure its solution. The following section deals with the linearization of Eq. 9.6 with respect to the unknown parameters of the pulse.

9.2.1.1 Linearization of the Observation Equations

Eq. 9.5 may be rearranged and written as,

$$\tilde{\tilde{\Pi}}(s) = \int_{-\infty}^{\infty} h_1 p(s-\Psi)d\Psi + \int_{x-w/2}^{x+w/2} (h_2-h_1)\, p(s-\Psi)d\Psi \qquad (9.7)$$

or

$$\tilde{\tilde{\Pi}}(s) = h_1 \int_{-\infty}^{\infty} p(s-\Psi)d\Psi + (h_2-h_1) \int_{x-w/2}^{x+w/2} p(s-\Psi)d\Psi \qquad (9.8)$$

since $\int_{-\infty}^{\infty} p(\Psi)\, d\Psi = 1.$, then

$$\tilde{\tilde{\Pi}}(s) = h_1 + (h_2-h_1) \int_{x-w/2}^{x+w/2} p(s-\Psi)d\Psi \qquad (9.9)$$

Now making a change of the variable $r = s - \Psi$, it follows that $dr = -d\Psi$ and hence, Eq. 9.9 can be rewritten as,

$$\tilde{\tilde{\Pi}}(s) = h_1+(h_2-h_1)\int_{s-x-w/2}^{s-x+w/2} p(r)dr \qquad (9.10)$$

Let $\mathbb{P}(r) = \int p(r)dr$, then

$$\tilde{\tilde{\Pi}}(s)= h_1+(h_2-h_1)\left[\mathbb{P}(s-x+w/2)-\mathbb{P}(s-x-w/2)\right] \qquad (9.11)$$

Substituting 9.11 in 9.6 and deriving the partial derivatives with respect to the unknown parameters of the pulse, yields the linearized form of Eq. 9.6. i.e.,

$$\Pi''(s)-\left\{\tilde{\tilde{\Pi}}(s)_o+\left[\frac{\partial\tilde{\tilde{\Pi}}(s)}{\partial h_1}\right]dh_1+\left[\frac{\partial\tilde{\tilde{\Pi}}(s)}{\partial h_2}\right]dh_2+\left[\frac{\partial\tilde{\tilde{\Pi}}(s)}{\partial w}\right]dw+\left[\frac{\partial\tilde{\tilde{\Pi}}(s)}{\partial x}\right]dx\right\}=v \qquad (9.12)$$

where dh_1, dh_2, dw and dx are corrections to initial values for the unknown parameters of the pulse; $\tilde{\tilde{\Pi}}(s)_o$ is evaluated using approximate values for the unknown parameters substituted in Eq. 9.11; and the partial derivatives are as follows,

$$\left[\frac{\partial\tilde{\tilde{\Pi}}(s)}{\partial h_1}\right] = -1+\left[\mathbb{P}(s-x+w/2)-\mathbb{P}(s-x-w/2)\right]$$

$$\left[\frac{\partial\tilde{\tilde{\Pi}}(s)}{\partial h_2}\right] = -\left[\mathbb{P}(s-x+w/2)-\mathbb{P}(s-x-w/2)\right]$$

$$\left[\frac{\partial\tilde{\tilde{\Pi}}(s)}{\partial w}\right] = -(h_2-h_1)\frac{\partial}{\partial w}\left[\mathbb{P}(s-x+w/2)-\mathbb{P}(s-x-w/2)\right]$$

$$= -1/2(h_2-h_1)\left[p(s-x+w/2)+p(s-x-w/2)\right]$$

$$\left[\frac{\partial\tilde{\tilde{\Pi}}(s)}{\partial x}\right] = -(h_2-h_1)\frac{\partial}{\partial x}\left[\mathbb{P}(s-x+w/2)-\mathbb{P}(s-x-w/2)\right]$$

$$= (h_2-h_1)\left[p(s-x+w/2)-p(s-x-w/2)\right]$$

Such a linearized equation (Eq. 9.12) can be written for each pixel over the region of interest. Thus, the system of the linear equations, will be given in matrix notation as,

$$G + G_o + AX = V \tag{9.13a}$$

where G contains the observations, i.e. the digitized density values of the pixels; G_o is evaluated by substituting the approximate values for the pulse parameters into Eq. 9.11; A is the coefficient matrix constructed from the partial derivatives of Eq. 9.11 with respect to the unknown parameters of the pulse; and X is the vector of corrections to the unknown parameters. Applying a least squares solution to Eq. 9.13a yields,

$$A^T W^{-1} A\, X = A^T W^{-1} f$$

$$\Rightarrow X = \left[A^T W^{-1} A\right]^{-1} A^T W^{-1} f \tag{9.13b}$$

where W is a weight matrix; and $f = -G - G_o$.

The solution of Eq. 9.13b yields values for the unknown pulse parameters. As can be seen, this approach determines the optimum matching solution by minimizing the grey level differences between the modelled pulse and the actual digitized pulse respectively.

9.2.2 Measurement of a Two-Dimensional Finite Line

Now consider an ideal two-dimensional finite line (Fig. 9.3). In the same way as was done for the one-dimensional pulse, the ideal target can be modelled mathematically using a certain number of parameters which define the target. For the finite line, these parameters are: the target and background density levels (h_2, h_1); the width and the length of the line (w, L); the angle of the orientation of the line with respect to the digitizer's coordinate system (θ); and the coordinates of the centre of the finite line (x, y).

Following the same procedures as were employed in the previous section, the ideal target can be convolved with the digitizer's psf.

However, in this case, there will be a two-dimensional psf p(s,t). Recalling Eq. 9.9,

$$g(s,t) = h_1+(h_2-h_1)\int_C^D \int_A^B p(s-\Psi, t-\tau)d\Psi\, d\tau \qquad (9.14)$$

where A, B, C and D are the limits of the integration and define the borders of the target in two dimensions.

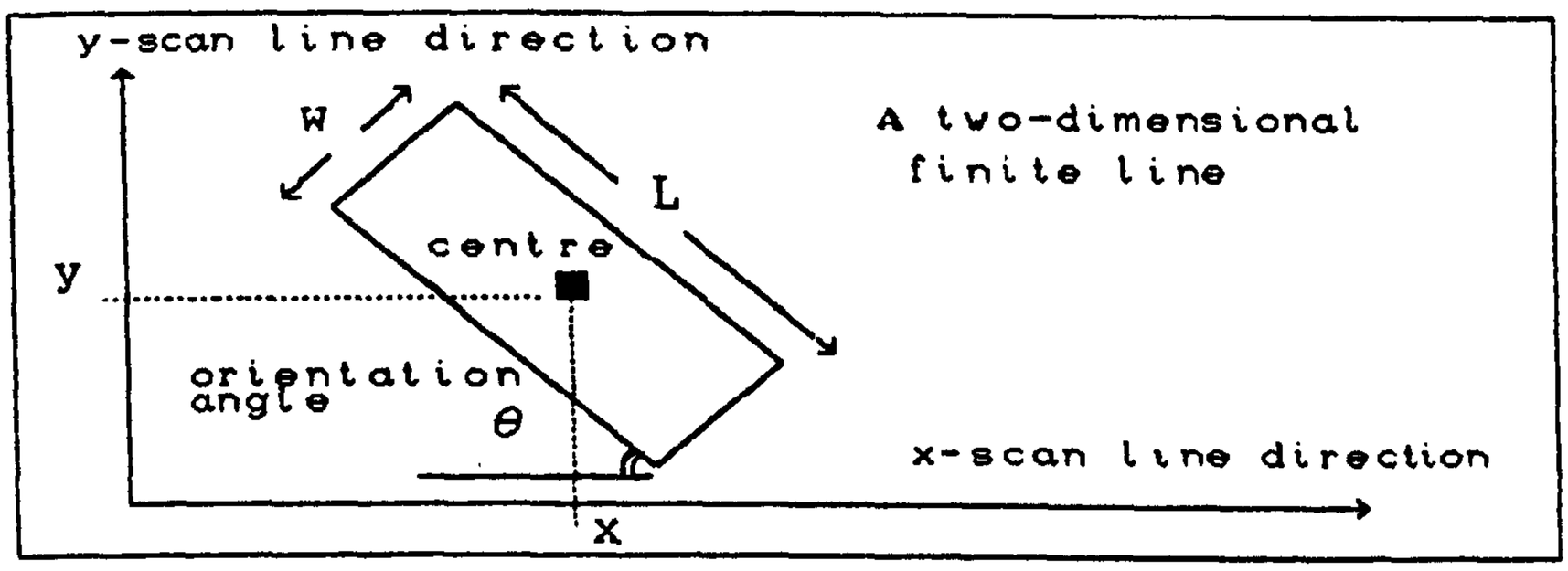


Fig. 9.3 A TWO-DIMENSIONAL FINITE LINE

To find the actual values for the integration limits (Eq. 9.14), the target may be rotated over its centre with an amount equal to θ. In this way, the integration limits are expressed in simpler form. After making the change of variables, $u=(s-\Psi)$ and $v=(t-\tau)$, Eq. 9.14 may be rewritten as

$$g(s,t) = h_1+(h_2-h_1)\int_{t'-L/2}^{t'+L/2} \int_{s'-w/2}^{s'+w/2} p(u,v)dudv \qquad (9.15)$$

where s' and t' are the coordinates of the centre of the target calculated using the following two-dimensional rotation:

$$\begin{bmatrix} s' \\ t' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} s-x \\ t-y \end{bmatrix}$$

If p(u,v) (Eq. 9.15) is separable into p(u) and p(v), then Eq. 9.15 may be written as,

$$g(s,t) = h_1+(h_2-h_1)\int_{t'-L/2}^{t'+L/2} p(u)\,du \int_{s'-w/2}^{s'+w/2} p(v)dv \qquad (9.16)$$

It follows that,

$$g(s,t) = h_1+(h_2-h_1)\left\{\left[\mathbb{P}(t'+L/2)-\mathbb{P}(t'-L/2)\right]\left[\mathbb{P}(s'+w/2)-\mathbb{P}(t'-w/2)\right]\right\} \qquad (9.17)$$

Eq. 9.17 is the general form of the non-linear observation equation which can be linearized by applying partial derivatives with respect to the unknowns h_1, h_2, w, L, θ, x and y, in a way similar to the one-dimensional pulse which was derived in the preceding section. The least squares solution then yields values for the unknowns, including the coordinates of the centre of the finite line.

9.2.3 Measurement of a Cross-Type Target

The relations derived in the last two sections, can now be applied to the real situation in photogrammetry where the targets to which measurements have to be made are usually cross-type fiducial marks or pre-marked cross-shaped control points. A fiducial cross is a combination of four finite line segments similar to the finite line segment presented in Fig. 9.3. The following parameters which define a cross need to be specified (see Fig. 9.4): the cross density and background density levels (h_2, h_1); the width and the length of the cross legs (w, L); the angle of the orientation of the cross with respect to the scan lines θ; and the coordinates of the centre of the cross (x, y).

Eq. 9.15 which was derived for a single two-dimensional line segment, can be written for each cross leg by considering the boundaries of each segment. This will eventually lead to,

$$g(s,t) = h_1+(h_2-h_1)\ .$$

$$\left\{ \int_{t'-w/2}^{t'+w/2}\int_{s'-L/2}^{s'-w/2} p(u,v)dudv + \int_{t'+w/2}^{t'+L/2}\int_{s'-w/2}^{s'+w/2} p(u,v)dudv \right.$$

$$\left. + \int_{t'-w/2}^{t'+w/2}\int_{s'+w/2}^{s'+L/2} p(u,v)dudv + \int_{t'-L/2}^{t'-w/2}\int_{s'-w/2}^{s'+w/2} p(u,v)dudv \right\}$$

(9.18)

Each double integration segment of the above equation represents a single two-dimensional line segment (i.e. the cross leg). Eq. 9.18 represents the general non-linear observation equation which may be used to measure the unknown parameters of the cross-type targets. It can be linearized in a way similar to the linearization of the one-dimensional pulse described in Section 9.2.1.1. Such a linearized equation can be written for each pixel within the area of interest to form a set of observation equations similar to Eq. 9.13a. The least squares solution (Eq. 9.13b) then gives the values for the unknown parameters including the x and y-coordinates of the cross centre.

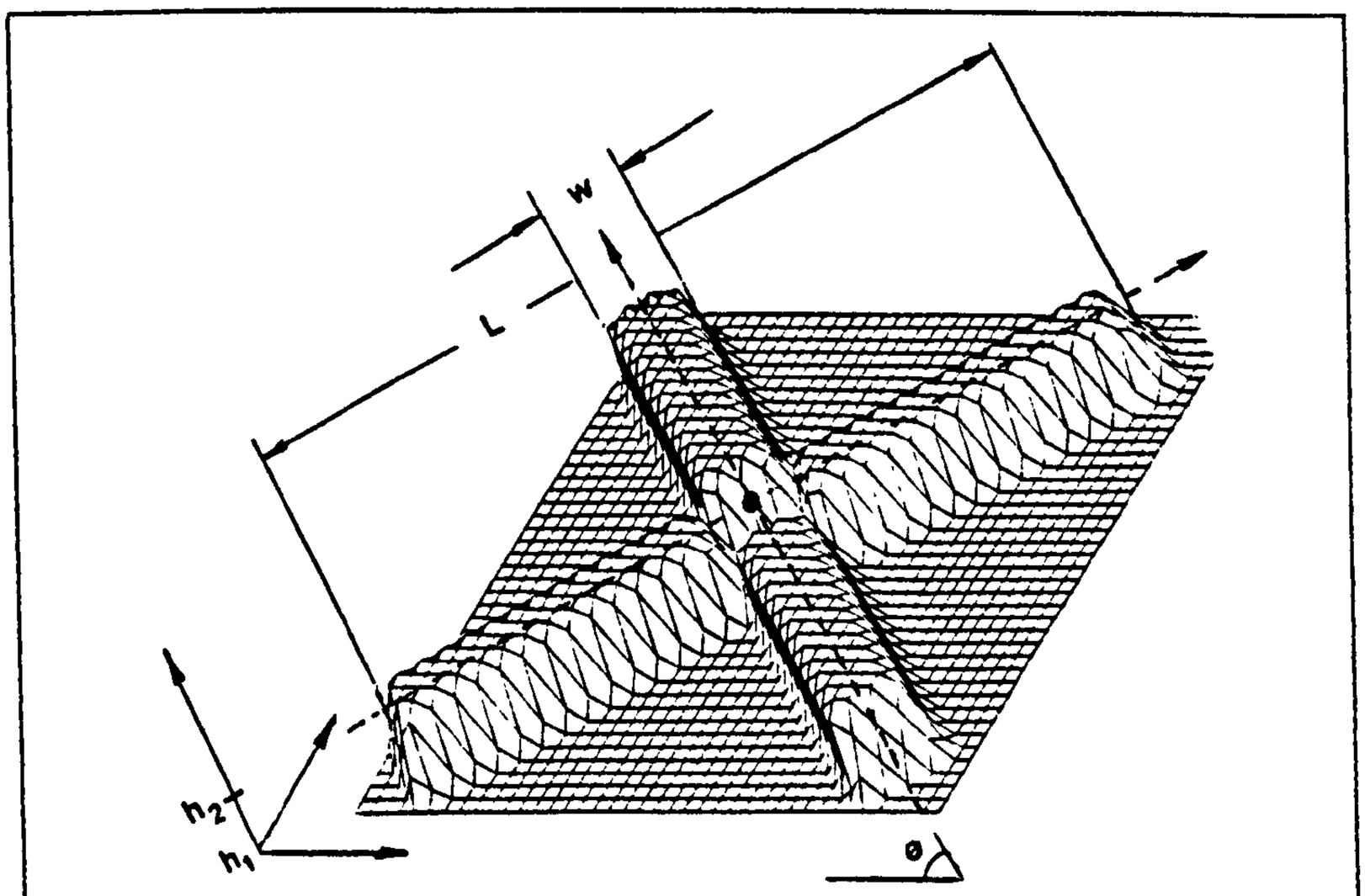


Fig. 9.4 A CROSS TARGET

Eq. 9.18 could be extended to cater for additional parameters, such as the width of the point spread function (psf), as unknowns. Moreover different kinds of psf (e.g., Gaussian, rectangular, etc) may be implemented in the equations. However, in this project, only the Gaussian psf which closely resembles the COSMOS psf has been implemented, and the unknown cross parameters are limited to: the coordinates of the cross centre (x, y); the cross and its background mean grey levels (h_2, h_1); and the cross-orientation angle (θ). The length and width of the legs of the crosses were determined manually. To determine the optimum size of the Gaussian psf (the distance between the two inflection points of the Gaussian spot), different values for the width of the Gaussian psf were input to the program. The best result was achieved for the Gaussian psf width of one pixel. This is in agreement with the width of the COSMOS scanning spot presented in the COSMOS manual.

As mentioned earlier in this chapter, the main body of the IML program used to determine the positions of each cross target was developed by Thurgood and Mikhail (1982) at Purdue University. However, the program ("PTSUB") was modified for this research work, to take into consideration the circular hole at the centre of the cross fiducial marks. This was achieved by the following modification.

A circular target may be defined by five parameters: background and target density levels (h_1, h_2); coordinates of the centre of the target (x,y); and finally the radius of the circle (R). Recalling the convolution procedures given in the previous sections, a blurred circle may be generated using the following expression:

$$g(s,t) = h_1+(h_2-h_1)\int_{t'-E}^{t'+E}\int_{s'-F}^{s'+F} p(u,v)dudv \qquad (9.19a)$$

where $E=(R^2-s'^2)^{1/2}$; and $F=(R^2-t'^2)^{1/2}$. If p(u,v) is separable into P(u) and p(v), then integrating Eq. 9.19a yields,

$$g(s,t) = h_1+(h_2-h_1)\left[\Phi(t'+E)-\Phi(t'-E)\right]\left[\Phi(s'+F)-\Phi(s'-F)\right] \quad (9.19b)$$

Equation 9.19a is added to Eq. 9.18 to give a cross with a central hole. In general, the program was modified to cater for three different cross targets -- these involving:

1. crosses with a circular hole at the centre (Fig. 9.5c);
2. crosses with a rectangular window at the centre (Fig. 9.5b); and
3. uniform crosses (Fig. 9.5a)

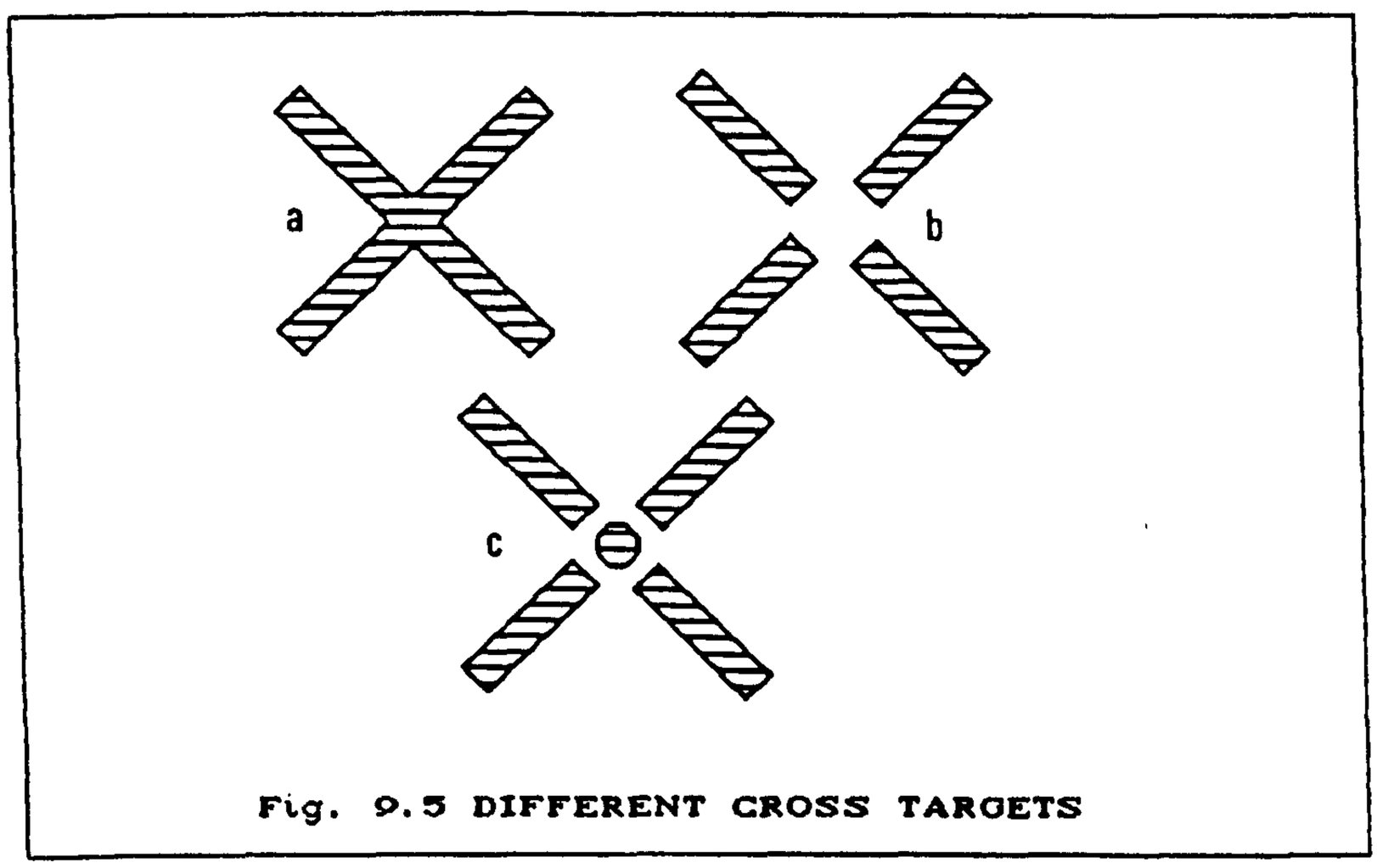


Fig. 9.5 DIFFERENT CROSS TARGETS

9.2.4 The IML Method's Pull-in Range (Convergence Radius)

Due to the high degree of non-linearity of the mathematical model (involving the observation equations, given in Eq. 9.18), quite expectedly, the use of these equations (Eq. 9.18) show a rather small pull-in range for convergency. Without having close approximations for the unknowns, the program started oscillating. In particular, it was found that the program is very sensitive to the initial value for the orientation angle of the cross. As mentioned before, this problem was overcome quite efficiently by using the cross parameters calculated by the Gradient method (program "GRAD", see Chapter 5) as the input approximations for the unknowns. Since the initial values were very

close to the actual ones, the program converged in one or, at the most, two iterations to give the final result. The listing of the program PTSUB is given in Appendix A1.

9.2.5 Test Results

To assess the performance of the IML method for the measurement of the position of cross targets, initially a set of crosses were simulated and their parameters were calculated. This was then followed by the automatic measurement of the real digitized fiducial crosses. The results and the accuracy figures for each case are presented in the following paragraphs.

(i) Test results for the simulated crosses

To produce crosses similar to those used as fiducial marks, different cross sizes and orientations were generated and convolved with a Gaussian psf. Different levels of uniformly distributed noise were then applied to the data (Figs. 9.6(A1), to 9.6(B9)). The centre of each cross and the other cross parameters were specified when the crosses were being generated. Therefore, the final accuracy assessments could be achieved simply by comparing the specified cross parameters with those calculated by the program.

The test results and the accuracy figures are given in Table 9.1. As the results show, the discrepancies between the known values and calculated values of the coordinates of the cross centres range from 0.01 to 0.07 of pixel size. The test results for the Gradient method (Chapter 5) are given in Table 9.2. As mentioned before, since the initial values for the cross parameters calculated by the Gradient method are very close to the true cross parameters, the least squares program converged with one or, at the most, two iterations to give the final result (Table 9.1).

(ii) Test results for the real digitized crosses

Following the interesting results obtained with the simulated

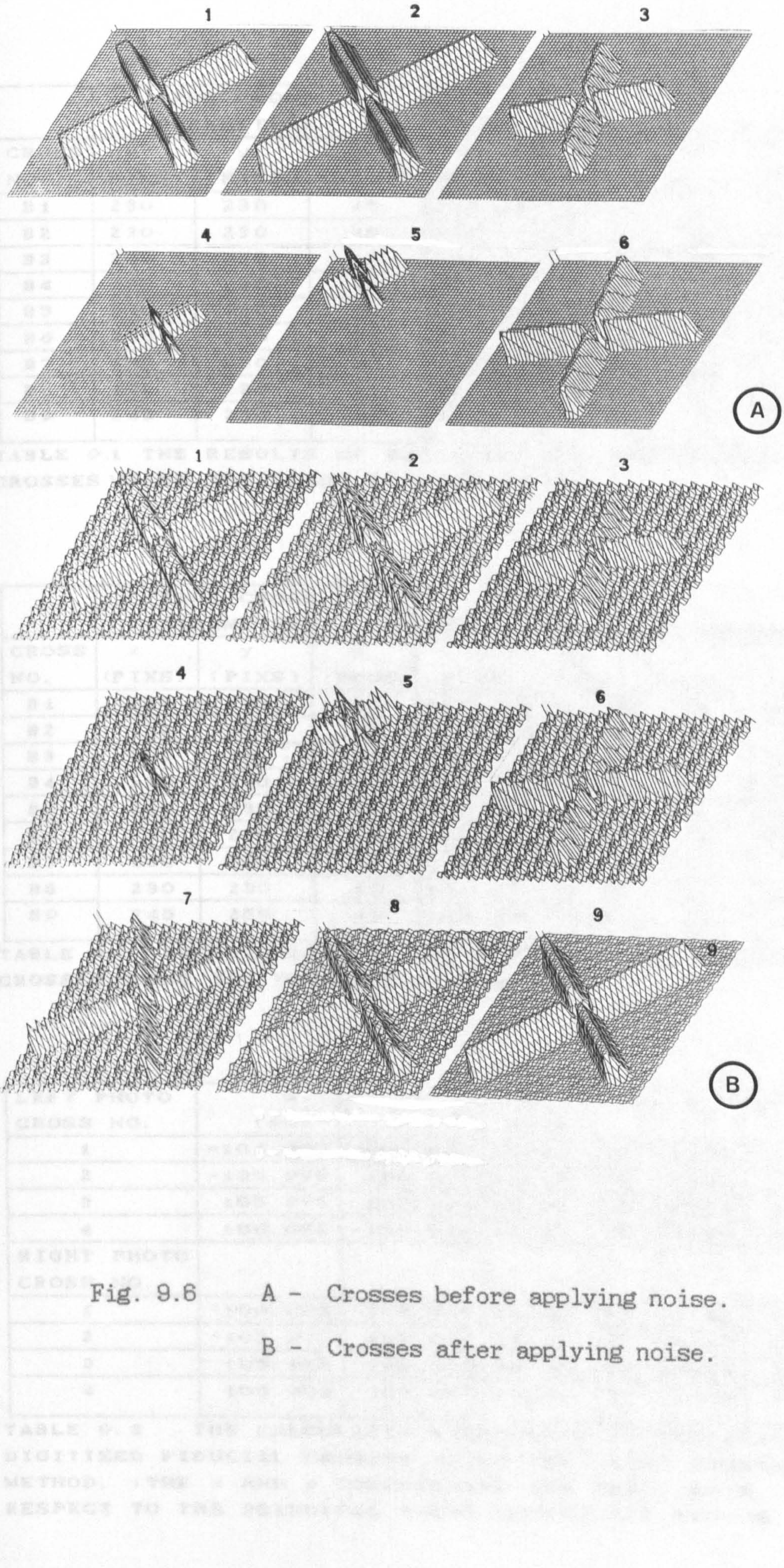


Fig. 9.6 A - Crosses before applying noise.

B - Crosses after applying noise.

	TRUE CROSS PARAMETERS			CALCULATED CROSS PARAMETERS					
CROSS NO.	x (PIXS)	y (PIXS)	θ (DEGS)	x (PIXS)	y (PIXS)	θ (DEGS)	H_1	H_2	IT
B1	230	230	45	229.99	229.99	45	14	133	2
B2	230	230	45	230.00	230.00	45	27	126	-
B3	230	230	45	230.00	229.99	45	18	137	2
B4	230	230	45	229.99	229.98	45	23	129	2
B5	230	230	60	230.01	229.98	60	23	90	1
B6	230	230	74	229.98	230.01	74	24	144	2
B7	230	230	80	230.03	229.96	80	24	143	2
B8	230	230	45	230.06	230.06	45	23	105	1
B9	245	250	45	245.03	250.07	45	23	132	2

TABLE 9.1 THE RESULTS OF THE TESTS FOR THE SIMULATED CROSSES USING THE LEAST SQUARES METHOD.

	TRUE CROSS PARAMETERS			CALCULATED CROSS PARAMETERS				
CROSS NO.	x (PIXS)	y (PIXS)	θ (DEGS)	x (PIXS)	y (PIXS)	θ (DEGS)	H_1	H_2
B1	230	230	45	229.94	229.38	45	18	120
B2	230	230	45	230.00	230.00	45	27	126
B3	230	230	45	230.07	229.89	45	22	124
B4	230	230	45	230.06	230.68	45	25	136
B5	230	230	60	230.03	229.98	60	25	130
B6	230	230	74	230.11	229.36	73	25	134
B7	230	230	80	229.77	230.26	77	25	133
B8	230	230	45	229.75	229.75	45	24	116
B9	245	250	45	245.00	250.87	45	24	120

TABLE 9.2 THE RESULTS OF THE TESTS FOR THE SIMULATED CROSSES USING THE GRADIENT METHOD.

LEFT PHOTO CROSS NO.	x (mm)	y (mm)	θ (DEGS)	H_1	H_2	IT.
1	-106.000	-106.008	44.86	40	5401	2
2	-105.978	106.011	44.86	63	5325	2
3	105.975	106.012	44.98	68	5520	3
4	106.002	-106.015	44.91	56	5268	2
RIGHT PHOTO CROSS NO.						
1	-106.002	-106.015	44.63	63	5781	2
2	-105.977	106.016	44.63	17	7140	2
3	105.975	106.006	44.75	81	6004	2
4	106.003	-106.007	44.63	83	5669	2

TABLE 9.3 THE CALCULATED PARAMETERS OF THE REAL DIGITIZED FIDUCIAL CROSSES USING THE LEAST SQUARES METHOD. (THE x AND y COORDINATES ARE GIVEN WITH RESPECT TO THE PRINCIPAL POINT COORDINATE SYSTEM.)

data, eight real digitized fiducial marks (four from each photograph) were input to the program. The calculated cross parameters using the least squares and Gradient methods are given in Tables 9.3 and 9.4 respectively.

To assess the accuracy of the calculated coordinates of the cross centres, the same sets of fiducial marks on the analogue diapositives were measured with the Zeiss Jena Stecometer stereocomparator. Each cross centre was measured several times (with a std. dev. of ±4 μm) and a mean value was adopted as the final value of the coordinates of each cross. To compare the two sets of the data, they were transformed into the principal point coordinate system using a conformal transformation. The discrepancies between the coordinates of the crosses measured by the Stecometer and the coordinates of the same crosses determined by the IML and the Gradient methods are given in Tables 9.5 and 9.6 respectively. As the results show, there are some discrepancies between the coordinates of the points measured automatically on the digital image and the coordinates of the same points measured by the Stecometer. These discrepancies could be due to the following reasons:

(a) Automatic methods (i.e. the Gradient and IML methods) define the centre of the cross by using the cross legs, while the cross centre when determined by a human operator does not necessarily coincide with the intersection of the lines fitted to the cross legs. In particular, for the images used for these tests, due to emulsion scratches and other reasons, the central hole of the cross, when viewed under a high magnification, did not appear as a perfect circle, so there were some ambiguities about the exact position of the centre of the cross circle.

(b) The second reason for the existence of the discrepancies between the points might be due to the systematic errors

LEFT PHOTO CROSS NO.	x (mm)	y (mm)	θ (DEGS)	H_1	H_2
1	-106.001	-106.006	44.88	69	6030
2	-105.981	106.008	45.28	80	5503
3	105.979	106.011	44.87	105	5573
4	106.003	-106.012	44.90	86	5466
RIGHT PHOTO CROSS NO.					
1	-106.000	-106.013	44.77	78	5931
2	-105.980	106.013	45.00	82	5311
3	105.980	106.007	44.93	95	6004
4	106.000	-106.007	44.99	77	5877

TABLE 9.4 THE CALCULATED PARAMETERS OF THE REAL DIGITIZED FIDUCIAL CROSSES USING THE GRADIENT METHOD. (THE x AND y COORDINATES ARE GIVEN WITH RESPECT TO THE PRINCIPAL POINT COORDINATE SYSTEM.)

	STECOMETER MEASUREMENTS		LEAST SQUARES METHOD		DISCREPANCIES	
LEFT PHOTO CROSS NO.	x (mm)	y (mm)	x (mm)	y (mm)	Δx (μm)	Δy (μm)
1	-105.997	-106.002	-106.000	-106.008	+3	+6
2	-105.986	106.012	-105.978	106.011	-8	-1
3	105.977	106.004	105.975	106.012	-2	+8
4	106.007	-106.014	106.002	-106.015	-5	+1
				RMSE	$\pm$5.0	$\pm$5.0
RIGHT PHOTO CROSS NO.						
1	-106.000	-106.002	-106.002	-106.015	+2	+13
2	-105.986	106.017	-105.977	106.016	-9	-1
3	105.972	105.996	105.975	106.006	+3	+10
4	106.015	-106.010	106.003	-106.007	-12	-3
				RMSE	$\pm$7.8	$\pm$8.4

TABLE 9.5 COMPARISON BETWEEN THE COORDINATES OF THE POINTS MEASURED BY THE STECOMETER AND THE SAME POINTS MEASURED BY THE LEAST SQUARES METHOD. (THE x AND y COORDINATES ARE GIVEN WITH RESPECT TO THE PRINCIPAL POINT COORDINATE SYSTEM).

	STECOMETER MEASUREMENTS		GRADIENT METHOD		DISCREP-ANCIES	
LEFT PHOTO CROSS NO.	x (mm)	y (mm)	x (mm)	y (mm)	Δx (μm)	Δy (μm)
1	-105.997	-106.002	-106.001	-106.006	+4	+4
2	-105.986	106.012	-105.981	106.008	-5	-4
3	105.977	106.004	105.979	106.011	+2	+7
4	106.007	-106.014	106.003	-106.012	-4	-2
				RMSE	±3.9	±4.6
RIGHT PHOTO CROSS NO.						
1	-106.000	-106.002	-106.000	-106.013	.0	11
2	-105.986	106.017	-105.980	106.013	-6	-4
3	105.972	105.996	105.980	106.007	+8	+11
4	106.015	-106.010	106.000	-106.007	-15	-3
				RMSE	±9.0	±8.2

TABLE 9.6 COMPARISON BETWEEN THE COORDINATES OF THE POINTS MEASURED BY THE STECOMETER AND THE SAME POINTS MEASURED BY THE GRADIENT METHOD. (THE x AND y COORDINATES ARE GIVEN WITH RESPECT TO THE PRINCIPAL POINT COORDINATE SYSTEM.

	STECOMETER MEASUREMENTS		CURSOR MEASUREMENTS		DISCREP-ANCIES	
LEFT PHOTO CROSS NO.	x (mm)	y (mm)	x (mm)	y (mm)	Δx (μm)	Δy (μm)
1	-105.997	-106.002	-105.998	-106.010	+1	+8
2	-105.986	106.012	-105.982	106.016	-4	+4
3	105.977	106.004	105.975	106.004	-2	0
4	106.007	-106.014	106.004	-106.011	-3	-3
				RMSE	±2.7	±4.7
RIGHT PHOTO CROSS NO.						
1	-106.000	-106.002	-105.998	-106.012	-2	+10
2	-105.986	106.017	-105.980	106.020	-6	+3
3	105.972	105.996	105.972	106.002	0	+6
4	106.015	-106.010	106.006	-106.010	-9	0
				RMSE	±5.5	±6.0

TABLE 9.7 COMPARISON BETWEEN THE COORDINATES OF THE POINTS MEASURED BY THE STECOMETER AND THE SAME POINTS MEASURED MANUALLY BY THE CURSOR ON THE DISPLAY SCREEN. (THE x AND y COORDINATES ARE GIVEN WITH RESPECT TO THE PRINCIPAL POINT COORDINATE SYSTEM).

introduced into the digital image because of the target shift within the pixel pattern discussed by Trinder (1987). This systematic shift becomes more significant as the pixel size approaches the size of the target. Trinder (1987) suggests the following simple formula for the magnitude of this systematic error:

RMSE in target position = 0.18 * pixel size

Trinder estimates an even larger error when the pixel size approaches that of the target. Regarding the pixel size of 32 μm for the digitized image, using the above formula, a systematic error with a magnitude of 6 μm may have been introduced to the image.

The results, when the cross positions were measured manually by a cursor on the display screen (see Chapter 6) and the comparison with the measurements of the crosses made using the Stecometer are also presented in Table 9.7.

9.3 Implementation of an Image Correlation Technique for Generating a DTM

In the previous sections, a method based on the image modelling and least squares approach for the measurement of cross-type targets was described. In this section, the implementation of an area-based image correlation method for the generation of a digital terrain model will be reported. The original intention was to adopt the IML approach and to modify it to produce an area-based correlation method. However, due to the time limitation for this research work, it was decided to adopt a more straight-forward solution, i.e., the systematic window shifting approach used with the image space primary solution (see Section 8.5.1). As was noticed during the course of the project, one of the advantages of the window shifting approach over the least squares one was the comparative simplicity of its implementation. To develop the correlation program, the author started from scratch and needed only a few weeks to develop the program. However, testing the program and assessing the accuracy of the generated DTM data took much longer than

the time spent developing the program itself.

The correlation technique which has been implemented is presented in two different sections; the first deals with the correlation algorithm and the second with the correlation method. This is then followed by a discussion of the correlation problems and the solutions which have been implemented. Finally an estimation of the accuracy of generated DTMs will be reported.

9.3.1 Implemented Correlation Algorithm

The extent of the similarity between the window array g and the search array f, was determined using the well known normalized cross-correlation function (Eq. 8.2). However, to compensate for differences in illumination, the mean grey values of the window array and the sub-search array are subtracted from the f and g values (Eq. 8.2). This yields,

$$\rho_{fg} = \frac{\sum\sum (f-\bar{f})(g-\bar{g})}{\left[\sum\sum (f-\bar{f})^2 \sum\sum (g-\bar{g})^2\right]^{1/2}} \qquad (9.20)$$

where $\bar{f}$ and $\bar{g}$ are the mean grey values of the window array and the sub-search array respectively. For computational convenience, Eq. 9.20 was rearranged as follows (see Wong and Wei-Hsin, (1986)):

$$\rho_{fg} = \frac{M\,N\sum\sum fg - \sum\sum f \sum\sum g}{\left\{MN\sum\sum f^2 - \left[\sum\sum f\right]^2\right\}^{1/2} \left\{MN\sum\sum g^2 - \left[\sum\sum g\right]^2\right\}^{1/2}} \qquad (9.21)$$

where M and N represent the number of rows and columns in the two arrays respectively. The value of ρ_{fg} ranges from +1 to -1. A value of +1 indicates exact similarity, while -1 represents reverse similarity.

9.3.2 Implemented Correlation Method

The correlation method which has been adopted is the one-dimensional systematic window shifting along epipolar lines. This, as mentioned

before, greatly accelerates the search operations. Moreover, this method reduces the possibility of producing gross errors because the search is restricted to only one dimension. One of the main characteristics of the implemented correlation method is that the correlation is achieved in two separate stages: rectifying and resampling the images to the epipolar lines (see Chapter 5) and then performing the correlation procedures. This means that the epipolar condition does not have to be solved simultaneously during the correlation stage.

To increase the image matching precision, the following relation formulated by Rosenholm (1985) was adopted:

$$dx = \frac{\rho(i+1,j)-\rho(i-1,j)}{4\rho(i,j)-2\rho(i+1,j)-2\rho(i-1,j)} \qquad (9.22)$$

where $\rho(i,j)$ is the observed maximum correlation coefficient. However, it should be mentioned that Eq. 9.22 can only improve the precision for the areas which have sharp density variations.

9.3.3 Implemented Techniques to Avoid Mismatching

To avoid mismatching, two approaches were considered, namely:

(i) a threshold value against which the maximum correlation coefficient was compared; and
(ii) optimization of the window array size.

1. **Threshold value selection.** One of the obvious remedies to avoid mismatching is the selection of an appropriate threshold value against which the maximum correlation coefficient is compared. Regarding the value of the threshold, it seems that each investigator tends to use his own criteria depending on the data he has learned from (Förstner, 1982). Regarding the data used in this research work, for the window size of 400 pixels, the use of a threshold value of 0.7 tended to give a better result. Those

correlation coefficients with values below the threshold value were rejected. However, even the selection of an appropriate threshold value may still allow the matching of non-identical points to take place if the density values of the points are similar.

2. **Window optimization.** The second remedy adopted to avoid mismatching in addition to the first approach, was to use an appropriate window size. As mentioned previously, the use of a small window size decreases the matching reliability while the use of a large window size reduces the accuracy. For window optimization, again an experimental approach was taken. Four different square windows with areas of 100 (10X10), 225 (15X15), 400 (20X20) and 625 (25X25) pixels respectively were selected. In each case, the DTM was generated and the accuracy was tested. The window size of 400 pixels provided the best result. The method of the accuracy estimation for the generated DTM points will be given in Section 9.3.5.1.

As it was discussed in the preceding chapter, in spite of the above mentioned remedies, it is not yet feasible for an image correlation technique to handle all the difficult situations arising during the correlation process without error. For this reason, post-processing methods have also been sought to detect and correct errors which occur in correlation process. A gross error detection algorithm which has been developed by my colleague Mr. Zhilin Li, (1990) was applied to a sample of the generated DTM data. Figs 9.7a and 9.7b show the contour lines generated before and after the removal of gross errors. The method involves comparing the height value of each point in the required area with the average height derived from the heights of its neighbours. This results in a set of residuals for the data points in the area. The mean and the standard deviation values are calculated for these residuals. If the calculated residual of any point in the area is larger than 3 X mean ± standard deviation, the point is then classified as a gross error (Li, 1990).

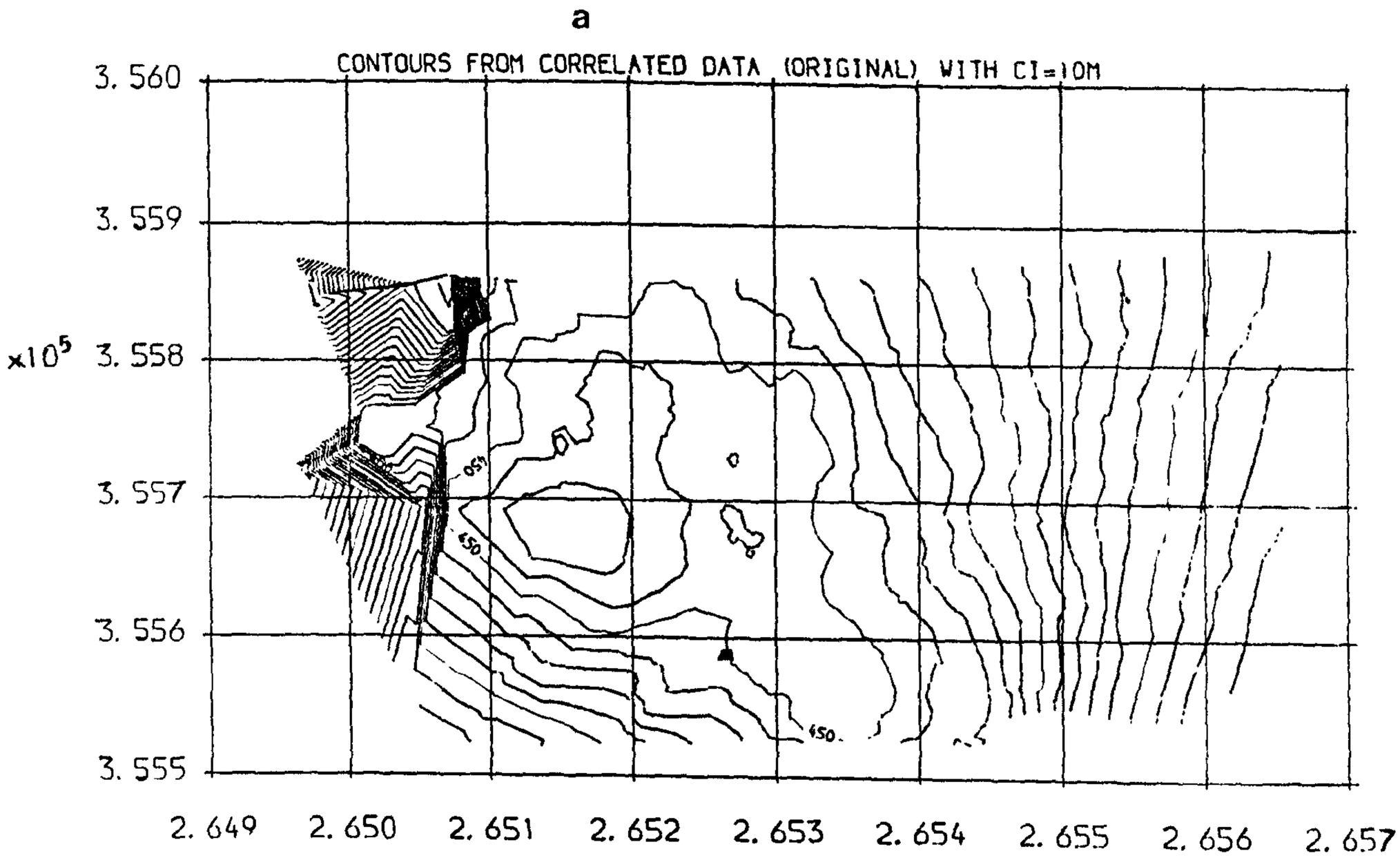


Fig. 9.7 a. Contour lines derived from the DTM data generated by image correlation (before gross error removal).
b. Contour lines after the removal of gross error.

9.3.4 Procedures of the Correlation Program

The program "CORRL" was developed to perform the correlation. The procedures for the digital matching of stereo-pairs as implemented in program CORRL can be divided into the following steps:

(a) The **window array** is shifted within the search array along the image rows (i.e. the epipolar lines) in steps of one pixel from border to border of the search array and Eq. 9.21 is computed for each position of the window array within the search array. The calculated **correlation coefficients** ρ_{fg} form a vector which is used in the next phase to determine the corresponding image points.

(b) The **maximum value** of ρ_{fg} in the vector is found in the position of column j and row i corresponding to the x and y image coordinates. If the value of ρ_{fg} is less than the threshold value, the point is rejected.

(c) To achieve sub-pixel image matching, a better estimation of the position of the **maximum correlation coefficient** is calculated using Eq. 9.22. The final x-coordinate of the point is, therefore: (x+dx).

(d) The **model coordinates** are generated using the coordinates of the left and right matched points by space intersection.

(e) The procedures are repeated for the next position of the window array. These procedures are continued until the **entire desired area** is covered.

(f) And finally, a three-dimensional conformal transformation is applied to the generated model points using the elements of the absolute orientation to generate the corresponding **terrain coordinates**.

The listing of the program CORRL is given in Appendix A8.

9.3.5 Test Results

The test images which were rectified in the previous stages (see Chapter 5), were used to automatically generate a digital terrain model. The size of these images are 2,000X2,000 pixels for each of the

left and right images. As before (see Chapter 6), the generated DTM data were transferred to the ICL 3980 mainframe machine on which the contour lines were generated using the contouring module of the PANACEA DTM package. The contour lines generated from an automatically derived DTM for a 400 pixels window array are given in Fig. 9.8.

9.3.5.1 The Accuracy Estimation of the Generated DTM

Different approaches can be used for the assessment of the geometric precision of the data generated by image correlation methods. The simplest approach is, perhaps, to consider the maximum value of the correlation coefficient ρ. The large value of ρ represents the higher signal-to-noise ratio and hence indicates a better match. However, Förstner (1982) shows that the precision of correlation also depends on the sharpness of the correlated object. This sharpness may be evaluated by using the second derivative at the maximum value of the correlation coefficient. Another well known approach for accuracy estimation for the generated data - provided that a least squares image matching method is implemented - is to use the quality criteria of the least squares solution, i.e. the standard error σ_o given by,

$$\hat{\sigma}_o^2 = \frac{\Sigma\, v^2}{N - u}$$

where v is the grey level difference between the left pixel and the corresponding right pixel; u is the number of unknowns, which is 1 for a single point least squares approach; and N is the number of pixels in the window array. From the standard error, an estimation of the translation error in the x-direction may be determined as, (see Förstner, 1982; Rosenholm, 1985)

$$\hat{\sigma}_x^2 = \frac{1}{N-u} \frac{\Sigma\, v^2}{\Sigma\, f'^2}$$

where $f' = d(f(x))/dx$.

However, all the approaches mentioned above give a rather indirect estimation of the accuracy and hence they have been rejected in this

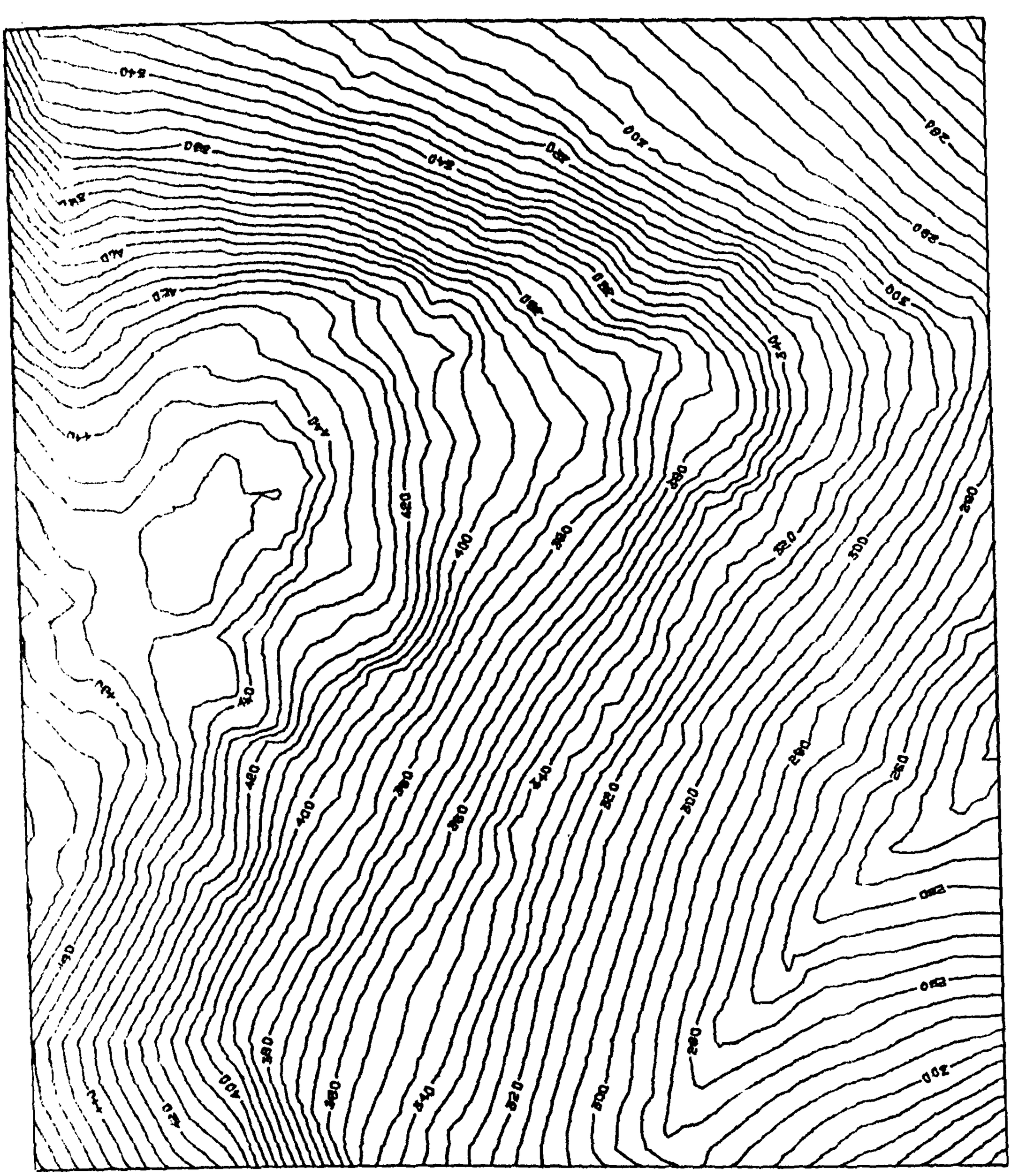


Fig 9.8 Contour lines derived from the DTM data generated by automatic image correlation.

research work. Instead a method has been adopted which is based on measuring a rather dense collect .on of check points using the original analogue images in their full resolution and a high precision stereocomparator. Thus, the same method as described in Chapter 6 for the accuracy estimation of the generated DSP data, was used for the assessment of the accuracy of the generated DTM points. Nearly 230 check points were measured by the Stecometer stereocomparator. The check points were transformed into the model coordinate system using independently generated orientation parameters. The distribution of the check points is given in Fig. (9.9). The comparison between the model coordinates of the check points and those of the DTM data were carried out as described in Section 6.8.

As mentioned earlier, different window sizes of 100, 225, 400 and 625 pixels respectively were used to decide on the optimum window size. The accuracy figures for each window are given in Table 9.8.

WINDOW SIZE (PIXEL)	MEAN RES. (m)	STD. DEV. (m)	RMSE (m)
10 * 10	- 3.88	± 14.45	± 15.08
15 * 15	- 0.67	± 1.10	± 1.29
20 * 20	- 0.19	± 1.02	± 1.04
25 * 25	- 0.10	± 1.15	± 1.15

Table 9.8

As the accuracy figures show, the window size of 20 X 20 (= 400 pixels) yields the best result with the relative height error of about 0.38 %.H (flying height). The accuracy figure for the generated data using this window size is slightly better than that of the DSP data (Chapter 6). The systematic errors for both the data sets generated by the correlation and DSP are almost equal. As before, this may be due

to the incomplete relative and absolute orientations.

The numerical values of the residual height discrepancies and their vector plots, for the window sizes of 100, 225, and 400 pixels are given in Figs. 9.9 to 9.14 respectively. The histograms of the residual height errors for the window sizes of 400 and 100 pixels are given in Figs. 9.15 and 9.16.

To obtain some estimations of the achieved accuracy in terms of pixel size on the image space, the procedures outlined in Section 6.8.1. were followed. This resulted in a pointing accuracy of about one pixel size on the image plane for the 20X20 pixels window array size. However, the accuracy figures for the area with high signal content approaches to half of the pixel size. This is almost the same result as was obtained by the DSP using a human operator. An accuracy improvement may be achieved if the image point measurements for the relative and absolute orientations are carried out with a higher accuracy. For example, an automatic image pointing method using image correlation may provide a more accurate image point measurements for the relative orientation.

The figures for the accuracy of the generated DTM data quoted above are lower than the results which one can usually find in the literature. The reason for this, in addition to the existence of systematic errors, may be that the accuracy estimation carried out in this project is quite different to those which use, for example, an indirect approach for the estimation of the accuracy.

9.4 Conclusion

In this chapter, the performance of two different approaches of image matching used for target measurement and for DTM generation respectively has been reported and discussed. The IML approach yielded a good result for the measurement of the digitized fiducial marks. In particular, it produced very high accuracy results for the simulated targets. Moreover, the convergency problem associated with the IML

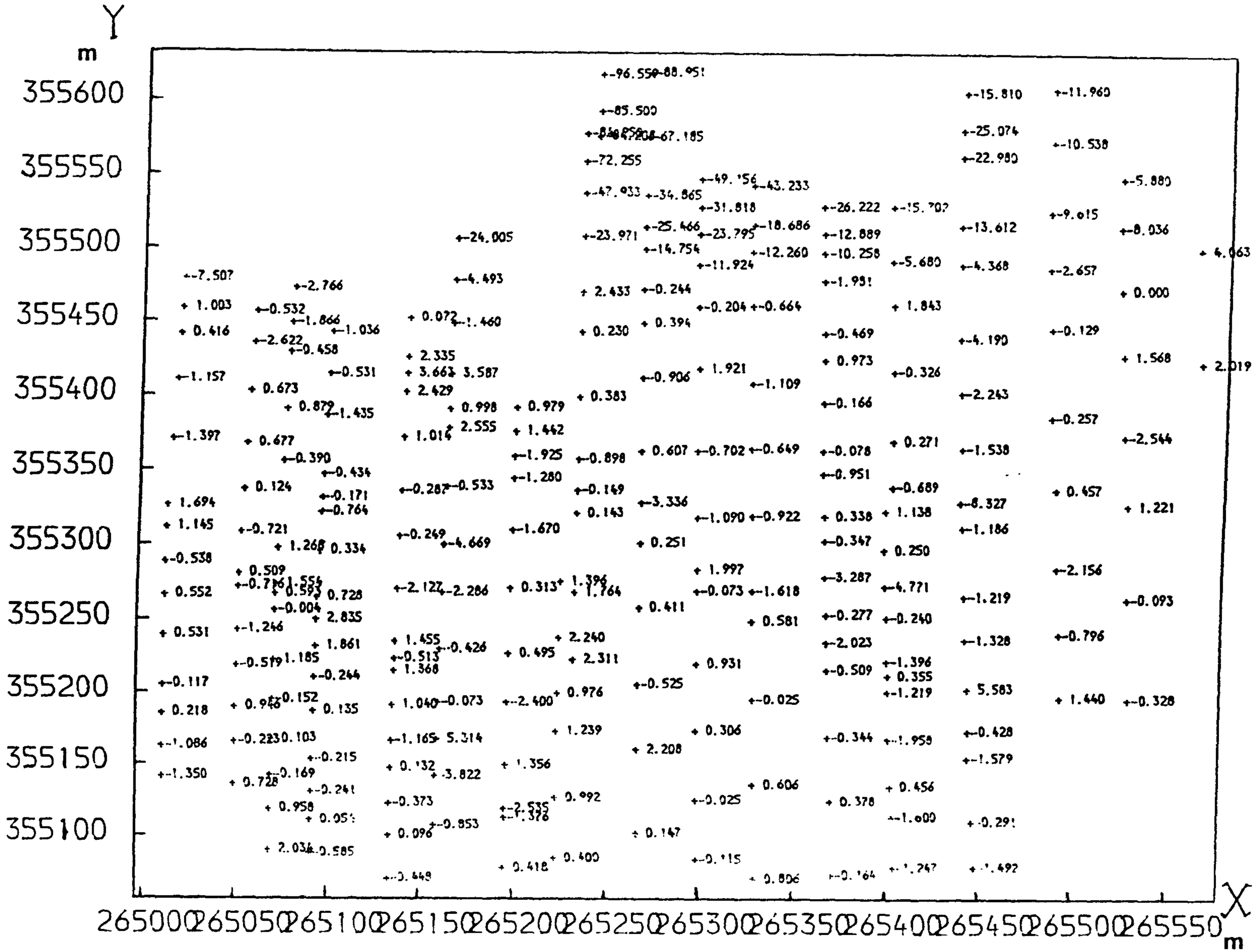


Fig. 9.9 The residual height errors (at the check points) for the DTM data generated by image correlation. (window size = 10x10 pixels)

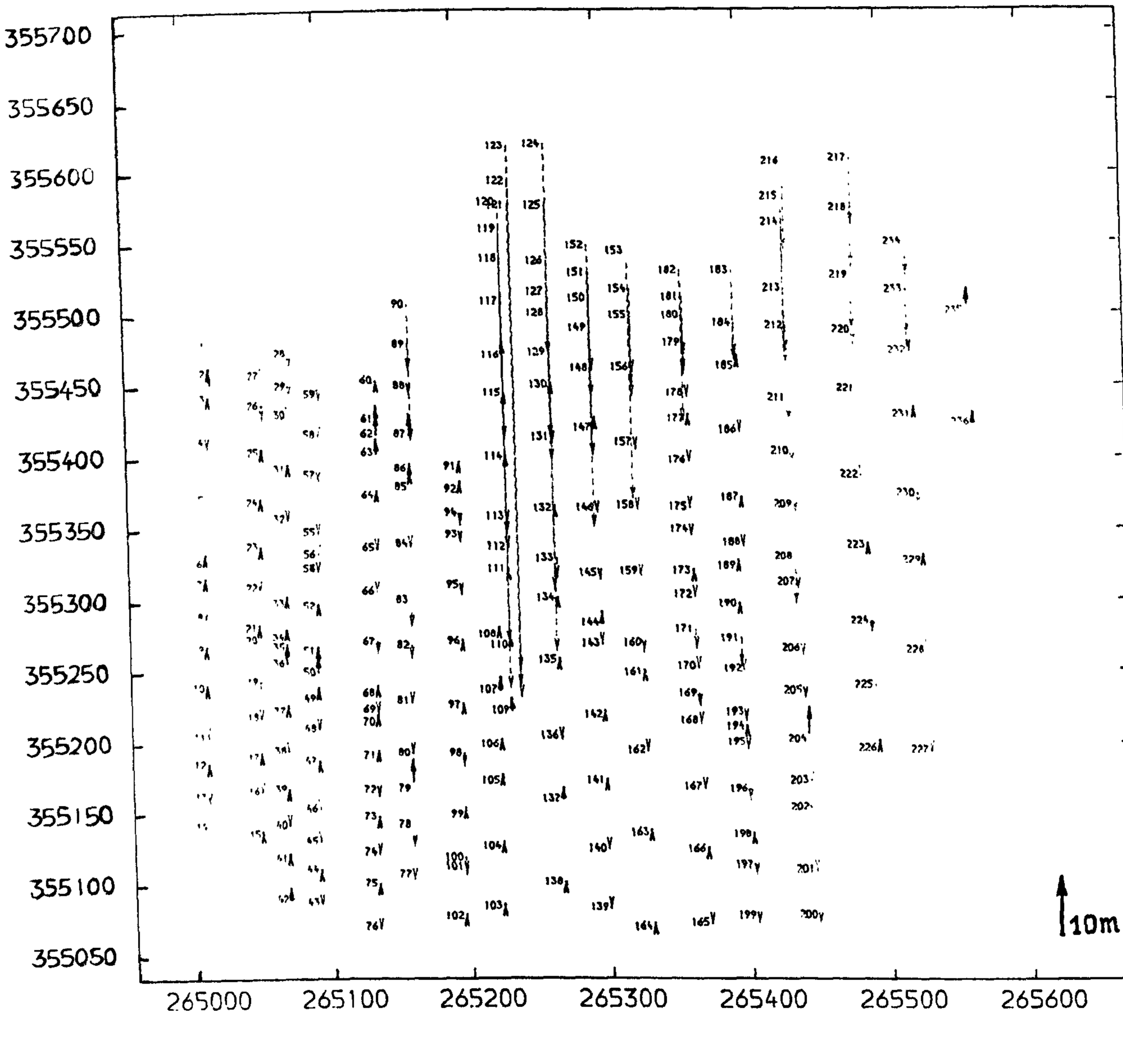


Fig. 9.10 Vector plot of the residual height errors of Fig. 9.9.

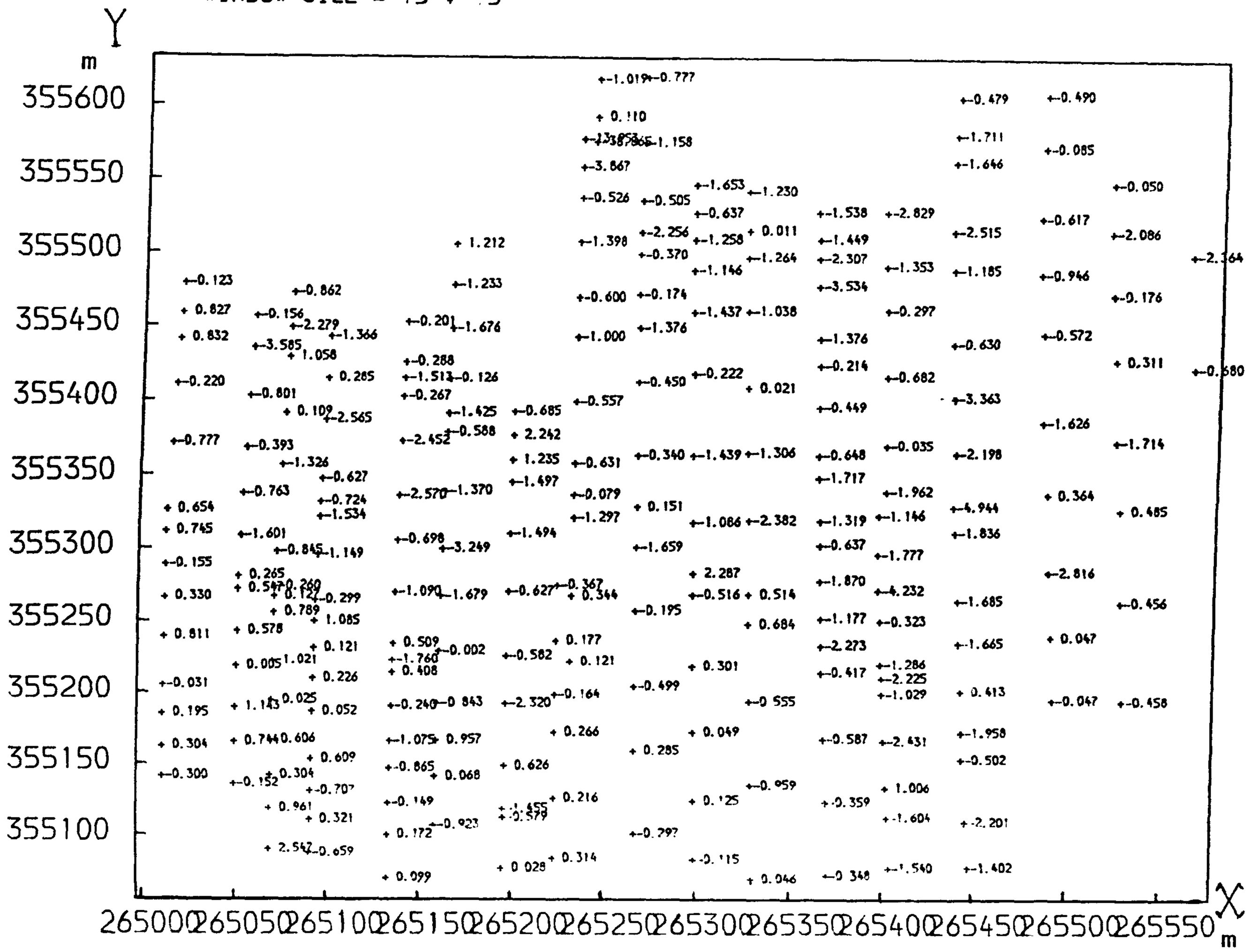


Fig. 9.11 The residual height errors (at the check points) for the DTM data generated by image correlation. (window size = 15x15 pixels)

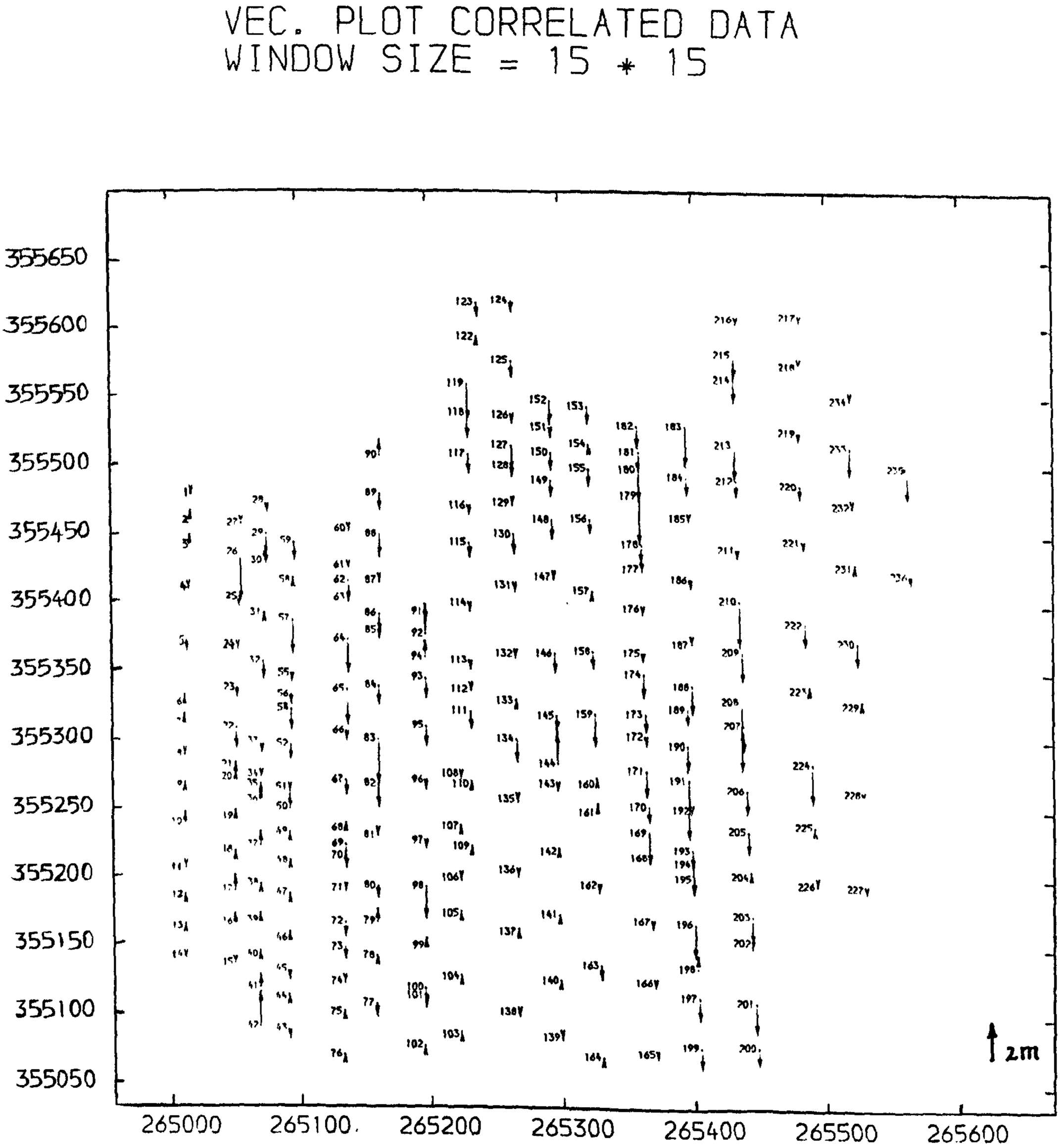


Fig. 9.12 Vector plot of the residual height errors of Fig. 9.11.

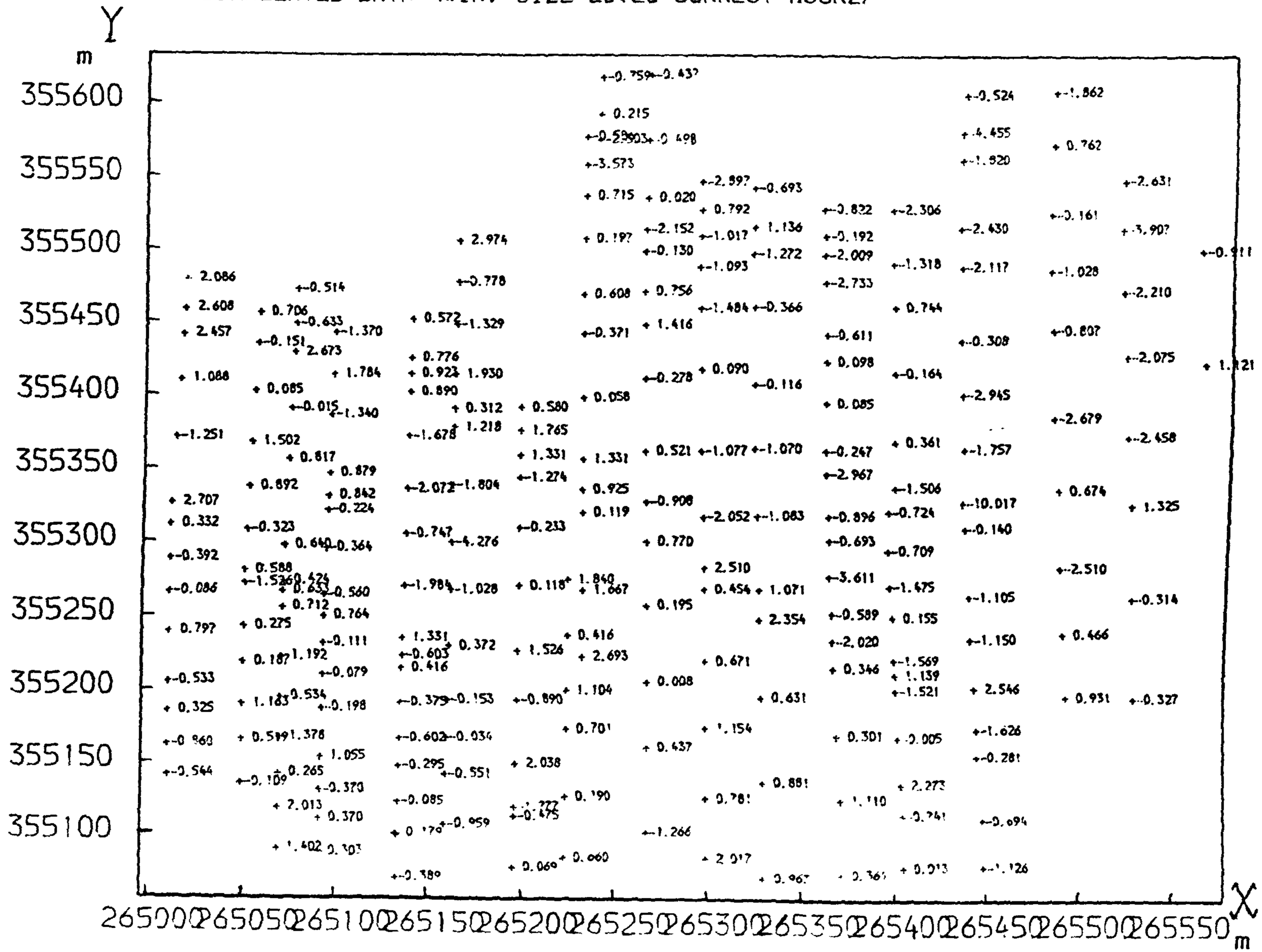


Fig. 9.13 The residual height errors (at the check points) for the DTM data generated by image correlation. (window size = 20x20 pixels)

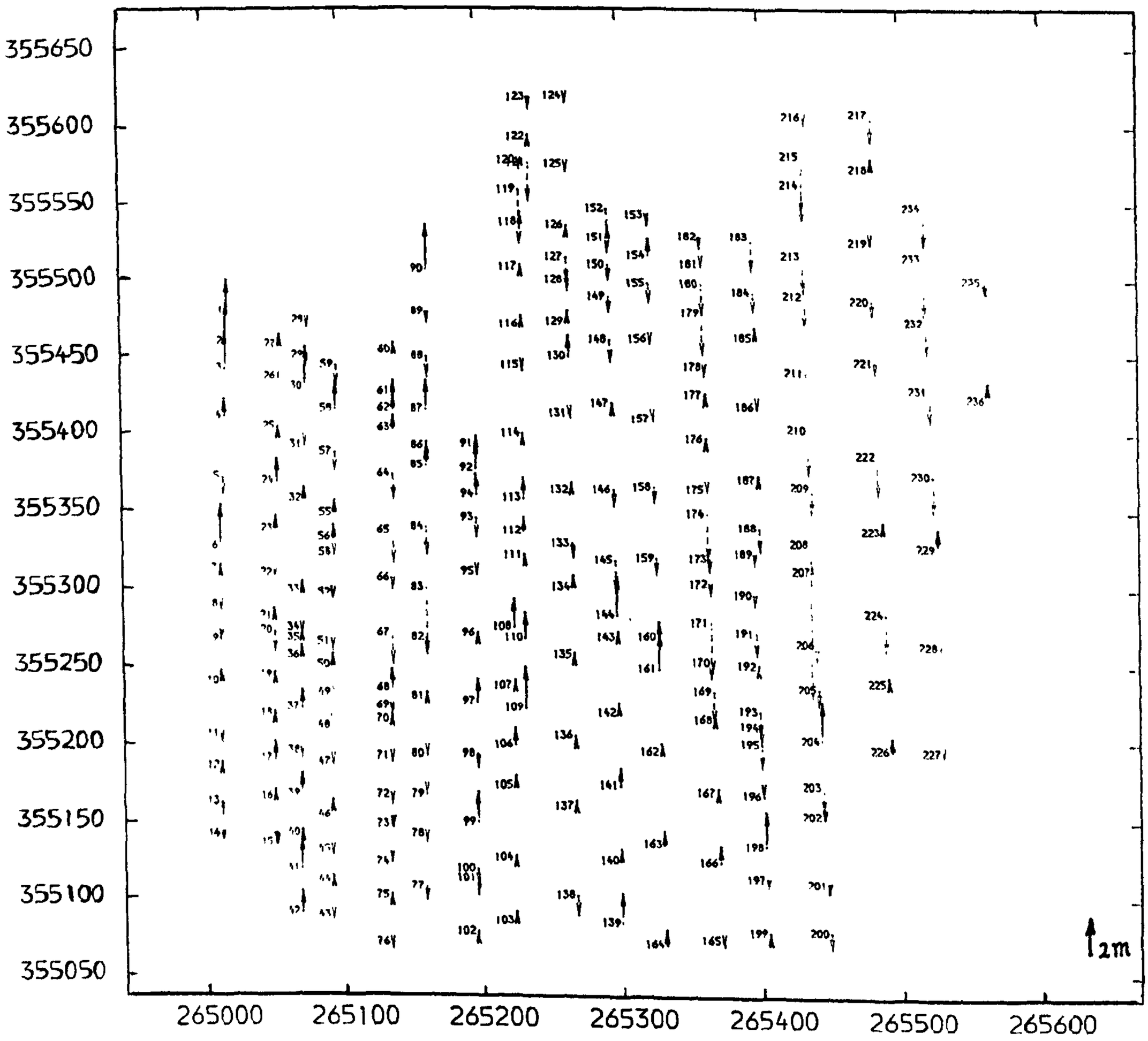


Fig. 9.14 Vector plot of the residual height errors of Fig. 9.13.

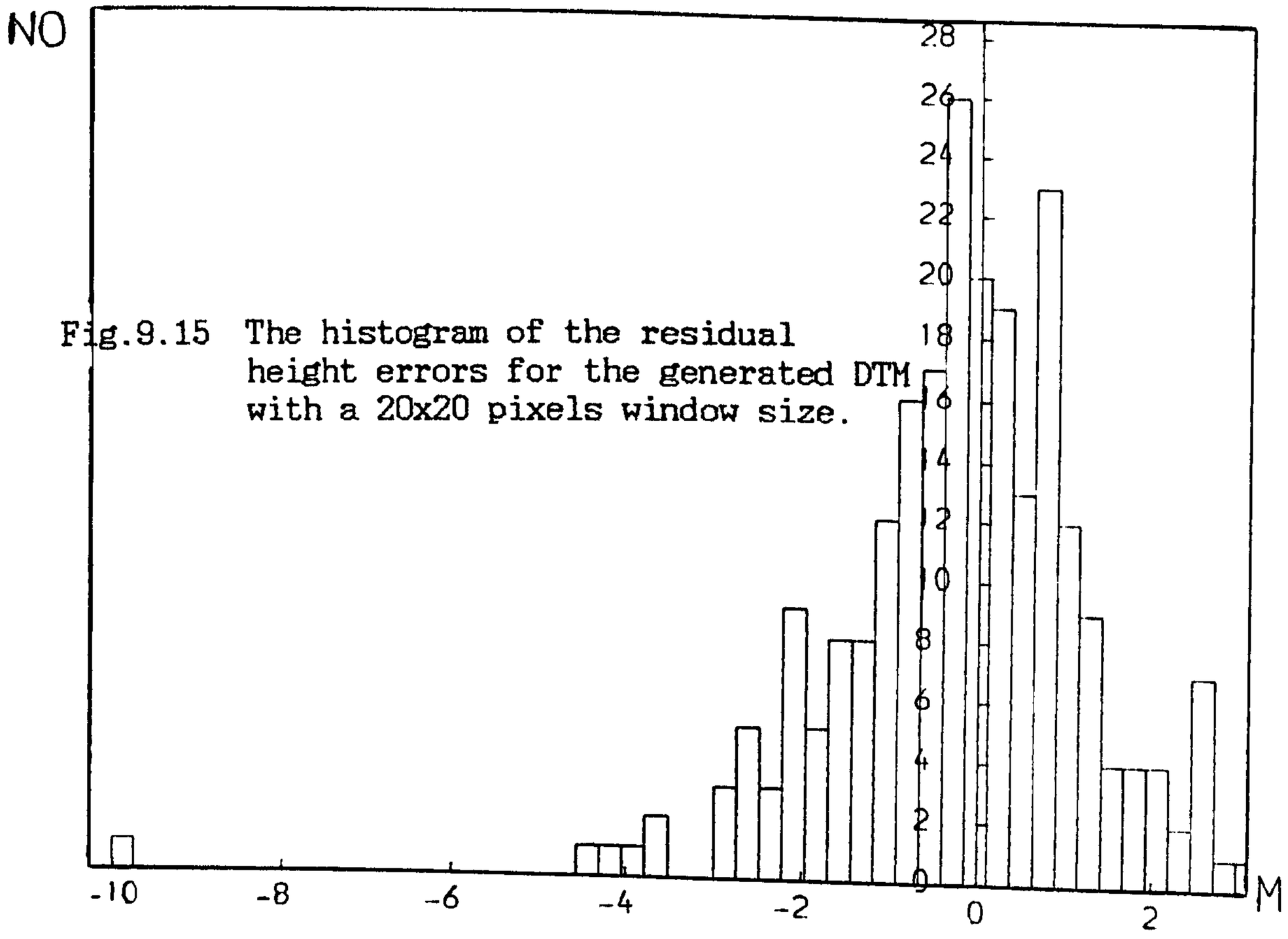


Fig.9.15 The histogram of the residual height errors for the generated DTM with a 20x20 pixels window size.

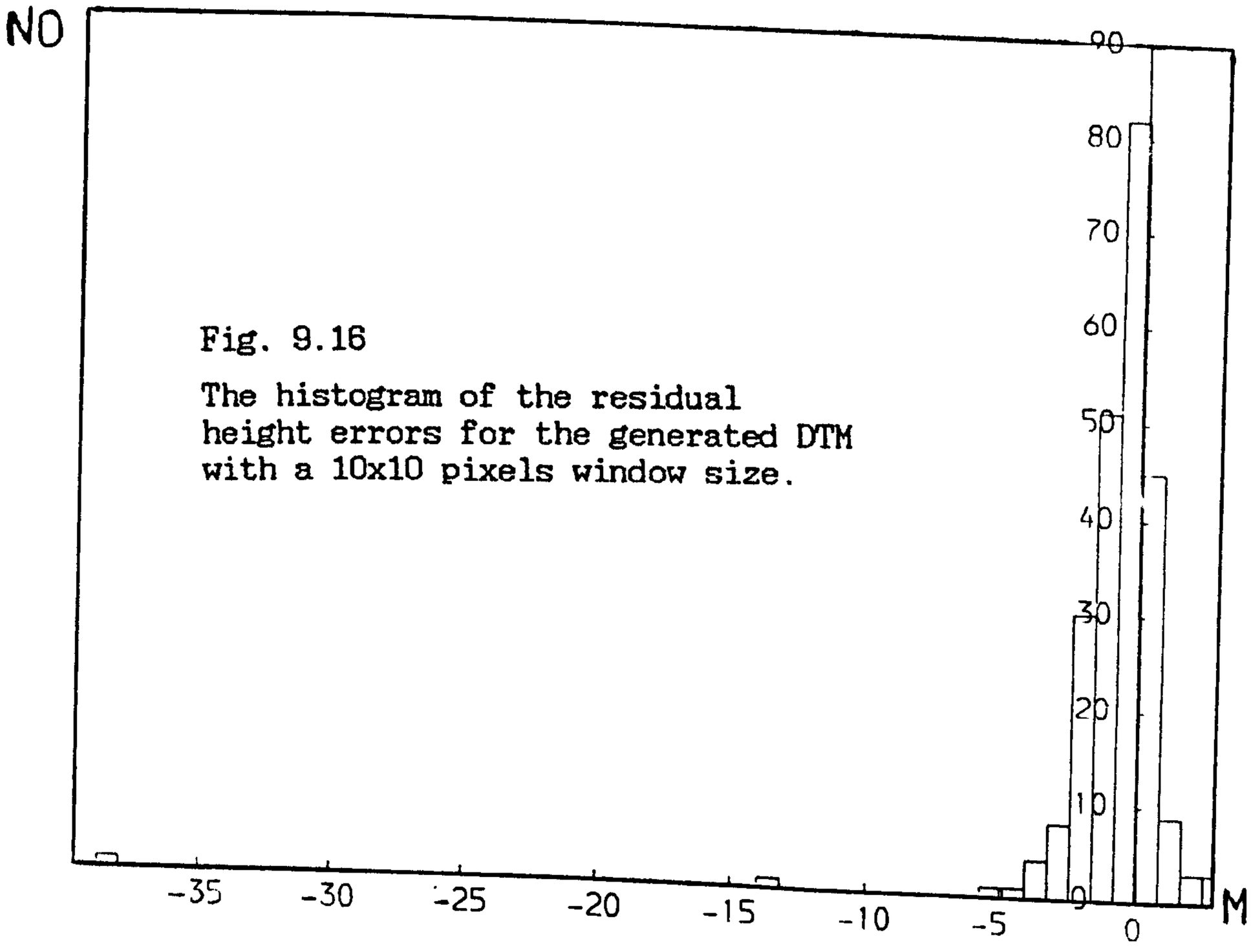


Fig. 9.16

The histogram of the residual height errors for the generated DTM with a 10x10 pixels window size.

method was solved quite efficiently by combining the Gradient approach with the IML method.

The window shifting image correlation method implemented for the DTM generation has the characteristic of using images which have been rectified and resampled along epipolar lines (i.e. images which are free from tilt displacements and other geometric distortions). Hence the method performs some kind of image shaping prior to the image correlation phase. Moreover, if a rough estimation of the terrain relief (a coarse DTM) is incorporated in the resampling stage, the problem of the terrain geometrical complexities may be solved.

In the next chapter, the achieved speed-up factors for the rectification and the correlation programs on the IBM 3090 computer are reported and discussed.

CHAPTER TEN

The Implementation of the Rectification and Correlation Programs on the IBM 3090 Computer with Vector Facility

review of the problems just quoted, but to give a brief introduction to the special purpose computers developed for numerically intensive computing and to report the results of the speed-up factors achieved on the University's IBM 3090 computer. These are covered in the sections which follow.

10.2 Classification of Digital Computers

According to whether the instruction or data streams are single or multiple, Flynn (1966) classified all computers into four groups which, broadly speaking, are still valid today. These different groups are discussed below.

(1) The Single Instruction Stream/Single Data Stream (SISD) Computer:

This is the conventional serial Von Neumann computer and sometimes referred to as a serial scalar computer.

(2) The Single Instruction Stream/Multiple Data Stream (SIMD) Computer:

This may be implemented in different forms, for example either as a parallel array of computers, or as a vector computer. In the former case, the control unit issues a single instruction to a number of identical processors, while in the latter case, the stream of data to be processed is broken into smaller sections, known as the stages in a pipeline, and individual hardware processors are allocated to perform different stages of the computation in parallel. Hence, a SIMD computer achieves parallelism in the data stream in these quite different ways. These will be discussed in more detail in the following sections.

(3) The Multiple Instruction Stream/Single Data Stream (MISD) Computer:

This achieves parallelism in the instruction stream rather than in the data stream. This type of operation is generally unrealistic for any kind of computer design and does not exist in practice.

(4) The Multiple Instruction Stream/Multiple Data Stream (MIMD) Computer:

This combines parallelism in both the instruction and data streams. This may be formed either by employing several small transputers (see Section 10.4) linked together, or it may be formed from a combination of several computers interfaced to each other.

Those computers which carry out concurrent processing operations fall into two categories known as parallel computers, and pipelined (or vector) computers. As mentioned above, the parallel and pipelined systems can either have N duplicated hardware devices or N partitionable stages, respectively. Thus, if a specific arithmetic function can be executed in a serial type computer in K nsecs, it will be executed repeatedly at rates of K/N nsecs in the parallel and vector computers. This represents an N-fold increase in performance. However, in practice, due to both hardware limitations and the nature of the function being executed, an N-fold increase can never be achieved (Hockney and Jesshope, 1981; Baz, 1984). Each of the parallel and pipelined machines will be discussed in the next sections.

10.3 Distributed Array Processors (DAP)

As mentioned earlier, one of the possible approaches for achieving concurrency, is by duplicating a hardware structure many times. One quite common type which operates on a large array of linked processors, is a SIMD type machine, and is termed a parallel or distributed array processor (DAP). The system contains an array of identical processors (called "processing elements" or PEs) capable of simultaneous arithmetic and logical operations. The PEs themselves may contain their own internal memories or they may use multiple shared memories, or they may have a combination of both. Generally at the time of execution, the various PEs will process similar data using a common algorithm. In most systems, the PEs do not contain their own programs, but obtain their common control signals from an external control unit known as the master control unit (MCU) (Baz, 1984). Fig. 10.1 shows the configuration of a typical DAP system.

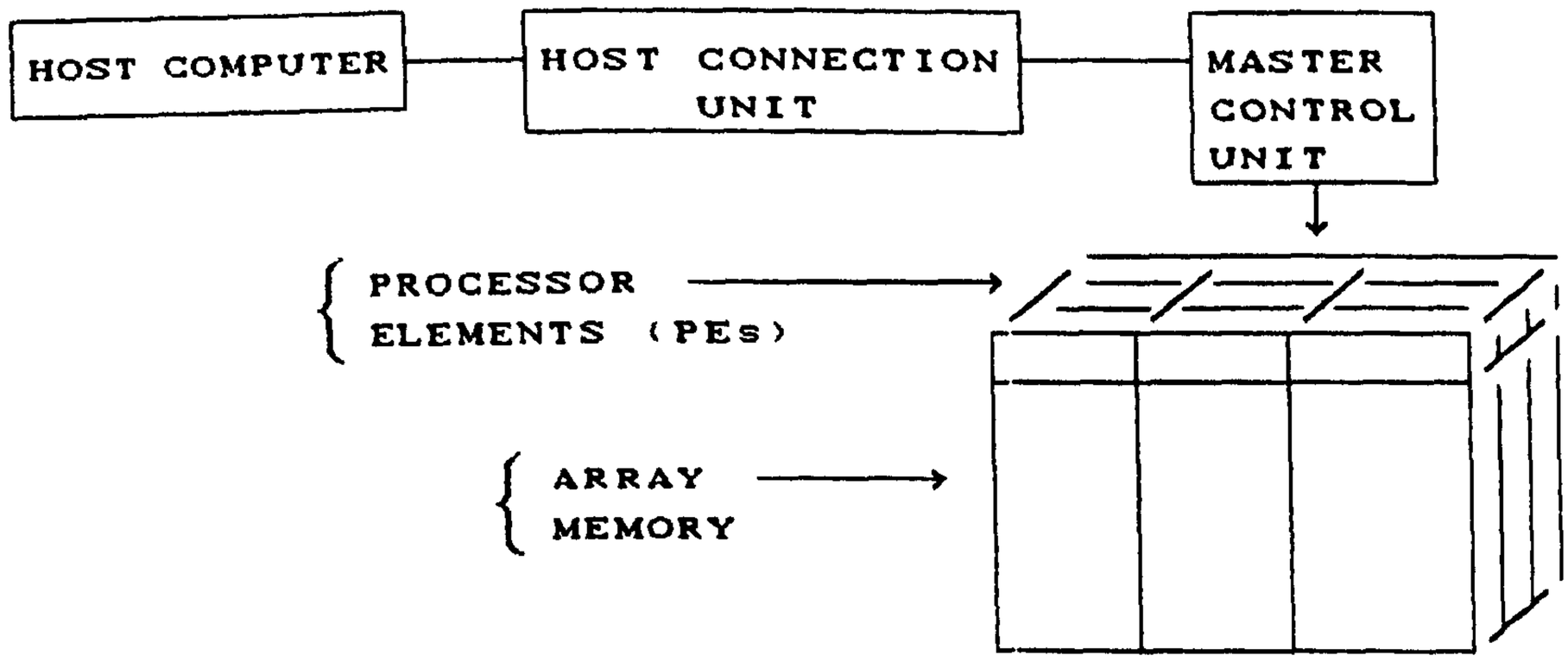


Fig. 10.1 GENERAL CONFIGURATION OF A DAP SYSTEM

Most distributed array processor systems are connected to or form part of another conventional computer which may be a mainframe, a mini-computer or a powerful graphics work station. The conventional computer acts as a front-end computer for the parallel array of processors. Several DAP systems are commercially available, e.g., the ICL DAP and its developments the DAP 510 and 610 (from Active Memory Technology AMT), the ILLIAC IV and the BURROUGHS PEPE. The recently developed AMT DAP 510 and DAP 610 machines are fine grain massively parallel computers with an square array of 32X32=1,024 and 64X64=4,096 PEs respectively. Each PE has connections to its four nearest neighbours. In addition, a bus system connects all the PEs in each row and all the PEs in each column. These row and column data paths provide rapid data broadcasting or fetching facilities and hence make this class of machines very appropriate for image processing operations which need to store huge amounts of data and to retrieve or exchange this data at a very high rate.

10.4 Transputer Arrays

Another possible approach for achieving concurrency is to realize almost the same concept presented above, but in an MIMD configuration. This technology, which is newer than the distributed array processors, is known as transputer array technology. The main commercially available transputer is the INMOS transputer which is essentially a computer on a chip. Its VLSI (Very Large Scale Integrated) chip comprises a 32-bit processor, 2 to 4k bytes of on-chip RAM and four

bi-directional asynchronous communication links with a transfer capacity of 20 Mbps. There are currently two primary versions of this chip. The first (T414) is a 32-bit integer-only processor while the second one (T800) is a 32-bit floating point chip. To achieve concurrency, a large number of such transputers may be linked together. One of the commercially available transputer-based array computers is the Meiko Computing Surface which has been installed very recently in the University of Glasgow.

There have been several attempts by photogrammetrists (e.g. Baz and Methley, 1985; Muller et al., 1988) to solve photogrammetric problems and to implement image processing operations both on the DAP and machines using transputer-based arrays. In particular, the most recent research work on the implementation of image matching using transputer arrays seems to be promising for solving the real-time image correlation problem (Muller et al., 1988). However since the main theme of this chapter is the solution of image processing operations on the more commonly available vector processors, further details regarding these technologies and their use for image processing operations are not given here. For the detailed treatment of the use of these machines for image processing and correlation, the reader is referred to Muller et al. (1988).

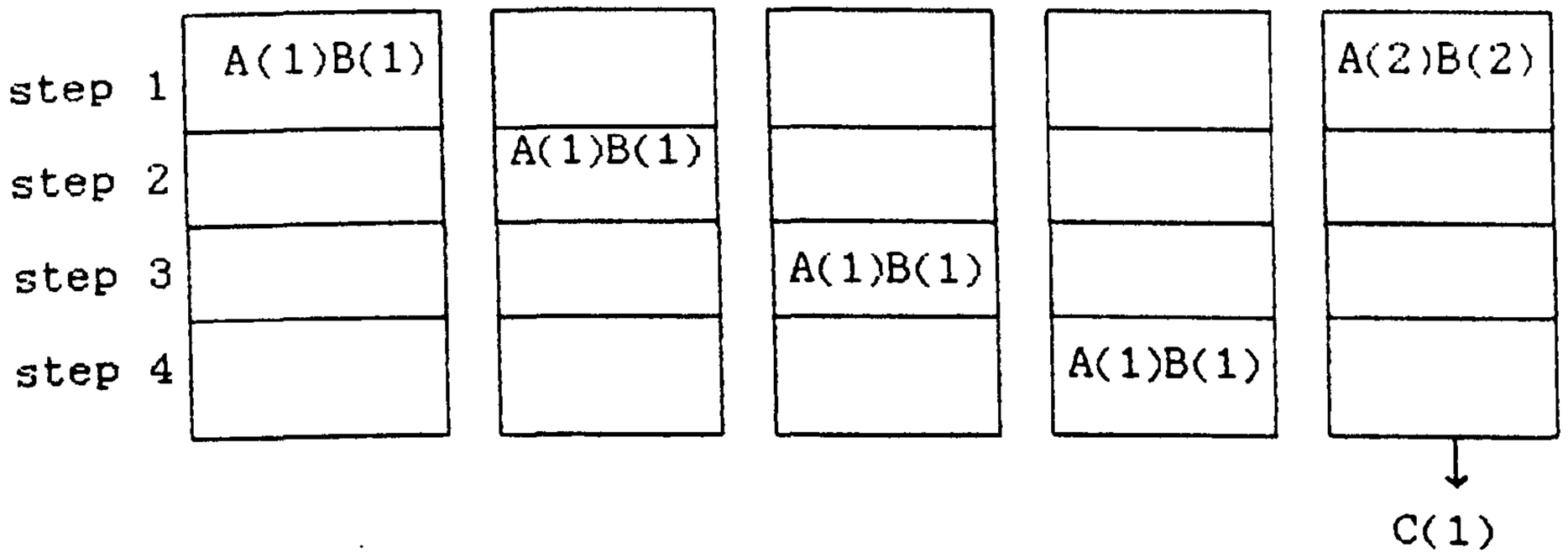
10.5 The Concept of Pipelining

As briefly discussed in Section 10.2, pipeline processing (sometimes referred to as vector processing) generally breaks down the arithmetic operations which have to be performed into a number of smaller stages and allocates individual **hardware processors** to execute each stage. Each can operate separately on the data values flowing through a specific stage. A data value which flows through the pipeline occupies only one stage at a time and all the stages of a functional pipeline are occupied by different data items. As soon as one stage is vacated by an item of data, it is occupied by the one immediately following it. For example, the floating point multiplication of two real numbers of 0.0298 X 10**4 and 0.176 X 10**5, would comprise the following steps:

step 1. normalize: 0.298 X 10**3 times 0.176 X 10**5
step 2. add exponents: 0.298 times 0.176 times 10**8
step 3. multiply mantissas: 0.05245 times 10**8
step 4. post normalize: 0.5245 times 10**7

The execution of this type of operation for arrays of data e.g., C(I)=A(I)*B(I) is described below in both scalar and vector modes.

SCALAR MODE:



VECTOR MODE:

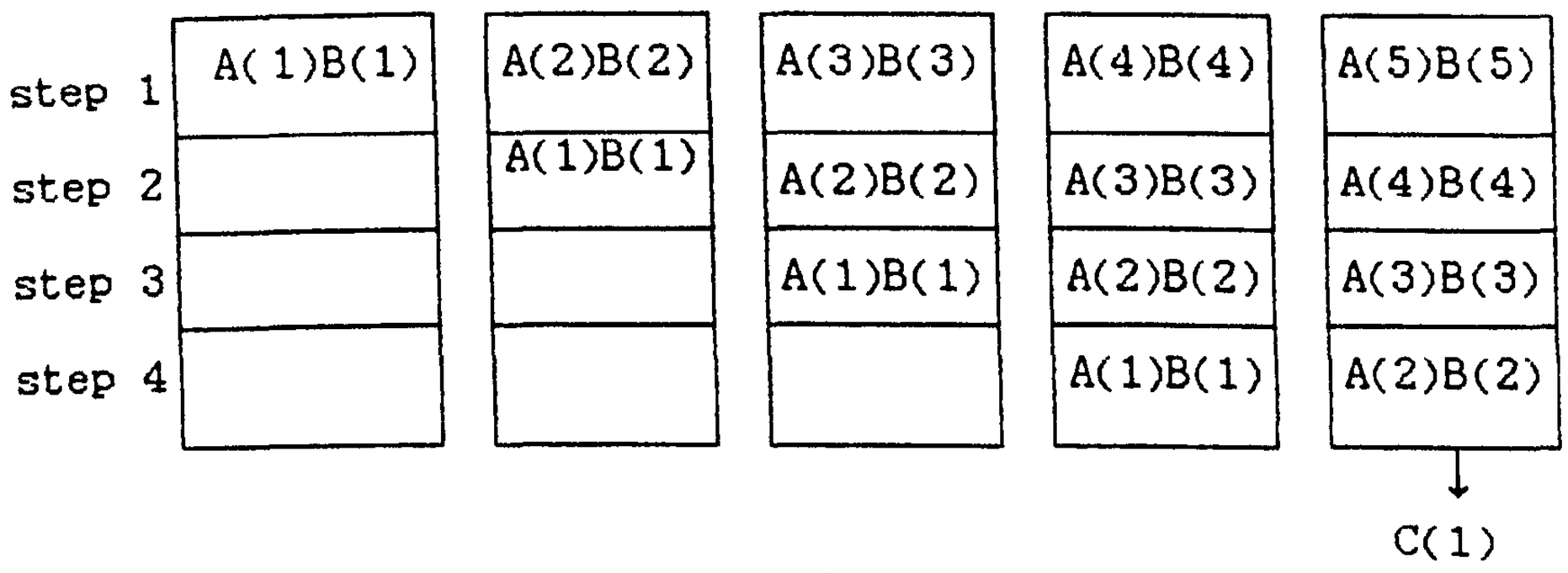


FIG. 10.2

The overlapping shown above (Fig. 10.2) is referred to as "Pipelining" or "vectorization", and the implementation of this concept is a major factor in the execution speeds of the vector instructions. In the example above, scalar processing would produce 3 results every 12 steps, whereas vector processing would produce 9 results during the same number of steps.

Pipelined computers may be classified into two groups. The first group containing for example, the IBM 3838 AP and CDC STAR-100, are those capable of performing only one basic function at a time, albeit on

many different items of data simultaneously. Thus it falls in the SIMD category. This type of pipelined computer is termed a vector or array processor. The second group are those computers which may occasionally perform more than one function in parallel, and are known as **multifunctional pipelined computers**. This series of machines may be classified as intermediate between the MIMD and SIMD computers. Besides the specialized vector and multifunctional computers, the conventional SISD computers may also employ various degrees of pipelining. For example the CRAY-1S supercomputer falls into the category of SISD computers, but since its hardware system contains explicit vector instructions, it is usually classified as a vector processor along with the true vector processors in the SIMD category. The IBM 3090 computer is also a serial-type general purpose scalar machine of the SISD type, but since it may have a hardware Vector Processor added to it, it may then be classified as a SIMD computer. Irrespective of the design goals, most of the current pipelined vector processors have an overall structure as shown in Fig. 10.3. This has a single main memory containing both the instructions and the data in both vector and non-vector form. As will be discussed later, there are some functions (e.g. I/O operations, operating system functions, etc.) which cannot be coded in vector form. Hence a scalar processor is included in the vector hardware.

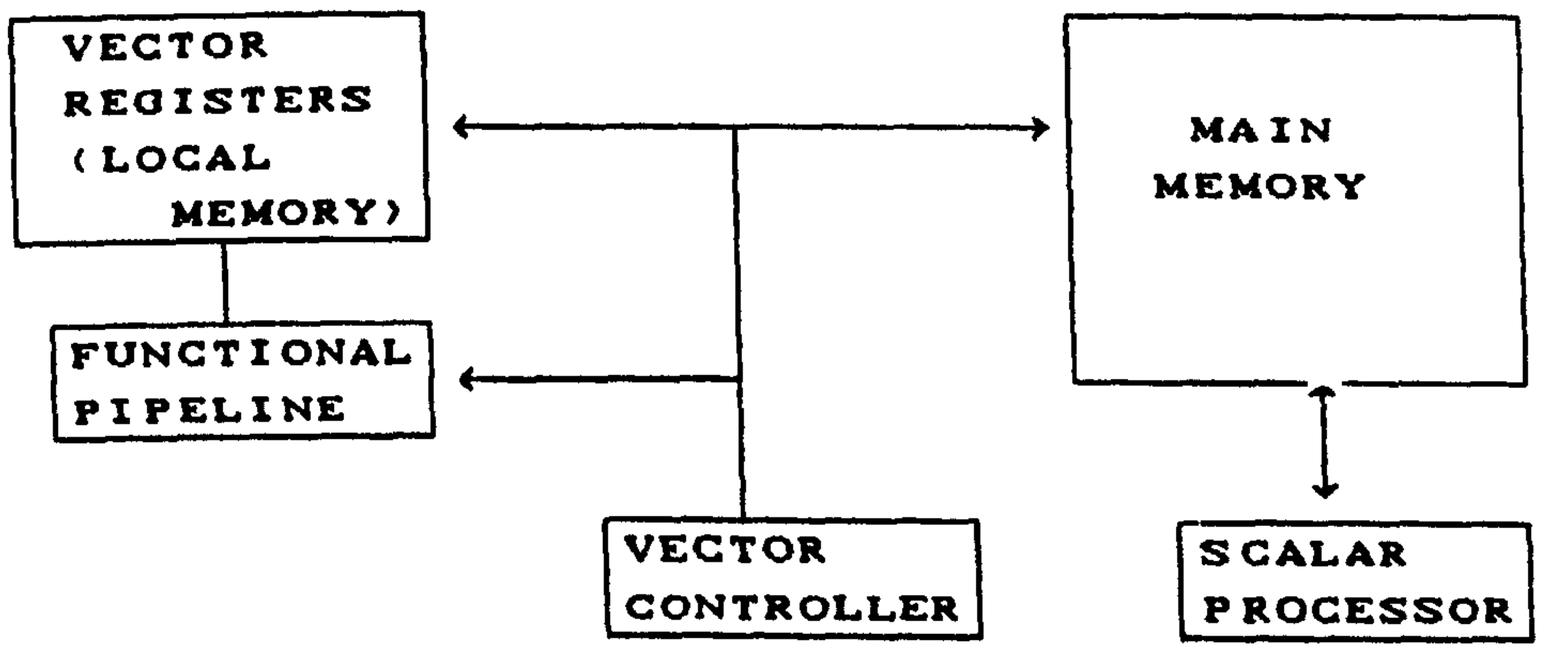


Fig. 10.3 STRUCTURE OF A VECTOR PROCESSOR (AFTER BAZ, 1984)

10.6 The IBM 3090 and its Vector Facility

The IBM 3090-150E mainframe computer used in the present research work is a scalar (SISD) computer to which a Vector Facility has been

added. This is a vector processor which provides the capability of processing vectors - a group of elements in an array - on the general purpose 3090 scalar machine. The IBM 3090 Vector Facility has 16 32-bit vector registers[1] or 8 64-bit registers for single or double precision data respectively. The vector registers can hold up to 128 data elements. The vector facility has three other registers which are termed the vector mask register, the vector activity count and the vector status register respectively. Fig. 10.4 shows the overall configuration of the IBM 3090 central processor with the Vector Facility.

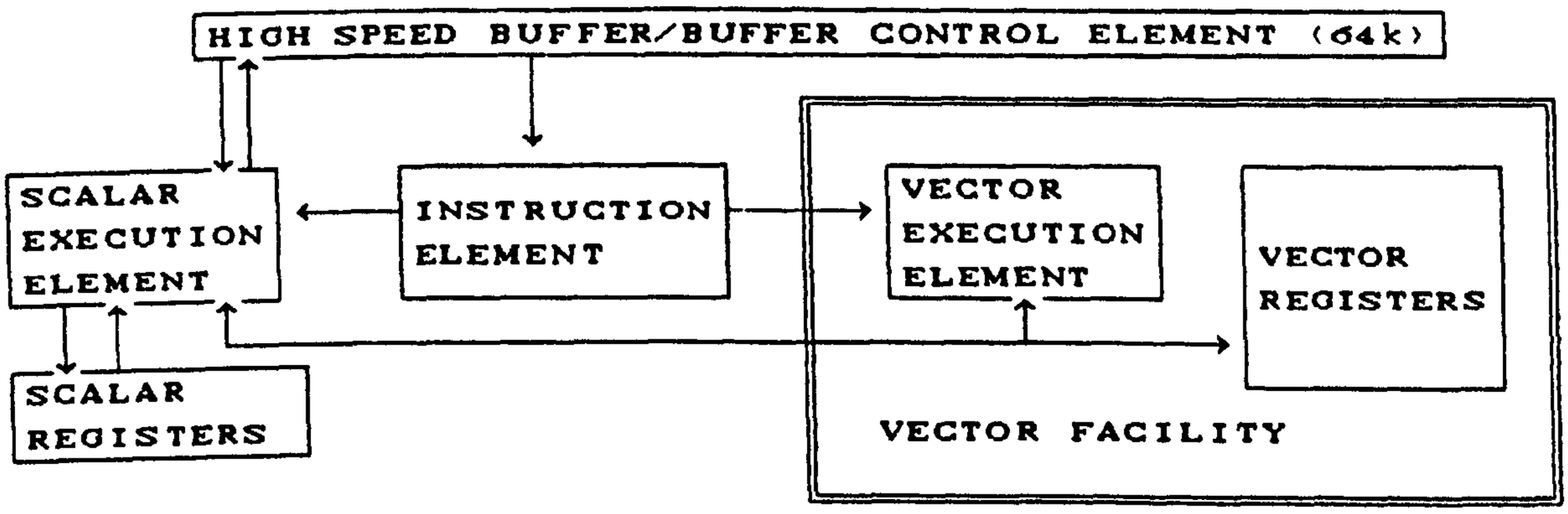


Fig. 10.4 IBM 3090 CENTRAL PROCESSOR

It is worth mentioning that the top machine in this range is the IBM 3090-600E which again is a scalar-type computer with fast scalar performance for computing intensive applications. However, in contrast to its smaller sister (the 150E) it has six processors, each with a Vector Facility. Thus, the IBM 3090-600E may be classified as a SIMD computer since it has multiple scalar and vector processors which can operate in parallel under a common single instruction. Its overall performance is such that it can be classed as a super-computer.

10.7 Vectorization Procedure on the IBM 3090 VF

Applications vary in their ability to take advantage of the IBM 3090-150E + VF. For example, those applications where a great deal of

1 *A register is a very small memory unit which can be accessed directly and rapidly through the logic of the computer. Some of the well known registers are the arithmetic register, the index register, the address register, etc.*

CPU time is spent performing iterative mathematical computations on arrays of numerical data, e.g. large scale engineering/scientific programs, may achieve more computational speed-up than other applications. When an application is found to be suitable for vector processing the following procedures, which were followed for vectorizing the RECT and CORRL programs, seem to offer a systematic approach for optimizing a program in order to make the highest benefit of the Vector Facility on the Model 150E.

(1) In the first stage, the program is run in scalar mode and the CPU time is recorded.

(2) Before running the program in vectorized version, it is essential to determine the key Do loops and the statements within these loops which are using the highest percentage of the CPU time. This enables the programmer to direct his tuning effort to those parts of the program which will give the best return for the effort expended. Determining the CPU usage of individual loops and statements within a program can be accomplished using a feature of the compiler's Interactive Debugging facility called "Program Sampling" or "Hotspot Analysis". This operates by interrupting the execution of the program at regular specified intervals and recording the percentage of CPU time required for each statement in the program. These records will comprise the output of the Hotspot Analysis in the form of an annotated listing.

(3) The next step involves compiling the program with the vector option activated and asking for a report. This invokes the VSFortran compiler on the IBM 3090 and produces a vector report which shows the loops which have been vectorized and the loops which cannot be vectorized. In general, the vector report generates the following massages:

   a. VECT - If a loop is vectorizable, it will be flagged as VECT.

   b. VECT/ELIG - This massage shows that a loop is eligible for vectorization, but the cost of vector execution exceeds the cost of scalar execution.

   c. UNAN - If language or control constructs are

encountered which hamper the compiler's ability to vectorize the loop, then the loop is flagged as unanalysable. For example, I/O statements within a loop make the loop unanalysable.

d. **UNSP** - The message shows that there are data and constructs for which there is no compiler and Vector hardware support. For example, INTEGER*2 fetches or stores governed by an IF statement are hardware-unsupportable.

e. **RECR** - If the vectorization of a loop would produce results which differed from the scalar version of the loop, then the compiler will assume that there exists some kind of recurrence, and will flag the loop as RECR (this matter will be discussed further in Section 10.8).

(4) If key loops have not been vectorized, the **inhibitors** may be removed by reorganisation of the program. The vectorization inhibitors and the possible ways of removing them are briefly discussed in Section 10.8.

(5) The program can now be run to obtain the CPU timings for vector execution and the **speed-up factor** can be calculated according to,

$$\text{speed-up} = T_s/T_v$$

where T_s and T_v are the CPU times taken by the scalar and vector versions respectively.

(6) If a satisfactory speed-up factor has not been achieved, the program may be further **modified** to improve the vector performance.

10.8 Vectorization Strategies

As briefly discussed in Section 10.7, there are certain factors which inhibit the vectorization of a Do loop. Therefore, when developing a program for vectorization, the following cases should be taken into consideration:

- Anything other than a **Do loop** cannot be vectorized.
- One of the major factors which affects performance improvement is **vector length**. As mentioned earlier, the vector registers can hold up to 128 elements. Therefore, vector lengths of 128 are excellent, and will give a speed-up which is close to optimum. A vector length exceeding 128 elements will be

"sectioned off" into "sectioning loops". For example, a vector length of 300 elements will be segmented into 3 sectioning loops with lengths of 128, 128 and 44 elements respectively. Array dimensions of less than 12 are likely to execute faster in scalar mode, although loops which are rich in floating point content can give good results with short vectors.

- IF statements are vectorizable; however the performance improvements can be limited. Hence conditional statements should be kept at a minimum level.
- Accessing storage contiguously provides the best result. Therefore, the **spacing between the data elements** in storage (known as the stride) is an important factor for accessing the data. This means that vectorization on the subscript which varies most rapidly (i.e. the left most) during loop iteration is preferred and gives "stride 1" processing.
- One of the key factors in the loop vectorization is the degree to which the data elements in the array can be processed independently. In general, if a value computed on one iteration of a loop is required as an input to a subsequent iteration of the loop, then the data elements cannot be processed independently and therefore, such a loop is not vectorizable. This is referred to as "**recurrence**". To give an example, consider the following Do loop, executed in both scalar and vector versions,

```
      DO 30 J = 1 , N
      A(J+1) = A(J)+B(J)
30    CONTINUE
```

* scalar execution	* vector execution
fetch A(1), fetch B(1)	fetch A(1), fetch B(1)
compute A(1)+B(1)	fetch A(2), fetch B(2)
store A(2)	etc.
fetch A(2), fetch B(2)	compute A(1)+B(1)
compute A(2)+B(2)	compute A(2)+B(2)
store A(3)	etc.
etc.	store A(2)
	store A(3)
	etc.

Note, that in scalar execution, A(2) is stored before it is fetched, whereas, in the vector version, A(2) would be fetched before it is stored. Therefore, the wrong value of A(2) would be used for the computation, and hence, vectorization is prohibited due to the recurrence on A.

10.9 Vectorization of the DIGREC and CORRL Programs

To vectorize the rectification and correlation programs, the procedures described in Section 10.7 were followed. For the program RECT, the first analysis of the vector report showed that the loops involving the three-dimensional transformations were being vectorized. However, the loops involved resampling were flagged as UNSP. This was due to the use of INTEGER*2 arithmetic of indirect addressing within an IF statement. Therefore, the overall result showed very little improvement in performance, i.e. the speed-up factor was very small.

To improve the performance, two strategies were implemented. First the variables were defined with a size of 128. As described earlier, this is due to the fact that the Vector registers can hold up to 128 elements. The second modification to the program was applied to the resampling loop, by extracting some of the segments of the statements involving the calculation of the coefficients of the bicubic interpolation function. These segments were then shifted to the three-dimensional transformation loops. In this way, the CPU time spent on the resampling loop was slightly reduced. This modification provided only a very small improvement in performance (i.e. a speed-up factor of only 1.1). This was due to the fact that still a large percentage of the CPU time was being spent on the remaining resampling loop.

To improve the computational time needed for the resampling loop, the bicubic interpolation function was replaced by the two alternative interpolation schemes of bilinear and nearest neighbour. The former improved the speed-up factor to 1.9; the latter to 2.4. However, this result, does not seem to be capable of significant further improvement, since a large percentage of the CPU time is spent on the I/O operations which are not vectorizable. The CPU times for an image

of the size 2,000X2,000 pixels using different interpolation schemes and the resulting speed-up factors for each case are given in Table 10.1.

INTERPOLATION METHOD	CPU TIME (NON-VECTOR) seconds	CPU TIME (VECTOR) seconds	SPEED-UP FACTOR
BICUBIC	62	56	1.1
BILINEAR	31	16	1.9
NEAREST NEIGHBOUR	19	8	2.4

TABLE 10.1 CPU TIME AND SPEED-UP FACTORS FOR THE PROGRAM RECT, USING DIFFERENT RESAMPLING INTERPOLATION FUNCTIONS

Regarding the program CORRL, the loops involving the calculation of the correlation coefficient function were simply vectorized. The program was then executed for different window array sizes. As a result, the speed-up factors varied with these different window sizes. Speed-up factors of 1.1, 1.5, 1.8 and 2.1 were obtained for the window sizes of 100, 225, 400 and 625 pixels respectively (Fig. 10.5a). The CPU time (Vector mode) for each window size is given in Fig. 10.5b. The CPU times for the 100 and 225 pixel windows were higher than expected. This appears to result from the fact that the use of a smaller window size increases the matching uncertainty and hence increases the number of search operations required.

10.10 Conclusion

In this chapter, the potential of a vector computer in speeding up some of the photogrammetric operations, was demonstrated. The present author does not claim that the speed-up factors achieved on the IBM 3090 for the programs RECT and CORRL, are the final boundary beyond which no further improvement can be achieved. It is likely that further modifications to the programs would improve the results presented in this chapter. However, it is unreasonable to expect to see a significant improvement in the speed-up factors, since a great

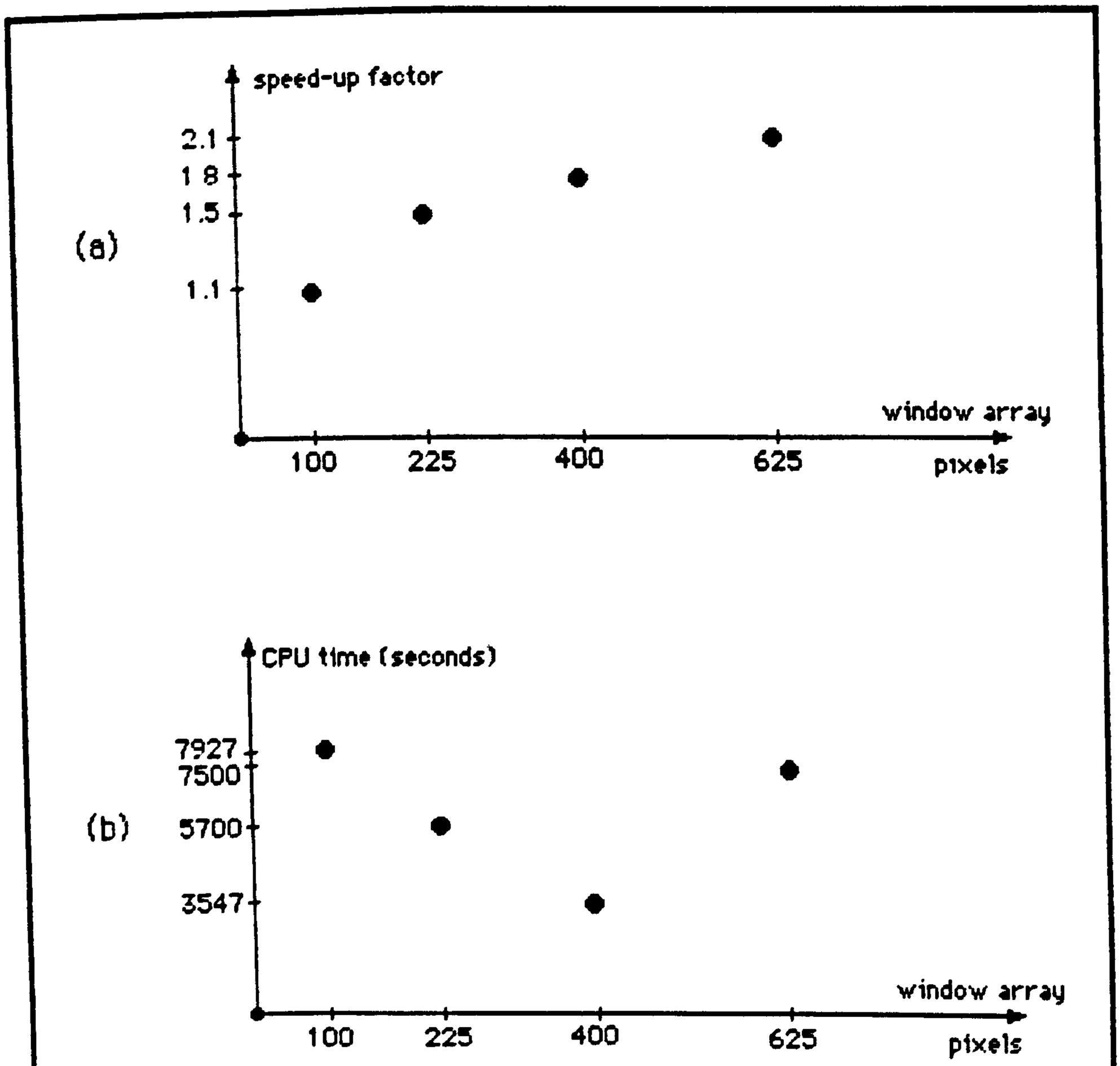


Fig. 10.5 (a) The speed-up factors for different window arrays.

(b) The CPU time (vector mode) for the correlation program carried out on 2000*2000 pixels images.

deal of the CPU time is spent on the I/O operations which, by their very nature, are unvectorizable. Nevertheless, it has been demonstrated in this chapter that the use of a vector processor can speed up the rectification and correlation programs with relatively little effort in optimizing the programs for vectorization. Therefore, in this respect, given the simplicity of programming on vector machines, the use of such a facility may be regarded as an advantage over the use of DAP and transputer arrays which need a much longer period of time to be spent on the development and modification of the programs. Nevertheless there is little doubt that at the present time, the only recognisable way forward in this area of all-digital photogrammetric systems is the adoption of a parallel computing architecture in the form of a DAP or transputer-based array since they do seem to offer the speed-up factors which are needed to make the adoption of all-digital machines really practical.

The next chapter which is the final one, draws some general conclusions regarding the present state of digital image photogrammetry and makes recommendations for future work.

CHAPTER ELEVEN

Conclusions and Recommendations

accuracy comparable to that achieved by analogue and analytical plotters. Of course this statement can only be made under the strict proviso that a high-quality image digitizer of the kind employed in this experimental work must be used to achieve the desired result.

In general, the implementation of the different photogrammetric operations in this project may be regarded as some contribution to defining and realizing systems which will become the future generation of photogrammetric instruments. It does seem that the ultimate goal of digital photogrammetric systems is to combine all photogrammetric tasks including the acquisition, storage, processing, etc. of terrain data into a single computer based system. Thus it is perhaps more appropriate to call these new instruments "**digital photogrammetric stations**". Besides the research going on in universities, there are clear signs that the mainstream photogrammetric system suppliers are also investigating various possibilities for implementing such a digital photogrammetric station, and in the case of Kern, Matra, and Autometric, they have indeed been constructing prototype systems with a view to producing a digital station of the type just quoted. This means that, in the near future, digital photogrammetric stations will probably find a market within the surveying and mapping community, especially in the more highly developed countries. If such a digital system is intended to fulfill all photogrammetric requirements and to fully exploit the highest possible level of the metric information extraction in real time, then the three major problems of data acquisition, data storage, and computational speed, all of which have been defined and examined throughout this experimental work, need to be solved. Each of these factors will be discussed individually in the following sections.

11.3 Data Acquisition

Regarding the problem of data acquisition, the present technology and architecture of digital cameras shows no sign that, in the near future, they can be developed to match the resolution, format size and specifications of the current metric photographic cameras, unless a totally new technology for acquiring digital images becomes available. This means that, in the near future, and indeed for the foreseeable future, the aerial photographic film camera will still reign supreme.

11.1 Introduction

Since the results obtained during the various stages of this research work have been reported and analyzed in the relevant chapters, it is not necessary to repeat them all in this concluding chapter. However, it does seem appropriate to first draw some general conclusions arising from this work and then to make some recommendations regarding the present state and the likely future development of digital image photogrammetry. These are discussed in the following sections.

11.2 General Conclusions

The future of photogrammetric mapping and modelling systems appears to lie with all-digital solutions of the type devised in this experimental work. The research work presented in the preceding chapters has demonstrated a prototype of the future generation of photogrammetric instruments. As will have become apparent from reading this thesis, it has been demonstrated that the successful implementation of a digital stereo-photogrammetric system eliminates totally the expensive mechanical stages and measuring systems and the high-class optical trains which are a feature of present-day analogue and analytical photogrammetric systems. The present project has shown that even a multi-user computer with no purpose-built accessories may serve as a digital photogrammetric system for generating a DTM. Of course, the clear disadvantage of such a system is the very slow rate of the measurement process. However, with the ever increasing power of modern computers, this particular defect can be expected to disappear.

This experimental work has also demonstrated the power and high potential of image modelling and the least squares image correlation technique when used in conjunction with the Gradient method for target measurement. Moreover, it has also been shown that, with the use of a quite simple computational procedure, a reasonably effective image correlation can be implemented to generate a DTM comparable to those measured by a human operator. Regarding the problem of the achievable accuracy using a digital stereo-photogrammetric system which was picked out in the introductory chapter, the measurements carried out during the different phases of the project showed that a digital stereo-photogrammetric system can generate output data with an

In this context, it should also be mentioned that remote sensing scanner technology, both in the form of sequential line-by-line scanning and the pushbroom system, has not proved capable of fulfilling the photogrammetric requirements of an airborne mapping system, due to its enormous geometric distortions and comparatively low resolution. Of course the situation of using these devices for small-scale low resolution mapping from space is a rather different matter, since they are then not subjected to the turbulence encountered in the Earth's atmosphere and experienced by airborne platforms.

Regarding these considerations, it may be concluded that, for some considerable time to come, for implementing digital photogrammetric systems, there will be no alternative but to use hard-copy aerial photos digitized with an appropriate image digitizer. In this respect, the image digitizers used for photogrammetric applications must satisfy the stringent requirements associated with the transformation of the **pictorial** and **metric** information of aerial photos into digital form.

(a) Considering the **pictorial side** (the image quality), the work carried out during this project has demonstrated the fact that the use of a medium size pixel (32 μm) has an insignificant effect on the image quality and consequently on stereo-viewing perception. This is in conformity with Trinder's report (1987). The work has also shown that the use of the full grey level variation of the digitized image does not seem to be a very critical factor either with regard to the image quality or to the extraction of the metric information. In particular, it has been shown that even a binary image may be used for stereo-viewing and measurement, provided that the reduction of the range of grey level variations in the image is performed with an appropriate algorithm.

(b) Turning next to the **metric side**, the measurements carried out during the project showed that a 32 μm pixel size is not enough if the full geometric information inherent in the

analogue image needs to be extracted. To fully transfer the information content of the aerial photograph (i.e. to replicate the analogue photographic information in digital form), one needs to employ a highly accurate image digitizer of the kind used in this project. Also the image should be digitized with a smaller pixel size than that employed in this research work. The adoption of these specific actions should guarantee the transformation of all the metric information of the image with no loss of resolution or geometric fidelity. Clearly such high quality microdensitometer instruments are too expensive to be owned by photogrammetric companies or even national mapping agencies. However, if there are a number of these machines available in a country, they may be used on a bureau basis to serve the users by digitizing the aerial photographs and transferring the digital images either using a national communications network or, more likely, by recording them on magnetic or optical media for despatch to customers. However, it may be argued that an image digitizer of the class quoted above is too accurate for photogrammetric applications and that less expensive image digitizers may perform the digitization with enough accuracy. However, to author's knowledge, a comprehensive investigation has not been carried out in this area to demonstrate the merit and adequacy of cheaper image digitizers for photogrammetric tasks. This is indeed one of the areas which demands more investigation.

11.4 Data Storage

Data storage is probably one of the main problems associated with the implementation of digital stations. As mentioned throughout the previous chapters, in this research work, magnetic tapes were used for the mass storage of the digital photographic data. This brings associated problems of low speed access to and archiving of data. In this research work, the data access was performed in a sequential mode i.e. from the magnetic tape to the disk and from there to the random access memory of the computer. This approach is extremely inefficient and restricts the photogrammetric operation to a small image patch at a time. Therefore, in the future generation of digital stations, some

kind of secondary random access memory such as a cache memory should be used to ensure rapid access to the data stored on magnetic or optical disks.

Regarding the storage problem, if digitized aerial photos are to be stored on **magnetic tapes** of the kind used in this experimental work (i.e. 6,250 b.p.i magnetic tapes), then 5 magnetic tapes are required to store a block of 10 photographs digitized with a 32 μm pixel size. This storage requirement will, of course, be further increased if the images are digitized with a smaller pixel size. A block of 10 photographs digitized with a 10 μm pixel size will occupy 50 magnetic tapes at 6,250 b.p.i density, which would be extremely inconvenient from the data handling point of view.

In this context, the recent developments in **optical disk** technology seem to be very promising. Optical disks store information by using a finely focused laser to heat the information layer of the disk, causing a change of reflectivity in this layer and hence providing the possibility of storing large amounts of data on a small area of the disk. Currently there are several types of optical disks available on the market. These include Compact Disk Read Only Memory (CD-ROM); Write-Once Read-Many (WORM); and the very recently developed magneto-optic disks. The image information written on either of the first two types of optical disk is unerasable. Therefore these types of optical disk best suit the archiving requirements of aerial or space photography since they can provide a more reliable and less fragile media for massive data storage than magnetic tapes and also allow rapid access to any required data set. A single CD-ROM can hold up to 600 Mbytes of data. This is equivalent to about 6 pairs of aerial photos digitized with a 32 μm pixel size. This capacity will be increased still more if larger sized compact disks are used. This means that these types of optical disk offer an excellent means for storing and archiving the digitized aerial photos.

In the third type of optical disk (i.e. the magneto-optic disk), the data may be written and erased as many times as needed and hence these disks may be used for the mass storage required during intermediate processing stages. Nevertheless it should also be mentioned that, at

the present time, the problem of rapid retrieval of data from this type of media still remains to be solved. The performance of optical disks for file retrieval is better than magnetic tapes for small or moderate amounts of data only. However, they still have a much poorer performance in terms of data access than magnetic hard disks. Therefore, regarding the present technology, optical disks can offer only a partial solution to the data storage and retrieval problem.

11.5 Computational Speed

Computational speed is another important requirement for the successful realization of a digital photogrammetric station. The tests carried out during this research work confirm the viewpoint that vector processors do not offer the same potential for photogrammetric work as a parallel computing architecture, for example as realized in the form of a DAP or transputer arrays. In particular, if the most recently developed image correlation techniques are to be employed, it seems to be quite obligatory that they be run in a parallel mode. Otherwise the computational times are simply too long and therefore uneconomic.

The DAP has already shown its enormous potential in such areas as an icing model; molecular graphics; chip routing; signal and image processing; etc. The two latter cases also help to demonstrate the suitability of the DAP for digital photogrammetric applications. In addition, the DAP has also proved its capability in providing very high speed input/output operations, which again is an important factor in digital image photogrammetry with its huge amount of data. As already mentioned in the preceding chapter, there are two models of DAP from AMT (Active Memory Technology) currently available -- the DAP 510 and the DAP 610. These machines are designed to be used as attached processors to Sun graphics work stations or DEC VAX microcomputers and would probably satisfy the next wave of experiments in this field. It has been claimed by AMT (Glasgow University DAP presentation, 1989), that, by the year 1993, the new series of DAP (the model 720c) may reach a speed which is 96 times greater than the speed of the currently available DAP 510.

In this context, of course, the potential power of transputer arrays

for carrying out photogrammetric and image processing applications should not be underestimated. Indeed, it is not quite clear which of these architectures (i.e. DAP and transputer arrays) will be best suited for photogrammetric applications. Since each has its own merits for different computational tasks, one might even imagine that a photogrammetric system should use a combination of both of these technologies. This is yet another area which calls for more investigation.

As a final note regarding computational speed, it is worth mentioning that **reduced instruction set computing (RISC)** processor technology, which is different to both the DAP and transputer arrays, is another newly developed technique for increasing computational speed. Since RISC systems have only recently become available in the market and their suitability for photogrammetric applications is as yet unknown, they were not considered in Chapter 10. However, it seems pertinent in these concluding remarks to present briefly the concept and potential of such systems for photogrammetric applications. The RISC concept is based on the processor having fewer and faster instructions. The removal of the complex instruction set (hence CISC) incorporated in a "normal" computer processor, allows the RISC processor to run at very high speed at the cost of making the processor more tightly focused towards particular applications. This has made the RISC technology quite controversial regarding its suitability for a wide range of applications.

The RISC technology was initially implemented on graphics work stations such as the IBM 6150 (PC-RT). However, very recently, several other computer manufacturers --notably Sun with its SPARC processor; Intergraph with its Clipper processor; and the MIPS devices used by DEC-- have produced processors based on RISC technology. These have been incorporated in newly built machines such as the Sun Sparcserver and Sparcstation, Digital's DECstation series, the ICL DRS 6000, and the IBM RS-6000.

Having regard to the fact that some photogrammetric operations may only require the execution of quite simple instructions, it seems that the use of RISC technology might speed up those operations quite

significantly. In particular, the Reality PC computer board from Real World Graphics based on the use of twin Intel i860 RISC processors which has been introduced to the market very recently (a few weeks before the final editing of this chapter) has features which make it particularly applicable to image processing and photogrammetric applications. Reality PC is in effect a 3-D supercomputer and image generator mounted on a PC/AT format board. In particular, it has the capability of displaying 3-D image data stereoscopically, since a second video output channel is provided as a standard feature allowing two separate images to be viewed on individual displays. Reality PC achieves its computing power by employing dual Intel i860 RISC microprocessors, each with one million transistors. It has data and instruction caches on-board and a 64-bit architecture. Regarding these features, it seems that the implementation of a digital photogrammetric system on such a computer is yet another research area which demands investigation.

11.6 Cost of a Digital Stereo-Photogrammetric System

Before ending this chapter, it also seems pertinent to say a few words about the costs of digital stereo-photogrammetric systems. Having regard to the fact that digital stations of the type mentioned in the preceding sections, need a very fast, powerful dedicated computer, the costs of such systems in the present-day market will be very high. Thus these systems cannot, as yet, be competitive with the present day analogue and analytical plotters. One can imagine that the digital photogrammetric systems, like analytical plotters, will be introduced first of all at the top end of the market, e.g. being sold to defence mapping and intelligence gathering agencies which are less price sensitive than commercial mapping companies. Therefore, in spite of all the investigations carried out in recent years and the undoubted progress in realizing a digital station, it should be made clear that still a considerable amount of research needs to be carried out and that computer technology has to be developed still further before the digital stations become production-type systems at a price which makes them competitive with present-day analogue and analytical plotters.

11.7 A Final Note

Ending this thesis on a personal note, this research work has greatly benefited the author in that it enabled him to enhance quite significantly his knowledge in different fields such as analytical photogrammetry, image processing, computer graphics, information theory, etc. Closely associated with this has been the requirement to define and formulate problems and to design experiments and to conduct analyses which provide solutions to these problems. Yet another important benefit which resulted from undertaking this research work, has been the experiences which have been gained during the various stages of the project. Undoubtedly they have enabled the author to tackle the numerous problems arising during the different stages of this research work. This project can also be regarded as being beneficial in that it enabled the author to participate in the Kelvin Project for numerically intensive computations on the IBM 3090 computer and to be regarded as one of the Kelvin Project pathfinders. The author has also had the opportunity to present papers and demonstrate the three-dimensional stereo-viewing of the overlapping digitized aerial photos implemented on the 5080 graphics terminal, during the Kelvin Project Symposia held at the University of Glasgow. Again this experience has been highly beneficial and very satisfying and undoubtedly will be helpful in the author's future career.

BIBLIOGRAPHY

Ackermann, F., 1984. Digital image correlation performance and potential application in photogrammetry. *Photogrammetric Record*, 11(64):429-439.

Ackermann, F. and Schneider, W., 1986. High precision aerial triangulation with point transfer by digital image correlation. *International Archives of Photogrammetry and Remote Sensing*, 26(3/1):18-27.

Albertz, J. and Koenig, G., 1984. A digital stereophotogrammetric system. *International Archives of Photogrammetry and Remote Sensing*, 25/A2:1-7.

Almroth, U., 1986. Digital photogrammetry: pixel size, image quality and noise considerations. *Fotogrammetriska Meddelanden*, 2(51):16 pages.

Andrews, H. C. and Hunt, B. R., 1977. *Digital image restoration*. Prentice-Hall, Englewood Cliffs, New Jersey.

Amin, A. M., 1986. *Geometrical analysis and rectification of thermal infrared video frame scanner imagery and its potential applications to topographic mapping*, Ph.D. thesis, University of Glasgow, 334 pages.

Azizi, A. and Petrie, G., 1989. The implementation of a purely digital stereo-photogrammetric system on the IBM 3090. *The Second Kelvin Project Symposium of Numerically Intensive Computing with the IBM Vector Facility*, University of Glasgow.

Bähr, H. P., 1980. Analogue versus digital image processing of photogrammetric imagery. *International Archives of Photogrammetry*, 23/B3:41-55

Baz, I., 1984. *The solution of aerial triangulation on vector and parallel array processors*. Ph.D. thesis, University of Glasgow, 1:237 pages.

Baz, I. and Methley, B. D. F., 1985. Block adjustment with array and vector processors. *Photogrammetric Record*, 11(65):543-566.

Benny, A. H., 1981. Automatic relocation of ground control points in Landsat imagery. *Proceedings of an International Conference on Matching Remote Sensing Technologies and their Applications*, London: 307-315.

Barnard, S. T. and Thompson, W. B., 1980. Disparity analysis of images, *IEEE Trans. Patt. Anal. Mach. Intel.*, PAMI-2(4):333-340.

Benard, M., 1984. Automatic stereophotogrammetry: a method based on feature detection and dynamic programming. *Photogrammetria*, 39:169-181.

Benard, M., Boutaleb, A. K., Kölbl, O. and Penis, C., 1986. Automatic stereo-photogrammetry: implementation and comparison of classical correlation methods and dynamic programming based techniques. *International Archives of Photogrammetry and Remote Sensing*, 26/1:10 pages.

Bernstein, R., Fernyhough, J.R. and Dallam, G., 1975. Digital image processing. *Photogrammetric Engineering and Remote Sensing*, 41(12):1465-1476.

Bertram, S., 1963. The automatic map compilation system. *Photogrammetric Engineering*, 29(4):659-675.

Bertram, S., 1965. The universal automatic map compilation equipment. *Photogrammetric Engineering*, 31(2):244-260.

Best, R. G., Scarpace, F. L. and Worrell, K. J., 1987. Producing simulated continuous tone images with a dot-addressable printer. *Technical Papers, ASPRS - ACSM Annual Convention*, 6:148-157.

Bethel, J., 1986. The DSR11 image correlator. *Technical Papers, ACSM - ASPRS Annual Convention*, 4:44-49.

Case, J. B., 1982. The digital stereo comparator/compiler (DSCC). *International Archives of Photogrammetry*, 24/2:23-29.

Castleman, K. R., 1979. *Digital image processing*. Prentice-Hall, Englewood Cliffs, New Jersey:429 pages.

Chen, C. F., Worrell, K. J., Bergen, W. R. and Scarpace, F. L., 1987. Application of frequency filtering in remotely sensed imagery. *Technical Papers, ASPRS-ACSM Annual Convention*, 6:98-107.

Claus, M., 1984. Digital terrain models through digital stereo correlation. *Photogrammetria*, 39:183-192.

Cogan, L., Gugan, D., Hunter, D., Lutz, S. and Peny C., 1988. Kern DSP1 - digital stereo photogrammetric system. Comm. II, *Presented Paper, 16th International Congress of Photogrammetry and Remote Sensing*, Kyoto: 13 pages.

Cowlishaw, M. F., 1985. Fundamental requirements for picture presentation. *Proceedings of the Society for Information Display*, 26/2:101-107.

Craig, M. D. and Green, A. A., 1987. One-dimensional resampling of digital imagery to sparse output grids. *Photogrammetric Engineering and Remote Sensing*, 53(5):501-106.

Davis, W. A. and Kenue, S. K., 1978. Registration of digital images. *Presented paper to the 5th Canadian Symposium on Remote Sensing*, Victoria, Canada.

Day, T. and Muller, J. P., 1988. Quality assessment of digital elevation models produced by automatic stereomatchers from SPOT image pairs. *Photogrammetric Record*, 12(72):797-808.

Dewitt, B. A. and Wolf, P. R., 1984. Experiences in automated generation of DTMs from aerial photos. *International Archives of Photogrammetry and Remote Sensing*, 25/A4:110-117.

Dorrer, E., 1977. The impact of digital image processing on photogrammetry and cartography: present realities, future potentials. *Proceedings of the International Symposium on Image Processing, Interactions with Photogrammetry and Remote Sensing*, Graz:41-45.

Dowman, I. J., 1984. Problems and some solutions in digital correlation for photogrammetric profiling. *Photogrammetria*, 39:155-167.

Dowman, I. J., 1987. Digital processing of SPOT data. *Proceedings of ISPRS Intercommission Conference on Fast Processing of Photogrammetric Data (Interlaken)*. Institut für Geodäsie und Photogrammetrie, ETH, Zürich:318-330.

Dowman, I. J. and Haggag, A., 1977. Digital image correlation along epipolar lines. *Proceedings of the International Symposium on Image Processing, Interactions with Photogrammetry and Remote Sensing*, Graz:47-49.

Ducher, G., 1971. Le Stéréomat Wild B-8 de IGN. *50 years Wild Heerbrugg Jubilee Volume*, Geodesy and Photogrammetry:68-73.

Eales, P. and Muller, J. P., 1988. The IAX image processing language and the IBM AT/370. *Small Computers in Image Processing and Mapping, Proceedings of Two One-Day Meetings at the Universities of Glasgow and Nottingham* :30-33

Ebner, H., and Heipke, C., 1988. Integration of digital image matching and object surface reconstruction. *International Archives of Photogrammetry and Remote Sensing*, 27/B11:III 534-III 545.

Ehlers, M., 1980. Filter techniques and their application in digital correlation. *International Archives of Photogrammetry*, 23/B3:193-201.

Fallvik, J. Y., 1986. Image restoration methods as preprocessing tools in digital stereo matching. *International Archives of Photogrammetry and Remote Sensing*, 26(3/1):223-236.

Floyd, R. W. and Steinberg, L., 1976. An adaptive algorithm for special greyscale. *Proceedings of the Society for Information Display*, 17/2(2):75-77.

Flynn, J. E., 1966. Very high speed computing systems. *Proceedings of IEEE*, 54(12):1901-1909.

Förstner, W., 1982. On the geometric precision of digital correlation. *International Archives of Photogrammetry*, 24/3:176-189.

Förstner, W., 1984. Quality assessment of object location and point transfer using digital image correlation techniques. *International Archives of Photogrammetry and Remote Sensing*, 25/A3a:197-219.

Gabel, R. A. and Roberts, R. A., 1973. *Signals and linear systems*. John Wiley and Sons:415 pages.

Gambino, L. A. and Crombie, M. A., 1974. Digital mapping and digital image processing. *Photogrammetric Engineering*, 40(11):1295-1302.

Gaskill, J. D., 1978. *Linear systems, Fourier transforms, and optics*. John Wiley and Sons:554 pages.

Gordon, M. R., 1981. Ground control pointing of the UK. *Proceedings of an International Conference on Matching Remote Sensing Technologies and Their Applications*, London:301-306.

Gruen, A. W., 1988. Digital photogrammetric processing system - current status and prospects, *International Archives of Photogrammetry and Remote Sensing*, 27/B10:II 342-II 351.

Gruen, A. W. and Baltsavias, E. P., 1986. High precision image matching for digital terrain model generation, *International Archives of Photogrammetry*, 26(3/1):284-296.

Gruen, A. W. and Baltsavias, E. P., 1987. Geometrically constrained multiphoto matching, *Proceedings ISPRS Intercommission Conference on Fast Processing of Photogrammetric Data, (Interlaken)*. Institut für Geodäsie und Photogrammetrie, ETH, Zürich.

Gugan, D. J. and Dowman, I. J., 1986. Design and implementation of a digital photogrammetric system. *International Archives of Photogrammetry and Remote Sensing*, 26/2:100-109.

Hamming, R. W., 1983. *Digital filters.* Prentice-Hall, Englewood Cliffs, New Jersy:257 pages.

Hannah, M. J., 1981. Error detection and correction in digital terrain models. *Photogrammetric Engineering and Remote Sensing*, 47(1):63-69.

Hardy, J. W. and Johnson, H. R., 1969. Electronic correlator for the Planimat. *Photogrammetric Engineering*, 35(8):780-788.

Havelock, D., 1984. Geometric precision in digital images. *International Archives of Photogrammetry and Remote Sensing*, 25:381-391.

Helava, U. V., 1976. Digital correlation in photogrammetric instruments. *International Archives of Photogrammetry*, 21(2):24 pages.

Helava, U. V., 1980. The concept of the analytical plotter. *Proceedings A. S. P. Analytical Plotter Symposium and Workshop*:12-29.

Helava, U. V., 1982. Fundamentals of stereo scanning. *International Archives of Photogrammetry and Remote Sensing*, 24/2:182-199.

Helava, U. V., 1987. Digital comparator correlator system. *Proceedings ISPRS Intercommission Conference on Fast Processing of Photogrammetric Data, (Interlaken)*. Institut für Geodäsie und Photogrammetrie, ETH, Zürich:404-418.

Helava, U. V., 1988a. Object-space least squares correlation. *Photogrammetric Engineering and Remote Sensing*, 54/1(6):711-714.

Helava, U. V., 1988b. On system concepts for digital automation. *Photogrammetria*, 43:57-71.

Helava, U. V. and Chapelle, W. E., 1972. Epipolar-scan correlation. *Bendix Technical Journal*, 5(1):19-23.

Hobrough, G. L., 1959. Automatic stereo plotting. *Photogrammetric Engineering*, 25:736-769.

Hockney, R. W. and Jesshope, C. R., 1981. *Parallel computers, programming and algorithms.* Hilger Press, Bristol.

Hössler, R., 1980. Differential rectification of digital or digitized imageries. *International Archives of Photogrammetry*, 23/B3:294-303.

Hueckel, M. H., 1971. An operator which locates edges in digitized pictures. *J. Ass. Comput. Mach.*, 18:113-125.

Hunt, D. J., 1989. AMT DAP - a processor array in a workstation environment. *Computer Systems Science and Engineering*, 4(2):107-114.

Jackson, D. H., 1984. *Experience with the IAX image processing system. Digital Signal Processing*, Elsevier Science Publishers B.V., North-Holland.

Keating, T. J. and Boston, D. R., 1979. Digital orthophoto production using scanning microdensitometers. *Photogrammetric Engineering and Remote Sensing*, 45(6):735-740.

Keating, T. J., Wolf, P. R. and Scarpace, F. I., 1975. An improved method of digital image correlation from scanned densitometric data. *Photogrammetric Engineering and Remote Sensing*, 41(8):993-1002.

Kennie, T. J. M. and Petrie, G., 1990. *Engineering surveying technology*. John Wiley and Sons, Inc. New York:485 pages.

Kibblewhite E. J., 1981. Automatic measurement of astronomical photographs. *Photogrammetric Record*, 10(58):427-433.

Koenig, G., Nickel, W. and Storl, J., 1986. Processing of scanning electron microscope imagery in a digital stereophotogrammetric system (DSS). *International Archives of Photogrammetry and Remote Sensing*, 26(2):130-138.

Koenig, G., Nickel, W. and Storl, J., 1988. Digital stereophotogrammetry - experimental system. *International Archives of Photogrammetry and Remote Sensing*, 27/B10:II 326-II 331.

Kölbl, O. and Silva, I. da, 1988. Derivation of a digital terrain model by dynamic programming. Comm. III, *Presented Paper, 16th International Congress of Photogrammetry and Remote Sensing*, Kyoto:10 pages.

Konecny, G., 1979. Methods and possibilities for digital differential rectification. *Photogrammetric Engineering and Remote Sensing*, 45(6):727-734.

Konecny, G., 1981. Correlation techniques and devices. *Photogrammetric Engineering and Remote Sensing*, 47(3):323-333.

Konecny, G., Bahr, H. P., Reil, W. and Schreiber, H. C., 1979. *Use of spaceborne metric cameras for cartographic applications.* Report of the Institute of Photogrammetry and Engineering Surveys by Order of Ministry of Research and Technology, Federal Republic of Germany:1-165.

Kowalski, D. C., 1968. A comparison of optical and electronic correlation techniques. *International Archives of Photogrammetry*, 17/6:9 pages.

Kraus, K., Otepka, G., and Loitsch, H. J., 1979. Digitally controlled production of orthophotos and stereo-orthophotos. *Photogrammetric Engineering and Remote Sensing*, 45(10):1353-1362.

Kunji, B., 1984. Experiments in digital processing of photogrammetric images. *International Archives of Photogrammetry*, 25/A2:298-304.

Lambert, B. P., 1971. Super-wide angle photogrammetry and orthophoto mapping in the Australian Federal Mapping Programmes. *50 years Wild Heerbrugg Jubilee Volume*, Geodesy and Photogrammetry:68-73.

Li, M. X., 1988. High precision relative orientation using feature based techniques. *International Archives of Photogrammetry and Remote Sensing*, 27/B3:456-465.

Li, M. X., 1989. *Hierarchical multi-point matching with simultaneous detection and location of break lines.* Ph.D. thesis, The Royal Institute of Technology, Stockholm:177 pages.

Li, Z., 1990. *Sampling strategy and accuracy assessment for digital terrain modelling.* Ph.D. thesis, University of Glasgow:229 pages.

Lin, Z., 1986. Multi-criterion for similarity assessment in photogrammetric image correlation. *International Archives of Photogrammetry and Remote Sensing*, 26(3/2):415-426.

Lohmann, P., Picht, G., Weidenhammer, J., Jacobsen, K. and Skog, L., 1989. The design and development of a digital photogrammetric stereo work station. *ISPRS Journal of Photogrammetry and Remote Sensing*, 44:215-224.

Lü, Y. and Zhang, Z., 1988. Fast implementation for generating epipolar line images with one-dimensional resampling. *International Archives of Photogrammetry and Remote Sensing*, 27/B3:511-520.

Luhmann, T. and Altrogge, G., 1986. Interest-operator for image matching. *International Archives of Photogrammetry and Remote Sensing*, 26/2:459-474.

MacGillivray, H. T. and Stobie, R. S., 1984. New results with the COSMOS machine. *Vistas in Astronomy*, 27(4):433-475.

McCullagh, M., 1983. *If you're sitting comfortably we'll begin.* Workshop notes on terrain modelling, Australian Computing Society, Siren Systems:158 pages.

Manual of Photogrammetry. 1980. *American Society of Photogrammetry*, 4th edition.

Makarovič, B., 1980a. Automatic off-line generation of digital terrain models. *International Archives of Photogrammetry*, 23/B2:139-157.

Makarovič, B., 1980b. Image correlation algorithms. *International Archives of Photogrammetry*, 23/B2:167-175.

Makarovič, B., 1984. Automatic production of DTM data using digital off-line technique. *ITC Journal*. (1984-2):135-141.

Masry, S. E., 1974. Digital correlation principles. *Photogrammetric Engineering*, 40(3):303-308.

Mather, P. M., 1987. *Computer processing of remotely sensed images (an introduction)*. John Wiley and Sons, New York: 352 pages.

Methley, B. D. F., 1986. Computational models in surveying and photogrammetry. Glasgow:346 pages.

Mikhail, E. M., Akey, M. L. and Mitchel, O. R., 1984. Detection and sub-pixel location of photogrammetric targets in digital images. *Photogrammetria*, 39:63-83.

Moffit, F. H. and Mikhail, E. M., 1980. *Photogrammetry*. Harper and Row, New York, Third Edition:648 pages.

Moik, J. G., 1980. *Digital processing of remotely sensed images*. NASA Scientific and Technical Branch, Publication No. SP-431, Washington DC: 330 pages.

Muller, J-P., Collins, K. A., Otto, G. P. and Roberts, J. B. G., 1988. Stereo matching using transputer arrays. *International Archives of Photogrammetry and Remote Sensing*, 27/B3:559-585.

Norvelle, F. R., 1981. Interactive digital correlation techniques for automatic compilation of elevation data. *Technical Papers, ASP 47th Annual Meeting* :554-567.

O'Neill, E. L., 1963. *Introduction to statistical optics*. Addison - Wesley, Reading, Massachusetts:179 pages.

O'Neill, M. A. and Dowman, I. J., 1988. The generation of epipolar synthetic stereo mates for SPOT images using a DEM. *International Archives of Photogrammetry and Remote Sensing*, 27/A1:587-598.

Or, W., 1987. A more expedient way to visit and travel in the frequency domain. *Technical Papers, ASPRS-ACSM Annual Convention*, 2:161-170.

Otto, G. P., 1988. Rectification of SPOT data for stereo image matching. *International Archives of Photogrammetry and Remote Sensing*, 27/A1:635-645.

Panton, D. J., 1978. A flexible approach to digital stereo mapping. *Photogrammetric Engineering and Remote Sensing*, 44(12):1499-1512.

Persoon, E. and Fu, K. S., 1977. Shape discrimination using Fourier descriptors. *IEEE Transactions on Systems, Man, and cybernetics*, CMS-7(3):170-179.

Pertl, A., 1985. Digital image correlation with the analytical plotter Planicomp C 100. *Photogrammetria*, 40(1):9-19.

Petrie, G., 1983. The philosophy of digital and analytical photogrammetric systems. *Proceedings of the 39th Photogrammetric Week*, University of Stuttgart:53-68.

Petrie, G., 1988. Developments in analytical instrumentation. Comm. II, *Invited Paper, 16th International Congress of Photogrammetry and Remote Sensing*, Kyoto:17 pages.

Petrie, G., 1990. *Analogue, analytical and digital photogrammetric systems applied to aerial mapping*. in Engineering surveying technology, edited by Kennie, T. J. M. and Petrie, G.:238-288.

Pratt, W. K., 1978. *Digital image processing*. John Wiley and Sons, New York.

Roetling, P. G., 1977. Binary approximation of continuous tone images. *Photographic Science and Engineering*, 21(2):60-65.

Rosenfeld, A. and Kak, A. C., 1976. Digital picture processing. *Academic Press*, New York: 457 pages.

Rosenholm, D., 1985. Digital matching of simulated SPOT-images. *Fotogrammetriska Meddelanden*, 2(50):134 pages.

Rosenholm, D., 1986. Accuracy improvement of digital matching for evaluation of digital terrain models. *International Archives of Photogrammetry and Remote Sensing*, 26(3/2):573-587.

Rosenholm, D., 1987. Multi-point matching the least-squares technique for evaluation of three-dimensional models. *Photogrammetric Engineering and Remote Sensing*, 53(6):621-626.

Sarjakoski, T., 1981. Concept of a completely digital stereoplotter. *The Photogrammetric Journal of Finland*, 8(2):95-100.

Scarano, F. A. and Brumm, G. A., 1976. A digital elevation data collection system. *Photogrammetric Engineering and Remote Sensing*, 42(4):489-496.

Sharp, J. V., Christiensen, R. L., Gilman, W. C. and Schulmann, F. D., 1965. Automatic map compilation using digital techniques. *Photogrammetric Engineering*, 31(2):223-239.

Stark, E., 1976. The effect of angular field on horizontal and vertical accuracy in photogrammetric plotting. *Proceedings 35th Photogrammetric Week*, Stuttgart:129-145.

Stokes, J., 1986. Are phase shift methods suitable for image matching in photogrammetry? *Fotogrammetriska Meddelanden*, 2(51):24.

Stokes, J., 1987. A photogrammetric monoplotter for digital map revision using the image processing system GOP-300. *International Archives of Photogrammetry and Remote Sensing*, 27(B2):424-432.

Tempfli, K. and Makarovič, B., 1978. Transfer functions of interpolation methods. *ITC Journal*, (1978-1):50-80.

Thurgood, J. D. and Mikhail, E. M., 1982. *Sub pixel mensuration of photogrammetric targets in digital images.* Technical Report, School of Civil Engineering, Purdue University, CE-PH-82-2:252 pages.

Torlegård, A. K. I., 1986. Some photogrammetric experiments with digital image processing. *Photogrammetric Record*, 12(68):175-196.

Trinder, J. C., 1971. Pointing accuracies to blurred signals. *Photogrammetric Engineering*, 37:192-202.

Trinder, J. C., 1984. Pointing precision on aerial photography. *Photogrammetric Engineering and Remote Sensing*, 50(10):1449-1462.

Trinder, J. C., 1986. Precision of stereoscopic height measurements. *Photogrammetric Engineering and Remote Sensing*, 52(1):75-79.

Trinder, J. C., 1987. Measurements on digitized hardcopy images. *Photogrammetric Engineering and Remote Sensing*, 53(3):315-321.

Whitehead, F. R., Krulikoski, J.R. S. J. and Kowalski, D. C., 1968. Coherent optical parallel processing. *International Archives of Photogrammetry*, 17/6:7 pages.

Wiesel, J. W., 1984. Image rectification and registration. *International Archives of Photogrammetry and Remote Sensing*, 25/A3b:1120-1126.

Wiesel, J. W., 1985. Digital image processing for orthophoto generation. *Photogrammetria*, 40(2)69-76.

Wiesel, J. W. and Behr, F. J., 1987. Digital orthophoto generation using the Kern DSR-11 analytical stereo restitution instrument. *Technical Papers, ASPRS-ACSM Annual Convention*, 2:216-224.

Williams, P. R. and Pratt, N. M., 1975. Looking at galaxies with Cosmos. *Spectrum*, (136):9-12.

Wolf, P. R., 1985. *Elements of photogrammetry*. MacGraw-Hill, Inc, Singapore, Second Edition:628 pages.

Wolf, P. R. and Dewitt, B. A., 1982. An automated photogrammetric mapping system. *International Archives of Photogrammetry*, 24:169-178.

Wolf, P. R. and Dewitt, B. A., 1984. Correction for systematic errors of a drum-type scanning microdensitometer. *International Archives of Photogrammetry* and *Remote Sensing*, 25/A3b:1147-1152.

Wong, K. W. and Wei-Hsin, H., 1986. Close-range mapping with a solid state camera. *Photogrammetric Engineering and Remote Sensing*, 52(1):67-74.

Zhou, G., 1986. Accurate determination of ellipse centers in digital imagery. *Technical Papers, ACSM-ASPRS Annual Convention*, 4:256-264.

APPENDICES

A1 Programs GRAD and PTSUB

The program GRAD was developed to measure the coordinates of the cross centre and other cross parameters (i.e. mean cross and mean background grey levels and cross orientation) automatically using the Gradient method.

The program PTSUB is a subroutine which is called from within the program GRAD. It is used to improve the cross parameters calculated by the first method. The program employs the image modelling and least squares (IML) image matching approach.

A1.1 Definition of Some of the Variables

MMR	Maximum number of image columns
NR	Maximum number of image rows
TMD	Cross mean grey level
BMD	Mean background grey level
TIOD, BIOD	Cross and background integrated optical densities
ARET, AREB	Cross and background area function
IHH	Initial threshold value
THRM	The second threshold value
B	The distance between the two cross legs with the same orientation (see Fig. A1).
EL	The length of a single cross leg (see Fig. A1).
SPCH	Width of the cross leg
THETA	Cross orientation with respect to the scan lines.
C1	The calculated Gradient along the row directions.
C2	The calculated Gradient along the Column directions.
GRAD	The calculated Gradient in two dimensions.
JF1	The column number of the starting position of the left cross leg.
JL1	The column number of the ending position of the left cross leg.
XLT, YLY	Row and column numbers of the midpoint of the left cross leg.
WID1	The calculated width of the left cross leg.
JF2	The column number of the starting position of the right cross leg.
JL2	The column number of the ending position of the right cross leg.
XRT, YRT	Row and column numbers of the midpoint of the right cross leg.
WID2	The calculated width of the right cross leg.

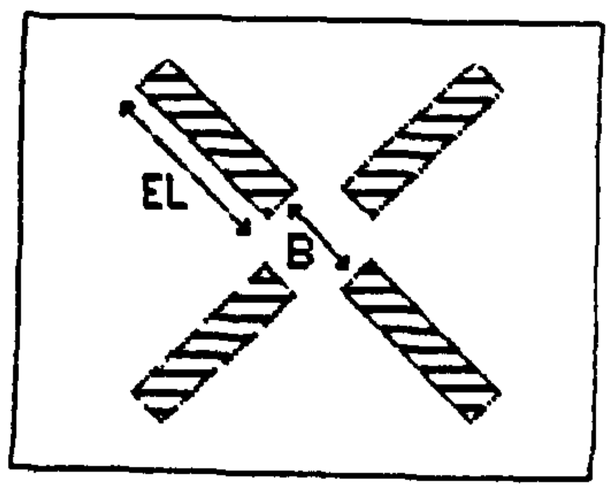


Fig. A1

A1.2 Definition of Some of the Arrays

IM	Matrix containing the digital image
CL, CR	Matrices containing the slope and constant of the lines

fitted to the midpoints with the same orientations.

XL,YL
XR, YR — Matrices containing the x and y-coordinates of the midpoints of the cross legs with the same orientation.

XL1, XR1 — Working matrics containing the x values of the left and right cross legs respectively.

U — Matrix containing the calculated cross parameters, i.e:

U(1) - Cross mean grey level
U(2) - Background mean grey level
U(3) - x coordinate of the cross centre
U(4) - y coordinate of the cross centre
U(5) - cross orientation angle

A1.3 Definition of Some of the Variables for Subroutine PTSUB

D — Width of the point spread function
W — Width of cross
A — Distance between two cross legs
SIGN — Sum of squared residuals between the grey levels of the modelled and original crosses.

A1.4 Definition of Some of the Arrays for Subroutine PTSUB

IMF — Matrix of modelled image

F — Matrix of discrepancies between the grey levels of the modelled and original image.

UX — Matrix containing the components of the simulated cross.

B — Coefficient matrix.

EN — Normal matrix.

A1.5 Explanation of Program GRAD and PTSUB

Block 1: Arrays are declared; the image patch containing the cross is input to the program.

Block 2: Minimum and maximum grey levels of the input image are determined and the integrated optical density and area function is calculated for the cross and its background; cross and its background mean grey levels are calculated and a threshold value based on these values is computed.

Block 3: The Gradient magnitude is calculated for the current pixel; the beginning and the ending position of the cross leg is identified; the midpoints for the cross leg is determined (the procedure is carried out for both the left and right cross legs).

Block 4: The calculated midpoints for the legs with the same orientations are identified and exchanged.

Block 5: A line is fitted to the midpoints with the same orientation; the coordinates of the cross centre are calculated

by the intersection of the lines; the cross orientation angle is determined.

Block 6: The calculated cross parameters are input to the subroutine PTSUB.

Block 7: The user selects the cross type; and the conditions for terminating the iterations are input to the program.

Block 8: The geometry of the simulated cross is established.

Block 9: The coefficients of the observation equations are calculated; the modelled cross is formed; the discrepancies between the modelled image and the original image are computed; the normal equation is formed.

Block 10: The normal equations are solved; if the conditions for terminating the iterations are fulfilled, the program is ended and the results are output.

Block 11: Contains the following subroutines:

1- subroutine AREA : calculates the area under a Gaussian curve.

2- subroutine SLVNRM : solves the normal equation

3- subroutine EXCHG : arranges the detected midpoints of the cross legs with the same orientation in a single array.

4- subroutine FIT : fits a line to the detected midpoints with the same orientation.

5- subroutines ZEROM1 and ZEROM2 : initialize one and two dimensional arrays respectively.

6- subroutine MXMN : determines the maximum and minimum values of an image.

7- subroutine CONT : calculates optical density and area function.

A1.6 Listing of Program GRAD

```
C     ************************************************************
C     **  Program GRAD measures the coordinates of the centre   **
C     **  of the cross and other cross parameters using the     **
C     **  Gradient operator. The calculated cross parameters are **
C     **  improved using subroutine PTSUB which employs the image **
C     **  modelling and least squares (IML) approach.          **
C     ************************************************************
C
C
      PROGRAM GRAD
      PARAMETER(NR=256,MMR=384)
      INTEGER*2 IM(NR,MMR)
      REAL CL(4),CR(4),U(10),                                    BLOCK(1)
     1XL(NR),YL(NR),XR(NR),YR(NR),XL1(NR),XR1(NR)
      CHARACTER*1 A
      DO 15 I=1,NR
   15 READ(2) (IM(I,J),J=1,MMR)
C ------------------------------------------------------------------
      CALL MXMN(IM,IH2,IH1,NR,MMR)
      IHH=(IH1+IH2)/2
      CALL CONT(IM,IHH,ARET,AREB,TIOD,BIOD,NR,MMR)
      TMD=TIOD/ARET
      BMD=BIOD/AREB
      THRM=SQRT(BMD*BMD+TMD*TMD)
      PRINT*,' '
      PRINT*,'##############################################'
      PRINT*,' '
      PRINT*,'MAX.DENSITY VALUE "H2"=',IH2
      PRINT*,'MIN. DENSITY VALUE "H1"=',IH1
      PRINT*,' '
      PRINT*,'----------------------------------------------'
      PRINT*,' '
      PRINT*,'TARGET INTEGRATED OPTICAL DENSITY',TIOD
      PRINT*,'TARGET THRESHOLD AREA FUNCTION',ARET
      PRINT*,'MEAN TARGET DENSITY',TMD
      PRINT*,' '
      PRINT*,'BACKGROUND INTEGRATED OPTICAL DENSITY',BIOD
      PRINT*,'BACKGROUND THRESHOLD AREA FUNCTION',AREB
      PRINT*,'MEAN BACKGROUND DENSITY',BMD
```

```
      PRINT*,' '
      PRINT*,'CALCULATED THRESHOLD VALUE',THRM                    BLOCK(2)
      PRINT*,' '
      PRINT*,'##################################################'
      PRINT*,' '
C     ------------------------------------------------------------
      PRINT*,'TYPE THE DISTANCE BETWEEN TWO CROSS LEGS'
      PRINT*,'AND THE LENGTH OF THE LEG'
      READ*,B,EL
      PRINT*,'TYPE "CROSS WIDTH"'
      READ*,SPCM
      SPC=SPCM
      PRINT*,'SELECT THE EDGE DETECTION OPERATOR:'
      PRINT*,'1- SMOOTHED GRADIENT'
      PRINT*,'2- GRADIENT'
      READ*,C
      PRINT*,'TYPE REDUCTION FACTOR FOR THRESHOLD VALUE'
      READ*,TH
C
      KK=0
      MC=MMR
      MR=NR
      THR=THRM
 17   SPC=SPCM
      THR=THR-TH
      IF(THR.LT.THRM/4.5) THEN
      PRINT*,'NO CROSS CAN BE DETECTED'
      STOP
      END IF
 11   I=3
      ACC=0
      II=0
 10   J=3
      JF1=0
      JL1=0
      JF2=0
      JL2=0
```

```
      XLT=0
      XRT=0
      WID1=0
      WID2=0
C
 20   IF(C.EQ.1) THEN
      C1=(IM(I+1,J)+IM(I+2,J))/2-(IM(I-1,J)+IM(I-2,J))/2
      C2=(IM(I,J+1)+IM(I,J+2))/2-(IM(I,J-1)+IM(I,J-2))/2
      ELSE
      C1=IM(I+1,J)-IM(I-1,J)
     C2=IM(I,J+1)-IM(I,J-1)                                    BLOCK(3)
      END IF
      GRAD=SQRT(C1*C1+C2*C2)
C
      IF(GRAD.GT.THR.AND.XLT.GT.0) THEN
      IF(C2.GT.0) JF2=J
      IF(C2.LT.0) JL2=J
      IF(JF2.GT.0.AND.JL2.GT.0) WID2=JL2-JF2
      END IF
      IF(WID2.GE.SPC) THEN
      II=II+1
      XR(II)=JF2+WID2/2
      YR(II)=I
      XR1(II)=XR(II)
      XRT=XR(II)
      WID2=0
      END IF
C
C
      IF(GRAD.GT.THR.AND.XLT.EQ.0) THEN
      IF(C2.GT.0) JF1=J
      IF(C2.LT.0) JL1=J
      IF(JF1.GT.0.AND.JL1.GT.0) WID1=JL1-JF1
      END IF
      IF(WID1.GE.SPC) THEN
      XLT=JF1+WID1/2
      YLT=I
```

```
      WID1=0
      END IF
C
C
      IF(XRT.GT.0) THEN
      XL(II)=XLT
      XL1(II)=XL(II)
      YL(II)=YLT
      GO TO 55
      END IF
C
      J=J+1
      IF(J.LT.(MC-2)) GO TO 20
 55   I=I+1
      IF(I.LT.MR-2) GO TO 10
      PRINT*,'CURRENT VALUES FOR THE CROSS AND THRESHOLD ARE:'
      PRINT*,SPC,THR
      IF(II.LT.2) THEN
      SPC=SPC-1
      IF(SPC.LT.1) GO TO 17
      GO TO 11
      END IF
C
C ----------------------------------------------------------------
      N=1
      IF(KK.EQ.2) N=2
 16   N=N+1
      IF(KK.EQ.3.OR.N.GT.II) THEN
      PRINT*,'++++++++++++++++++++++++++++++++++++++++'
      PRINT*,'NO BETTER RESULT CAN BE OBTAINED'
      STOP
     END IF                                          BLOCK(4)
      SL=XL1(N)-XL1(N-1)
      IF(SL.LT.0.OR.ACC.GE.1) THEN
      ACC=ACC+1
      M=N
      IF(KK.EQ.1) M=N-1
```

```
      IF(KK.EQ.2) M=N-2
      CALL EXCHG(M,XL,XR,NR)
      END IF
      IF(N.LT.II) GO TO 16
      IF(KK.EQ.1.OR.KK.EQ.2) CALL EXCHG(II,XL,XR,NR)
      III=II-1
      IF(KK.EQ.2) CALL EXCHG(III,XL,XR,NR)
      PRINT*,' '
      WRITE(*,800)
      DO 23 I=1,II
 23   WRITE(*,200) XL(I),YL(I),XR(I),YR(I)
      PRINT*,' '
C
C ------------------------------------------------------------------
      CALL FIT(XL,YL,II,CL)
      CALL FIT(XR,YR,II,CR)
C
      PRINT*,'CL',CL(1),CL(2)
      IF((CL(1)-CR(1)).EQ.0) THEN
      PRINT*,'THE COEFFICIENT MATRIX IS SINGULAR'
      PRINT*,'THE PROGRAM WILL BE TRIED AGAIN BY INTERCHANGING'
      PRINT*,'THE LEFT AND RIGHT CENTRE POINTS'
      KK=KK+1
      GO TO 11
      END IF                                              BLOCK(5)
C
      IF(CL(1).EQ.0.AND.CL(2).EQ.0) THEN
      FX=XL(1)
      FY=CR(1)*FX+CR(2)
      T1=1.571
      T2=ATAN(CR(1))
      GO TO 222
      ELSE IF(CR(1).EQ.0.AND.CR(2).EQ.0) THEN
      FX=XR(1)
      FY=CL(1)*FX+CL(2)
      T1=ATAN(CL(1))
      T2=1.571
```

```
      GO TO 222
C
      END IF
      FX=(CR(2)-CL(2))/(CL(1)-CR(1))
      FY=CL(1)*FX+CL(2)
      T1=ATAN(CL(1))
      T2=ATAN(CR(1))
  222 THETA1=-T1*180/3.14
      THETA2=-T2*180/3.14
      PRINT*,'-------------------------------------------------------'
      WRITE(*,300) FX,FY,THETA1,THETA2
      WRITE(*,400) BMD,TMD
      PRINT*,' '
      PRINT*,'-------------------------------------------------------'
C
      DEL=180+THETA1-THETA2
      IF(DEL.GT.93.OR. DEL.LT.87) THEN
      PRINT*,'THE PROGRAM WILL BE TRIED AGAIN BY INTERCHANGING'
      PRINT*,'THE LEFT AND RIGHT CENTRE POINTS'
      KK=KK+1
      GO TO 11
      END IF
C ------------------------------------------------------------------
      U(1)=BMD
      U(2)=TMD
      U(3)=FX
      U(4)=-FY
      U(5)=(1.5714+T1+T2)/2
      PRINT*,'U5',U(5)
      W=SPCH                                          BLOCK(6)
C
      CALL PTSUB(U,IM,B,EL,W)
C
C     PRINT*,'DO YOU WANT TO RECORD THE COORDS.(Y,N)?'
C     READ(*,700) A
C     IF(A.EQ.'Y') THEN
C     PRINT*,'ENTER THE NAME OF THE FILE'
```

```
C       READ(*,700) NAME
C       OPEN(12,FILE=NAME,STATUS='NEW',FORM='UNFORMATTED')
C       WRITE(12) U(3),U(4)
C       END IF
 800  FORMAT(/5X,'XL',9X,'YL',6X,'XR',9X,'YR')
 200  FORMAT(4F9.2)
 300   FORMAT(/1X,'APPROXIMATE VALUES FOR UNKNOWNS:',
     1//5X,'X',9X,'Y',7X,'THETA1',7X,'THETA2',//2F9.2,F11.2,F14.2)
 400   FORMAT(/6X,'H1',9X,'H2',//F9.0,F11.0)
 700   FORMAT(A)
       STOP
       END
C ------------------------------------------------------------------
       SUBROUTINE PTSUB(U,IM,A,EL,W)
C      PTSUB CALCULATES CENTRE OF TARGET WITH SUB PIXEL ACCURACY
C ------------------------------------------------------------------
       PARAMETER(NR=500,MMR=500)
       REAL EN(5,5),SP(4),DL(5),T(5),ARE(4),U(10),G,SIGN,SIGO,SIGCH
        REAL IM(NR,MMR),IMF(NR,MMR),F(NR,MMR),WU(5),
      1P(4),UX(5,4),B(5),P1(4)
C
       SPREAD(X)=EXP(-X*X/(D*D*0.5))/(1.2533141*D)
C
       DO 1 I=1,5
   1   WU(I)=0.0
C
       PRINT*,'ENTER THE TYPE OF CROSS:'
       PRINT*,'1- CROSS WITH CIRCULAR HOLE'
       PRINT*,'2- CROSS WITH RECTANGULAR HOLE'
       PRINT*,'3- UNIFORM CROSS'
       READ*,C                                              BLOCK(7)
       IF(C.EQ.1) THEN
       PRINT*,'ENTER THE RADIUS OF THE CIRCULAR HOLE'
       READ*,R
       END IF
       PRINT*,'ENTER WIDTH OF SPREAD FUNCTION'
       READ*,D
```

```
      PRINT*, 'TYPE MINIMUM ALLOWABLE CHANGE FOR "SIGN"'
      READ*,CRITIC
      PRINT*, 'TYPE MAXIMUM ALLOWABLE ITERATION'
      READ*,MAXITR
      PRINT*, 'TYPE MINIMUM VALUE OF CORRECTION TO "X AND Y"'
      READ*,SPECI
   25 PRINT*, 'TYPE INCREMENT FOR CALCULATING THE AREA UNDER THE CURVE'
      READ*,H
      IF(H.LE.0.0) GO TO 25
      ICONVG=0
      LN=40
      BR=40
      ITR=0
      GAIN=1.0
C
      CALL ZEROM1(U(6),5)
      CALL ZEROM1(DL,5)
C
      SIGN=1.0E+15
C ------------------------------------------------------------------------
  101 ITR=ITR+1
      SIGO=SIGN
      SIGN=0.0
C
      CALL ZEROM2(EN,5,5)
      CALL ZEROM1(T,5)
C
C     SPECIFY IMAGE GEOMETERY
C
      UX(1,1)= +0.5*W
      UX(1,2)= -0.5*W                                        BLOCK(8)
      UX(1,3)= -0.5*A
      UX(1,4)= -EL
C
      UX(2,1)= +EL
      UX(2,2)= +0.5*A
      UX(2,3)= +0.5*W
```

```
      UX(2,4)= -0.5*W
C
      UX(3,1)= +0.5*W
      UX(3,2)= -0.5*W
      UX(3,3)= +EL
      UX(3,4)= +0.5*A
C
      UX(4,1)=-0.5*A
      UX(4,2)=-EL
       UX(4,3)= +0.5*W
       UX(4,4)= -0.5*W
      IF(C.EQ.3) UX(4,1)=+0.5*A
C
C ----------------------------------------------------------------
      SN= SIN(U(5))
      CS= COS(U(5))
C
      LLL=ABS(U(3))-LN
      KKK=ABS(U(4))-BR
      JJJ=ABS(U(3))+LN
      III=ABS(U(4))+BR
      DO 106 I=KKK,III
      DO 106 J=LLL,JJJ
C
      S1= FLOAT(J)
      T1= -FLOAT(I)
C
C
      SS= CS*(S1-U(3))-SN*(T1-U(4))
       TT= SN*(S1-U(3))+CS*(T1-U(4))
C
      IF(C.EQ.1) THEN
      TTT=ABS(TT)
      SSS=ABS(SS)                                          BLOCK(9)
      IF(TTT.GT.R) TTT=R
      IF(SSS.GT.R) SSS=R
      UX(5,1)=SQRT(R*R-SSS*SSS)
```

```
      UX(5,2)=-UX(5,1)
      UX(5,3)=SQRT(R*R-TTT*TTT)
      UX(5,4)=-UX(5,3)
      LL=5
      END IF
C
      CALL ZEROM1(B,5)
C
      LL=4
      IF(C.EQ.1) LL=5
      DO 103 IL=1,LL
C
      P(2)=TT
      P(1)=P(2)
      P(4)=SS
      P(3)=P(4)
C
      DO 102 K=1,4
      P(K)=P(K)+UX(IL,K)
      P1(K)=P(K)
      IF(P(K).GT.5.0.OR.P(K).LT.-5.0) P(K)=5.0
      SP(K)=SPREAD(P(K))

C
      CALL AREA(P1(K),D,ARE(K),H)
C
 102  CONTINUE
C
C     FORM COEFFICIENT MATRIX
C
      B(1)=B(1)+(ARE(1)-ARE(2))*(ARE(3)-ARE(4))
      DST=(SP(1)-SP(2))*(ARE(3)-ARE(4))
      DSS=(SP(3)-SP(4))*(ARE(1)-ARE(2))
C
      B(3)=B(3)-(U(2)-U(1))*(-SN*DST-CS*DSS)
      B(4)=B(4)-(U(2)-U(1))*(-CS*DST+SN*DSS)
      B(5)=B(5)-(U(2)-U(1))*(+SS*DST-TT*DSS)
```

```
C
 103   CONTINUE
C
          B(1)=-1.0+B(1)
          B(2)=-1.0-B(1)
C
C       IMF - MODELLED IMAGE  I.E. IDEAL FUNCTION CONVOLVED WITH -
C             GAUSSIAN SPREAD FUNCTION.
C       F   - DISCREPANCIES BETWEEN MODELLED AND ORIGINAL IMAGE
C       SIGN- SUM OF SQUARED RESIDUALS (DISCREPANCIES)
C
          IMF(I,J)=U(1)-(U(2)-U(1))*B(2)
          F(I,J)=-IM(I,J)+IMF(I,J)
          SIGN=SIGN+F(I,J)*F(I,J)
C ------------------------------------------------------------------
C       FORM NORMAL EQUATIONS
C
          DO 105 II=1,5
          DO 104 JJ=1,5
          EN(II,JJ)=EN(II,JJ)+B(II)*B(JJ)
C
 104   CONTINUE
C
          T(II)=T(II)+B(II)*F(I,J)
C
 105   CONTINUE
 106   CONTINUE
C
          SIGN=SIGN/(2*LN*2*BR-5)
C
C       CHECK IF SIGN CHANGE (SIGCH) INCREASES, DROP THE GAIN FACTOR
C
          SIGCH=(SIGO-SIGN)/SIGO*100
          IF(SIGCH.GT.-10000) GO TO 108
C
          GAIN=GAIN*0.5
          PRINT*,'$$$$$$$$$$$$$$$$$$'
```

```
      PRINT*, 'CORRECTIONS ARE REDUCED, DUE TO NEG.VALUE OF SIGCH'
      PRINT*, '$$$$$$$$$$$$$$$$$$'
      DO 107 I=1,5
      U(I+1)=U(I+1)-DL(I)*GAIN
      U(I)=U(I)-DL(I)*GAIN
  107 CONTINUE
      ITR=ITR-1
C
      GO TO 101
C
  108 IF(ITR.GT.1) THEN
      WRITE(*,135)
      WRITE(*,113) W,EL,D,(U(I),I=1,5)
      WRITE(*,114) (ITR-1),(DL(I)*GAIN,I=1,5)
      WRITE(*,119) SIGN,SIGCH,GAIN
      WRITE(*,135)
      END IF
C
      GAIN=1.0
      IF(SIGCH.LT.CRITIC) ICONVG=ICONVG+1
      IF(ITR.EQ.1) WRITE(*,115) SIGN
C
C     CHECK FOR MAX. ALLOWABLE ITERATION
C
      IF(ITR.EQ.MAXITR.OR.ICONVG.EQ.1) GO TO 111
C
C     UNIFIED APPROACH
C
      DO 109 I=1,5
      EN(I,I)=EN(I,I)+WU(I)
  109 T(I)=T(I)-WU(I)*U(I+5)
C
C     SOLVE NORMAL EQUATIONS                                  BLOCK(10)
C
      CALL ZEROM1(DL,5)
      CALL SLVNRM(EN,T,DL)
C
```

```
      THT=U(5)+DL(5)
C
C     ADD CORRECTIONS TO THE APPROXIMATE VALUES
C
      DO 110 I=1,5
      U(I)=U(I)+DL(I)*GAIN
      U(I+5)=U(I+5)+DL(I)*GAIN
  110 CONTINUE
C
C
      ICONVG=0
      IF(ABS(DL(3))*GAIN.LT.SPECI.AND.ABS(DL(4))*GAIN.LT.SPECI) ICONVG=1
      GO TO 101
C
C     PRINT THE RESULT
C
  111 WRITE(*,136)
      WRITE(*,116) (U(I),I=6,10),(U(J),J=1,5),SIGN
      WRITE(*,135)
C
      RETURN
C
  113 FORMAT(/5X,'CROSS POINTING WITH',1X,'W=',F5.2,1X,'L=',F7.2,
     14X,'AND SPREAD FUNCTION OF WIDTH=',F5.2,//1X,
     2//5X,'PARAMETERS APPROXIMATION:',//9X,'H1',9X,'H2',10X,'X',8X,'Y',
     37X,'THETA',//4X,2F10.2,2F10.3,F10.5)
  114 FORMAT(/1X,'ITERATION',6X,'CORRECTIONS:',/4X,I3,5X,2F10.2,2F10.3
     1,F10.5)
  119 FORMAT(/1X,'VARIANCE',7X,'VAR.CH.',3X,'GAIN',/,2F10.3,F7.1)
  115 FORMAT(97X,F10.3)
  116 FORMAT(/6X,18HTOTAL CORRECTIONS:,/10X,2F10.2,2F10.3,F10.
     15,/6X,16HFINAL ESTIMATES:,/10X,2F10.2,2F10.3,F10.5,5X,F14.4)
  131 FORMAT(/25X,'RESIDUALS',/)
  117 FORMAT(12X,9F5.0)
  130 FORMAT(//15X,14HORIGINAL IMAGE,28X,14HMODELLED IMAGE)
  121 FORMAT(9I4,6X,9I4)
  135 FORMAT(/1X,'----------------------------------------')
```

```
 136  FORMAT(/1X,'$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$')
      END
C ---------------------------------------------------------------------
      SUBROUTINE AREA(P,D,DD,H)
      REAL DD,DH,F,H
      SPREAD(X)=EXP(-X*X/(D*D*0.5))/(1.2533141*D)
C
      DD=P/D+0.5
      IF(DD.LT.0.0) GO TO 200
      IF(DD.GT.1.0) GO TO 250
C
      DH=0.0
      IF(ABS(P).LT.H) GO TO 150
      A=H
      F=0.0
 90   F=F+H
      DH=DH+2*SPREAD(ABS(P)-F)
      IF(ABS(P)-F.GT.H) GO TO 90
      GO TO 100
 150  A=ABS(P)
 100  DD=0.5*A*(DH+0.7978845/D+SPREAD(P))
      IF(P.LT.0.0) DD=0.5-DD
      IF(P.GE.0.0) DD=0.5+DD
      GO TO 70
 200  DD=0.0
      GO TO 70
 250  DD=1.0
 70   CONTINUE
      RETURN
      END
C
      SUBROUTINE SLVNRM(EN,T,DL)
      PARAMETER(NR=5)
      DIMENSION EN(NR,NR),T(NR),DL(NR),ENI(NR,NR)
C
      DO 6,I=1,NR
      DO 6,J=1,NR
```

```
6     ENI(I,J)=0.0
      ENI(1,1)=1.0/EN(1,1)                                    BLOCK(11)
      DO 40,N=2,NR
     K=N-1
     EK=EN(N,N)
     DO 10,I=1,K
     DO 10,J=1,K
 10   EK=EK-EN(N,I)*ENI(I,J)*EN(J,N)
      ENI(N,N)=1.0/EK
      DO 30,I=1,K
      DO 20,J=1,K
 20   ENI(I,N)=ENI(I,N)-ENI(I,J)*EN(J,N)/EK
 30   ENI(N,I)=ENI(I,N)
      DO 40,I=1,K
      DO 40,J=1,K
 40   ENI(I,J)=ENI(I,J)+ENI(I,N)*ENI(N,J)*EK
C
      DO 50,I=1,NR
      DO 50,J=1,NR
 50   DL(I)=DL(I)+ENI(I,J)*T(J)
      RETURN
      END
C
      SUBROUTINE EXCHG(M,XL,XR,NR)
      DIMENSION XL(NR),XR(NR)
      X1=XL(M)
      X2=XR(M)
      XL(M)=X2
      XR(M)=X1
      RETURN
      END
C
      SUBROUTINE FIT(X,Y,N,D)
     PARAMETER(NR=150)
     REAL EN(2,2),F(2),D(2),AX(NR,2),AXT(2,NR),X(NR),Y(NR),DT
     CALL ZEROM1(F,2)
     CALL ZEROM2(EN,2,2)
```

```
      DO 10 I=1,N
      AX(I,1)=X(I)
      AX(I,2)=1
      AXT(1,I)=X(I)
10    AXT(2,I)=1
      DO 30 II=1,2
      DO 30 JJ=1,2
      DO 30 L=1,N
30    EN(II,JJ)=EN(II,JJ)+AXT(II,L)*AX(L,JJ)
      DO 40 I=1,2
      DO 40 J=1,N
40    F(I)=F(I)+AXT(I,J)*Y(J)
C
      DT=EN(1,1)*EN(2,2)-EN(1,2)*EN(2,1)
      IF(DT.EQ.0) DT=0.01
      D(1)=EN(2,2)*F(1)-EN(1,2)*F(2)
      D(2)=-EN(2,1)*F(1)+EN(1,1)*F(2)
      D(1)=D(1)/DT
      D(2)=D(2)/DT
      RETURN
      END
C
      SUBROUTINE ZEROM1(A,N)
      DIMENSION A(N)
      DO 100 I=1,N
100   A(I)=0.0
      RETURN
      END
C
      SUBROUTINE ZEROM2(A,N,M)
      DIMENSION A(N,M)
      DO 200 I=1,N
      DO 200 J=1,M
200   A(I,J)=0.0
      RETURN
      END
C
```

```
      SUBROUTINE MXMN(AM,IH2,IH1,NR,MMR)
      PARAMETER(JR=500,JJR=500)
      INTEGER*2 AM(JR,JJR)
      IH2=1
      IH1=1000
      DO 25 I=1,NR
      DO 25 J=1,MMR
        IF(IH2.LT.AM(I,J)) IH2=AM(I,J)
   25   IF(IH1.GT.AM(I,J)) IH1=AM(I,J)
        RETURN
        END
C
        SUBROUTINE CONT(AM,IHH,ARET,AREB,TIOD,BIOD,NR,MMR)
        PARAMETER(JR=500,JJR=500)
        INTEGER*2 AM(JR,JJR)
        ARET=0
        AREB=0
        TIOD=0
        BIOD=0
        DO 35 I=1,NR
        DO 35 J=1,MMR
        IF(AM(I,J).GT.IHH) THEN
        TIOD=TIOD+AM(I,J)
        ARET=ARET+1
        ELSE
        BIOD=BIOD+AM(I,J)
        AREB=AREB+1
        END IF
   35   CONTINUE
        RETURN
        END
```

A2 Program COPLAN

This program was developed to perform relative orientation *using the* coplanarity condition equations.

A2.1 Definition of Some of the Variables

DY, DZ — Initial values for the differences between the Y and Z coordinates of the left and right projection centres respectively.

DDY DDZ — Calculated improvement values to DY and DZ respectively.

IT — Number of iterations

F — Calibrated focal length

XL,YL,ZL — The coordinate values of the left projection centre.

BS — The base value

A2.2 Definition of Some of the Arrays

A — Matrix of the measured coordinates of image points reduced to the principal point coordinate system.

DD — Matrix which contains the initial and improved values of phi2, omega2 and kappa2.

C — The working matrix which contributes to the formation of the coefficient matrix P.

P — The coefficient matrix.

D — The normal matrix.

AM — The matrix of the calculated model coordinates.

A2.3 Explanation of Program COPLAN

Block 1 : Arrays are declared; total numbers of image points, check points, focal length, base value, and the coordinates of the left projection centre are input to the program.

Block 2 : Initialization of variables.

Block 3 : Formation of the coefficient and normal matrices.

Block 4 : Solution of the normal equations.

Block 5 : Formation of the rotation matrix and transformation of the image coordinates of the right hand photo.

Block 6 : Checking the specified condition to stop the iteration.

Block 7 : Formation of model coordinates and output of the results.

A2.4 Listing of Program COPLAN

```
C     * ************************************************************ *
C     * The program performs relative orientation using the          *
C     * coplanarity condition. It stores the following elements:     *
C     * BY , BZ , PHI , OMEGA , KAPPA , ROTATION MATRIX , BASE ,     *
C     * and model coordinates.                                       *
C     ************************************************************ *
C
      PROGRAM COPLAN
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (NR=600)
      DIMENSION A(NR,9),C(4,3),CC(6,3),D(5,6),P(6),DD(3),AM(NR,4)
      CHARACTER*32 NAME
      CHARACTER*1 E
C
      PRINT*,'PLEASE ENTER THE NO. OF IMAGE POINTS (ONE PHOTO)'
      READ*,N
      PRINT*,'PLEASE ENTER THE NO. OF CHECK POINTS'
      READ*,NNN
      PRINT*,'PLEASE ENTER CALIBRATED FOCAL LENGTH'
      READ*,F
774   PRINT*,'PLEASE ENTER IMAGE POINTS FILE NAME'
      READ(*,100) NAME
      OPEN(1,FILE=NAME,STATUS='OLD',ERR=114)
      DO 11,I=1,N+NNN
      READ(1,200) (A(I,J),J=1,5)
11    A(I,6)=F                                                  BLOCK (1)
      GO TO 116
114   PRINT*,'IMAGE FILE CAN NOT BE FOUND'
      GO TO 774
116   PRINT*,'PLEASE ENTER AN OPTIONAL VALUE FOR THE BASE'
      READ*,BS
      PRINT*,'PLEASE ENTER XL,YL,ZL '
      READ*,XL,YL,ZL
      DO 999 I=1,N+NNN
      DO 999 J=4,6
999   A(I,J+3)=A(I,J)
C ------------------------------------------------------------------
```

```
      PIX=31.25
      DY=0.0
      DZ=0.0
      DDY=0.0
      DDZ=0.0
      IT=0.0
      DO 54 I=1,3
 54   DD(I)=0.0
C ------------------------------------------------------------
C       Form coefficient matrix and normal equations.
C
 111  IT=IT+1
      DO 55 I=1,5
      DO 55 J=1,6
55    D(I,J)=0.0
      DO 101 I=1,N
      C(1,1)=DY*F-DZ*A(I,3)
      C(1,2)=BS*F-DZ*A(I,2)
      C(1,3)=BS*A(I,3)-DY*A(I,2)
      C(2,1)=A(I,6)*C(1,1)
      C(2,2)=-A(I,4)*C(1,3)
      C(2,3)=A(I,6)*C(1,2)
      C(3,1)=A(I,5)*C(1,3)
      C(3,2)=-A(I,5)*C(1,1)
      C(3,3)=-A(I,4)*C(1,2)
      BB=BS*(A(I,3)*A(I,6)-F*A(I,5))
C
      P(1)=-A(I,2)*A(I,6)+F*A(I,4)
      P(2)=A(I,2)*A(I,5)-A(I,3)*A(I,4)
      P(3)=C(2,1)+C(2,2)
      P(4)=C(2,3)+C(3,1)
      P(5)=C(3,2)+C(3,3)
      P(6)=-BB
C
      DO 22 L=1,5
      DO 22 K=1,6
 22    D(L,K)=D(L,K)+P(L)*P(K)
```

BLOCK (2)

BLOCK(3)

```
  101  CONTINUE
C ------------------------------------------------------------------
C     Solve normal equations
C
C     Forward solution
C
      DO 15 I=1,5
      SQR=SQRT(D(I,I))
      DO 25 J=1,6
   25 D(I,J)=D(I,J)/SQR
      IF(I-5) 45,65,65
   45 IP=I+1
      DO 15, L=IP,5
      DO 15, J=L,6
   15 D(L,J)=D(L,J)-D(I,L)*D(I,J)
C
C     Backward solution
C                                                        BLOCK(4)
   65 D(5,6)=D(5,6)/D(5,5)
      DO 75 I=1,4
      NM=5-I
      NMP=NM+1
      DO 85 J=NMP,5
   85 D(NM,6)=D(NM,6)-D(J,6)*D(NM,J)
   75 D(NM,6)=D(NM,6)/D(NM,NM)
C ------------------------------------------------------------------
C     Form rotation matrix
C
      DO 33 I=1,3
      DD(I)=DD(I)+D(I+2,6)
      CC(1,I)=SIN(DD(I))
   33 CC(2,I)=COS(DD(I))
      CC(3,1)=CC(2,1)*CC(2,3)
      CC(3,2)=-CC(2,1)*CC(1,3)
      CC(3,3)=CC(1,1)
      CC(6,1)=-CC(1,1)*CC(2,3)
      CC(6,2)=CC(1,1)*CC(1,3)
```

```
      CC(6,3)=CC(1,2)*CC(1,1)
      CC(4,1)=CC(2,2)*CC(1,3)+CC(6,3)*CC(2,3)
      CC(4,2)=CC(2,2)*CC(2,3)-CC(6,3)*CC(1,3)
      CC(4,3)=-CC(1,2)*CC(2,1)
      CC(5,1)=CC(1,2)*CC(1,3)+CC(6,1)*CC(2,2)
      CC(5,2)=CC(1,2)*CC(2,3)+CC(6,2)*CC(2,2)
      CC(5,3)=CC(2,2)*CC(2,1)
C
      DYY=D(1,6)-DY
      DZZ=D(2,6)-DZ                                          BLOCK(5)
      DY=D(1,6)
      DZ=D(2,6)
      DO 27 I=1,N+NNN
      DO 27 J=4,6
 27   A(I,J)=0.0
C
C     Transform the image points of the second photograph
C
      DO 24 I=1,N+NNN
      DO 24 J=4,6
      DO 24 K=1,3
 24   A(I,J)=A(I,J)+CC(J-1,K)*A(I,K+6)
C ------------------------------------------------------------
C     Check the condition to terminate the iteration
C
      R=.000001                                              BLOCK(6)
      IF(DYY.LT.R.AND.DZZ.LT.R.AND.D(3,6).LT.R.
     *AND.D(4,6).LT.R.AND.D(5,6).LT.R) GO TO 555
      GO TO 111
C ------------------------------------------------------------
C     Form the model coordinates
C
 555  YR=YL+DY
      ZR=ZL+DZ
      DO 123 I=1,N+NNN
      AL=A(I,6)*BS-A(I,4)*DZ                                 BLOCK(7)
      DEN=A(I,2)*A(I,6)-A(I,4)*F
```

```
      ALAM=AL/DEN
      AM(I,3)=ZL+ALAM*F
      AM(I,2)=YL+A(I,3)*(AM(I,3)-ZL)/F
      YRR=YR+A(I,5)*(AM(I,3)-ZR)/A(I,6)
      AM(I,1)=XL+A(I,2)*ALAM
  123 AM(I,4)=YRR-AM(I,2)
C
      DO 124 I=2,4
      DO 124 J=1,3
  124 C(I,J)=CC(I+1,J)
      C(1,1)=(XL+BS)*PIX
      C(1,2)=YR*PIX
      C(1,3)=ZR*PIX
C
      PRINT*
      WRITE(*,1001)
      PRINT*
      DO 1002 I=3,5
 1002 WRITE(*,102) (CC(I,J),J=1,3)
      PRINT*
      PRINT*,'  ---------- IT         =',IT,' --------------------'
      PRINT*
      WRITE(*,300) DY,DZ,(DD(I),I=1,3)
      PRINT*,' '
      PRINT*,'  ------------------------------------------------------'
      PRINT*,' '
      WRITE(*,400)
      PRINT*,' '
      DO 2 I=1,N+NNN
    2 WRITE(*,500) A(I,1),(AM(I,J),J=1,4)
      PRINT*
      PRINT*,'DO YOU WANT TO RECORD THE ROTATION MATRIX ?'
      READ(*,800) E
      IF(E.EQ.'Y') THEN
      PRINT*,'ENTER THE NAME OF THE OUTPUT FILE'
      READ(*,100) NAME
      OPEN(2,FILE=NAME,STATUS='NEW')
```

```
      DO 127 I=1,4
127   WRITE(2,102) (C(I,J),J=1,3)
      END IF
      PRINT*,'DO YOU WANT TO RECORD THE MODEL POINTS ?'
      READ(*,800) E
      IF(E.EQ.'Y') THEN
      PRINT*,'ENTER THE NAME OF THE OUTPUT FILE'
      READ(*,100) NAME
      OPEN(8,FILE=NAME,STATUS='NEW')
         DO 951 I=3,5
951      WRITE(8,102) (CC(I,J),J=1,3)
         WRITE(8,300) DY,DZ,(DD(I),I=1,3)
      DO 128 I=1,N+NNN
128   WRITE(8,500) A(I,1),(AM(I,J),J=1,4)
      END IF
C
100   FORMAT(32A)
800   FORMAT(A)
2500  FORMAT(3F15.4)
102   FORMAT(3F15.5)
200   FORMAT(F6.0,4F15.4)
300   FORMAT('          BY          BZ          PHI        OMEGA        KAPPA',
     1//2X,5F10.5)
400   FORMAT(/,'       MODEL COORDINATES: (mm)',
     1//2X,'PT.NO.        X                Y                Z          PY')
500   FORMAT(2X,F4.0,4F15.5)
1001  FORMAT(/,'    FINAL ROTATION MATRIX')
      STOP
      END
```

A3 Program DIGREC

This program was developed to perform digital rectification using the indirect transformation and resampling on a pixel by pixel basis.

A3.1 Definition of Some of the Variables

MC	Maximum column number of the input image
MR	Maximum row number of the input image
ABSX, ABSY	*The coordinates of the origin of the input image patch with respect to the origin of the complete image (in terms of pixels)*
FL	The calibrated focal length
BS	The base value
DY	The difference between the Y coordinates of the left and right projection centres.

A3.2 Definition of Some of the Arrays

IM	Matrix of the original image
IM2	Matrix of the rectified image
LIM	Working matrix which contributes to the calculation of the bicubic convolution
B	Matrix of the coefficients which transform the image points from the principal point coordinate system into the microdensitometer coordinate system.
BB	Matrix of the coefficients which transform the image points from the microdensitometer coordinate system into the principal point coordinate system.
C	Matrix containing the coordinates of the camera projection centre and the rotation elements.
CT	Transposed rotation matrix.
ID	Matrix containing the image coordinates of the corners of the input image.
XY1, XY2 IXX, IYY	x and y image coordinates of the output pixel transformed to the plane of the input image.
CB1 CB2 CD1 CD2 CE1 CE2 CF1 CF2	These arrays contribute in the calculation of the coefficients of the bicubic convolution.

A3.3 Explanation of Program DIGREC

Block 1 : Arrays are declared; variables are initialized; the input image, the orientation parameters, the coefficients of the inner orientation and the coordinates of the corners of the input image are read into the program.

Block 2 : To determine the coordinates of the corners of the output image, the coordinates of the corners of the input image are first transformed into the principal point coordinate system and then into the plane of the output matrix.

Block 3 : The pixels of the output matrix which are located on a single row and occupy the area between the corner points of the output matrix are transformed backward into the plane of the input image using the following transformations:

1- transformation into the epipolar plane;

2- transformation into the principal point coordinate system;

3- transformation into the microdensitometer coordinate system.

Block 4 : Density values are calculated and assigned to the output pixels; the rectified image row is recorded; the procedures of the block 3 are repeated until the entire output matrix is recorded.

A3.4 Listing of Program DIGREC

```
@PROCESS DIR('IBM')
      PROGRAM DIGREC
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER(M=2000,N=2000)
      INTEGER   LIM(4)
      INTEGER*2 IM(M,N),IM2(M)
      DIMENSION IXX(128),IYY(128),
     1 CB1(128),CD1(128),CE1(128),CF1(128),
     2 CB2(128),CD2(128),CE2(128),CF2(128)
      DIMENSION B(6),C(4,3),CT(3,3),ID(8),
     1D(2,6),XY(10),CC(6),BB(6)
C.....
C.....SPECIFY THE DIMENSION AND ORIGIN OF THE INPUT IMAGE
C.....
      MC=512
      MR=512
C     ABSX=896.0
C     ABSY=2624.0
      ABSX=1408
      ABSY=3136
C.....
      MARGN=50
      FL=4773.44
      BS=2312.5
      DY=88.34
C.....                                                    BLOCK(1)
      DO 10 I=1,MR
 10   READ(2) (IM(J,I),J=1,MC)
      DO 20 I=1,4
 20   READ(3,200) (C(I,J),J=1,3)
      READ(4,300) (B(J),J=1,6)
      READ(7,300) (BB(J),J=1,6)
      READ(8,400) (ID(J),J=1,8)
C ---------------------------------------------------------------
C.....      TRANSFORM FOUR IMAGE CORNERS INTO THE OUTPUT PLANE
C.....
      CC(3)=-FL
```

```
      DO 25 I=1,8,2
      D(1,1)=1
      D(1,2)=ID(I)
      D(1,3)=ID(I+1)
      DO 40 K=1,3
      D(1,K+3)=0.0
      D(2,K)=0.0
40    D(2,K+3)=D(1,K)
      DO 55 L=1,2
      XY(L)=0.0
      DO 55 K=1,6
55    XY(L)=XY(L)+D(L,K)*BB(K)
      CC(1)=XY(1)
      CC(2)=XY(2)
      DO 63 K=1,3
      CC(K+3)=0.0
      DO 63 L=1,3
63    CC(K+3)=CC(K+3)+C(K+1,L)*CC(L)
      XY(I+2)=-C(1,3)*CC(4)/CC(6)
      XY(I+3)=-C(1,3)*CC(5)/CC(6)
25    CONTINUE
      DO 39 I=1,3
      DO 39 J=1,3                                          BLOCK(2)
39    CT(I,J)=C(J+1,I)
C.....................................................................
      IIR=XY(4)+C(1,2)+MARGN
      IRR=XY(6)+C(1,2)-MARGN
      IIC=XY(3)+C(1,1)+MARGN
      ICC=XY(9)+C(1,1)-MARGN
      CC(3)=-C(1,3)
      C11=C(1,1)
      C12=C(1,2)
      CC3=CC(3)
C ---------------------------------------------------------------------
      DO 31 J=IIR,IRR
      DO 30 II=IIC,ICC,128
      IIM=II-1
```

```
C       ITI=MIN(II+127,ICC)
        ITI=II+127
        IF(ITI.GT.ICC) ITI=ICC
        DO 28 I1=1,ITI-IIM
        I=I1+IIM
C.....
C.....APPLY 3-D CONFORMAL TRANSFORMATION TO THE OUTPUT PIXELS
C.....
        CC1=I-C11
C       IF(CAT.LT.0) CAT=0
        CCC1=I-3593.75
        CC2=J+(CCC1*DY/BS)-C12
        CC4=CT(1,1)*CC1+CT(1,2)*CC2+CT(1,3)*CC3
        CC5=CT(2,1)*CC1+CT(2,2)*CC2+CT(2,3)*CC3
        CC6=-FL/(CT(3,1)*CC1+CT(3,2)*CC2+CT(3,3)*CC3)
        X=CC4*CC6
        Y=CC5*CC6                                          BLOCK(3)
C ...
C.....FORM COEFFICIENT MATRIX FOR AFFINE TRANSFORMATION
C.....
        XY1=B(1)+B(2)*X+B(3)*Y-ABSX
        XY2=B(4)+B(5)*X+B(6)*Y-ABSY
        IXX(I1)=XY1
        IYY(I1)=XY2
        C1=XY1-IXX(I1)
        C2=XY2-IYY(I1)
        C1S=C1**2
        C2S=C2**2
        CB1(I1)=-C1*(1.-C1)**2
        CD1(I1)=1.+C1S*(C1-2.)
        CE1(I1)=C1*(1.+C1-C1S)
        CF1(I1)=C1S*(C1-1.)
        CB2(I1)=-C2*(1.-C2)**2
        CD2(I1)=1.+C2S*(C2-2.)
        CE2(I1)=C2*(1.+C2-C2S)
  28    CF2(I1)=C2S*(C2-1.)
C ---------------------------------------------------------------
C.....CALCULATE DENSITY VALUES FOR OUTPUT PIXELS
C.....
```

```
      DO 29 I2=1,ITI-IIM
      IX=IXX(I2)
      IY=IYY(I2)
      IXPP=IX+2
      IF(IX.GT.1.AND.IY.GT.1.AND.IXPP.LE.MC.AND.IY+2.LE.MR) THEN
         IXM=IX-1
         IXP=IX+1
         IYMM=IY-2
         DO 60 LL=1,4                                          BLOCK(4)
         L=LL+IYMM
C        LIM(LL)=-C1*(1-C1)**2*IM(IX-1,L)
C     1  +(1-2*C1**2+C1**3)*IM(IX,L)
C     2  +C1*(1+C1-C1**2)*IM(IX+1,L)
C     3  -C1**2*(1-C1)*IM(IX+2,L)
   60    LIM(LL)=CB1(I2)*IM(IXM,L) + CD1(I2)*IM(IX,L)
     1   +CE1(I2)*IM(IXP,L) + CF1(I2)*IM(IXPP,L)
         III=I2+IIM-IIC+1
C        IM2(III)=-C2*(1-C2)**2*LIM(1)
C     1  +(1-2*C2**2+C2**3)*LIM(2)
C     2  +C2*(1+C2-C2**2)*LIM(3)
C     3  +C2**2*(C2-1)*LIM(4)
         IM2(III)=CB2(I2)*LIM(1) + CD2(I2)*LIM(2)
     1   +CE2(I2)*LIM(3) +CF2(I2)*LIM(4)
      END IF
   29 CONTINUE
   30 CONTINUE
      WRITE(10) (IM2(IQJ),IQJ=1,ICC-IIC+1)
   31 CONTINUE
      PRINT*,'THE NO. OF ROWS & COLUMNS FOR THE ORIGINAL IMAGE ARE:'
      PRINT*,'******',ID(2),ID(4),ID(1),ID(7),'********'
      PRINT*,'THE NO. OF ROWS & COLUMNS FOR THE RECTIFIED IMAGE ARE:'
      PRINT*,'******',IIR,IRR,IIC,ICC,'********'

  200 FORMAT(3F15.5)
  300 FORMAT(F15.5)
  400 FORMAT(I6)
      STOP
      END
```

A4 Program FIDPT

This program was developed to be used as the first stage of the inner orientation operation and involves measurement of the position of the fiducial marks. The fiducial marks are measured in both of the normal size and enlarged size.

A4.1 Definition of Some of the Variables

DIM2	Dimension of the constructed window around the cross centre
DIM	Dimension of the window which is used to display the enlarged cross.
XX, YY	The column and row numbers of the input image
SMXN, SMYN SMXL, SMYL	Sum of the x and y measurements of the centre of the cross for the normal size and enlarged crosses respectively.
MXN, MYN MX, MY	Mean values for the x and y measurements on the normal size and enlarged crosses respectively.
SIGX, SIGY	Square sum of the discrepancies between the manually and automatically measured crosses.
RESX, RESY	RMSE of the discrepancies between the manually and automatically measured crosses.

A4.2 Definition of Some of the Arrays

CRS	The matrix containing the cross target.
AA	The matrix containing the measured coordinates of the normal-size cross.
DD	The matrix containing the measured coordinates of the enlarged cross.
MM	The matrix containing the mean value of the measured coordinates of the enlarged cross.

A4.3 Listing and explanation of Program FIDPT

```
/* PROGRAM MEASURES THE COORDINATES OF FIDUCIAL MARKS */
SAY'**************************************************************'
SAY'**    <<<<<<<<<<< INNER ORIENTATION PROGRAM >>>>>>>>>>>>>       **'
SAY'**                  FIDUCIAL MARK MEASUREMENT                   **'
SAY'**************************************************************'

                                        /* input the required information */

SAY'ENTER THE NO. OF CROSSES AND THE NO. OF OBSERVATIONS TO EACH CROSS'
SAY'>'
```

```
 PULL NUM ITR
 'R='NUM'*'ITR''
 'RR='R'+1'
 'NM='NUM'+1'
 'AA=O(0:1,1:'RR')'                          /* initialize variables */
 'DD=C3(O(0:1,1:'RR'))'
 'MM=C3(O(0:1,1:'NM'))'
 K=0
 L=0
 M=0
 'DIM=C3(400)'
 'DIM2=C3(40)'
 'MAG=C3('DIM'/'DIM2')'
 'DIM1='DIM'/2'
 'CURSOR(SMALL)'
/*$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$*/
DO I=1 TO NUM BY 1
 'SMXN=0'
 'SMYN=0'
 'SMXL=0'
 'SMYL=0'
 'SIGX=0'
 'SIGY=0'
SAY'ENTER THE NAME OF THE FILE CONTAINING THE CROSS'
SAY'>'
PULL CRS
SAY'TYPE THE COORDS. OF THE ORIGIN OF THE IMAGE'
SAY'>'
PULL XORG YORG
SAY'ENTER THE COORDS. OF THE CROSS CALCULATED BY AUTOMATIC METHOD'
SAY'>'
PULL TRUX TRUY
'GET('CRS',IAX2,A)'            /* input the image containing the cross */
'XX=XS('CRS')';'YY=YS('CRS')'    /* extract the number of the columns */
                                 /* and rows of the image */
'SCREEN DEF 1 0 'XX' 0 'YY''     /* define the logical screen */
SAY' '
```

```
SAY' '
SAY'THE IMAGE WILL BE DISPLAYED IN NORMAL SIZE'
SAY' '
SAY' '
'DISP 'CRS' IN 1'
/*-------------------------------------------------------------*/
DO J=1 TO ITR BY 1
K=K+1
'TRANSFER CURSOR'
PULL XN YN BOT                                  /* start the measurement */
SAY'-------------------------------------------------------------'
SAY' '
SAY' '
SAY'X & Y WITH RESPECT TO THE SCREEN COORDS. SYSTEM:'     XN  YN
                                  /* calculate the coordinates of the */
                                  /* cross centre with respect to the */
                                  /* microdensitometer coordinate system */
'AA(0,'K')='XN'+'XORG''
'AA(1,'K')='YN'+'YORG''
'D1=AA(0,'K')'
'D2=AA(1,'K')'
'TRANSFER D1'
'TRANSFER D2'
PULL D1
PULL D2
SAY'-------------------------------------------------------------'
SAY' '
SAY'X & Y WITH RESPECT TO MICRODENS. COORDS. SYSTEM:'    D2  D1
SAY' '
SAY' '
END J
/*-------------------------------------------------------------*/
'XF='XN'-'DIM2'/2';'YF='YN'-'DIM2'/2'
'XL='XF'+'DIM2'';'YL='YF'+'DIM2''
'CC='CRS'('XF':'XL','YF':'YL')'     /* construct a window around the */
                                    /* centre of the cross */

'EE=RRANGE 'CC''
```

```
'SCREEN DEF 1 0 'DIM' 0 'DIM''          /* define the logical screen */
SAY' '
SAY' '
SAY'THE IMAGE IS BEING ENLARGED'
SAY' '
SAY' '
'EE=CHOP('EE',TO,'DIM','DIM')'                    /* enlarge the image */
'CURSOR(SMALL)'
'DISP 'EE' IN 1'                                  /* display the enlarged image */
DO LL= 1 TO ITR BY 1
M=M+1
'TRANSFER CURSOR'
PULL XR YR BOT          /* start the measurement on the enlarged cross */
SAY' '
SAY'X & Y WITH RESPECT TO THE SCREEN COORDS. SYSTEM          XR  YR
SAY' '
SAY'----------------' XR       YR '--------------------'
SAY' '
'XRR='XR'/'MAG'+'XF''; 'YRR='YR'/'MAG'+'YF''
'DD(0,'M')='XRR'+'XORG''
'DD(1,'M')='YRR'+'YORG''
'E1=DD(0,'M')'
'E2=DD(1,'M')'
'TRANSFER E1'
'TRANSFER E2'
PULL E1
PULL E2
SAY'X & Y WITH RESPECT TO MICRODENSITOMETER COORDS. SYSTEM'
SAY' '
SAY'--------------------'  E2         E1 '-----------------'
SAY' '
END LL
'FREE 'EE''
'FREE 'CRS''
/*-------------------------------------------------------------------*/
DO J=1 TO ITR
L=L+1
```

```
 SMXN= SMXN +AA(0,'L')                    /* calculate mean value for the */
                                          /* coordinates of the cross */
'SMYN='SMYN +AA(1,'L')'
'SMXL='SMXL +DD(0,'L')'
'SMYL= SMYL'+DD(1,'L')'
'SIGX='SIGX'+('TRUX'-DD(0,'L'))**2'
'SIGY= SIGY'+('TRUY'-DD(1,'L'))**2'
 END J
 'MXN='SMXN'/'ITR''
 'MYN='SMYN'/'ITR''
 'MX='SMXL'/'ITR''
 'MY='SMYL'/'ITR''
 'RESX=('SIGX'/('ITR'))**0.5'
 'RESY=('SIGY'/('ITR'))**0.5'
 'MM(0,'I')='MX'*0.032'
 'MM(1,'I')='MY'*0.032'
 'TRANSFER MXN '
PULL MNX
 'TRANSFER MYN'
PULL MNY
 'TRANSFER MX'
PULL MLX
 'TRANSFER MY'
PULL MLY
 'TRANSFER RESX'
PULL RX
 'TRANSFER RESY'
PULL RY
SAY ' '
                                          /* display the result */
SAY'=============================================================='
SAY'                              CROSS NO.' I
SAY'          MEANX (NORMAL SIZE)       MEANY (NORMAL SIZE)'
SAY'        ' MNX'                     'MNY
SAY'          AUTOMATIC X                 AUTOMATIC Y'
SAY'          ' TRUX'                     'TRUY
SAY'          MEANX (ENLARGED)          MEANY (ENLARGED)'
```

```
SAY'        ' MLX'           'MLY
SAY'         RMSE X                   RMSE Y'
SAY'        'RX '           'RY
SAY'===========================================================' 
SAY'
END I
/*$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$*/
'PUT( AA'.CRSCRDS,DATA,A)'            /* record the measured points */
'PUT('DD'.CRSHECD,DATA,A)'
'PUT('MM'.MFDML,DATA,A)'
SAY'NORMAL TERMINATION'
EXIT
```

A5 Program IMPT

The program was developed to measure the coordinates of the image points for relative or absolute orientation. The program allows both stereoscopic and monocular measurement.

A5.1 Definition of Some of the Variables

N	Number of image points.
NNN	Number of observations to each image point.
XLL, YLL XRR, YRR	The x and y coordinates of the origins of the left and right image patches (in terms of pixels).
MM, NN	The number of columns and rows of the image patches.
E1, E2 E3, E4	Sum of the x and y values of the measured image points on the left and right images.
MEAN1, MEAN2 MEAN3, MEAN4	The calculated mean values of x and y coordinates of the left and right image points.

A5.2 Definition of Some of the Arrays

LEFT, RIGHT Matrices holding the left and right image patches.

A	Matrix containing the coordinates of image points measured stereoscopically.
RR	Matrix containing the coordinates of image points measured monocularly.
ADD	Matrix containing zero digits which is used to suppress the effect of the blue electron gun.

A5.3 Listing and Explanation of Program IMPT

```
/* PROGRAM MEASURES THE COORDINATES OF IMAGE POINTS */
/* BOTH STEREOSCOPICALLY AND MONOCULARLY */
SAY'**************************************************************'
SAY'*        <<<<<<<< RELATIVE ORIENTATION PROGRAM >>>>>>>>>          *'
SAY'*                        IMAGE POINT MEASUREMENT                 *'
SAY'**************************************************************'

SAY'--------------------------------------------------------------'
SAY'------------- FIRST STAGE: STEREOSCOPIC MEASUREMENT ------------'
SAY'--------------------------------------------------------------'
SAY'ENTER THE NO. OF IMAGE POINTS AND THE NO. OF OBSERVATIONS TO'
SAY'EACH IMAGE POINT'
```

```
SAY '? ?
PULL N NNN
SAY 'ENTER THE NAME OF THE LEFT AND RIGHT IMAGES FOR DISPLAYING'
SAY '? ?'
PULL LEFT RIGHT
SAY 'ENTER THE COORDS. OF THE ORIGINS OF LEFT AND RIGHT IMAGES'
SAY 'XL,YL,XR,YR'
SAY '? ? ? ?'
PULL XLL YLL XRR YRR
'GET('LEFT',IAX1,A)'          /* read the left and right image patches */
'GET('RIGHT',IAX1,A)'
'MM=XS('LEFT');'NN=YS('LEFT')'  /* extract the x and y sizes of the */
                                /* image patch */
'A=C3(0(0:3,0:'N'))'            /* initialize the matrices */
'F=0(0:3,0:'NNN')'
'ADD=0(1:'MM',1:'NN');'ADD'=C1('ADD')'
'CURSOR(SMALL)'
'STDCOL'                      /* set the display screen in colour mode */
'SCREEN DEF 2 700 800 700 800'        /* define the logical screens */
'SCREEN DEF 1 0 'MM' 0 'NN''
'DISP SRGB 'RIGHT' 'LEFT' 'ADD' IN 1'  /* display the right and left */
                                       /* images in red and green */
'GTEXT(+ 50 50 IN 2 )'                 /* generate the symbol "+" to */
                                       /* be used in the measurement stage */
'DIM=C3(120)';'DIM2=C3(40)'
'MAG=C3('DIM'/'DIM2')';'DIM1='DIM'/2'
'DIMM='DIM'*8'
SAY' '
SAY' '
/*<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<>>>>>>>>>>>>>>>>>>>>>>>>>>>>*/
SAY' ----------------------------------------------------------'
SAY 'PUT THE RED MOVEABLE CURSOR IN CONTACT WITH THE GREEN FIXED CURSOR'
SAY' ----------------------------------------------------------'
SAY' '
SAY' '
DO II=1 TO N BY 1                      /* start the measurement */
'E1=0';'E2=0'
```

```
'E3=0'; 'E4=0'
'VV1=0'; VV2=0
'VV3=0'; 'VV4=0'
SAY '************** IMAGE POINT NO.' II '*****************'
DO JJ=1 TO NNN BY 1
'TRANSFER CURSOR'
PULL XL YL BOT
'SX='XL'-68'
'SY='YL'-23'
'SCREEN MOVE 2 TO 'SX' 'SY''
'TRANSFER CURSOR'
PULL XR YR BOT
'F(0,'JJ')='XL'+'XLL''; 'FF1=F(0,'JJ')'
'F(1,'JJ')='YL'+'YLL''; 'FF2=F(1,'JJ')'
'F(2,'JJ')='XR'+'XRR''; 'FF3=F(2,'JJ')'
'F(3,'JJ')='YR'+'YRR''; 'FF4=F(3,'JJ')'
'E1='E1'+F(0,'JJ')'
'E2='E2'+F(1,'JJ')'
'E3='E3'+F(2,'JJ')'
'E4='E4'+F(3,'JJ')'
'TRANSFER FF1';PULL FF1
'TRANSFER FF2';PULL FF2
'TRANSFER FF3';PULL FF3
'TRANSFER FF4';PULL FF4
SAY FF1 FF2 FF3 FF4 '>>>>>>>>>>>>>> OBSERVATION NO.' JJ
'SCREEN MOVE 2 TO 800 800'
END JJ
'MEAN1='E1'/'NNN''; 'MEAN2='E2'/'NNN''   /* calculate mean and std for */
'MEAN3='E3'/'NNN''; 'MEAN4='E4'/'NNN''   /* each set of measurements */
'ME1='MEAN1''; 'ME2='MEAN2''
'ME3='MEAN3''; 'ME4='MEAN4''
DO LL=1 TO NNN BY 1
'SB1='MEAN1'-F(0,'LL')'; 'SB2='MEAN2'-F(1,'LL')'
'SB3='MEAN3'-F(2,'LL')'; 'SB4='MEAN4'-F(3,'LL')'
'VV1='VV1'+'SB1'**2'
'VV2='VV2'+'SB2'**2'
'VV3='VV3'+'SB3'**2'
```

```
'VV4='VV4'+ SB4'**2'
END LL
'RMX1=SQRT('VV1'/'NNN')';'RMY1=SQRT('VV2'/'NNN')'
'RMX2=SQRT('VV3'/'NNN')';'RMY2=SQRT('VV4'/'NNN')'
'RX1='RMX1'';'RY1='RMY1''
'RX2='RMX2'';'RY2='RMY2''
'TRANSFER ME1';PULL ME1
'TRANSFER ME2';PULL ME2
'TRANSFER ME3';PULL ME3
'TRANSFER ME4';PULL ME4
'TRANSFER RX1';PULL RX1
'TRANSFER RY1';PULL RY1
'TRANSFER RX2';PULL RX2
'TRANSFER RY2';PULL RY2
SAY' '                                    /* display the mean and std */
SAY'=============================================================='
SAY'   MEANXL  MEANYL  MEANXR  MEANYR'
SAY        ME1            ME2            ME3            ME4
SAY'       STDXL          STDYL          STDXR          STDYR'
SAY      RX1            RY1            RX2          RY2
SAY'=============================================================='
SAY' '
'A(0,'II')='MEAN1'';'A(1,'II')='MEAN2''
'A(2,'II')='MEAN3'';'A(3,'II')='MEAN4''
END II
'PUT('A',STCONDL6,DATA,A)'                /* record the measured points */
'KEEP 'RIGHT' 'LEFT' 'DIM' 'DIMM' 'DIM2' 'MAG' 'MM' 'NN''
SAY'**************************************************************'
STDMONO
SAY'--------------------------------------------------------------'
SAY'--------------- SECOND STAGE: MONOCULAR MEASUREMENT ----------'
SAY'--------------------------------------------------------------'
SAY'ENTER THE NO. OF OBSERVATIONS TO EACH IMAGE POINT'
SAY' '
SAY'YOU CAN POINT TO 4 IMAGE POINTS'
SAY'-------------------------------'
SAY'?'
```

```
SAY' '
PULL MMM
M=4
'B=0(0:3,0:'MMM')'                          /* initialize the matrices */
'BB=C3(0(0:3,0:'MMM'))'
'R=C3(0(0:3,0:'M'))'
'RR=C3(0(0:3,0:'M'))'
'F11=(0 0 0 0 0)';'F22=(0 0 0 0 0)'
'F11=C3('F11')';'F22=C3('F22')'
'F33=(0 0 0 0 0)';'F44=(0 0 0 0 0)'
'F33=C3('F33')';'F44=C3('F44')'
'XX=2*'MM''
'SCREEN DEF 1 0 'XX' 0 'NN''
'DISP 'LEFT'||'RIGHT' IN 1'                 /* display the left and right */
                                            /* images side-by-side */
DO JJ=1 TO M BY 1                           /* start the measurement */
'H1=0';'H2=0'
'H3=0';'H4=0'
'V1=0';'V2=0'
'V3=0';'V4=0'
SAY'*************** IMAGE POINT NO.' JJ '*****************'
DO II=1 TO MMM BY 1
'TRANSFER CURSOR'
PULL XL YL BOT
'TRANSFER CURSOR'
PULL XR YR BOT
'B(0,'II')='XL'+'XLL'';'X1=B(0,'II')'
'B(1,'II')='YL'+'YLL'';'X2=B(1,'II')'
'B(2,'II')='XR'-'MM'+'XRR'';'X3=B(2,'II')'
'B(3,'II')='YR'+'YRR'';'X4=B(3,'II')'
'H1='H1'+B(0,'II')';'H2='H2'+B(1,'II')'
'H3='H3'+B(2,'II')';'H4='H4'+B(3,'II')'
'TRANSFER X1';PULL X1
'TRANSFER X2';PULL X2
'TRANSFER X3';PULL X3
'TRANSFER X4';PULL X4
SAY X1 X2 X3 X4 '--------------- OBSERVATION NO.' II
```

```
'XF1='XL'-'DIM2'/2';'YF1='YL'-'DIM2'/2'
'XL1='XF1'+'DIM2'';'YL1='YF1'+'DIM2''
'XF2='XR'-'MM'-'DIM2'/2';'YF2='YR'-'DIM2'/2'
'XL2='XF2'+'DIM2'';'YL2='YF2'+'DIM2''
IF JJ=1 THEN;'CC='LEFT'('XF1':'XL1','YF1':'YL1')'     /* construct a */
                                                      /* window around each image point */
IF JJ=1 THEN;'F11(1)='XF1''
IF JJ=1 THEN;'F22(1)='XF2''
IF JJ=1 THEN;'F33(1)='YF1''
IF JJ=1 THEN;'F44(1)='YF2''
IF JJ=1 THEN;'DD='RIGHT'('XF2':'XL2','YF2':'YL2')'
IF JJ=2 THEN;'WW='LEFT'('XF1':'XL1','YF1':'YL1')'
IF JJ=2 THEN;'F11(2)='XF1''
IF JJ=2 THEN;'F22(2)='XF2''
IF JJ=2 THEN;'F33(2)='YF1''
IF JJ=2 THEN;'F44(2)='YF2''
IF JJ=2 THEN;'ZZ='RIGHT'('XF2':'XL2','YF2':'YL2')'
IF JJ=3 THEN;'QQ='LEFT'('XF1':'XL1','YF1':'YL1')'
IF JJ=3 THEN;'F11(3)='XF1''
IF JJ=3 THEN;'F22(3)='XF2''
IF JJ=3 THEN;'F33(3)='YF1''
IF JJ=3 THEN;'F44(3)='YF2''
IF JJ=3 THEN;'UU='RIGHT'('XF2':'XL2','YF2':'YL2')'
IF JJ=4 THEN;'GG='LEFT'('XF1':'XL1','YF1':'YL1')'
IF JJ=4 THEN;'F11(4)='XF1''
IF JJ=4 THEN;'F22(4)='XF2''
IF JJ=4 THEN;'F33(4)='YF1''
IF JJ=4 THEN;'F44(4)='YF2''
IF JJ=4 THEN;'PP='RIGHT'('XF2':'XL2','YF2':'YL2')'
END II
'MN1='H1'/'MMM'';'MN2='H2'/'MMM''      /* calculate mean and std */
                                       /* for each set of measurements */
'MN3='H3'/'MMM'';'MN4='H4'/'MMM''
'ME1='MN1''
'ME2='MN2''
'ME3='MN3''
```

```
'ME4='MN4''
DO LL=1 TO MMM
'V1='V1'+('MN1'-B(0,'LL'))**2';'V2='V2'+('MN2'-B(1,'LL'))**2'
'V3='V3'+('MN3'-B(2,'LL'))**2';'V4='V4'+('MN4'-B(3,'LL'))**2'
END LL
'SD1=SQRT('V1'/'MMM')';'SD2=SQRT('V2'/'MMM')'
'SD3=SQRT('V3'/'MMM')';'SD4=SQRT('V4'/'MMM')'
'S1='SD1'';'S2='SD2''
'S3='SD3'';'S4='SD4''
'TRANSFER ME1';PULL ME1
'TRANSFER ME2';PULL ME2
'TRANSFER ME3';PULL ME3
'TRANSFER ME4';PULL ME4
'TRANSFER S1';PULL S1
'TRANSFER S2';PULL S2
'TRANSFER S3';PULL S3
'TRANSFER S4';PULL S4
SAY' '
SAY'IMAGE POINT NO.' JJ
SAY'=============================================================='
SAY'  MNX1     MNY1     MNX2     MNY2'
SAY  ME1 ME2 ME3 ME4
SAY'       STDX1            STDY1            STDX2            STDY2'
SAY  S1 S2 S3 S4
SAY'=============================================================='
SAY' '
'R(0,'JJ')='MN1'';'R(1,'JJ')='MN2''
'R(2,'JJ')='MN3'';'R(3,'JJ')='MN4''
END JJ

                /* enlarge the window constructed around each image point */

'CC=CHOP('CC',TO,'DIM','DIM')';'DD=CHOP('DD',TO,'DIM','DIM')'
'WW=CHOP('WW',TO,'DIM','DIM')';'ZZ=CHOP('ZZ',TO,'DIM','DIM')'
'QQ=CHOP('QQ',TO,'DIM','DIM')';'UU=CHOP('UU',TO,'DIM','DIM')'
'GG=CHOP('GG',TO,'DIM','DIM')';'PP=CHOP('PP',TO,'DIM','DIM')'
'SCREEN DEF 1 0 'DIMM' 0 'DIM''

                /* display the enlarged images side-by-side */
```

```
'DISP 'CC'||'DD'||'WW'||'ZZ'||'QQ'||'UU'||'GG'||'PP' IN 1'
'DIM3=0'
'DIMN='DIM'*2'
DO JJ=1 TO M                                    /* start the measurement on */
                                                /* the enlarged image */

SAY '****************** IMAGE POINT NO.' JJ' ********************'
 'VV1=0';'VV2=0'
 'VV3=0';'VV4=0'
 'HH1=0';'HH2=0'
 'HH3=0';'HH4=0'
DO II=1 TO MMM BY 1
 'TRANSFER CURSOR';PULL X11 Y11 BOT
 'TRANSFER CURSOR';PULL X22 Y22 BOT
 'BB(0,'II')=('X11'-'DIM3')/'MAG'+F11('JJ')+'XLL'';'XX1=BB(0,'II')'
 'BB(1,'II')='Y11'/'MAG'+F33('JJ')+'YLL'';'XX2=BB(1,'II')'
 'DIM4='DIM3'+'DIM''
 'BB(2,'II')=('X22'-'DIM4')/'MAG'+F22('JJ')+'XRR'';'XX3=BB(2,'II')'
 'BB(3,'II')='Y22'/'MAG'+F44('JJ')+'YRR'';'XX4=BB(3,'II')'
 'HH1='HH1'+BB(0,'II')';'HH2='HH2'+BB(1,'II')'
 'HH3='HH3'+BB(2,'II')';'HH4='HH4'+BB(3,'II')'
 'TRANSFER XX1';PULL XX1
 'TRANSFER XX2';PULL XX2
 'TRANSFER XX3';PULL XX3
 'TRANSFER XX4';PULL XX4
SAY XX1 XX2 XX3 XX4'-----OBSERVATION NO.' II
END II
'DIM3='DIM3'+'DIMN''                            /* calculate mean and std for */
                                                /* each set of measurements */
 'MNN1='HH1'/'MMM'';'MNN2='HH2'/'MMM''
 'MNN3='HH3'/'MMM'';'MNN4='HH4'/'MMM''
 'RR(0,'JJ')='MNN1'';'RR(1,'JJ')='MNN2''
 'RR(2,'JJ')='MNN3'';'RR(3,'JJ')='MNN4''
DO LL=1 TO MMM BY 1
 'VV1='VV1'+('MNN1'-BB(0,'LL'))**2';'VV2='VV2'+('MNN2'-BB(1,'LL'))**2'
 'VV3='VV3'+('MNN3'-BB(2,'LL'))**2';'VV4='VV4'+('MNN4'-BB(3,'LL'))**2'
END LL
```

```
'SDD1=SQRT('VV1'/'MMM')';'SDD2=SQRT('VV2'/'MMM')'
'SDD3=SQRT('VV3'/'MMM')';'SDD4=SQRT('VV4'/'MMM')'
'MEE1='MNN1''
'MEE2='MNN2''
'MEE3='MNN3''
'MEE4='MNN4''
 'TRANSFER MEE1';PULL MEE1
 'TRANSFER MEE2';PULL MEE2
 'TRANSFER MEE3';PULL MEE3
 'TRANSFER MEE4';PULL MEE4
 'SS1='SDD1'';'SS2='SDD2''
 'SS3='SDD3'';'SS4='SDD4''
 'TRANSFER SS1';PULL SS1
 'TRANSFER SS2';PULL SS2
 'TRANSFER SS3';PULL SS3
 'TRANSFER SS4';PULL SS4
SAY' '
SAY'IMAGE POINT NO.' JJ                    /* display the mean and std */
SAY'=============================================================='
SAY'      MEANXL1        MEANYL1        MEANXL2        MEANYL2'
SAY  MEE1 MEE2 MEE3 MEE4
SAY'      STDLX1         STDLY1         STDLX2         STDLY2'
SAY  SS1 SS2 SS3 SS4
SAY'=============================================================='
SAY' '
END JJ
SAY'**********************************************************************'
'PUT('R',MONCOND1,DATA,A)'/* record the coordinates of */
                           /* the measured image points */
                           /* for the enlarged and normal size images */
'PUT('RR',MNLCOND1,DATA,A)'
SAY'RUN THE PROGRAM READCONT TO TRANSFER THE MEASURED POINTS TO'
SAY'THE PRINCIPAL POINT COORDINATE SYSTEM'
SAY' ------------------------------------------------------------'
SAY'          _____________ END OF MEASUREMENT _____________      '
SAY'                        NORMAL TERMINATION                    '
SAY' ------------------------------------------------------------'
EXIT
```

A6 Program PROF

This program was developed to measure the image points along parallel lines in the image space and to generate the model coordinates for the measured image points. The measurement interval between the successive parallel lines is defined by the user.

A6.1 Definition of Some of the Variables

FL	The calibrated focal length
XORGL, YORGL	The coordinates of the origin of the left rectified input image patch (in terms of pixels).
XORGR	The x-coordinate of the origin of the right rectified input image patch (in terms of pixels).
INITX, INITY FINX, FINY	The coordinates of the top left and the bottom right of the area of image to be measured (with respect to the origin of the display screen -- in terms of pixels).
BASE	The base value

A6.2 Definition of Some of the Arrays

LEFT RIGHT	Matrices of the left and right input images.
ADD	Matrix containing zero values which is used to suppress the effect of the blue electron gun.
A	Matrix of the generated model points

A6.3 Listing and Explanation of Program PROF

```
/* PROGRAM MEASURES THE COORDINATES OF THE LEFT AND RIGHT IMAGE */
/* POINTS ALONG X AND Y DIRECTIONS AND CALCULATES THE MODEL */
/* COORDINATES FOR EACH MEASURED POINT */

SAY'     ************************************************************'
SAY'     **                                                        **'
SAY'     **     IMAGE POINT MEASUREMENT ALONG PARALLEL LINES       **'
SAY'     **                                                        **'
SAY'     ************************************************************'

SAY' '                                  /* input the required information */

SAY'ENTER THE LEFT AND RIGHT IMAGES FOR DISPLAYING'

SAY' '

PULL LEFT RIGHT

SAY' '

SAY'ENTER THE COORDINATES OF THE ORIGINS OF THE IMAGES'
```

```
SAY'WITH RESPECT TO THE RECTIFIED IMAGES:(XL YL XR)'
SAY' '
PULL FXL FYL FXR
SAY' '
SAY'ENTER THE INCREMENTS FOR CURSOR SHIFTING ALONG "Y & X" DIRECTIONS'
SAY' '
PULL I J
SAY' '
SAY'ENTER THE INITIAL POSITION OF THE CURSOR  X-Y'
SAY' '
PULL INITX INITY
SAY' '
SAY'ENTER THE TERMINAL POSITION OF THE CURSOR X-Y'
SAY' '
PULL FINX FINY
'FL=4773.4375'                                  /* initialize variables */
'XORGL=3133'
'YORGL=2603'
'XORGR=3133'
'BASE=2312.5'
'DDD=0.038201081'
'ADD=0(1:512,1:512)'
'A=C3(0(1:3,1:800))'                        /* define the matrix A as real */
''ADD'=C1('ADD')'                       /* define the matrix ADD as integer */
'GET('LEFT',IAX1,A)'                        /* input left and right images */
'GET('RIGHT',IAX1,A)'
SAY'-------------------------------------------------'
SAY' '
SAY'PUT THE RED MOVEABLE CURSOR IN CONTACT WITH THE GREEN FIXED CURSOR'
SAY' '
SAY'-------------------------------------------------'
'STDCOL'
'CURSOR(SMALL)'
'SCREEN DEF 2 0 100 0 100'          /* define the logical screen to be  */
/*'TCOLOUR(107)'*/                  /* used for displaying the symbol + */
'GTEXT(+ 50 50 IN 2)'
'DISP SRGB 'RIGHT' 'LEFT' 'ADD' IN 1'  /* display the right and left */
```

```
                                        /* images superimposed in red */
                                        /* and green */
K=0
/*<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<*/
'INITX='INITX''; 'INITY='INITY''
DO II=INITX TO FINX BY J                /* start the measurement along */
                                        /* parallel lines */
XL=II
'FF='XL'-68'
DO JJ=INITY TO FINY BY I
YL=JJ
'HH='YL'-23'
'SCREEN MOVE 2 TO 'FF' 'HH''
'TRANSFER CURSOR'
PULL XR YR BOT
K=K+1
'SR='XR'+'FXR'+'XORGR'-256-5906.25'  /* calculate the model coordinates */
'SL='XL'+'FXL'+'XORGL'-256-3593.75'
'PX=('SL'-'SR')+(('SL'-'SR')*1.69)/'BASE''; 'LAM='BASE'/'PX''
'CRT='SL'*'DDD''
'A(0,'K')=('SL'*'LAM'*0.032)+115'
'A(1,'K')=((('YL'+'FYL'+'YORGL'-256-3593.75)+'CRT')*'LAM'*0.032)+115'
'A(2,'K')='152.75'+(-'FL'*'LAM')*0.032'
END JJ
END II
/*'PUT('A',PRF236,DATA,A)'*/          /* record the measured data */
SAY 'NORMAL TERMINATION'
EXIT
```

A7 Program CONT

This program was developed to perform measurement along contour lines using a single cursor.

A7.1 Definition of Some of the Variables

FL	Calibrated focal length
XORGL, YORGR	The coordinates of the origin of the left rectified input image patch (in terms of pixels).
XORGR	The x-coordinate of the origin of the right rectified input image patch (in terms of pixel).
BASE	The base value

A7.2 Definition of Some of the Arrays

LEFT, RIGHT	Matrices containing the left and right digital images.
ADD	Matrix of zero digits which is used to suppress the effect of the blue electron gun.
A	Matrix of the calculated model points.

A7.3 Listing and Explanation of Program CONT

```
/* THE PROGRAM PERFORMS DIRECT CONTOURING */
/* AND CALCULATES THE MODEL COORDINATES FOR EACH POINT */

SAY'    @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@'
SAY'    @********************************************** @'
SAY'    @*  ---------- DIRECT CONTOURING PROGRAM (DCP) ----------   * @'
SAY'    @********************************************** @'
SAY'    @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@'

SAY'     ___________________________________________________________  '
SAY'    |    PROCEDURES:                                            |'
SAY'    |       1- ESTABLISH THE EXACT HEIGHT VALUE FOR THE         |'
SAY'    |          CONTOUR LINE TO BE TRACED.                       |'
SAY'    |       2- MOVE THE CURSOR TO THE RIGHT CORNER OF THE SCREEN|'
SAY'    |          AND PRESS ANY BUTTON.                            |'
SAY'    |       3- TRACE THE CONTOUR LINE.                          |'
SAY'    |       4- REPEAT FROM STEP 1 UNTIL ALL SPECIFIED           |'
SAY'    |          CONTOUR LINES HAVE BEEN TRACED.                  |'
SAY'    |       5- TO TERMINATE, MOVE THE CURSOR TO THE LEFT CORNER |'
SAY'    |          OF THE SCREEN AND PRESS ANY BUTTON.              |'
SAY'    |___________________________________________________________|'
SAY''

                                   /* Input the required information*/
```

```
SAY'ENTER THE NAME OF THE LEFT AND RIGHT IMAGES FOR CONTOURING'
SAY'? ?'
PULL LEFT RIGHT
SAY'ENTER THE HEIGHT OF THE FIRST CONTOUR LINE & THE CONTOUR INTERVAL'
SAY'? ?'
PULL ZZ DZ
SAY'ENTER THE HEIGHT OF THE LAST CONTOUR LINE'
SAY'?'
PULL LZ
SAY'ENTER THE COORDINATES OF THE ORIGIN OF THE IMAGES'
SAY'WITH RESPECT TO THE RECTIFIED IMAGES: (XL YL XR)'
SAY'? ? ?'
PULL FXL FYL FXR
SAY' '
SAY'-------------------------------------------'
SAY' PUT THE RED MOVEABLE CURSOR IN CONTACT                '
SAY' WITH THE GREEN FIXED CURSOR                           '
SAY'-------------------------------------------'
SAY' '
/*-------------------------------------------------------*/
'FL=4773.4375'                              /* initialize variables */
'XORGL=3132'
'YORGL=2603'
'XORGR=3132'
'BASE=2312.5'
'DDD=0.038201081'
'GET('LEFT',IAX1,A)'
'GET('RIGHT',IAX1,A)'
'SIZX=XSIZE('LEFT')'
'SIZY=YSIZE('RIGHT')'
'ADD=0(1:'SIZX',1:'SIZY')'    /* construct a matrix with zero digits */
'A=C3(0(1:3,1:900))'                  /* define the matrix A as real */
''ADD'=C1('ADD')'                  /* Define the matrix ADD as integer */
'DIF=0'
'K=0'
'STDCOL'
'CURSOR(SMALL)'
```

```
/*'TCOLOUR(107)'*/
'SCREEN DEF 2 0 100 0 100'          /* define the logical screens */
'SCREEN DEF 3 0 270 0 270'            /* for displaying the texts */
'SCREEN DEF 4 0 400 0 400'
'SCREEN DEF 5 0 400 0 400'
'GTEXT(+ 50 50 IN 2)'
'SCREEN MOVE 2 TO 300 300'  /* shift screen 2 to position 300  300  */
'TXT="Z - MM"'
'TXT2="POINT NO."'
'THEIGHT(20)';'TCOLOUR(45)'
'GTEXT(TXT 70 80 IN 4)'                    /* display text in screen 4 */
'TCOLOUR(80)'
'GTEXT(TXT2 70 130 IN 4)'
'THEIGHT(50)';'TCOLOUR(127)'
/*-------------------------------------------------------------------*/
DO Z=ZZ TO LZ BY DZ                                 /* start the first loop */
'DISP SRGB 'RIGHT' 'LEFT' 'ADD' IN 1'  /* display the right and left */
                                       /* images superimposed in */
                                       /* red and green */
DO FOREVER
 'TRANSFER CURSOR'                     /* measure the image points */
  PULL XU YU BUTTON                    /* stereoscopically and calculate */
                                       /* the model points */
   IF XU>800 THEN LEAVE          /* if the required height is obtained */
                                 /* move to the contour tracing mode */
  'XL='XU'';'YL='YU''
  'FF='XL'-68';'HH='YL'-23'
  'SCREEN MOVE 2 TO 'FF' 'HH''
  'TRANSFER CURSOR'
  PULL XR YR BUTTON
 'SCREEN CLEAR(3)'
 'SR='XR'+'FXR'+'XORGR'-256-5906.25'
 'SL='XL'+'FXL'+'XORGL'-256-3593.75'
 'PX=('SL'-'SR')+(('SL'-'SR')*1.69)/'BASE''
 'LAM='BASE'/'PX''
 'MDLZ=152.75+(-'FL'*'LAM')*0.032'
 'TRANSFER MDLZ';PULL MDLZ
 'GTEXT('MDLZ' 50 50 IN 3)'
```

```
   END
'SCREEN MOVE 2 TO 100 900'
SAY''
                                        /* start the direct contouring mode */
SAY <<<<<<<<<<<<<<<<< TRACE THE CONTOUR LINE >>>>>>>>>>>>>>>>>'
SAY'                        FOR THE HEIGHT' MDLZ
SAY''
'DIF='XL'-'XR''              /* shift the right image with respect */
'DIS=512-'DIF''              /* to the left one so that the x-separation */
    'RGT='RIGHT'(1:'DIS',0:511)'   /* for the measured point in the */
    'BBB='1(0(1:'DIF',0:511))'     /* first mode, is reduced to zero */
    'RRIGT='BBB'||'RGT''
    'DISP SRGB 'RRIGT' 'LEFT' 'ADD' IN 1'
'LL=0'
DO FOREVER                                  /* trace the contour line */
  'K=K+1'
    'LL=LL+1'
'TRANSFER CURSOR'
    PULL XU YU BUTTON
    'SCREEN CLEAR(5)'
    IF XU > 800 THEN LEAVE  /* if tracing the contour line */
                            /* is finished, move to the first */
                        /* mode to start the next contour line */
    IF XU<100 THEN DO               /* if no more contour line */
    'PUT('A',ERDCN4,DATA,A)'        /* is required, terminate */
                                    /* the program and record */
                                    /* the measured points */

     SAY' ----RUN PROGRAM PTRD TO PRODUCE ASCII OUTPUT---'
      SAY'  '
      SAY'  '
      SAY'  '
     'MENUE2'
      EXIT
    END
'GTEXT('LL' 230 140 IN 5)'
'XL='XU''; 'YL='YU''
'SL='XL'+'FXL'+'XORGL'-256-3593.75'
```

```
        'CRT=('SL'*'DDD')'
        'YLL=('YL'+'FYL'+'YORGL'-256-3593.75)'
        'YMM='YLL'+'CRT''
        'A(0,'K')=('SL'*'LAM'*0.032)+115'
        'A(1,'K')=(('YLL'+'CRT')*'LAM'*0.032)+115'
        'A(2,'K')='MDLZ''
   END
END Z
                                    /* record the calculated model points */
'PUT( A',ERDCN4,DATA,A)'
SAY''
SAY'             @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@'
SAY'             @@                                               @@'
SAY'             @@   -------------  END  OF  TRACING  --------   @@'
SAY'             @@                                               @@'
SAY'             @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@'
SAY''
SAY 'NORMAL TERMINATION'
EXIT
```

A8 Program CORRL

This program was developed to perform one-dimensional image correlation along epipolar lines using the shifting window method and the normalized cross-correlation algorithm.

A8.1 Definition of Some of the Variables

INTN, APPP	Initial position of the left and corresponding right image points.
IWINSZ, IWINY	Maximum column and row numbers of the window array.
SRCHSZ	Maximum column number of search array.
NUMX, NUMY	Maximum column and row numbers of the input image.
SIGW	Sum of the grey levels of the window array
SIGS	Sum of the grey levels of the search array
SIGW2	Sum of the squares of grey levels of the window array
SIGS2	Sum of the squares of grey levels of the search array.
SIGWS	Sum of the products of the window and search arrays.

A8.2 Definition of Some of the Arrays

IML, IMR	Matrices containing the left and right digital images.
COEF	Matrix containing the calculated normalized cross correlation function.

A8.3 Explanation of Program CORRL

Block 1 : Arrays are declared; variables are initialized; parts of the left and right digital image patches are input to the program.

Block 2: Sum of the grey levels of the window and search arrays are computed; sum of the products of the window and search arrays are determined; sum of the squares of each of the window and search arrays are calculated.

Block 3 : Normalized cross correlation function is calculated

Block 4 : If the maximum cross correlation function is above a specified threshold value, the row and column numbers of the left and right images corresponding to the maximum cross correlation function, are accepted as the image coordinates of the matched points and a sub-pixel shift to the matched points is calculated and added to the image coordinates of the matched points.

Block 5 : The model coordinates are calculated and recorded.

Block 6 : The program returns to the first stage and reads the next image lines and continues the procedures described above until the whole area of interest is covered.

A8.4 Listing of Program CORRL

```
C@PROCESS DIR('IBM')
C     ************************************************************
C     **      The program performs image correlation using the    **
C     **              window shifting method and                  **
      **          normalized cross correlation algorithm          **
C     ************************************************************
C
      PROGRAM MATCH
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      INTEGER*2 IML(2000,21),IMR(2000,21)
      DIMENSION  COEF(-3:60)
C-----------------------------------------------------------------
      INTN=100
      APPP=100
      IWINSZ=15
      IWINY=15
      SRCHSZ=25
      NUMX=1920
      NUMY=1800
      LAPS=0
      DLZB=-100
C
          XORGL=3133-1
          XORGR=3133-1
          YORGL=2603
          FXL=0
          FYL=0
          FXR=0
          FL=4773.4375
          BASE=2312.5
          EPI=0.038201081
          CCCC=FXR+XORGR-5906.25
          DDDD=FXL+XORGL-3593.75
          EEEE=FYL+YORGL-3593.75
C-----------------------------------------------------------------
C                     READ THE LEFT AND RIGHT IMAGES
C-----------------------------------------------------------------
```

```
                SIGW2=SIGW2+IML(KKK,J)*IML(KKK,J)
                SIGS2=SIGS2+IMR(I,J)*IMR(I,J)
   30         CONTINUE
C- ------------------------------------------------------------------
C          CALCULATE NORMALIZED CROSS CORRELATION FUNCTION
C-----------------------------------------------------------------
                ANUM=IWINSZ*IWINY*SIGWS-SIGW*SIGS
                DEN1=IWINSZ*IWINY*SIGW2-SIGW*SIGW
                DEN2=IWINSZ*IWINY*SIGS2-SIGS*SIGS
                LL=LL+1
                IF((DEN1*DEN2).EQ.0.0)  GO TO 1234
1010
                COEF(LL)=ANUM/SQRT(DEN1*DEN2)
                  IF(AMAX.LT.COEF(LL)) THEN
                  AMAX=COEF(LL)
                  RANK=LL                                    BLOCK(3)
                  RR=KK
                  END IF
1001        CONTINUE
C ---------------------------------------------------------------
C          SET A THRESHOLD VALUE AS CORRELATION CRITERION
C---------------------------------------------------------------
        IF(AMAX.GE.0.73) THEN
C ------------------------------------------------------------------
C       CALCULATE SUB PIXEL SHIFT USING SECOND DERIVATIVE
C-----------------------------------------------------------------
              IF(COEF(RANK+1).EQ.0.0.OR.COEF(RANK-1).EQ.0.0) THEN
              DX=0.0
              GOTO 1212
              END IF
              DER=COEF(RANK+1)-COEF(RANK-1)
              SDER=-4*COEF(RANK)+2*COEF(RANK+1)+2*COEF(RANK-1)
              IF(SDER.EQ.0.0) GOTO 1212
              DX=DER/(-SDER)
 1212         XR=RR-0.5+DX
              XL=INT-0.5
C             WRITE(*,555) XL,XR,III
C             PRINT*,COEF(RANK-1),COEF(RANK),COEF(RANK+1)
              APP=RR-3
```

```
      NK=NK+1                                                    BLOCK(4)
      IF(NK.EQ.1) THEN
      IF(XL.LT.220.AND.XL.GT.80) THEN
      APPP=RR-5
      INTN=XL-5
      ELSE
      APPP=90
      INTN=100
      END IF
      END IF
C ------------------------------------------------------------------------
            IF(LAPS.LE.7) GOTO 1234
            SR=XR+CCCC+IWINSZ/2
            SL=XL+DDDD+IWINSZ/2
            CRT=SL*EPI
            PX=(SL-SR)+((SL-SR)*1.69)/BASE
            ALAM=BASE/PX
            DLX=SL*ALAM*0.032+115
            DLY=(III+7.5+EEEE+CRT)*ALAM*0.032+115
            DLZ=(-FL*ALAM)*0.032+252.75
C
            SLP=ABS(DLZ-DLZB)
      IF(SLP.GT.0.015) THEN
      WRITE(10,444) DLX,DLY,DLZ
      WRITE(11,777) DLX,DLY,COEF(RANK)                           BLOCK(5)
      WRITE(12,777) DLX,DLY,(-SDER)
      END IF
         DLZB=DLZ
         ELSE
         APP=RR-IWINSZ/2
         END IF
 1002 CONTINUE
C ------------------------------------------------------------------------
 1234      DO 515 I=1,NUMX
           DO 515 J=1,IWINY-1
               IML(I,J)=IML(I,J+1)
 515           IMR(I,J)=IMR(I,J+1)                               BLOCK(6)
```

```
 1003  CONTINUE
C
 444   FORMAT(3F15.4)
 555   FORMAT(2F10.3,I8)
 777   FORMAT(3F15.5)
C
       STOP
       END
```