Microgravity Measurement In Space Using Imaging Techniques

Mark John Steedman

The Robert Gordon University, Aberdeen

September 1996

Microgravity Measurement In Space Using Imaging Techniques

Mark John Steedman

The Robert Gordon University, Aberdeen

Submitted for the partial fulfilment of the requirements for the degree of Doctor of Philosophy

Collaborating Establishment : NASA, Goddard Space Flight Centre.

September 1996

Abstract

The proposal is made that very small changes in gravitational field could be detected by monitoring a cell of heated fluid. Variations in gravity would be observed by their effect on the convection in the fluid cell. It is proposed that the convection patterns be observed by using an interferometer to image the temperature gradient in the fluid by utilising the effect of temperature variations on the refractive index of the fluid. It was proposed that this system may be able to detect changes in the Earth's gravitational field from orbit. The possibility of using the NASA GAS programme to perform a space flight test of this proposal was suggested. Activities in this programme were therefore surveyed and a proposal made.

The primary experimental results were recorded in the form of convection images which were later processed on the ground to extract information. Research has been carried out in this area as investigation has shown that existing image processing techniques are not suitable to process the anticipated fringe images. Meanwhile design and development of a GAS experiment was performed. This was undertaken in liaison with NASA in order to achieve the required safety approvals for flight. Subsequently the flight experiment was performed aboard the Space Shuttle Endeavour during the STS-77 mission launched on the 19th May 1996.

As a result of this project an image processing system for the analysis of interferogram images of convection has been developed and an experiment to image convection in microgravity, with a view to analysing its use for the detection of changes in gravity has been performed.

Declaration

I hereby declare that this thesis is a record of work undertaken by myself, that it has not been the subject of any previous application for a degree, and all sources of information have been duly acknowledged.

Acknowledgements

I would like to the assistance of the following in the research programme and experiment described in this report.

Professor N.D.Deans, as my project supervisor.

Mr Tom Dixon, NASA Technical Manager for the space flight experiment.

The technical support staff of the School of Electronic and Electrical Engineering at the Robert Gordon University for assistance with the preparation of the flight experiment.

I would like to thank the Robert Gordon Universtiy for finacially supporting this project.

Contents

List Of Figures

Chapter 4

Chapter 5

Chapter 6

Chapter 7

Abbreviations

NASA	National Aeronautics and Space Administration
STS	Space Transport System (often refered to as 'Space Shuttle')
PLD	Programmable Logic Device
EPROM	Electronically Programmable Rear Only Memory
GAS	Generate Activities in Space (Also: 'Get Away Special')
GCD	GAS Control Decoder
SDP	Safety Data Package
PAR	Payload Accomodations Requirements
EMI	Elecromagnetic Interference
EPLD	Electronically Programmable Logic Device
Matlab	A Computer Mathematics Package
APC	Autonomous Payload Controller
PPC	Payload Power Contector
G-490	Get Away Special experiment, reservation number 490, the Robert Gordon Universities experiment.

Chapter 1

Introduction

1.1 Project Summary

1.1.1 Overview and Objectives

A programme of research and development work has been carried out on image processing techniques suited to the analysis of interferometrically generated microgravity convection images. It is intended that these techniques be used to study interferogram images recorded in space by the Robert Gordon University NASA project.

The NASA project had as its principal objective, the flight in space of an experimental payload intended to test a proposed novel method for measuring gravity. The development and implementation of the space flight experiment to test the proposed measurement has been completed and a successful flight has taken place.

Existing approaches to image processing are unable to extract meaningful features from images of the type it was anticipated convection in microgravity would generate. The research results presented in this thesis therefore cover investigations carried out on image processing techniques, developed to process the visual data that the space experiment was designed to collect for later analysis on Earth.

The project objectives included :

- A literature review of the GAS programme.
- A review of techniques for measuring low gravitational fields.
- A review of transport phenomena in fluids.
- Liaison with NASA to prepare the payload and documentation for flight.

- Payload design, development and integration with the carrier hardware.
- The compilation of a database of interferogram fringe images.
- Performing an evaluation of existing image processing techniques.
- Processing of sample, and flight images to determine information content.

1.1.2 The NASA Project

The Robert Gordon University NASA Project commenced in 1989 to formulate a submission to an IEE run competition to win an opportunity to fly a payload aboard the NASA Space Shuttle. Conceptual testing of the proposed experiment, as discussed in Section 1.4.1, and addressed in more detail in Chapter 2, Section 2.5.1, was performed in 1990.

The proposal proceeded to win the competition and a series of student project were performed on the proposed payload. The author joined the project in late 1993 to provide full time work on the payload design, co-ordination and paperwork preparation, and to perform research on the processing of the anticipated images. At this time the experiment as introduced in Section 1.2.2, and experimental platform had been defined and an initial draft of the first requirements document (See NASA documentation in Chapter 6, Section 6.5) explaining this to NASA had been written.

1.2 Gravity Measurement

1.2.1 Existing Methods

The measurement of gravity is of interest for a number of reasons. Knowledge of variations in gravity, when combined with other geographical and geological information, can provide an insight into the composition of terrain. For example, information on the underlying rock formations may be deduced, that purely visual, or contour data cannot provide. This is due to various sources of gravity variations, for example changes in underlying rock density, and therefore gravity, and variations in the surface profile of the terrain. It has been proposed that gravity

maps could be used for navigation, and that the monitoring of gravity over time could be used to monitor changes both on, and below, the surface of the Earth. Some limited monitoring of gravity is already carried out using delicate precision accelerometers. Similarly, gravity mapping of the oceans has been performed using ship-borne instruments. Given the large geographical area to be covered, however, a faster scanning system is preferable, especially a profiling technique that could be used on land as well as at sea. The mapping of gravitational fields using orbital vehicles therefore has enormous advantages.

Variations in the gravity field experienced by a spacecraft could be expected to be generated by several sources, thruster firings (10^{-1} - 10^{-3} g), residual atmospheric drag (10^{-2} - 10^{-7} g) and variations in gravitational acceleration (at right angles to the drag, 10^{-3} - 10^{-5} g).

1.2.2 Proposed New Method

The proposal is made that very small changes in gravitational field could be detected by monitoring a cell of heated fluid. Variations in gravity would be observed by the effect on the convection pattern observed in the fluid cell. To observe the convection patterns, an interferometer is used to image the temperature gradient in the fluid by utilising the effect of temperature variations on the refractive index of the fluid. This results in variations in the refraction of the light beam passing through the heated cell. This distortion then produces interferogram fringe patterns from optical interference between light that has followed different paths through the fluid, and differing levels of refraction. Variations in gravitational field that result in an acceleration vector anywhere in the two dimensional plane perpendicular to the heating element are observed by the displacement of the interference fringes which form in the cell as the result of varying refractive indices. This provides an indication of the gravitational field, as the fringes correspond to contours of equal refractive index.

1.3 Space Flight

1.3.1 Reasons For Testing In Orbit

There are three principal reasons for testing the system in Earth orbit. Firstly it has been suggested that the new system may be able to detect changes in the Earth's gravitational field from orbit, something which has previously only been achieved using indirect methods (as discussed in Section 2.1.4). Secondly, low Earth orbit is a 'sustained free fall' environment and therefore the dominant affect of the Earth's overall gravitational field will be absent, and concerns about this masking the variations from much smaller disturbances and variations in the overall gravity field will be removed. This will allow a much more effective test of the true sensitivity of the proposed new gravity detector. Finally, low Earth orbit provides a scanning track across the Earth's surface which provides for far better coverage per unit time of the Earth's surface than any other viewpoint.

1.3.2 Experimental Platform

The payload was scheduled for flight in the NASA Get Away Special (GAS) programme. This programme was conceived by NASA while developing the Space Shuttle. It was intended as the junior partner of an initial two small payloads programmes, 'Get Away Special' and 'Hitchhiker', that could provide access to space for very low cost. Payloads fly in the programme on a space and weight-available basis in a first come, first served, queue of payloads that have been approved by NASA. The experimenters are required to provide the power supplies, control and data logging facilities. The basic GAS carrier hardware configuration is a sealed canister provided by NASA to isolate the experiment from the worst features of the space environment and the Shuttle itself. Three relay inputs from the astronauts are the only external inputs to the canister and one of these must be dedicated to control power supply to the payload so that NASA can disable the supply and thereby render the payload inert in the event that an emergency occurs.

1.4 Research Objectives

1.4.1 Simulation Testing

Preliminary testing was done during the experiment definition and development phase in 1990 to obtain a conceptual indication of the results format. These experiments used a heated resistor (in air) to generate simulated fringe patterns indicating the form of the results to be expected. Varied results were obtained, some simulating zero gravity and others giving rise to more typical convection patterns. Other attempts to give 'low' gravity results resulted in distorted, but still complete, concentric fringe patterns (anticipated to be generated in space). A more detailed overview of the initial concept testing work is presented in Chapter 2, Section 2.5.1.

The initial ground test images were an important first step as they indicated the types of images that might be expected and proved that the proposed experiment could be implemented and therefore tested. This preliminary testing enabled the development of the experimental platform to perform the space flight test and the development of image processing algorithms to process the results, to be carried out.

The testing of the experimental equipment itself directly prior to the flight produced concentric fringes but only imaged part of the fringe pattern. It was noted however that the contrast in these images was much superior to previous ground test images. The data from the initial tests are used exclusively in Chapters 2 to 5 as the only images available during research and development of the image processing system. Work on processing the results from the images generated by the flight equipment is discussed in Chapter 7.

1.4.2 Image Processing

The primary experimental results are recorded in the form of convection images which are later processed on the ground to extract information. Research has been performed in this area as investigation into several existing image processing techniques has shown that existing image processing techniques are not suitable to process the anticipated fringe images. However as discussed in more detail in

Section 3.1.7, techniques from a number of existing approaches to image processing were used at a later stage in the research programme.

The image processing research has been undertaken to provide a means of processing the expected experimental results. The core of the gravity detector is a cylindrical cell of fluid with a heating element running along its centre line. This is intended to produce convection currents in any direction in a two dimensional plane which are then imaged by placing this in one arm of a Michelson interferometer which projects a fringe image onto the imaging plane of a CCD video camera. The resultant fringe image will in the case of pure conduction be formed of concentric circles. In practice however distortion of this nominal image will be generated by any convection induced in the fluid by variations in the gravity field the experiment is passing through, see the ground simulation work in Chapter 2. Chapter 3 reports upon the results of an extensive literature survey carried out in order to identify any previously existing research capable of processing fringe images of the type the experiment is expected to generate. A number of possible solutions identified by the literature survey have been implemented and tested, but in all cases have either failed completely or required far too much operator assistance to be capable of processing a representative sample of the images to be generated. Although only a fraction of these may have to be processed in practice, a fully automatic fringe processing system is still desired to minimise operator bias and effort.

1.5 Subsidiary Aspects

1.5.1 Liaison With NASA

Due to the space flight opportunity for the experiment being in the NASA Shuttle Programme, extensive liaison with NASA was required. This was principally in order to ensure that the experimental equipment that was developed for the GAS payload was suitable for the constraints placed on it by the carrier system and that it met the required standard of safety. The NASA safety procedures are explained in

detail in Chapter 6 and have been the cause of a considerable amount of work during the development of the flight experiment.

1.5.2 Thermal Modelling

Thermal modelling of the payload was a requirement of the safety approval process to prevent over-temperature of the experiment's environmental control system. Combining this modelling work and the experiment's own thermal monitoring system also provides the possibility of comparing the model, with actual flight results.

1.6 Thesis Overview

This thesis describes three areas in which work has been carried out. Prior to reviewing the details of the work that has been undertaken Chapter 1 has introduced the background of the project. Chapter 2 considers the development of the experiment and the conclusions of the literature reviews undertaken. Detailed development of the novel image processing techniques research, and related investigations into existing research on image processing is covered in Chapter 3, Chapter 4 discusses the development of processing schemes and Chapter 5 the final image processing algorithm. Work undertaken to achieve safety approval for the space flight experiment in liaison with NASA to obtain permission to fly the experimental equipment is discussed in Chapter 6 along with the design and construction of the payload. Chapter 7 reviews results that have been collected, and finally Chapter 8 discusses the conclusions to be drawn from the project.

Chapter 2

Measurement Of Microgravity

2.1 Review of Existing Methods For Measuring Microgravity

2.1.1 Aims of the Review

Investigations of existing methods of measuring microgravity were undertaken in order to establish if any previous work had been performed to measure microgravity using convection currents as a measuring tool. The similarity of any techniques which have been proposed, or actually flown in space, and their capability is of particular interest to enable comparisons of the different techniques presently available with the proposed new approach to be made.

This chapter discusses methods of gravity measurement from geological work and techniques relating to the space experiment. The concept of the gravity measurement experiment is then introduced. The possibility of modelling the experiment, from a gravity vector to simulated experimental results is then considered. Finally, the source of the images used during the research is discussed and a proposed technique for extracting a gravity vector from these images is presented.

2.1.2 Current Ground-Based Methods Of Measuring Gravity

Gravitational forces are commonly measured on the Earth's surface with portable precision accelerometers. This approach is used by geologists, particularly in work with volcanoes, although it only provides readings for individual locations and requires a ground team to take precision instruments to each reading site in turn. This work is established in the field of geology but far removed from the

measurement of gravitational field variations in the microgravity environment of low Earth orbit. More closely related research using linear accelerometer systems in space is discussed below in Section 2.1.3. More recent attempts to improve basic ground-based gravity measurement techniques have been reported. The measurement of variations in the Earth's gravitational field from aircraft and moving vehicles has been documented by Hehl [1]. Hehl discusses the measurement of gravity on a profile by mounting a linear accelerometer on a stabilised platform and then moving the whole assembly while recording. Reference is made to long-established measurements from ships but notes that such work is time consuming as ships are expensive to operate and very slow, and the obvious fact that the data sets are limited to the oceans. These measuring techniques are still under development but have the capability to provide gravity profiles as opposed to simply multiple readings. The ground-based approach, using a moving vehicle as the experimental platform, is limited to locations that can be reached by vehicle. Reasonable roads are desirable for mobile recording in order to record gravity profiles without interference from the vibration of the recording platform. No work has been reported with respect to the use of convection currents as a method of measuring microgravity.

2.1.3 Space Microgravity Measurement Experiments

The review of space-based microgravity measurement work detailed herein encompasses research activities ranging from work on vibration in space, to techniques for measuring gravitationally induced convection. The principal areas reported have been the measurement of residual atmospheric drag, Howell [2], and vibration from crew movements, or platform equipment, Jones [3].

The measurement of vibrations aboard the Space Shuttle using linear accelerometers has been reported. The aim of the work in these investigations, with the exception of STEP, Jafry [4] which is still a proposal, has been to measure vibration levels from the accelerations produced rather than actually measure the level of the gravitational field itself. Work has been carried out and reported to measure the effects of vibration from equipment, Arrott [5], Jones [3], atmospheric drag, Howell [2] and general drift and thruster compensation for it on the Shuttle,

Werner [6]. The direct measurement of fluctuations in the Earth's gravitational field from space remains undocumented. No reports from either the Soviet/Russian space programme or the European Space Agency on microgravity measurement exist. It is reasonable to assume however that the ESA and the Soviets examined the problems that vibration causes with microgravity crystal growth experiments and that the Russian space programme continues to formulate experiments in this area.

The proposed STEP payload, Jafry [4] will include gravity mapping of the Earth from space using linear accelerometers, but its primary purpose is an investigation of the gravitational force itself, and this experiment is still far from being launched. The gravity fields of other planets in the solar system have been measured from space by an indirect method based on measuring fluctuations in the orbital velocity of a spacecraft placed into a gravity sensitive orbit about the target planet. Variations in the propagation delay of its data down-link to Earth are then used to determine variations in the orbital velocity of the satellite from which the fluctuations in gravity may be determined.

The principal method reported for measuring vibration and acceleration in microgravity is the use of a triplet of linear accelerometers. This is discussed for the development of a microgravity monitoring instrument and its application to the guidance, navigation and control of a vehicle, Howell [2]. Howell details NASA's activities in the area of microgravity measurement and covers three experiments. These include, the 'Small Expendable Deployer System/ End-Mass Payload' (SEDS/EMP), which uses a tri-axis accelerometer assembly to monitor accelerations to 10^{-4} g on a payload suspended into the upper atmosphere on a 20km tether from an orbiting Delta rocket. The second experiment, the 'Rarefied-Flow Aerodynamic Measurement Experiment' (RAME), will use a similar tri-axis accelerometer assembly but with a 10^{-6}g measurement capability. Both of these systems are intended to measure accelerations on payloads arising from contact with the rarefied upper atmosphere (which is effectively a vibration and drag measurement). The third experiment, the 'Space Acceleration Measurement System' (SAMS) is intended to have a measurement capability in the 10^{-9} g range for measuring very small disturbances in microgravity science and technology experiments. The combination of low frequency (<5Hz) and high accuracy measurement capability of this system is

intended to study the aerodynamic behaviour of the STS Orbiter in the rarefied flow environment at orbital altitudes. High accuracy accelerometer systems (10^{-6} g), are also required by Jones [3], though the interest in this case is in eliminating vibrations in a space environment as part of a feedback control loop, rather than measuring them. The accelerometers required in this instance to provide the feedback to the active vibration control systems.

2.1.4 Conclusions of the Review of Gravity Measurement

The principal result of the review of previous work is that no reports exist indicating that the proposed method has already been tested. Attempts have been made to measure the gravity of other planets from space using a different approach, and references have been made, Jafry [4] to previous data on the Earth's gravitational field measured using these satellite tracking based approaches. Several references to the inclusion of linear accelerometers in previous space payloads have been found, Arrott [5], Howell [2], Jones [3]. This is to be expected considering the importance of the level of even the tiny residual gravitational forces in orbit to experiments such as crystal growth. The primary purpose of the accelerometer systems in most cases has been to investigate the effect of spacecraft systems on the quality of an experiment's microgravity environment. The typical instantaneous disturbances have invariably proved to be the operation of mechanical devices aboard the vehicle carrying the experiment, or the movements of its crew. The conclusion of this review is that no experiments with a similar aim to the proposed microgravity convection experiment have flown.

2.2 Literature Survey Of Fluids And Convection

2.2.1 Objectives of Reviewing Fluids and Convection

An investigation of reported work on the behaviour of fluids in microgravity conditions has been carried out to establish if any previous experimental work has

been performed on the method proposed. Initially, an investigation of the Marangoni convection effect was undertaken. The review of the existing research was used to assess whether this phenomena could be expected to occur in the proposed experiment. It was important to establish at an early stage whether Marangoni convection was likely to be measurable by the experiment as a secondary effect or if it would dominate the conventional convection patterns, thereby making it an undesirable phenomena. The principal aim of the investigation on classical convection was to check if the convection patterns to be anticipated in space could be interpolated from Earth-based experiments that had previously been carried out during the conceptual phase of the experiment design.

2.2.2 Behaviour Of Fluids In Convection

Extensive searches for reports on the behaviour of fluids, and on techniques for measuring the behaviour of fluids under microgravity conditions have been conducted. The onset of convection, and the frontier between oscillatory and stationary convection have been investigated using a mathematical approach, Mardones [7]. The existence of the convection threshold, Mardones [7], is a principal reason why a test of the convection phenomena in space under microgravity conditions is of interest. Having assumed that convection does indeed occur, the modelling of natural convection in a 'low-gravity' environment is dealt with mathematically, Jue [8]. This simulation considers natural convection in the presence of thermocapiliary and gravity modulation effects in low-gravity environments. The study of oscillatory convection in a gravitationally modulated layer is expanded to three dimensions by Clever [9], which, although relating to a dissimilar condition to the anticipated microgravitational field, is more representative than the typically analysed steady state condition. This is despite the fact that in this case, the experimental cell arrangement is with heating from below. The development of convection currents in microgravity is discussed by Vetashen [10], with reference made to acceleration measurements made on Salyut spacecraft and the Mir space station (however the convection currents are not being imaged or used to measure the gravitational field). The nature of buoyancy-driven liquid flows is described by

Kessemi [11], in a reduced gravity environment. Natural convection interactions in 1-g and low-g environments have been analysed by Kessemi [12] by examining the interactions between radiation and convection based heat transfer using the discrete exchange factor method. This work however provides little direct insight into the likely nature of the flow to be anticipated in a particular set of conditions that is dissimilar to that addressed directly. Similarly it does not provide an easy means of determining the onset point for convection. The principal finding is that these factors are too specific for general conclusions to be drawn.

The measurement of temperature gradients in fluids has been extensively documented. Holographic interferometry has been used, Ito [13] to investigate flame spread using fringe patterns. The interpretation presented however is entirely reliant on human knowledge and expertise and the required holographic interferometer system is considerably more complex than the Michelson arrangement proposed herein. Holographic interferometry is again used to observe the temperature field near to condensing bubbles by Chen [14] though no processing of the resulting interferogram images is attempted. The only previous work on thermal gradients in fluids in space, which has been found by the author, is the mapping of thermal gradient using an infra red camera, Verga [15] relating to a payload flown in the European Space Agency's spacelab in 1993. The system was intended primarily to investigate Marangoni convection and a large gas-liquid interface was used to maximise this effect.

Relevant to the current experimentation is the reporting on the movement of fluids and methods of measuring that movement. Particle tracking, Vshijima [16] has been used to monitor fluid flow in a simulated reactor system, although this has been improved upon by a three dimensional particle velocity monitoring system, Coupland [17] using holographic techniques. Pattern-matching has also been used by Yan [18] for the tracking of fluid movements in ocean currents using sequential thermal satellite images.

2.2.3 Marangoni Convection

The study of Marangoni convection under varying liquid properties has been addressed by simulation, Hapolitano [19] and likewise the onset point of the phenomena has been addressed, Ybarra [20] by calculation. The analysis is extended by looking at Marangoni convection in microgravity, Bauer [21], with G-jitter being considered by Ramachandran [22]. Further work by Bergoon [23] considers other disturbances and unusual container shapes. The suppression of the phenomena has also received attention by Dressler [24]. Further conditional cases are also covered, including coaxial cylindrical surfaces, Napolitano [25] and spherical surfaces, Hoefsloot [26]. The effect of Marangoni convection on mass transfer, but not the resulting convection itself, was studied during the D1 Spacelab mission by Lichtenbelt [27].

Marangoni convection in space was recorded aboard the USML-1 Spacelab, Kamotani [28]. Streamline patterns of the resulting convection were imaged by the illumination of particles in the fluid by a sheet of laser light but these were not analysed. Instead, they were compared visually with computer simulations of the experiment and found to be in good agreement.

2.2.4 Summary Conclusions On Fluid Behaviour

The behaviour of fluids during convection, both conventional and Marangoni, is a subject on which a considerable amount of work has been carried out.

The applicability of Marangoni convection to the proposed experiment has been assessed and it has been concluded that this effect is unlikely to occur in the liquid cell in the proposed experiment. The reasoning is that Marangoni convection is a phenomena that is induced when a temperature gradient exists across a gas-liquid interface, and it is intended that there be no bubbles in the liquid in the proposed experiment (as the separation of the liquid and gas phases in low gravitational fields cannot be guaranteed unless the liquids surface tension is relatively high).

The possibility of phase mixing occurring between the liquid and a gas pocket is unknown, but it could be expected to make predicting the fluid behaviour

more difficult. It could lead to the heating element overheating, but not failing, were a gas bubble to settle on it, considering the much lower thermal capacity of gasses as compared to liquids.

The effect of gas-to-liquid interfaces on diffraction in the cell could possibly hide the temperature gradient changes the experiment needs to be able to detect were a bubble to settle in the cell centre. Reported research suggests that if Marangoni convection were to occur in the experiment, that it would probably mask out the residual buoyant convection effects, particularly as the higher surface tension fluids that would not mix with gasses in microgravity are the same ones in which Marangoni convection would be strongest. However the same literature suggests that Marangoni convection will not occur as, due to the cylindrical cell shape, any small bubbles will be evenly heated and will therefore not support the phenomena.

2.3 Experiment To Measure Gravity

2.3.1 Experiment Concept

The aim of the proposed space research programme is to investigate and develop a novel technique for measuring microgravity with the intention that this can be used to measure small changes in the Earth's gravitational field. The suggested measuring phenomena is the creation of convection currents in a cylindrical fluid cell due to the presence of gravity fluctuations. As the first stage in this process it is proposed to test the concept of the experiment. The proposed experiment is therefore intended to answer the question 'is there a visible convection effect', and no attempt is made to calibrate the equipment. If the processing of the images demonstrates an effect can be seen a second experiment with a calibration system would be required. This however could be designed utilising lessons learnt during the conceptual test which will highlight any problems to be overcome.

Nominally there should be no convection in space due to the free-fall nature of a low Earth orbit. Any convection currents that are present will therefore occur because of a variation in the Earth's gravitational field or the presence of a local

disturbances. The experimental method is to be sensitive to very small changes, there being no dominant 'one Earth's gravity' force to hide small variations. The convection currents will be induced in a fluid as this is considered to be the most suitable convective medium for experimental purposes. The convection currents will be monitored by observing the temperature variations in the fluid through their effect on the refraction of light passing through the fluid. The variation in refraction can be detected by the fringe patterns that will result due to optical interference in an interferometer containing the fluid cell. A laser source will be used to illuminate the interferometer, and Figure 2.1 shows the conceptual view of the resulting imaging system, with a laser illuminating the cell and a video camera capturing the image produced.

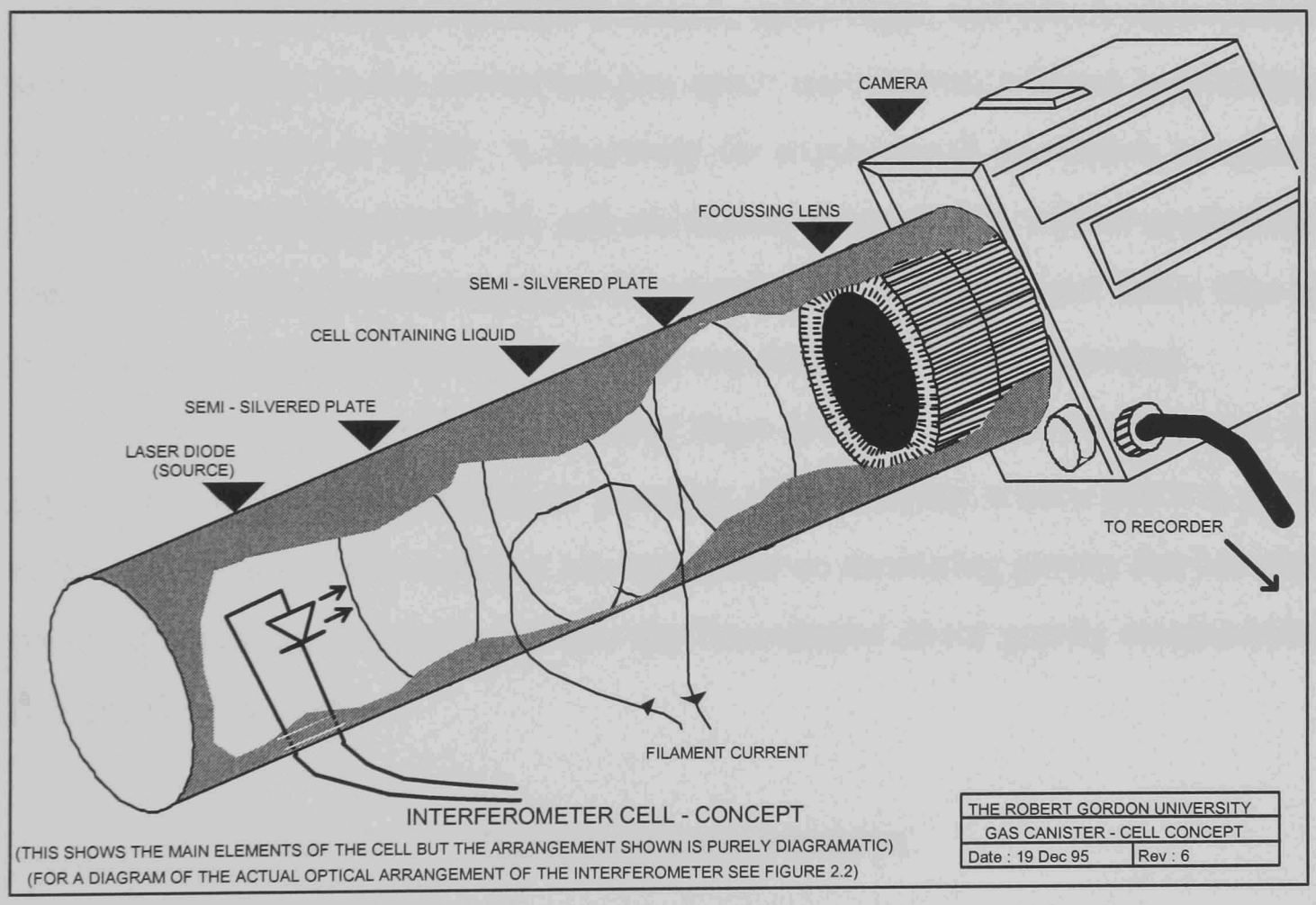


Figure 2.1

2.3.2 Experiment originality

The conclusions drawn from the various aspects of the review were that although both lasers and interferometers have been used before in space they have not been used before in the manner proposed in this experiment. However some experimental work similar to that in this experiment has been proposed. A micro precision interferometer test bed for space based interferometry is being developed by Hines [29]. This development is intended to result in an interferometer for use in space for astronomical observations. The present state of progress in this experiment is the development of a ground simulation system to develop the design of the instrument. The operation of a laser in space was carried out in G-449, Mukerheide [30] aboard the STS61-C mission. This demonstrated that a rugged 'high peak power' laser could survive the Shuttle launch, space flight, and return environment, and operate safely in the convection-less space environment without overheating. Development work on STEP, 'a laboratory for experimental gravitation in space', Jafry [4] is also being carried out, and one aspect of this system will be acceleration based gravity variation measurement. This system uses accelerometers rather than an interferometer imaging system and is some way from reaching flight testing.

The conclusion to be drawn from these investigations is therefore that no experiments of the type proposed are presently close to flying or have flown in space at this time. Further, the work that has been done on measuring gravity has not used optical systems and has yet to include any documented direct gravity measurement systems for space use.

2.3.3 Experiment Implementation

For the purpose of imaging the convection currents in the experiment, the cell of experimental fluid is placed in one arm of a Michelson interferometer. This type of interferometer is well suited to achieving in practice the aim of the conceptual imaging system shown in Figure 2.1, producing an interference fringe pattern corresponding to the temperature profile of the liquid in the experimental cell. A silicone oil [158], was selected as the experimental medium for the convection cell.

The selection was made on the basis of the oil having suitable thermal characteristics. Of particular relevance was a melting point below -40°C and a boiling point of above 140°C, both beyond the temperature range that it was anticipated the payload could encounter (see Chapter 6, Section 6.3.1) even under worst case conditions.

The basic Michelson interferometer arrangement is shown in Figure 2.2. This detects detailed differences in the paths of the light beams in its two 'arms' by generating an interference fringe pattern of light and dark fringes at its output. In this case, the distortion of one light path is caused by temperature variations in the fluid cell which affect the refractive index of the fluid, and therefore the amount by which it distorts a light beam that is passed through it. As the pattern of warm and cool areas in the fluid is dependant on conduction, radiation and convective heat transport by the fluid, zero gravity conditions should result in a series of circular fringes corresponding to radiation and conduction only. A gravity vector applied to the cell will cause convection leading to a distortion of the pattern as the warmer, and therefore less dense, fluid rises. For this reason, a cylindrical cell shape is used, with the heating element and light path running along the axis of the cylinder so that the observed two dimensional cross section of the cell has no corners. The result is that the cell appears identical in all directions in the plane across the cell imaged by the interferometer.

Interferometer : Optical Arrangement

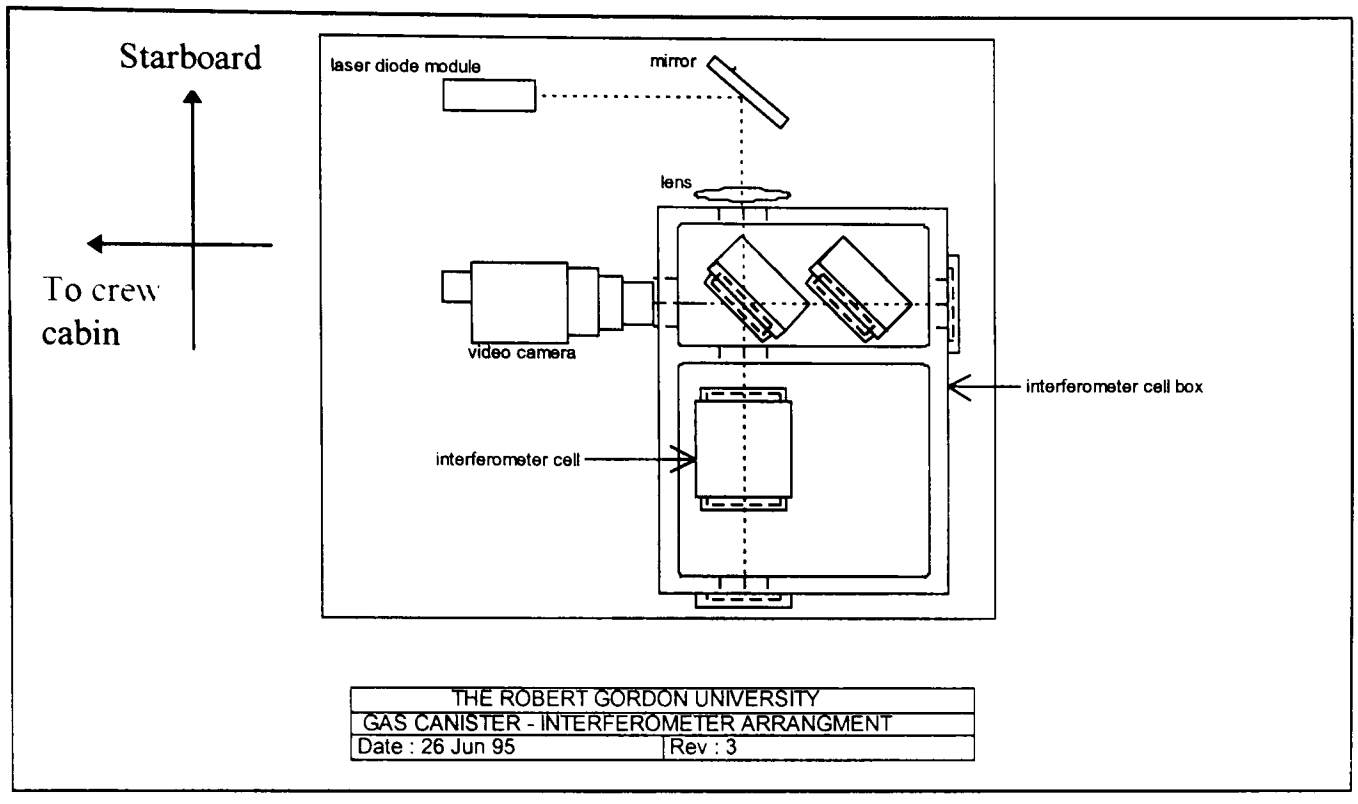


Figure 2.2

2.4 Modelling Fluid Flow

2.4.1 Modelling Proposal

As the principal aim of the image processing work is to extract a gravity vector from a fringe image, it has been proposed that the experiment be modelled mathematically to see if the fringe patterns can be generated from the gravity vectors produced by processing them. This would provide a bi-directional mathematical approach to the image processing system and allow the generation of many more simulated images for processing with the images processing system under a wide range of conditions.

2.4.2 Review Of Modelling Techniques

A review of work reporting upon fluid behaviour modelling, finite element analysis, and related discrete modelling techniques was undertaken. The aim was to assess the degree of research work on the behaviour of fluids and to obtain an

evaluation of the work that has already been undertaken in order to determine if any existing modelling technique could be applied to the task required.

The effects of gravity on fluids are discussed in the literature for numerous specific cases. These include the effect of gravity modulation on a body of fluid which is considered by Gresho [31], although the principal interest of the research is the effect the modulation has on the stability of the fluid layer, rather than the movement of convection currents within it. Computation of convective flow in a cavity is detailed, Biringen [32], for the case of a rectangular cavity with the temperature gradient from wall to wall. The combination of both buoyant and thermocapiliary flows is discussed by Bradley [33]. The modelling of natural convection and radiation interactions is addressed by Yang [34], but the method of solution using finite element analysis although formulated is not detailed. Thermal convection aboard a spacecraft due to vibration has been addressed, Kamotani [35]. Again the problem is formulated, but not detailed, though some conclusions are drawn from the author's work.

Heat transfer has also been investigated, as this forms the basis of the 'hot wire in a liquid' nature of the experiment. Radiative heat transfer using a nodal distribution modelling technique is discussed by Saltiel [36]. Use of the discrete exchange factor method for radiative transfer is covered by Naraghi [37], but only for gasses in rectangular enclosures. A different approach has been tried by Fiveland [38], using discrete-ordinates to solve for radiative heat transport, while, Ramaswamy [39], uses finite element analysis to investigate oscillatory flow and heat transfer.

The numerical analysis of natural convection is benchmarked, Davis [40], for the basic case of air in a square cross section cavity. Gravity modulation is introduced, Biringen [32], using the Navier-Stokes equations for fluid flow. The true complexity of natural convection in enclosures is discussed by Ostrach [41], giving a view of why the problem poses such a challenge from both a physical and mathematical point of view. Lauriat [42], meanwhile brings together radiation and convection for the case of a rectangular cavity with a transverse temperature gradient.

2.4.3 Conclusions On Fluid Flow

The review of modelling fluid flow, using finite element analysis, and related techniques was intended to ascertain the extent of any previous work undertaken on modelling the flow of liquids in low gravitational fields. The work discovered deals almost exclusively with the case of a cuboid of fluid between two plates at different temperatures, although it considers a variety of cases and both convection and radiation in these circumstances. This is related to what is required to model the experimental fluid cell, but it is a simpler system analytically than the required cylindrical arrangement. In the cylindrical arrangement, the heated surface is a cylinder in the centre of the fluid cell, which provides 360 degrees of convection to allow the possibility of measurement of fluid movement without the lack of symmetry in the fluid vessel impinging on the results.

However, no work on the situation encountered in the proposed microgravity experiment has been found, namely that of a cylindrical cell with a convectively flowing fluid in a semi-steady but varying, as opposed to oscillating, gravitational field. Computation of convective flow in a cavity is detailed, Biringen [32], and although this is a good starting point for the development of a model for the experimental cell, the case covered is for a rectangular cavity with the temperature gradient being from wall-to-wall rather than from a central heating element.

The conclusion to be drawn from the review is that a mathematical solution could be developed from the work found. The general consensus of the search on numerical modelling of the experiment however is that without a software package capable of doing the job without development, the modelling task alone would be a considerable undertaking. Further it would contribute little to the current research.

2.5 Proposed Imaging Scheme

2.5.1 Sources Of Images

A series of experiment were carried out in 1990 [43] to verify the proposed method of viewing convection currents using interferometry techniques. These used a heated resistor in air for the experiment cell. Although this arrangement is dissimilar to the flight experiment the overall shape of the resulting fringe patterns is basically the same as the patterns produced in orbit as it is the same physical phenomena which determines the results. These results also acted as a guide to the development of the flight instrument.

2.5.2 Sample Images

Varied results were obtained from the ground simulation images, some of them simulating zero gravity and others giving rise to more typical laboratory convection patterns. Other attempts to give 'low' gravity results resulted in distorted but still complete concentric fringe patterns and it is this sort of result which it is anticipated will be generated in space. Figure 2.3 shows simulated results for zero gravity while Figure 2.4 is more typical of convection on Earth and Figure 2.5 is a simulation of microgravity convection and is similar to the zero gravity image but offset due to a small gravity field acting. Further images were also obtained and were used to get an overall idea of possible results, and as samples for image processing trials.

Simulated Results For Zero Gravity [point heat source]

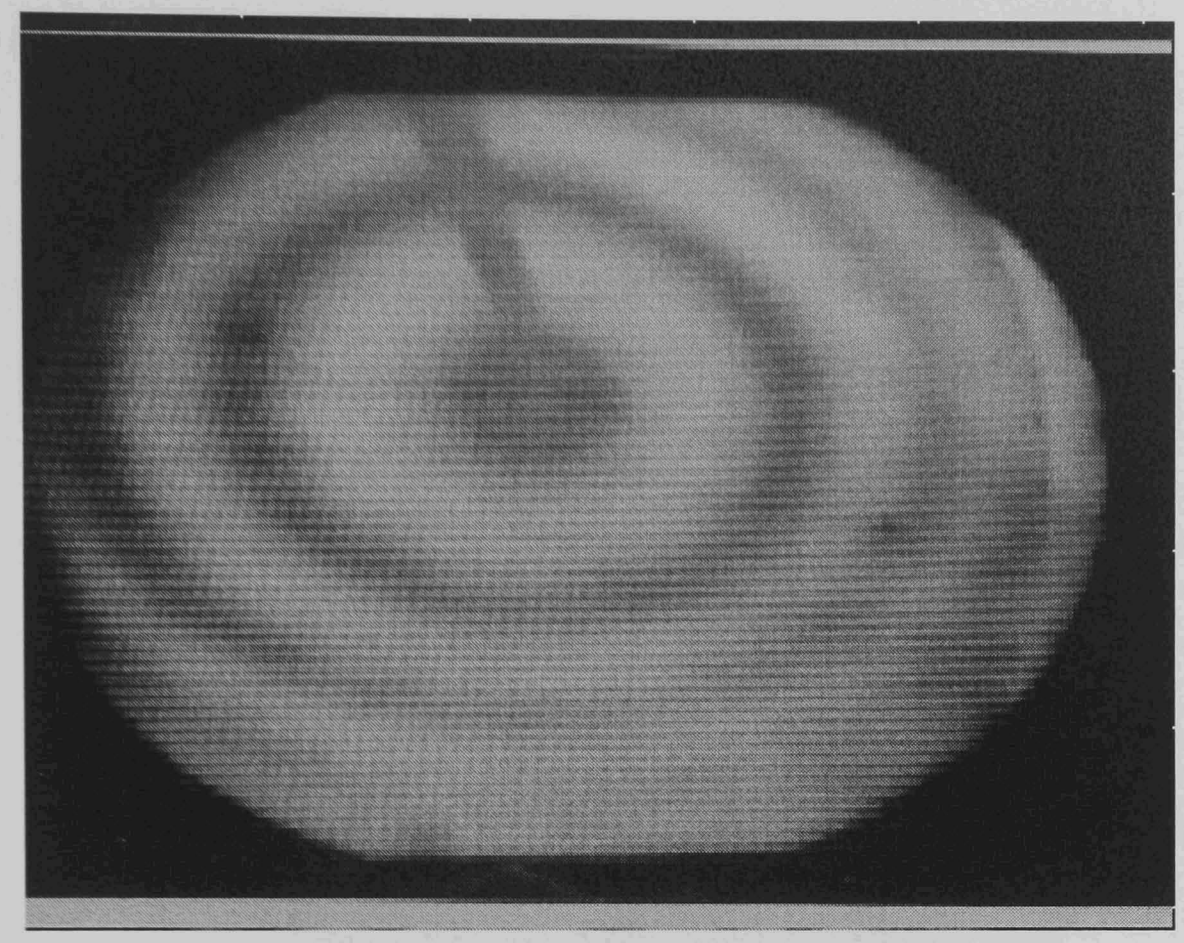


Figure 2.3

Typical Laboratory Convection Pattern [cylindrical heat source]

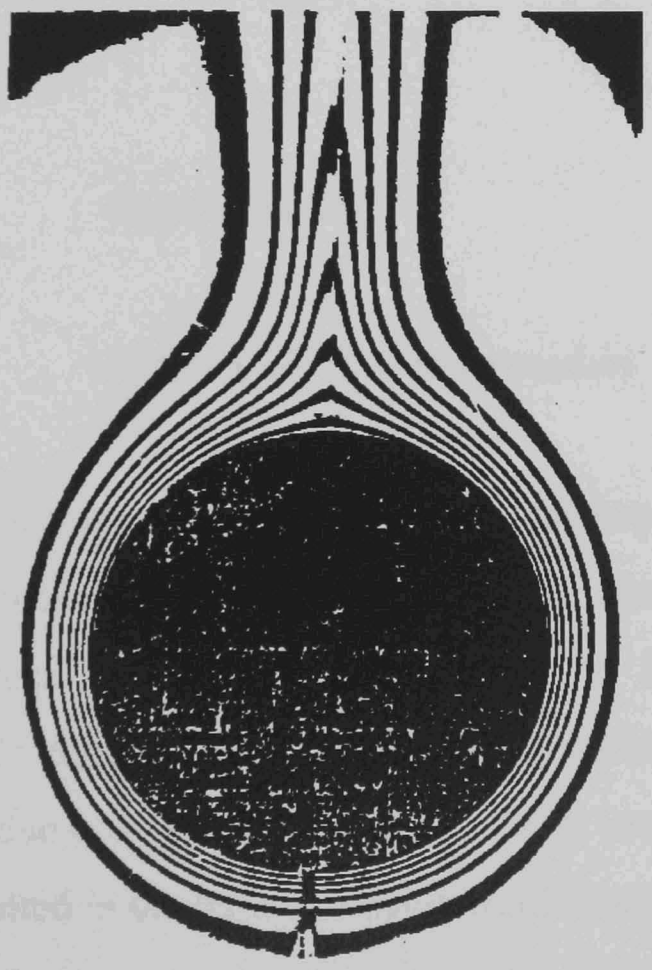


Figure 2.4

Simulated Microgravity Convection Image [point heat source]

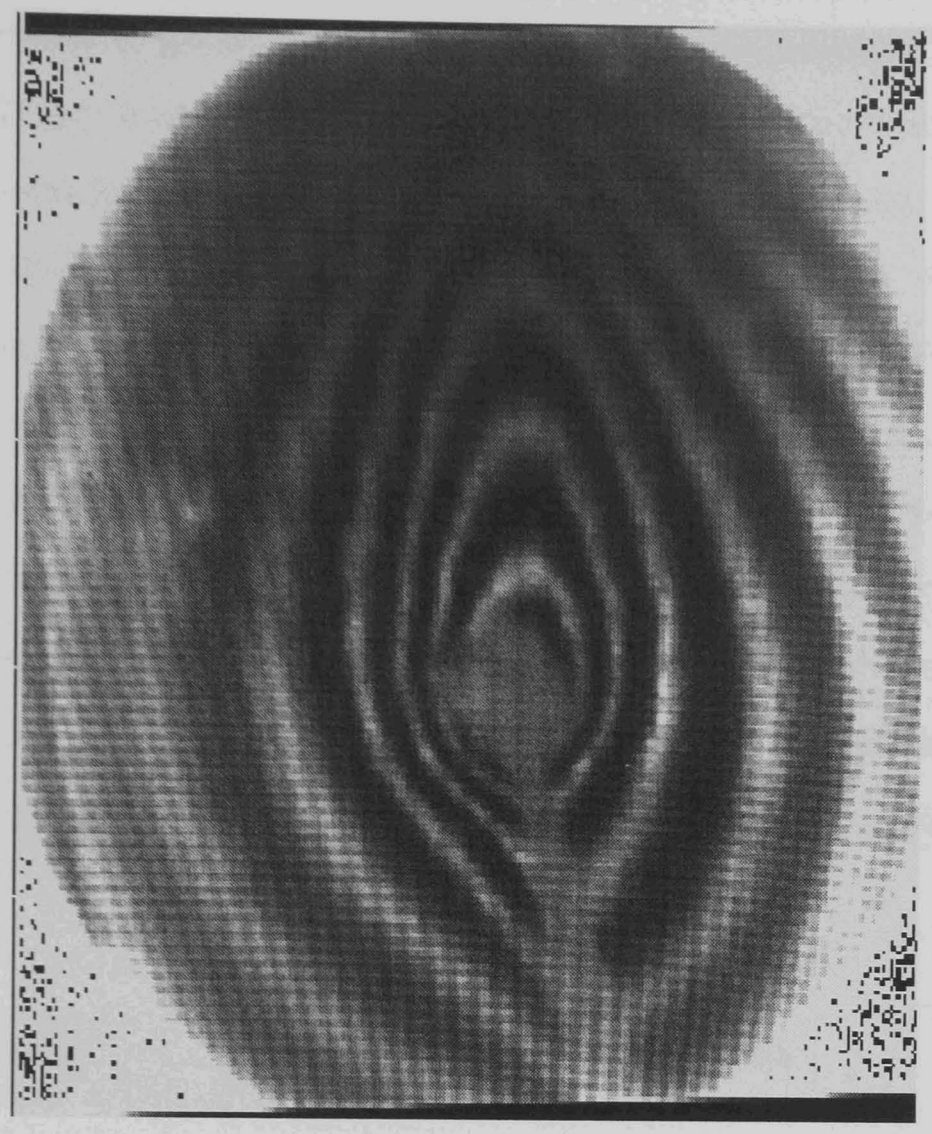


Figure 2.5

2.6 Gravity Vector

2.6.1 Reasons For Vector Extraction

In order to obtain a measure of gravity from the interferogram images it is necessary to extract a numerical representation of their distortion due to gravity-induced convection. From Figure 2.5 it is clear that the fringes as a whole are extended in an upwards direction and it follows from knowledge of convection phenomena that this is due to a gravitational acceleration towards the lower edge of the image. This has resulted in the lighter, warmer fluid heated by the cell heater, (in the centre) rising due to buoyancy as it is less dense than the surrounding liquid. The interferometer has produced fringes corresponding to lines of equal refractive index

which have resulted in this pattern due to the dependence of refractive index on temperature. In order to get an image processing system to measure this distortion, a feature of the fringes must be extracted from the images. As in some of the sample images, the patterns are very distorted and shape alone is insufficient. The one image feature that is obviously moved with the expanding fringe side is the centre of gravity of a shape matched to the peak of the fringe. If all the fringe centres of gravity are extracted, a series of points that should nominally lie on a line indicating the direction of movement of the fringes. A vector can then be generated by averaging these movements which corresponds to the buoyant rise of the fluid. This is directly related to the gravitational acceleration acting upon the cell that is producing the convective fluid flow and hence by extracting the vector indicating the movement of the centres of gravity of the fringes, the acting gravitational force may be determined.

2.6.2 Image processing Requirements

The basic requirement for processing the images produced by the interferometer system is to extract the vector corresponding to the movement of the centres of gravity of the fringes in the images from the stream of images on a video tape which recorded the interferometer output during flight. The problem can be split into a number of stages. Firstly the algorithm must identify the location of the fringes, then group the image points on each fringe together, so that they may be processed as individual fringes in order to extract their centres of gravity. Finally these centres of gravity are used to compute the translation vector for their centres of gravity and hence deduce the gravity vector acting upon the fluid cell.

The obvious visible feature to locate on the fringes is the fringe peak, which corresponds to a point of locally maximum image brightness in the fringe images. This requires a 'roof' edge detector which is not the most common image processing operator. Therefore the detection of the sides of the fringe, which are equivalent to edges in the image has also been considered. Once a map of fringe points, or detections has been achieved, a fringe or line following algorithm is required to trace along the fringe sections one at a time and label the points in some way so that

succeeding processing stages know which points belong to which fringes. This gives the individual fringes as separate computer identified and recognised features. The fringe centres of gravity can then be found from the [1,0] and [0,1] moments of the points comprising each fringe. In order to minimise errors though any missing or undetected fringe sections should be interpolated before applying the moment operator as a best guess is better than a gap that will lead to an offset fringe centre point detection. The final stage then is simply a matter of averaging the x and y co-ordinate movements of the different fringe centres with respect to one another.

Chapter 3

Image Processing, An Introduction

3.1 Literature Survey On Image Processing

3.1.1 The Image Processing Task

Image processing techniques are used to recover information from the interferogram images stored on videotape during the experimental flight. In all, the video tape recorder was intended to record an hour of images, generating approximately 180,000 image frames. The interpretation of the experimental results requires that only a fraction of the images be processed, but even this involves a considerable amount of processing. Therefore attempting to manually process the images would be time consuming. Manual identification of image features is also liable to inconsistencies in processing due to operator bias and systematic errors. An automatic method of processing the results from this experiment is therefore required so that the data can be analysed easily and consistently.

The first preparatory task before starting development of new work was to search the published literature for previous work on image processing and analysis techniques. Image processing as a subject covers a variety of approaches to altering images. The type of approach taken is generally determined by whether the results are intended to produce an improved image for human analysis, or are intended to provide a direct machine processed result. The flexibility to accept new images and freedom from human error correction in existing techniques varies considerably. Many tasks can be greatly simplified if the approach 'the operator will say no to false positive results' may be taken. The literature review has therefore been split into a number of subsections, with each of the techniques represented in the search discussed separately.

The literature review started with investigations into the Hough transform. The Hough Transform was taken as a starting point due to its high noise tolerance and the noisy nature of the then existing sample images. It expanded to include amongst others, algorithms for image analysis, processing, and enhancement operations. Extensive surveys of the Hough Transform are given by Illingworth [44], and Leavers [45]. Surveys of edge detection techniques, Larry [46], image analysis methods, Pavlidis [47], and simulated annealing, Hajek [48] assisted the search for material on other methods of image processing.

The sections that follow subdivide the image processing field into image enhancement and image restoration techniques (designed to 'improve' images for human viewing) and pattern matching and edge finding techniques, including the Hough Transform (designed to produce results for automatic analysis). The reasons why these techniques are, or are not, of assistance in processing the sample interferogram convection images are addressed. Finally, techniques intended specifically for phase information and fringe extraction, from interferogram type fringe images, are discussed.

3.1.2 Image Enhancement

Much reported research discusses algorithms and methods for 'enhancing' images. The usual aim of these methods is to 'improve' one of the image parameters, for example contrast, to assist the presentation of an image to a human observer. These techniques generally make no attempt to interpret the image. The analysis is left to a human observer who takes advantage of the image processing, to enhance contrast in the images or smooth noise. This subdues the effects of unwanted noise in the image and makes the desired data easier to see and therefore interpret. Early image enhancement techniques generally apply a uniform transformation to the whole image. These methods include for example, basic histogram equalisation. More recently these techniques have been improved upon, for example Colour Map Contrast Enhancement, Carpenter [49], and transform-based image enhancement techniques, Aghagolzadeh [50]. The most recent papers typically describe algorithms that use local information to vary the enhancement transform from one part of an

image to another Hung [51], depending on the local conditions. Alternatively they may use local information more directly to adjust the parameters of a global transform Leu [52], rather than just collecting the images overall statistics and using these as a guide to its enhancement without regard to the fact that some image areas may be adversely affected by such a treatment. Another technique is to use multi-scale image enhancement, Deng [53] where the image is split into images of different scales so that features present on different scales in an image may be processed differently. Finally with more processing intensive estimation methods, Sworder [54] enhancement techniques using image sequences, Ko [55] have been developed.

3.1.3 Image Restoration

In addition to image enhancement techniques used to improve poor images, there are techniques better described as image restoration. Although these are closely related image enhancement tends to imply improving poor contrast images, or coping with lighting conditions. Image restoration techniques are specifically designed to recover data from a noise-corrupted image which had acceptable contrast and similar attributes, but which has been corrupted by a noise source which can at least in theory be modelled and cancelled out. The techniques used for image restoration vary but filters, and inverse operators, Yamashita [56] which try to reverse the noise corruption mathematically are common. A comparison of some popular restoration filters has been carried out by Cannon [57]. Geometric filters, Crimmins [58] and adaptive restoration, Kuan [59] are also used, for the restoration of images degraded by speckle noise. The type of restoration that is most applicable depends on the type of noise (for example signal dependent noise, Henri [60] is tackled using homomorphic transformation techniques). Some work has been done using more complex approaches such as renormalisation, Gidas [61] and stochastic relaxation, Gibbs distributions, and Bayesian restoration, Geman [62]. This latter trades simplicity, speed and generality against effectiveness, though typically performance is the most desirable feature. However the nature of the corruption process affecting an image can be difficult, if not impossible, to calculate from the image, and approximations are generally required. Much of the work that is

undertaken on the different techniques described is aimed at determining which imprecise restoration technique is the best at approximating the requirements of a particular case. An associated problem is that a meaningful method of measuring the 'goodness' of the resulting final image is difficult to achieve. As Cannon [57] discusses, computers have to use a mathematical closeness measure, while humans compare subjective image similarity. Attempting to minimise both of these difference measures at once has a tendency to cause conflicts of interest which makes a computer-based technique for calculating how 'good' a restoration attempt is for a human observer a complex problem.

3.1.4 Pattern Matching

The term 'Pattern Matching Algorithms' is taken to describe a class of algorithms which carry out recognition of 'objects' in images by matching them with some stored bank of known template objects. This was one of the original lines of investigation in attempts to carry out automated scene analysis. Applications of pattern matching are varied. Komai [63] investigates the recognition of fracture surface morphologies. The generic case of pattern detection using matching techniques is addressed, Links [64], and the modelling of a group of shapes to give a more general matching algorithm is also presented, Hsiung [65]. Natural shape detection, Samal [66] has also been carried out using matching techniques, by matching the principal components of shapes. More specialised work investigates improving stages of the recognition process, for example the fringe distance measure, Brown [67], and an improved moment computation technique, Philips [68]. String matching, Bunke [69], and picture component labelling, Hambrusch [70] are also developed to provide enhanced solutions to some of the problems posed by pattern matching approaches.

The principal drawback with pattern matching is that it suffers from only being able to identify things if it can match them to templates it has seen before. This makes the technique impracticable where the possible images are unknown and very limited in complex environments as the number of different object types becomes impossibly large. Despite this, some applications are suited to pattern matching

techniques and edge replacement in occluded objects, Ullmann [71] and shape extraction using genetic algorithms Nagao [72] are both effective techniques, within certain limitations on the types of images they can process.

3.1.5 Edge Detection

Considerable work has been undertaken on finding edges in images, and many of the detection operators have as a result become established. The best results of much of the older work are now detailed in books covering the subject, for example Gonzalez [Bib 1] and Davies [Bib 2]. Methods of finding edges in an image reported more recently in the literature generally consider edge extraction from grey scale images. A wide variety of different operators have been proposed to accomplish this task. The theory of edge detection has itself received attention, Marr [73]. The algorithms proposed generally extract an 'edge map' from an image though some work has continued in an attempt carry out processing of the resulting 'edge image'. The object-background optimised edge detector, Kammoun [74], is an example of an operator intended to find shapes in grey level images. Histogram shifting, Crooks [75] uses a local image histogram to minimise double detection of edges, while the optimising threshold operator, Hebert [76], improves on thresholding techniques. Constrained regularised differentiation Taratorin [77], works to improve the noise rejection of a differential operator. Zhou [78], uses directional derivatives with a zero crossing operator and, Gruen [79] uses edge matching procedures. The robust morphological edge detector, Wang [80] and search techniques, Koundinya [81] approaches are shown to be more robust in the presence of noise. More unusual operators include the use of second order derivatives, Xu [82], segmentation using the Deriche filter, Meribout [83], and fuzzy if-then operators Tao [84] for edge finding. The specific detection of curved edges with parabolic equations, Chen [85] and curved edge multiresolutional frameworks, Calway [86], are unusual but shown to be possible. Contour tracing using the Kalman filter, Becker [87], and stochastic boundary detection, Geman [88] demonstrate the following up of the detection stage with attempts to group the detection points into features. The more general case of finding edges of any 'type'

receives less attention but the recursive optimal filter, Petrou [89] finds roof and valley type edges in addition to more typical step edges.

Some work concentrates on trying to do a 'passable' job of edge detection in a reasonable amount of time. As some operators use large masks, particularly older gaussian based detectors, workstations and supercomputers can be required to cope with the processing even if hours are spent processing a single image. The requirement for 'computer vision' meanwhile is 'real time' results. This typically requires the processing of multiple frames per second and generally uses simple algorithms running on specialist systems. Some reported research has taken the mathematical route, trying to develop and implement a numerically 'best' compromise between finding edges and avoiding false detections. Implementing these algorithms, or close approximations thereof, without using a supercomputer to run the code is also discussed.

A second group of reported results are those based exclusively around the Hough transform, and as a result, this transform has been surveyed, Leavers [45]. The Hough transform has proved to be successful at finding straight lines in binary images or grey scaled images which are easily converted to a binary representation with a simple threshold operator. The original Hough transform was exclusively a line-finding operator but various advances have expanded its scope to include many further situations. The basic Hough transform technique has been used in many different ways, from the basic detection of lines and curves in pictures, Duda [90], generation of local contour feature data, Vaz [91] to the tracking of targets with a search radar, Carlson [92,93,94], and the estimation of multimodal motion, Bober [95]. The detection of corrupting features in straight fringes has also been demonstrated, Pieralli [96]. The expansion of the Hough transform to a curve finding operator, Casasent [97] has resulted in a variety of approaches including a connectionist approach, Arrue [98] which can cope with high order curves. This is the first step to direct circle and ellipse detection. Circle detection using the basic Hough transform can be performed using a two-stage process but the results are typically poor when applied to complex images compared to the lines detection results. Many methods of overcoming this limitation have been proposed. The improvement of the original two stage method using multiprocessors, Kavianpour

[99] uses the processing power of a multiprocessing system. The detection of multiple circles using the Hough transform is improved with labelling techniques, Cao [100]. Kumar [101] utilises parallel processing approaches to tackle the problem. Better is the direct one-stage processing system, Kimme [102] in which circles rather than lines are plotted on the Hough plane, accumulating 'votes' for circle centre points, as opposed to line positions and angles. The next generalisation step is the detection of ellipses, Raymond [103] which requires further transform adaptations. To cope with the resulting computation requirements, proposals include searching for detections in two parameter spaces simultaneously, Tsuji [104] and splitting the problem into two stages, Yuen [105]. These stages are based on finding some parameters during a first pass and the remainder on pass two, as opposed to the more typical Hough technique of searching for all parameters of an object simultaneously. In addition to developing these extension techniques, the Hough transform has been successfully applied to the detection of circular geographic features, Cross [106] and to the detection of circular structures on satellite images, Taud [107]. Although the Hough transform was originally intended for the detection of analytic shapes, it can also be extended to the detection of arbitrary shapes, Ballard [108] with the Generalised Hough Transform. Overcoming the principal problem associated with the basic implementation of the Generalised Hough Transform, (namely speed) has been approached, Tsai [109] with a fast version of the transform. Over a thousand iterations can otherwise be required, Leavers [110]. Another approach has been the development of the Adaptive Hough Transform, Illingworth [111] which uses a coarse-to-fine scaling of the accumulator over multiple passes to concentrate effort where it is required.

The principle drawbacks of the Hough transform technique in general image processing are the pre-processing operations required to achieve binary 'line' images and extracting data from the complex responses of the transform to general shapes. The algorithm's robustness in the presence of noise and occluded objects makes it a popular choice where circumstances are suitable.

The similar Radon transform, Leavers [112] both in its original form and later extensions, for example using a geometric approach to processing delta function curves reported by Leavers [113], has similar limitations.

In addition to work on improving the actual edge and feature detection processes, considerable attention has been applied to the task of turning the detection points into the recognition of features in images. A wide variety of techniques have been used. The techniques used depend to some extent on the amount of prior information know about the contents of the image. For example, where it is known that only one image feature is present, Friedland [114] simple edge detectors may be used with the knowledge that all points detected will lie on the target feature. In this case a powerful second stage is applied, using simulated annealing, in the knowledge that all the detected points are valid. Another technique for ensuring that all the points grouped are part of the same feature, is edge following, which can be carried out using several methods, for example with the Kalman filter, Becker [87]. Curve fitting, Tsai [115] has been demonstrated where the target image contains isolated analytical curves. Curves can also be detected by following connected edge points, Rosin [116]. With the simultaneous consideration of multiple points on an edge feature, the effective 'cost' Tan [117] of edge positions can be used as a comparative measure using a defined comparison function to better localise the edge detections. Hierarchical relaxation, Chang [118] uses a multiresolutional approach to finding complete edges, and fills in gaps using a minimised best-fit-by-relaxation, operation. Another approach taken is constrained optimisation, Geman [119], which utilises area segmentation to identify boundary lines between areas. Recursive optimal filters have been extended to search for roof edges and thin lines, Petrou [89]. If the input consists of image sequences, contours may be tracked from image to image, Nagao [120]. This tracking of features provides multiple images of each feature which assists noise suppression.

Whole feature detection algorithms are typically multistage, with the initial edge detector providing a pre-processing stage for subsequent operations. Algorithms which extract whole objects by following extracted points and grouping these points into contours are typically much less computationally intensive than techniques utilising a relaxation approach. Relaxation techniques are typically more robust however, particularly in the presence of fragmented detections.

Early work on edge finding generally assumed that the edges were in the form of perfect steps with at most the possibility of the edges being smoothed or

corrupted by noise being considered. This approach has the limitation that only a small fraction of the edge types in real images are of this type, and explains why these algorithms miss some 'edges'. Some more recent research has considered the problem of finding edges in the true, general sense, including 'ramp' edges, 'roof' and 'valley' edges, Petrou [89] and composite edges, Ghosal [121]. Methods used to improve on traditional edge detection include local energy minimisation, Kisworo [122], multi-type detectors which run several edge search strategies and then try to combine them, Eom [123], and robust consensus based edge detection, Mintz [124].

Rather than relying on the data contained in a single image for input, some algorithms have been developed to work with image sequences. The tracking of boundaries in images sequences, Prasad [125] to provide their velocity vectors was achieved by comparing boundaries detected in each image. However this requires locating the boundaries in each image. The tracking of point velocity vectors, Mailloux [126] demonstrates this is not required to extract image flow fields. This approach however does not provide feature tracking as a result. Boundary tracking has also been demonstrated using co-ordinate transformation and co-occurance matrix thresholding, Zhang [127] and by searching for the zero crossings of the derivatives of gaussian smoothing, Duncan [128].

Multiple frame processing has several advantages. It enables the results to be averaged, assisting in the elimination of random noise and false feature detection. Many problems however have been encountered in identifying the same image feature in subsequent frames where motion has occurred between one frame and the next.

3.1.6 Phase Information And Fringe Extraction

The majority of image processing algorithms are either ineffective at locating the diffuse, curved and noisy 'roof' and 'valley' edges that comprise the sample interferogram images, or at localising their positions. For the limited case of straight fringes there are some solutions, for example using the Hough transform, Pieralli [96], or phase unwrapping, with line detection, Lin [129]. These approaches only provide solutions to a limited set of situations however.

When relatively noise-free fringes are encountered, operations such as phase unwrapping, Servin [130] and phase detection in high contrast fringes, Crescentini [131] have been used. The extension of phase unwrapping to contour reconstruction with high contrast fringes for Moiré contouring, Cline [132] and automatic Moiré contouring, from binary fringe patterns, Cline [133] demonstrate the advantages of the prior assumption of quality input data.

Work on fringe images has not been limited to attempting to detect fringes. A circular and concentric spots detector, Robinson [134], working on parallel fringes has been reported as has fringe-finding by a region-growing method, Osten [135]. This technique is used for fault detection using interferometry. A further general fault detector operating on fringe images without performing fringe detection, Glunder [136] and fringe maxima and minima counting systems for noise-free fringes, Tichenor [137] have also been demonstrated.

A number techniques that achieve fringe detection only when applied to noise-free fringe images are reported. With sinusoidal fringes a sinusoidal grating, Chang [138] has been demonstrated, while a holographic interferometer that detects fringes by thresholding, Suezou [139] and fringe extraction using the 1D FFT, Mastin [140] are demonstrated. Similarly the 2D Fourier transform has been used, Bone [141] for 'fringe object' extraction from perfect fringe patterns. The error analysis of a zero crossing fringe side detector, Gasvik [142] and difference measurements using fringe detection, Hofling [143] have also been reported as techniques for analysing the performance of operators on images of this type.

In an attempt to expand the range of fringe images that can be processed, some techniques for improved fringe detectors have been suggested. Techniques proposed include segmentation of fringes followed by region growing, Osten [144] and lines from interferograms by region growing, Eichhorn [145] which is good but assumes that the fringes being processed are of even width. Fringe minima detection in presence of speckle noise, Paler [146] has been demonstrated but operator assistance is required to minimise errors. The Gabor filter has been proposed for fringe detection, Bowier [147] but no results are given.

In some cases, whole fringe object extraction has been achieved for example with perfect fringes, Funnell [148], but operator error correction is required in all

cases. A thresholding technique using a floating image background fringe detector, Becker [149] has been demonstrated on imperfect fringes. Fringe analysis using a peak detector has also been reported, Yatagai [150] but the algorithm is poor in the presence of noise and requires assistance with error correction.

3.1.7 Summary of the Review of Image Processing

The interferometer cell produces interferograms containing a series of concentric fringes radiating from a central point where the heating element is located. In image processing terms, these fringes are formed from 'slope', 'roof' and 'valley' edges, in a grey level image. These edges are of types which many image processing edge detectors find difficult or impossible to either detect, or at least accurately localise the location of. From a detection point of view, the requirement is therefore to locate a point on each fringe and highlight it so that the 'position' of the fringe can be used for calculations to extract information from the image. The direct approach is to detect the 'roof' edges as these correspond to points of maximum image intensity and there is only one per fringe. The principal requirement is that multiple concentric fringes should be detected and accurately differentiated from each other. The simple detection of points is not sufficient however. The operator is required to identify points, 'a,b,c' are being on Fringe 1 and 'x,y,z' as being on Fringe 3 and so forth. This enables the fringes to be separated and processed in isolation so that operations that require the relationships to be known can be applied to the images despite each fringe only containing a fraction of the detected fringe maxima points. This enables a much wider range of data to be determined from the detected fringes.

Having defined the requirement for an image processing system capable of extracting data from the interferograms produced by an interferometer cell, the requirement is to assess the possibility performing the task with methods described in the literature. Image enhancement algorithms, although possibly of use for image manipulations to aid a human observer in extracting data manually from images, are of little assistance as they do not provide direct information extraction from an image. Pattern matching algorithms could potentially be useful in detecting features

that are nearly circular, but in practice, with feature orientation unknown and size unknown, the computational requirements would be excessive. Testing of the circle-detecting Hough transform also noted that circle finding algorithms are easily upset by concentric features unless they are defined so as to have negligible tolerance to imperfect matches. These problems therefore make pattern-matching methods unsuitable as a solution.

Edge finding techniques are readily sub-divided into a number of sections which are discussed in turn. The first, oldest, and simplest class of edge detectors are those applicable to binary images only. These are by default not suitable unless a pre-processing step to threshold the image is carried out. In practice, this is of little assistance as general interferogram results do not in typically have even contrast across the image field, hence simple thresholding fails. This has been demonstrated successfully where the input images are suitable, but is generally a limited technique. The Hough transform suffers as transforming the grey scale interferograms into a form it can process is difficult and even the work reported on the General Hough Transform does not overcome this. Likewise the Circular Hough Transform, though very effective, is fooled by concentric image features unless made so stringent that the flexibility and noise immunity that normally makes the Hough transform so effective is lost. This renders it useless for finding anything but perfect circles. Papers describing more recent work on grey scale edge detectors are more suited to the requirements, but most do not detect 'roof' edges and others require excessive computation. The general edge-detecting algorithms, relaxation methods and image sequence processing papers however are possibly of direct use, though no work describing their use on interferometer results has been reported. The results on medical images which are comparable to the simulated microgravity interferograms for noise or on complicated natural scenes presented in some of the papers does however indicate possibilities. Finally some work on interferogram processing has been reported but this has been limited. Phase unwrapping methods are potentially useful but do not extract any information on the location of fringes or another equivalent phase position measure. Some fringe detectors, particularly differential zero-crossing Gasvik [142], have been reported, but in each case there are problems. These algorithms either require fringes with considerable intensity variations or an

even background level. Although some techniques overcome this latter limitation, they have other limits on their applicability. This work gives a good basis from which to proceed, but the conclusion is that no single existing operator is capable of coping directly with the sample images that have been generated to date.

Despite the lack of operators capable of tackling the processing of the simulated microgravity convection images directly, a number of the sub-sections and approaches used in the papers identified have been referred to, in order to complete part of the task. The principal limitation with the majority of algorithms dealing with whole fringe extraction is that either the algorithms as presented are not combined with a sufficiently powerful pre-processing stage to cope with general interferogram images, or they rely on an operator to interactively delete false positive detections. The region-growing method particularly has been noted as a very powerful technique for tidying up raw detections. The use of both roof and valley detection largely overcomes the principal limitation of dilation based techniques, namely that they generally result in merging fringes before they fill gaps in fringes resulting from image noise.

3.2 Testing Existing Techniques

3.2.1 The Requirement to Perform Testing

An extensive literature search has revealed several relevant image processing methods that have already been developed. These were tested and evaluated to assess their performance with regard to the task required. Several edge detectors have been examined using both simulated results and idealised images containing much sharper edges than the actual test images. The results with idealised images containing clean edges are predictably good, but whenever grey scale interferogram results are used, the accuracy of the detectors becomes poor at best. The problem is that the small operators used cannot find the centre of an extended slope, and although large masks might be workable, the operators available are limited by the need to keep the computational load within reasonable bounds.

3.2.2 The Hough Transform

The possibility of using the Hough Transform, Illingworth [44] for extracting features from the interferogram images has been investigated, as this is a proven technique and is noted for being both robust and very tolerant of noise. A typical Hough transform detection result for a pair of straight lines is shown in Figure 3.1. Although in this case perfect lines were used to demonstrate the transform, peak finding routines can readily locate the peaks in images of this sort with considerable background noise present and gaps in the lines.

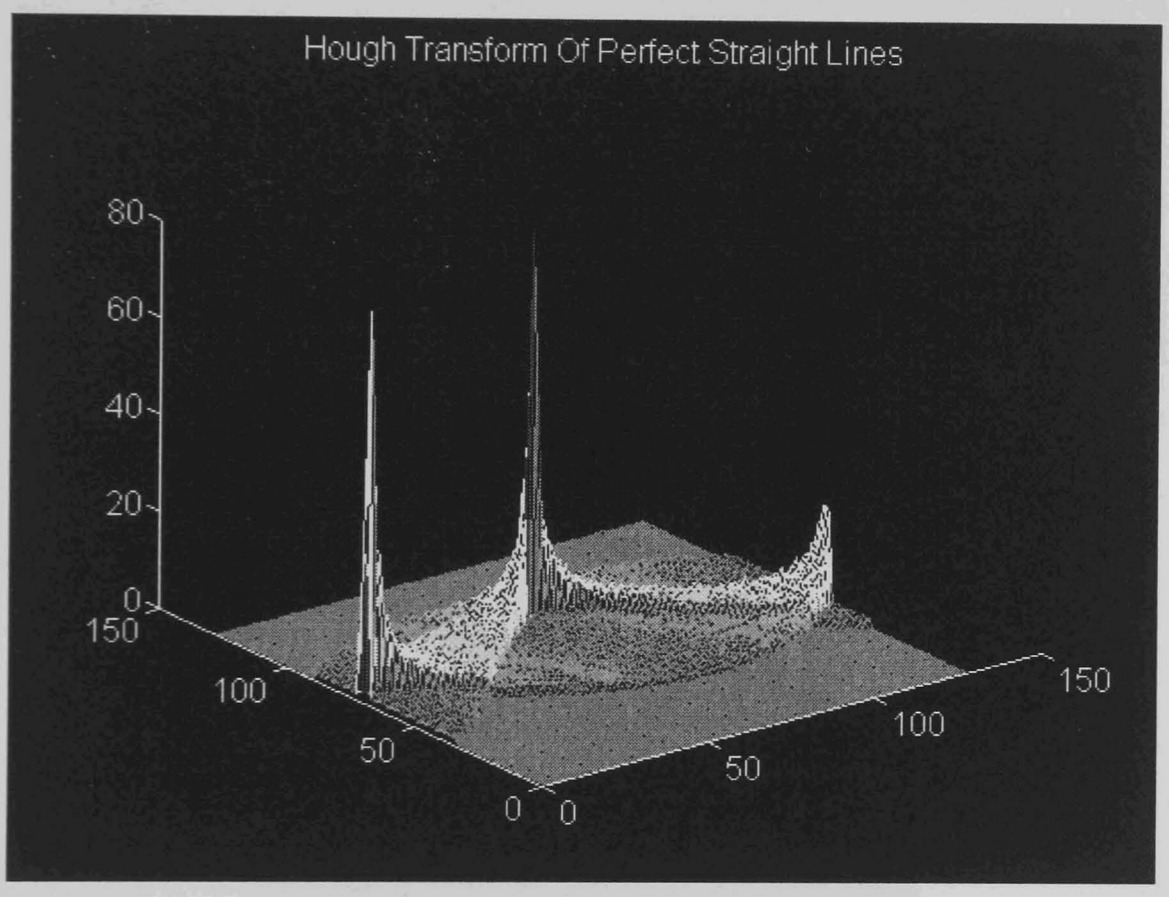


Figure 3.1

The Hough Transform was applied to thresholded images and to binary edge images resulting from the application of grey scale edge detectors to sample images. The Hough Transform approach suffered from problems in that thresholding the interferograms either gave very wide detected fringes or missed large sections of them. This is well demonstrated in Figure 3.2 which is the Hough transform plane corresponding to an interferogram image, Figure 3.3, which has been thresholded. The Hough plane bears a considerable resemblance to the image it corresponds to

with minimal fringe amplification having been achieved, resulting in no meaningful information extraction as a result of the transform.

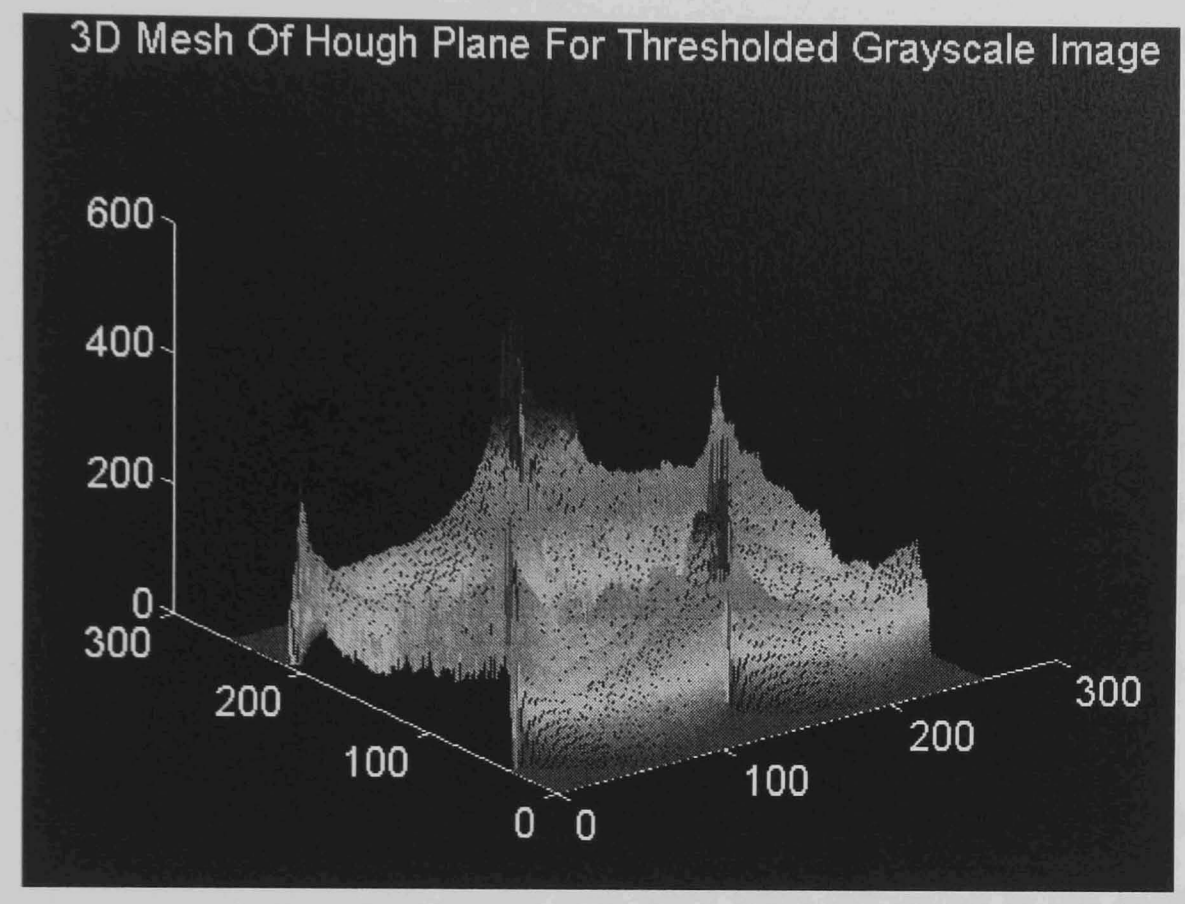


Figure 3.2

Thresholded Interferogram Image

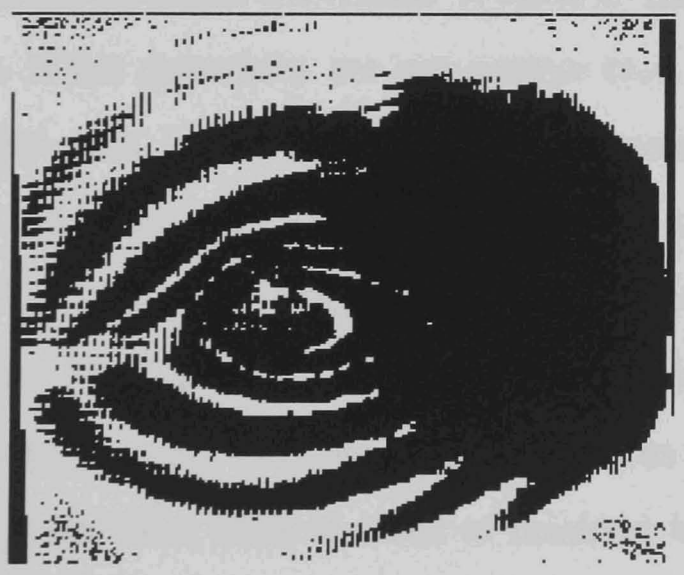


Figure 3.3

Testing demonstrated that generally no threshold detector is able to cope with the intensity variation in the sample images. This results in a poor quality input to the transform stage. If the threshold is set low enough to separate the fringes in the dark parts of the images, then all of the fringes are linked together in the bright parts. Conversely, if the threshold level is set high enough to avoid connecting fringes then

only parts of the fringe pattern were recovered. The second problem with the Hough Transform is that it that it either has to be used in multiple passes to detect intrinsically circular interferogram fringes, or it has to be applied in the form of the Circular Hough Transform, Kimme [102]. The problem with multiple Hough applications is that the straight-line detecting Hough Transform produces a mass of detections in the Hough plane as seen in Figure 3.2. This is due to the wings of one 'butterfly' detection pattern overlapping the next such as to make peak detection almost impossible. This is demonstrated by noticing the difference in signal to noise ration between Figure 3.1 where the isolated peaks are clean and sharp, and Figure 3.2 where despite using a fairly simple input image, Figure 3.3, the peaks that do exist are surrounded by overlapping short line detections in which any amount of data could have been lost. The Circular Hough Transform has other problems which are examined below.

Possible solutions to the thresholding problems due to the variation in the basic image intensity across the image area have been presented in the literature. In order to get clear peaks, the Hough transform requires that the basic feature to be detected is essentially narrow. This means that thresholds must be iteratively applied generally using an operator to assess the results in order to find the optimum setting for a particular image. If the detections are not narrow then all possible lines that pass through the 'wide' line in the image plane are 'voted for' in the Hough transform plane. This produces a broad detection feature and many smaller votes results from all the line fragments crossing the main feature. This is amply demonstrated in Figure 3.2. Due to the nature of the Hough technique, these problems are amplified when more than two or three features are present, forming a mass from which tall peaks from lines in areas of minimal interference are almost impossible to separate from peaks created by several overlapping detections of small line fragments. Attempts to use the basic Hough Transform have therefore been abandoned in the search for an algorithm that is more able to cope with multiple lines or features in the same image.

3.2.3 The Circular Hough Transform

Due to the failure of the attempts to use conventional Hough transform techniques being largely due to the need to find essentially circular type features with a line-detection algorithm an attempt was made to use a form called the Circular Hough Transform. The normal Hough transform has to detect circles by a two stage process, taking advantage of the fact that the many small tangents to a circle that a Hough transform detects, lie along two straight lines in Hough space. The circular version of the transform detects circles directly by using a circular mapping rather than a line mapping into the transform plane. This works by plotting circles about each point in the image plane into the transform plane. The circles plotted about each point on a circular feature overlap at its centre. The resulting peak indicates the circle detection as the peaks in the standard Hough plane indicate line detections. This is demonstrated in Figure 3.4.

Operation Of The Circular Hough Transform

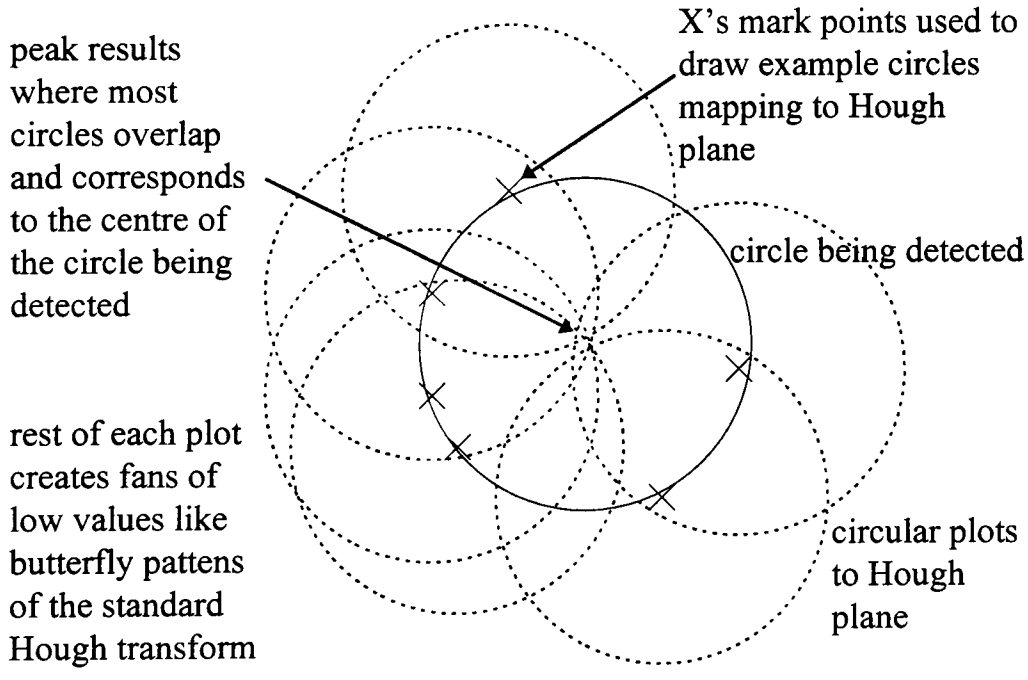


Figure 3.4

In testing, the Circular Hough Transform proved able to match to parts of ellipses as shown for perfect ellipses in Figure 3.5. However it proved unable to cope with concentric features. If the detection threshold, (the size of peak taken as

representing a circle detection) is lowered enough for a circle to 'match' an ellipse, the transform detects circles comprising sections of multiple fringes better than it does those made out of sections of a single fringe. Figure 3.6 demonstrates a typical result for 'thin' ellipses where many of the detections have picked up on part of an ellipse and another image feature, in this case mostly the innermost ellipse rather than just on a single ellipse.

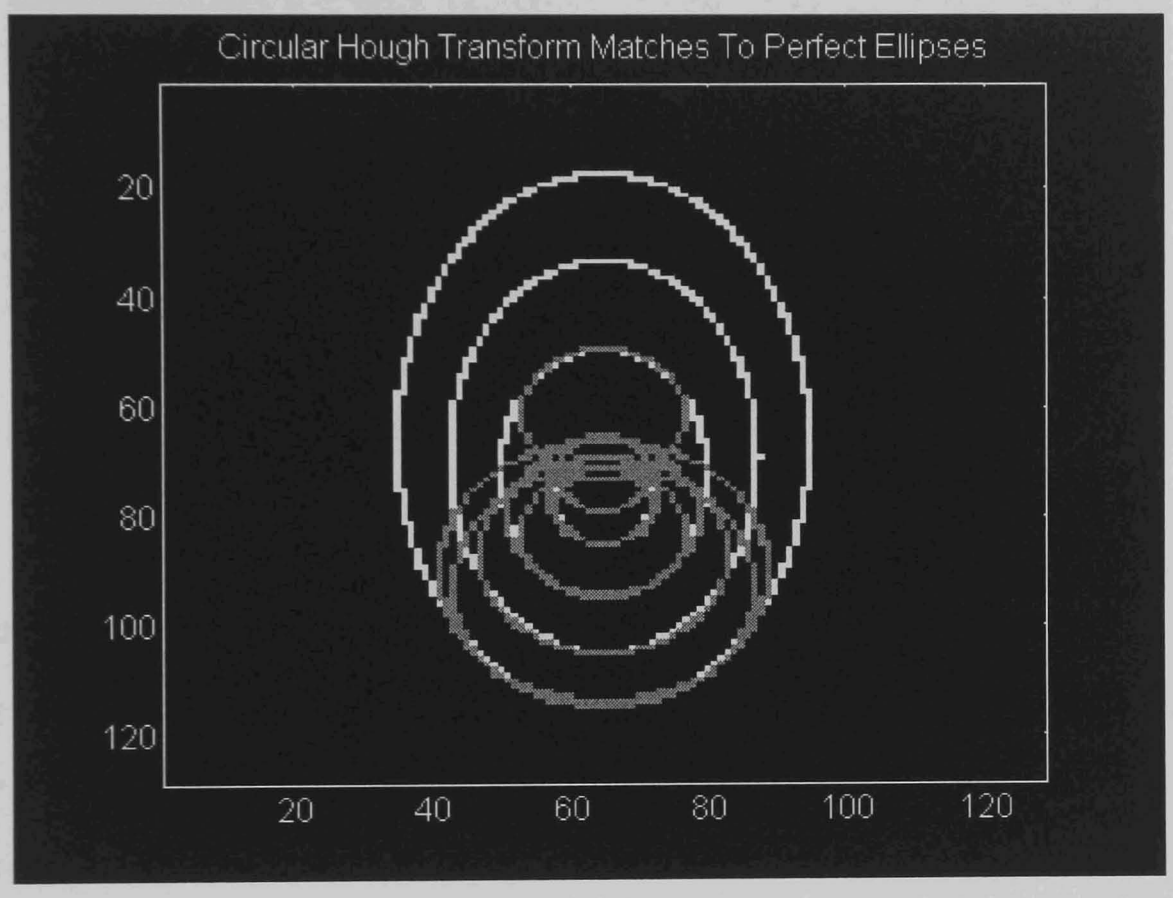


Figure 3.5

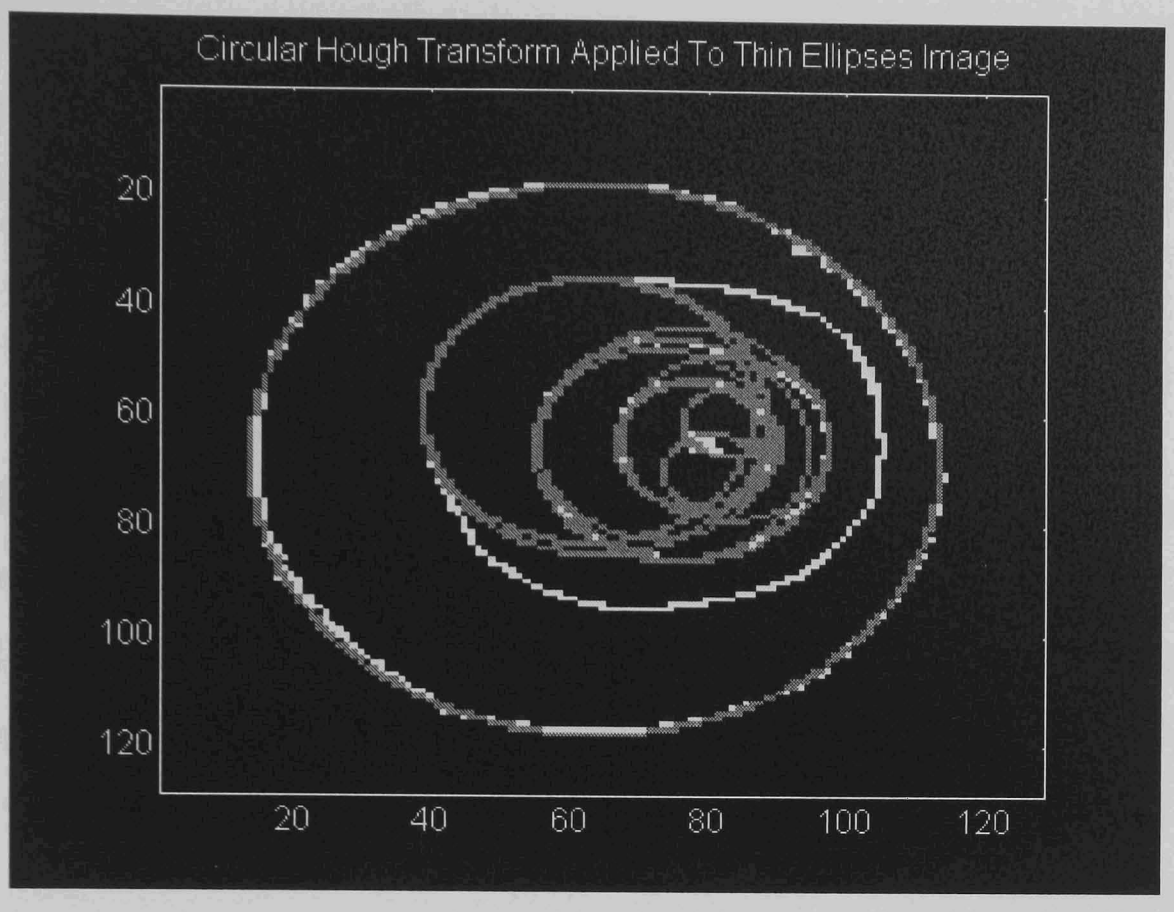


Figure 3.6

It is notable that when applied at multiple radii to a complete image in order to detect circles of unknown radius, the Circular Hough is a computationally intensive algorithm. The Circular Hough Transform algorithm may therefore be of use for extracting the centres of features given a suitable pre-processing to isolate individual features (or at least the concentric ones), but such an operator is not presently available. The problems above are further compounded by the requirement that only 'thin' (1-2 pixel wide) features are used, as the Hough Transform also fails when presented with 'wide' (4 pixel or more) features. These result in an indecipherable set of peaks in the transform plane from which little if anything can be determined. The number of times the operator matches to different parts of the same feature rises rapidly with the width of the features being detected.

In conclusion, despite using this circular transform approach the Hough transform has still proven to be of little use. The technique was discarded, as even if it could be made to work, the problem of producing a suitable binary input image of fringe features is still outstanding. This problem is not likely to be solved by Hough techniques as only the Generalised Hough Transform can cope with grey scale images and it functions simply as a detector of known shapes. Attempts to provide a

pattern-matching solution would still fail in the usual manner of such algorithms, as the number of templates to be tried will rapidly exceed the processing power available in all but the most trivial cases.

3.2.4 Edge Detection Masks

The standard Sobel and Robert's Cross operators were tested on the sample images to provide a guide as to the suitability of mask-based operators. Having obtained these results for comparison, the utility of other operators described in the literature could be better deduced from the results presented by their authors. This avoided the requirement of deducing implementations for each of these operators in order to obtain assessments of their suitability. Figure 3.7 is a typical Sobel result and Figure 3.8 is a typical result for the Robert's Cross operator.

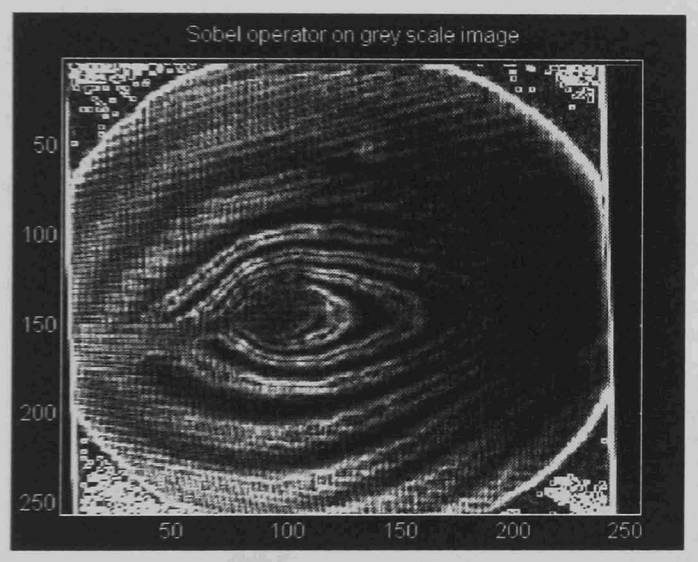


Figure 3.7

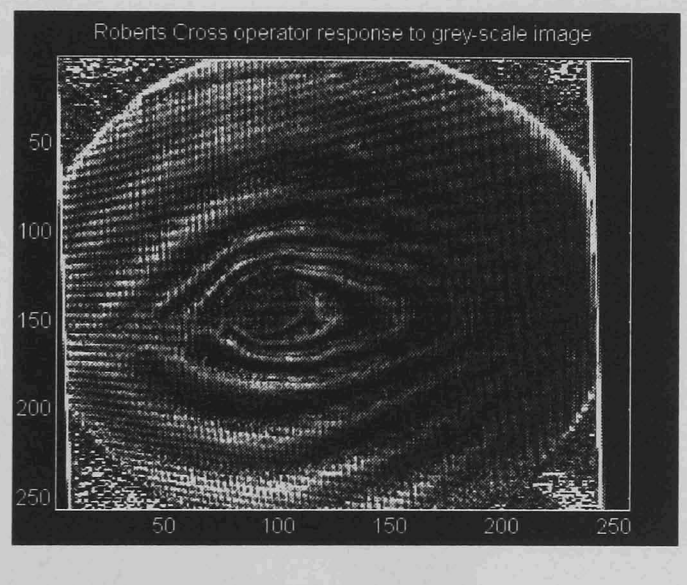


Figure 3.8

The results in both cases are poor. Although the clear fringe sections have been detected, the results are only distinct for the high contrast fringe sections. The noise in the image has also been amplified.

As these images represent 'edginess' as opposed to definitive edge detection, they demonstrate edge detection using small mask-based operators to be of minimal use in extracting fringes from the interferogram images. This type of operator is well covered in the earlier literature but as demonstrated, it is only really suitable for use on good edge images or in cases where supercomputers are available to process large

masks. A further problem posed by using large masks in the presence of multiple features, is the lack of resolution that results if the masks become larger than the edge features to be detected.

3.2.5 Other Techniques

Apart from the Hough transform and its derivatives, a thorough search of the literature has discovered a number of algorithms that could be suitable for handling images containing multiple concentric fringes. The result of this search however has been that virtually no current detectors can find the peaks of fringes. The few operators found are unable to localise the fringe sides enough for an edge detector of any type except a roof edge detector to be of use. An example of poor edge localisation by an edge-detecting operator is shown in Figure 3.9. In this example, multiple detections for each edge are avoided by searching for edges of one slope direction until an edge is located and then searching only for edges with the opposite slope. Despite this restriction which prevented many points on each edge from being detected, the detected fringe edge points are not good indications of the positions of the fringes.

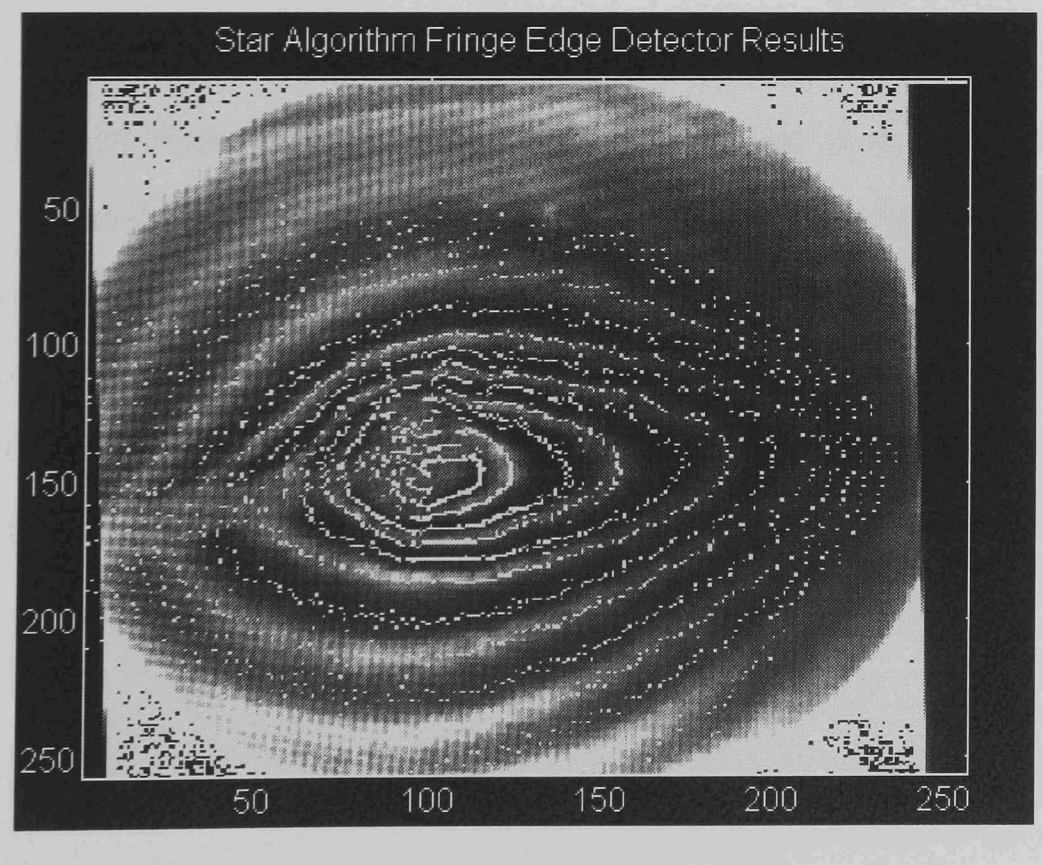
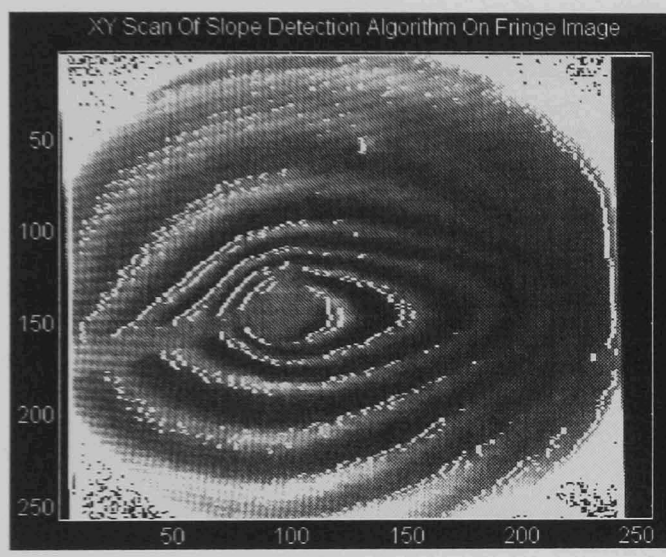


Figure 3.9

This image is better than could be expected of a 'rows-and-columns' scan of the image as it has used a radial image scanning technique, Friedland [114]. This scan of slices radiating from a central point approach is discussed in detail in Chapter 4 where work building on the effectiveness of this operator is discussed. The results using a scan along both the rows and columns of the image of Figure 3.9 is shown in Figure 3.10 for comparison. Although this looks acceptable, apart from lack of sensivity causing it to miss most of the fringes, increasing the sensitivity causes other problems. These include excessive noise detection and multiple detection of the same fringe as illustrated in Figure 3.11.



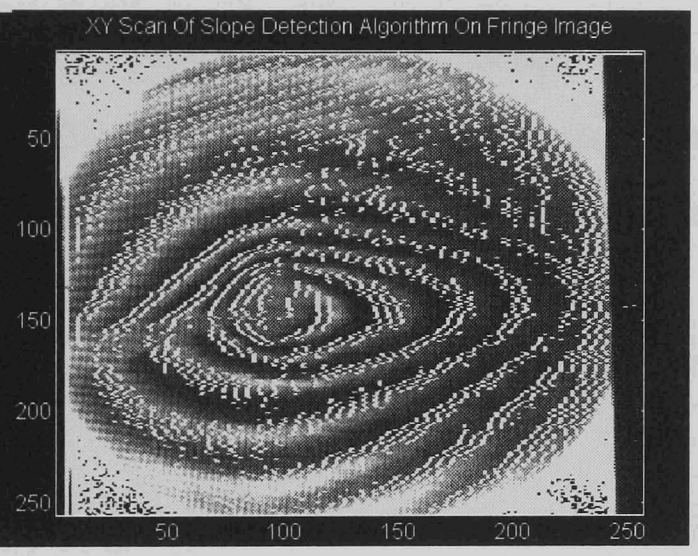


Figure 3.10 Figure 3.11

Another problem with edge detection of this type is that the results can be very image dependant. Figure 3.12 demonstrates the results of using a lower sensitivity detector on another image. This is a good demonstration of the lack of detector accuracy, as both rising and falling detections for one fringe appear on the same side of the fringes. The amount of false detections in other image areas is also rather high, and inevitably gets even worse as the sensitivity is increased as shown in Figure 3.13.

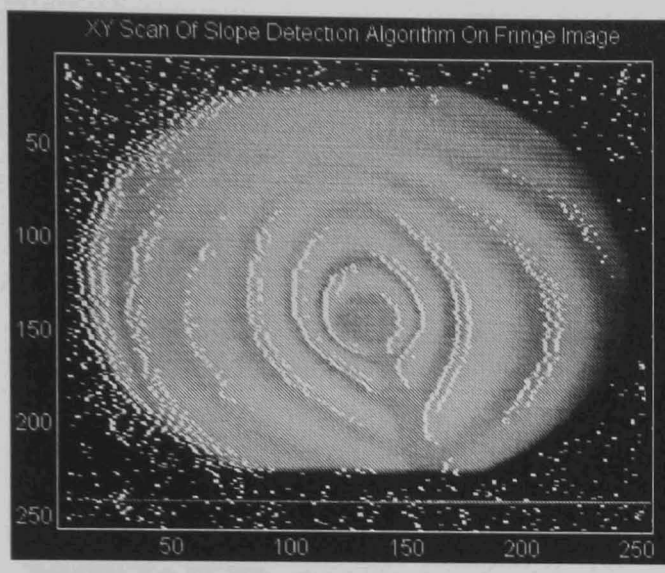


Figure 3.12

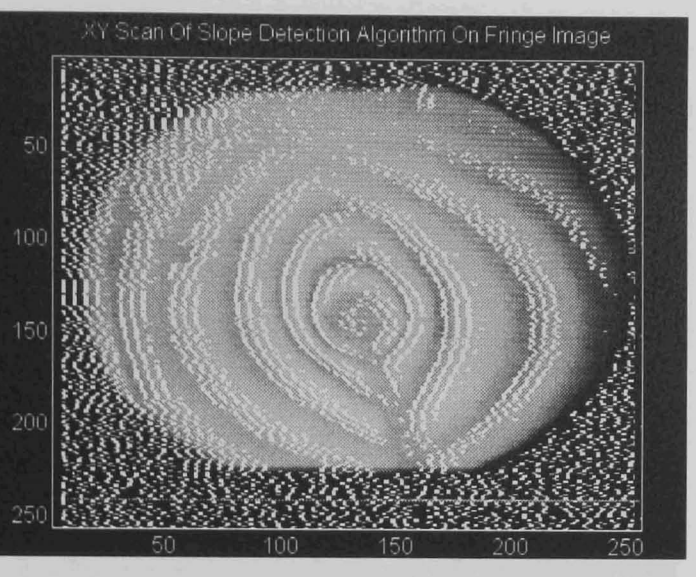


Figure 3.13

Although none of the existing algorithms identified to date has been accepted as capable of performing the task required, some algorithms cope with simpler forms of the problem or can assist an operator. Operator assistance is achieved by following fringes and then accepting interactive assistance or correction from an observer. A selection of these operators are worth closer investigation as to what they do achieve, and why this is not acceptable.

The application of image processing to the interactive analysis of interferometric fringes by Funnell [148] uses a fringe peak-following algorithm. It is demonstrated operating on images consisting of multiple concentric fringes with the results being complete separate fringes. Appearances can be deceptive however, for although this is a good operator-guided algorithm, it has considerable manual assistance requirements. The starting requirement that the operator specifies the fringe pattern centre point and a line crossing all the fringes in the image is permissible as these are fixed features of the equipment arrangement used to generate the images. They are therefore invariant for a set of images from a single experimental arrangement. The operator used to locate the peaks of the fringes in one dimension along this line is however limited. It accepts a point as a peak simply if it is of higher intensity than either of its neighbours. This is quite acceptable with the smooth narrow fringes shown in the example results, but not if the operator is applied to the sample convection images. These contain too much random noise as a simple examination of a cross-section demonstrates, see Figure 3.14. The resulting detection problems caused by the simple peak detector are illustrated in Figure 3.15.

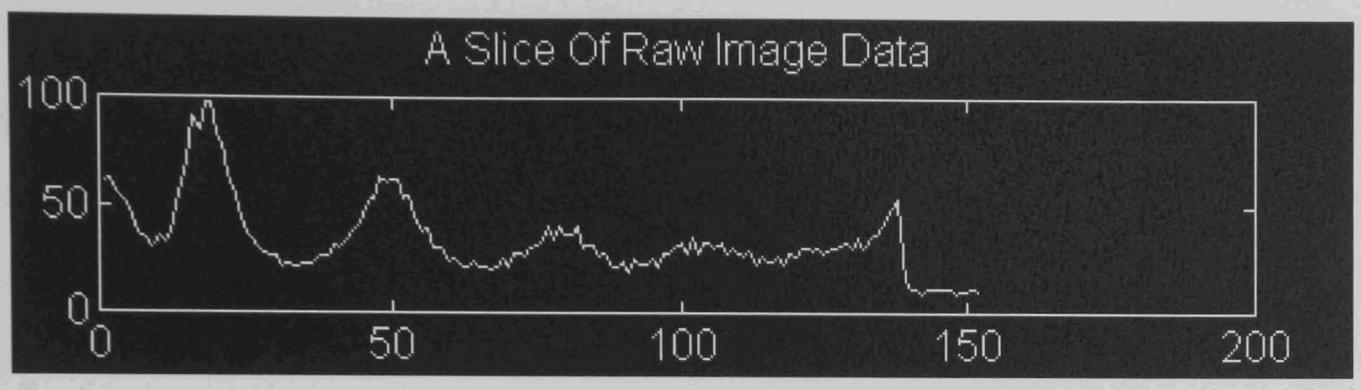


Figure 3.14

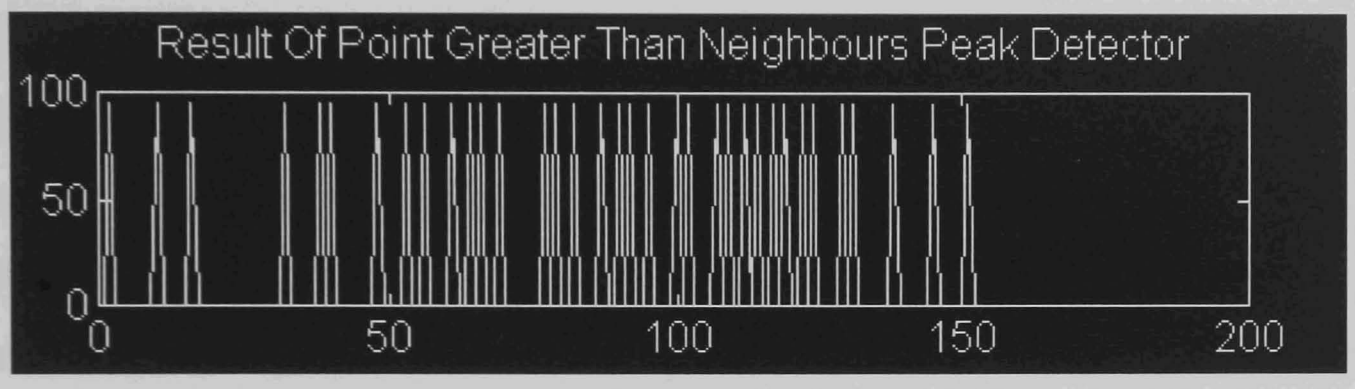


Figure 3.15

Simply smoothing the image slices, as in Figure 3.16, does not remove the problem, as flat peaks which result in two adjacent similar values are no longer detected, as shown in Figure 3.17. Many types of features which are clearly not fringe peaks are also detected in error.

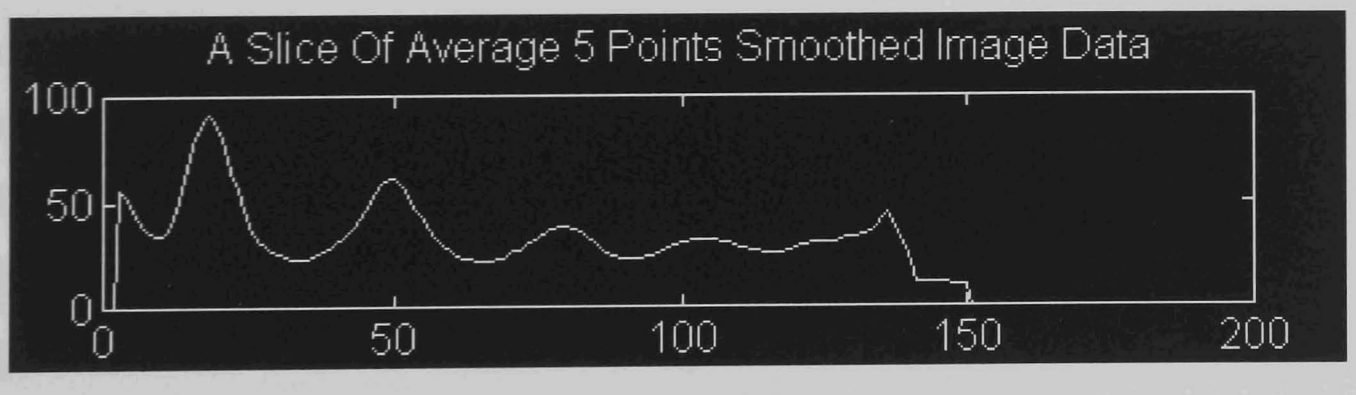


Figure 3.16

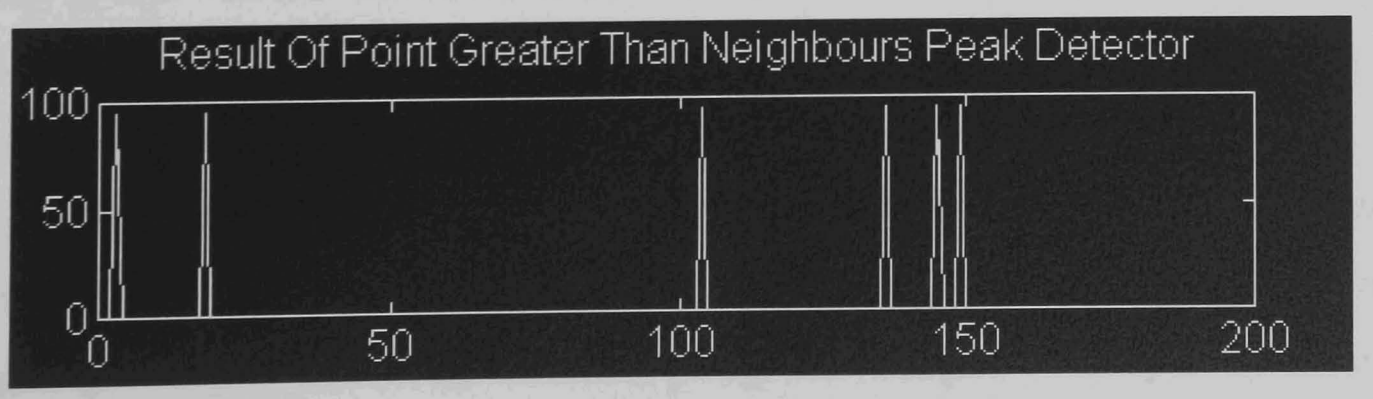


Figure 3.17

From these images, fairly simple solutions such as searching for both plateau edges is a possibility. In general however the result will be double peaks that require many smoothing passes to merge them into a single peak, and large areas of small variation that actually contain fringes occur in images, and these will still not be detected reliably.

The second major problem is that the next stage of the algorithm, Funnell [148], is a fringe-following operator that follows the maximum points. Although it takes advantage of the known circular style of the images to control stepping, the algorithm still can easily jump from one fringe to the next in the presence of noise. In this case, interactive manual intervention is used for correction. A further problem with this method is that the fringe-following stage would be confused by the presence of wide areas of similar grey level. These are not present in the original image but are common in the simulated interferogram images that require processing. The right hand side of Figure 3.10 illustrates this problem.

Robinson [134] examines automatic fringe analysis for three applications. Full interferogram fringes are considered but under specific conditions. The first application, strain measurement, produces straight parallel fringes which enable assumptions to be made. Particularly, a simple mathematical model of the expected fringe shape is available, and the requirement is to search for fringes in the form of lines only. The second case is three-dimensional position measurement, but again the fringes although circular are known in advance to be perfect circles. This permits assumptions to be made and mathematical advantage taken of a known result. Finally Robinson [134] examines fault finding in non-destructive testing. Although the target artefacts are sets of concentric fringes, they are again perfect circles, and for this application all that is required is the position of the centre of each concentric fringe group. Another limitation is that this application uses binary thresholding of the images which in this case works due to the high contrast of the fringes making subsequent processing much simpler than with grey scale images.

Further work using thresholding, Osten [135] and carrying out fringe counting, Tichenor [137] as opposed to fringe location has also been found. The region growing operation in, Eichhorn [145] which segments images by dilating fringe detections is good and has been useful in development of further work but it is

linked with a very simple fringe detector that cannot cope with images unsuitable for thresholding type detectors.

3.2.6 Lessons Learnt During Testing

An important result of the work discussed was that in realistic interferogram images, the detection of fringe sides as rising or falling edges has proven difficult, if not impossible, while the possibility of detecting the fringe 'roofs' and 'valleys' is promising. Similarly the direct extraction of analytic features by either pattern matching or probability voting (Hough Techniques) cannot cope with the noise levels encountered. The fringe peaks however are much better defined as maxima in an image rather than simply vague points on a slope, and attempts to detect this type of edge feature appear to offer a better solution, despite the limited amount of work performed on this task previously. Two useful advantages are that this approach provides only one detection for a fringe, (assuming valleys are separated from roofs) and that it produces a detection where a human can see a fringe, which makes checking the subjective quality of the results easier. In fact, given the images with gently sloping fringes, it is difficult to locate the half way point on a fringe side, although it can be defined mathematically by the point of inflection of the image intensity curve.

In summary, a review of the existing literature on image processing techniques has been presented. A selection of techniques have been analysed and tested for suitability for processing interferogram fringe images. The conclusion as a result is that the existing techniques are not suitable for this task. Finally some techniques that can be developed to assist in the development of a system to perform fringe detection have been discussed.

Chapter 4

Image Processing, Proposed Algorithm Development

4.1 Proposed Solution

4.1.1 Previous Methods

Having established that no pre-existing techniques were suitable for the complete analysis of the interferometer fringe data anticipated from the experiment, the image processing effort was directed from testing existing algorithms to developing an algorithm capable of the task. This algorithm development enhances the capabilities of previously published algorithms, as few researchers have reported algorithms capable of detecting 'roof' type edges. Some examples of previous systems include, composite edge detection, Ghosal [121], a local energy approach to sub pixel edge location, Kisworo [122] and robust edge detection by consensus, Mintz [124]. Similarly, the previous work on detecting concentric image features similar to those anticipated, as discussed in the last section, does not provide an algorithm capable of extracting complete fringes from noisy images unassisted. Fringes may be extracted with either operator assistance, or considerable prior knowledge of the format of the fringe pattern to be expected. Previous work analysing concentric features is unusual, though a number of researchers have examined the occlusion of features by other features of similar size, (for example edge replacement of occluded objects, Ullmann [71]). Attempts have been made, but the identification of fringe minima in speckle patterns, Paler [146] for example, is not capable of processing the images expected. Some work has produced algorithms which operate on concentric fringe patterns given good imaging conditions. This is illustrated by the image processing of interferometric fringes, Funnell [148]. Work on images of this type includes the knowledge based evaluation of fault detection, Osten [135], fast line derivation from interferograms, Eichhorn [145], automatic

moire contouring, Cline [133], interferogram data reduction, Osten [144] and holographic 3D deformation measurement, Nakadate [139]. The automatic computer analysis of fringes, Robinson [134] and automatic fringe analysis, Yatagai [150] are examples of the limits of existing algorithms.

4.1.2 The Circular Approach

The initial approach selected to designing a new algorithm, in an attempt to overcome the difficulties of the existing operators, was to improve upon the radial gradient algorithm, Friedland [114]. Although this algorithm requires an image centre point to be given, this data need only be entered once for a particular set of experimental apparatus. In the case of the microgravity experiment, the centre point is known in advance, as the location of the heating element wire, (Some of the existing processing algorithms reported could most likely determine its location automatically if required. For example the automatic computer analysis of fringes, Robinson [134] and knowledge based evaluation of fault detection, Osten [135]). The principal intrinsic advantage of the radial gradient algorithm is that by taking slices of an image from the centre of 'circular' fringes, the slices naturally intersect the fringes close to right angles. This occurs in all parts of the image, (rather that at just a few locations) thereby reducing the two dimensional image to a series of one dimensional slices (containing fringe cross-sections) without the problem that some of the image slices are parallel to, rather than intersecting, the fringes in the image. The latter is the inevitable result if a rows and columns scanning pattern is adopted. This is the principal reason for the variability of the performance of such approaches depending on the orientation of image features.

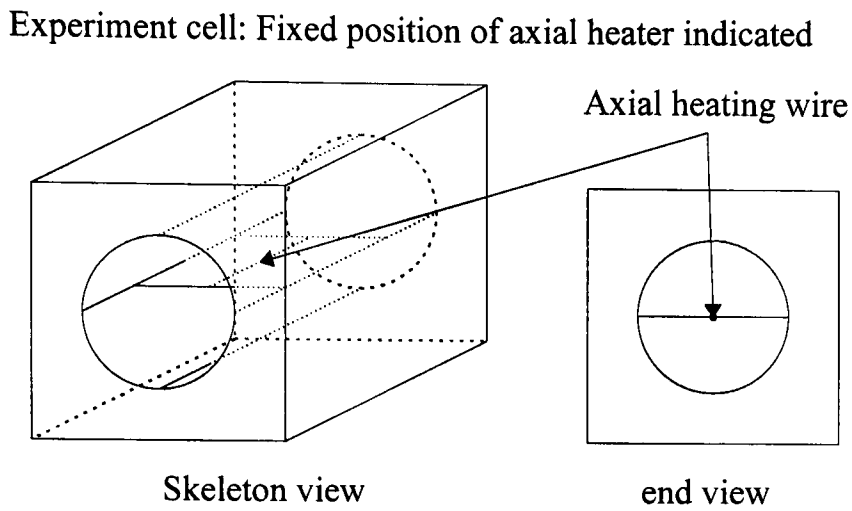


Figure 4.1

As the nature of the fringe patterns is known in the NASA experiment some assumptions may be made. The images are of a circular vessel with a stationary heating element in the centre, as shown in Figure 4.1. The starting point of the operator can therefore be positioned on the heating element and is valid for all the results. The radial operation is also known to be a valid approach. A 'rows and columns' scan of the image could be substituted for other experimental arrangements making developments based on this concept more widely applicable than the algorithm developed utilising this assumption. Testing has been performed to compare the results of a radial scan and a rows-and-columns scan approach to processing the simulated microgravity images with the conclusion that the radial approach is better (see Chapter 3, Section 3.2.5).

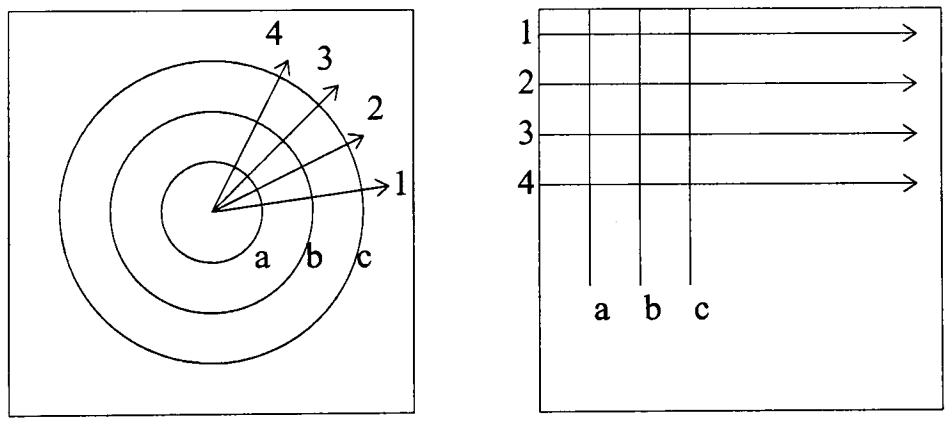
4.2 Fringe Detection Development

4.2.1 The Star Algorithm

The 'star algorithm' concept originated from a study of the automatic detection of ventricular cavity boundaries from sequential ultrasound images using simulated annealing, Friedland [114]. There are two basic ideas behind this method of scanning images. Firstly it provides a radial scan which, as explained above, is helpful in realising a good fringe cross section in the direction of the image

processing operator. Taking the centre of the image features as a starting point also results in the slices, of the image being scanned, intersecting the complete fringes in the same order and only once each. This ordering of the slice fringe intersections means that theoretically, the first fringe point detected in each slice of a radial pattern should be on the first fringe. In practice, this detection ordering need not be true due to noise. Testing has demonstrated this assumption to be of no assistance as in practice it causes more problems than it solves. It can be assumed though that fringe detections will be in order. As fringes may be missed at random in a particular slice, this information would be very difficult to take direct advantage of however. The second advantage of taking slices at multiple angles about the centre of the image, is the possibility of a reduction in the processing requirements. There is no requirement that the number of angles scanned be sufficient for all pixels in the original image to be covered by a radial slice. This potentially can reduce the data set to be processed, and thereby save processing time assuming the resulting accuracy of fringe location is sufficient.

Conversion of Fringes Array to Array of Slices For Fringes a,b,c



Radial slices, numbers 1,2,3,4 Resulting array of slices

Figure 4.2

The result of the star algorithm conversion is a new definition of the image, shown in Figure 4.2. The resulting array is a two dimensional matrix where the distance of a point from the centre is the x-axis and its angle number around the starting point is the y-axis. The initial image is in rows and columns format, and the angular spacing of the slices taken about the centre point is equal to the number of

total slices divided by 360 in degrees. Points in the original image are mapped to the points in the array of slices that overlap them.

Having created the array of slices, a one-dimensional operator is now applied to extract the desired image features from each slice in turn. The initial feature detectors proposed were of the form of step edge-detecting masks, as shown in Figure 4.3. This operator is designed to give a high numerical output when the step is centred on the location of a fringe. Details of the operator can be found in Friedland [114]. The results with this operator were excellent on idealised images of concentric circular ring-like features, (Figure 4.4), which have either sharp or only slightly stepped edges. More realistic sample images, for example those shown in Figure 4.5 however demonstrate the limitations of this edge detector. The improved detection shown in Figure 4.6 resulted from starting at the centre and switching the detector from rising to falling edge detection and vice-versa each time an edge was detected. This approach minimises multiple detections of the same edge, although the result is still worse than for the ideal images. Multiple fringe processing is handled in these images by searching for fringe #2 from the position at which fringe #1 was located and so forth. Problems are also encountered in setting the edge detection threshold for the operator output. Even with interactive human assistance, the results with the grey-scale sample interferograms are poor. The edge detectors fail to distinguish the rising grey level of a fringe edge from other image features of a similar nature.

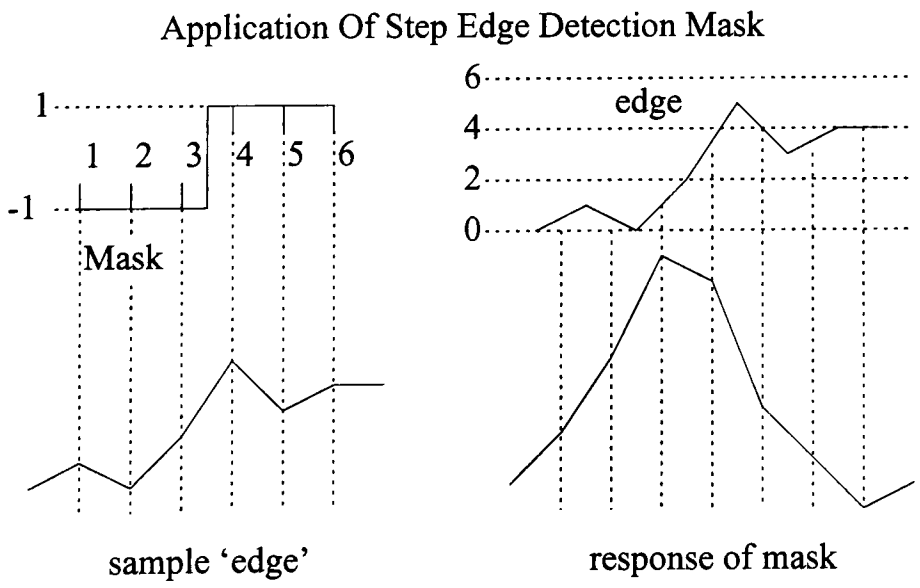


Figure 4.3

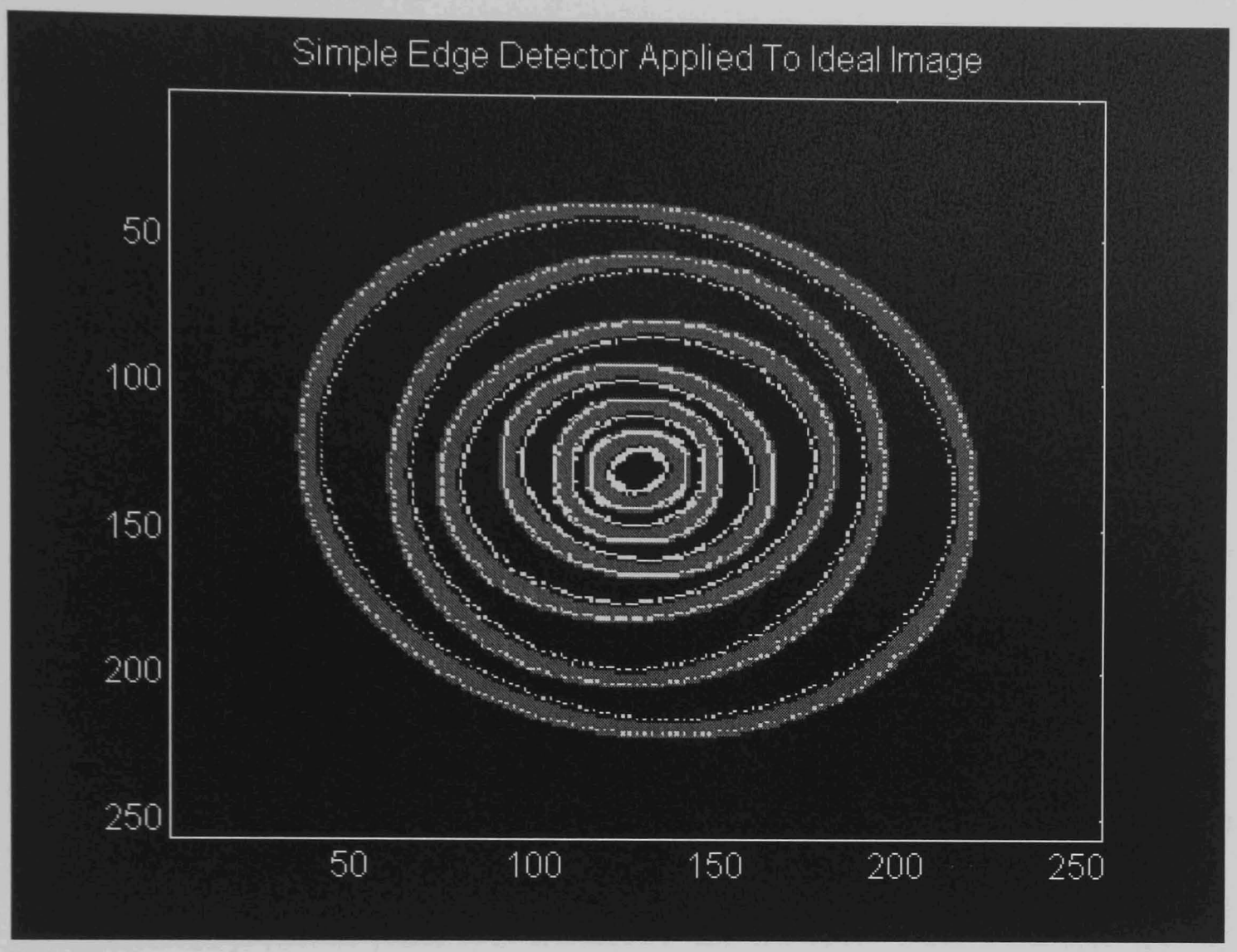


Figure 4.4

Simple Edge Detector Applied to Interferogram Image

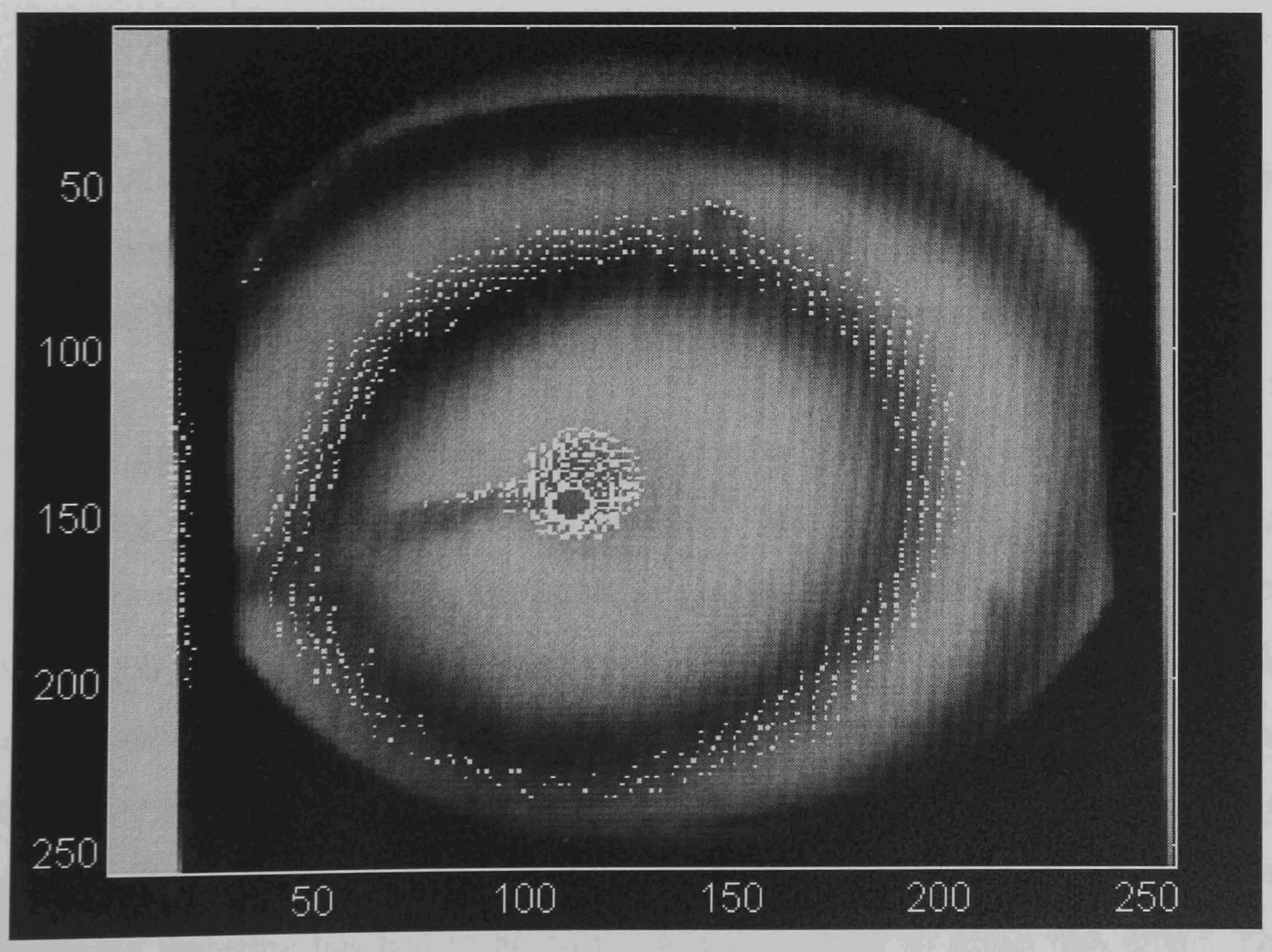


Figure 4.5

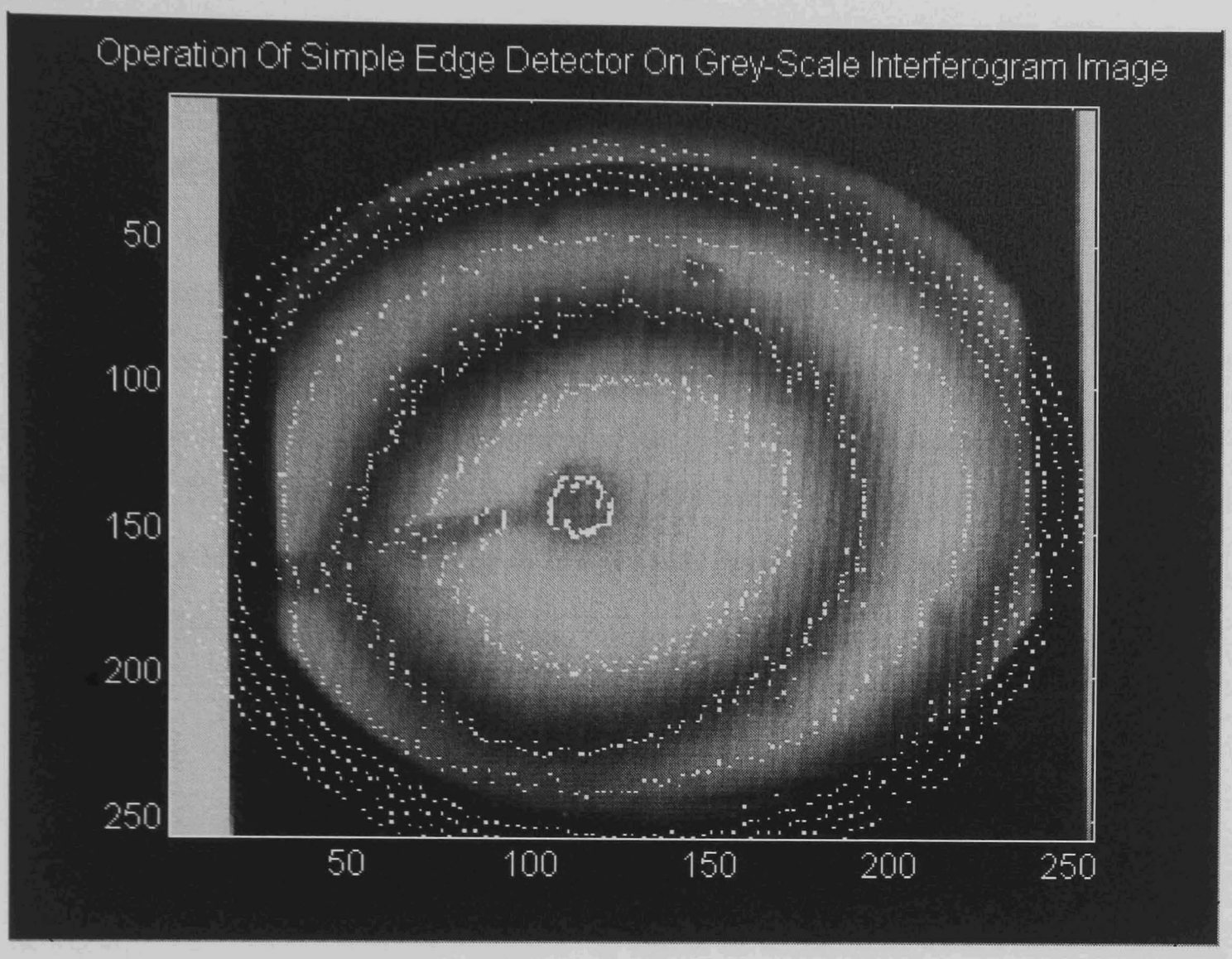


Figure 4.6

Further research on methods of detecting fringes has suggested that the best approach to fringe finding is to use a first derivative fringe locator. As the sides or 'edges' of fringes are slopes of undefined but continuously varying slope, any detector attempting to locate them is faced with an impossible task, namely trying to give a set location in space to a feature which extends over an area. Even with a 'step' edge only two pixels wide, the simple detector of Figure 4.3 provides a high response for the full edge width. In the case of edges sloping gently over tens of pixels, the results, as in Figure 4.6, are poor. The advantage of taking the first derivative of the image is that the resulting gradient profile crosses zero at both the fringe maxima and the fringe minima. This enables feature types to be detected and differentiated from one another either by utilising the direction of the change of sign of the gradient, or the second derivative. The principal limitation of derivative based detectors though is that they are well known for amplifying noise in an image. Image smoothing therefore has had to be built into the operator to offset this effect. This

must however be accomplished without making the smoothing such that it is image dependant, or blurring the image to such an extent that the accuracy of feature location in the image is lost. The implementation of this is discussed in more detail later, but it places a definitive limit on the amount of unguided blurring that can be performed for noise control. Much of the more recent work has been done in order to overcome the false detection and 'wide' feature detection problems that noise points at the detection stage cause for later stages in the processing cycle. At later processing stages more information is available to make decisions on noise on a feature basis rather than on each individual pixel which is of unknown relationship to its neighbours.

4.2.2 Gradient Based Peak Detection

This section discusses the testing of gradient based methods of fringe peak detection. The term 'Gradient operator' is taken to mean an operator that examines the gradient of an image slice as a contour of an image, regardless of whether the operator actually carries out mathematical differentiation or not.

The 'starslice' algorithm, which has been developed, is based on feature location using the gradients in a grey scale image along the image slices taken by the slicing operation. It is important to distinguish points which are local maxima due to noise spurs from the real fringe peaks in the image as early as possible in the processing sequence. In order to assist with noise suppression, the detector checks the points each side of a zero crossing in order to locate points where the gradient goes form +ve to -ve (change from rising to falling slope distinguishes a fringe peak from a fringe valley) without turning back through zero right away. This latter feature is typical of sharp noise spurs but not rounded interferogram fringes. In order to achieve smoothing to offset the noise amplification inherent in taking the gradient of a grey-scale image, the algorithm uses the average of the slope between four sequential pairs of points as the gradient. The resulting vector of locally averaged gradient values is then scanned for zero crossings. In the scanning pass, it is assumed that if the five gradient values before the present point sum to a gradient value greater than a small positive noise threshold (set by testing on various sample images

to find a suitable value) that the image has been increasing in intensity up to this point along the vector. The five points beyond the present scan point are likewise summed and assumed to represent a drop if their sum is below a similar small negative noise threshold. A fringe (a `bright` ridge or area maximum in an image) is declared to have been located if both the above tests are true for a centre point which has a gradient value of approximately zero. The actual value of the gradient at a nominal zero crossing point is unlikely to be zero as the image has been discretised into pixels, and the true fringe peak will most likely lie between two of the pixels. This operator is latterly referred to as being eleven pixels wide, one for the point under test (as a fringe peak) and five for the slope detection each side of this point.

Various limits have been used to avoid detecting the image edges as fringes, and to cope with the problems of practical implementation. For example the fact that in practice the slices have to contain points beyond each end of the array of image points that have been mapped into them. This is generally in order to allow all the image points to be processed by multi-point wide operators without incurring computer errors as a result of operators overlapping the ends of the slice arrays.

This algorithm has been improved in a number of ways after testing on the sample interferogram images. Notable improvements include the removal of the limitation on the detected zero crossing point having an actual value in a pre-set range. Due to noise this could exclude perfectly valid fringe peaks. Various measures were also taken to improve the coding of the algorithm to minimise processing time requirements.

The operation of this algorithm on an image containing a small number of very wide fringes is shown in Figure 4.7 and its operation unchanged on an image with much narrower fringes is demonstrated in Figure 4.8. This shows the wide input image applicability of the approach.

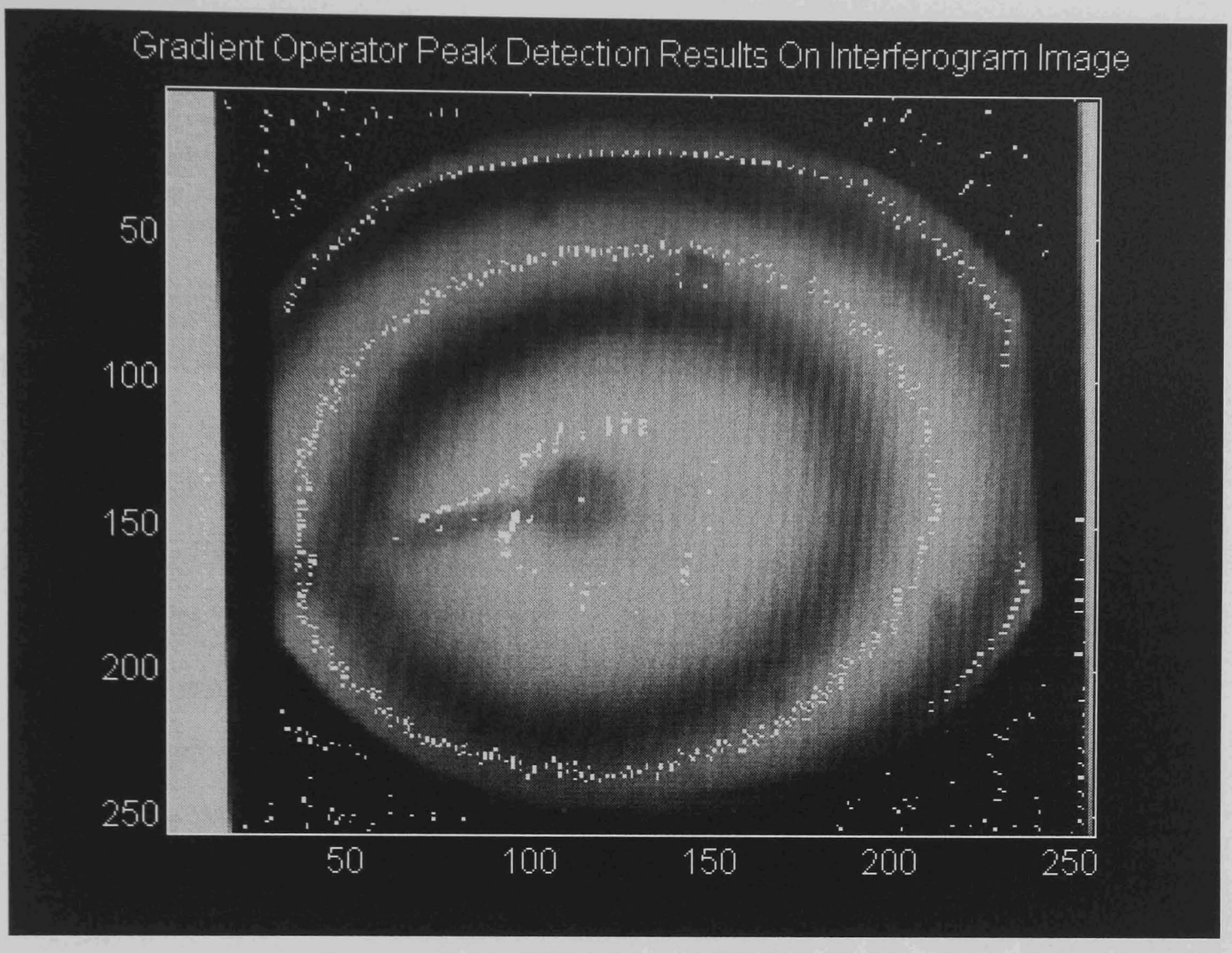


Figure 4.7

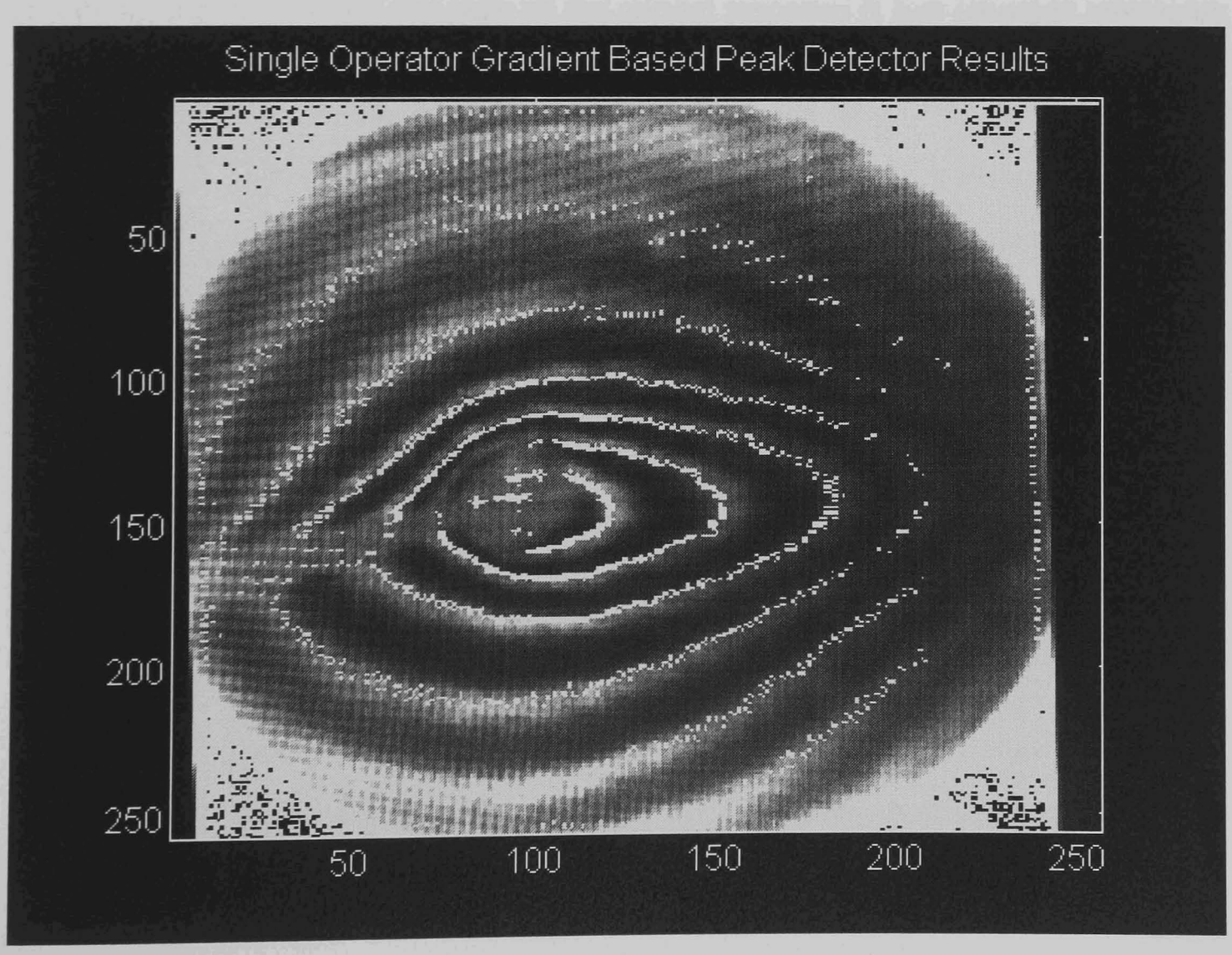


Figure 4.8

4.2.3 Multiscalar Operation

In order to further improve the peak detection, the possibility of using a multiscalar algorithm was investigated. The basic zero crossing detector has a set operator width. The slice values for an arbitrarily pre-determined number of points each side of the point under examination are summed to find the average gradient. This is used to test for the presence of an interferometer fringe at that location. The problem with using a set size of operator on unknown input images is that the operator performs best when the spacing between fringes is at a certain ratio with respect to the width of the operator. If the fringes are more closely spaced than expected, two fringes could occur within the width of the operator. This would render the assumption that the points it is looking at represent the two sides of a fringe peak invalid, even when a fringe peak lies at the detector centre. Likewise, wide fringes, with respect to the operator, could be so flat on top as to appear level. Eventually a point is reached where either a threshold is set below which data is considered flat, or any noise in such an area is detected as a fringe peak. This also results in failure to locate the peak as the correct response is lost amongst erroneous detections (as the detector sensitivity is to low for the signal to noise ratio).

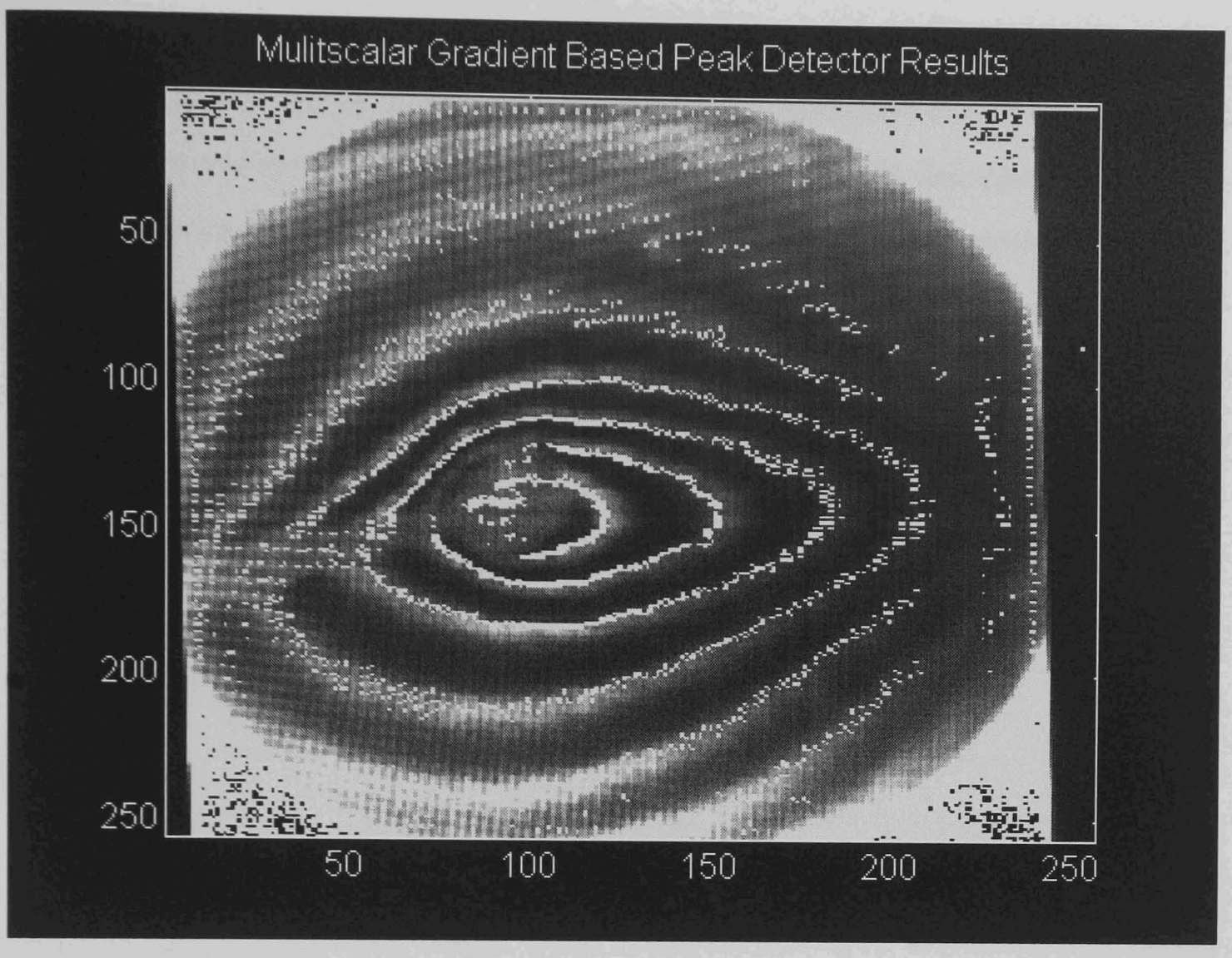


Figure 4.9

The drawback of a multiscalar algorithm where multiple passes of several operators based on the same mathematical approach are made over the same data, is that the time requirements for processing are multiplied by the number of passes. However, changing from the basic operator of 11 points (one possible peak point and five each side over which the gradient was averaged, to get a rising, falling or no slope result) to multiple widths, proved beneficial. The initial operator widths used were 13, 19, 25, 31 and 37, with the result that the peak detection performance in images consisting of relatively few very wide fringes improved dramatically. Even with images more suited to the original narrow operator, an improvement was seen as demonstrated by comparing Figure 4.9 with Figure 4.8. The detection of the low contrast fringes to the right of the image has been considerably improved. Another advantage of using a set of wider operators is that the setting of the no-slope noise level cut-off becomes much less critical. The cut-off can be set at a cautious value on the assumption that, missing a peak with a narrow operator as a result, is better that getting false detections when this failing will be compensated for by the wider

operators which are intrinsically more sensitive. The narrowest operator required is obviously determined by a compromise between the detection of too many isolated noise points and the minimum fringe width to be detected. In this case it was set to suit the test images but for general operation, a further 7 wide operator would be appropriate. There is in practice little point in searching for fringes much less than ten pixels wide using zero crossing techniques, as fringes and noise spikes become too similar in shape and eventually the nyquist sampling limit is reached. In practice the theoretical limit accessible is noise dependant, and the closer to the limit the algorithm operation point is the less generally applicable the algorithm is likely to be.

4.2.4 Turning Peak Detection's Into Fringes

Discussion so far has concentrated on the initial detection of the potential fringe maxima points. This results in an image consisting of circles of fringe peak detections that, although appearing as lines on a plot, have no relationship to each other in the computer. Having acquired point-wise fringe detections the problem is to get the computer to realise that point 'a' and point 'b' are actually part of the same fringe, not merely adjacent to each other. The peak point detection alone however does not directly assist image analysis operations. It is though a required first stage of the process of detecting and identifying fringes in an image.

Converting peak detection to fringe detection is where the, radius against angle, nature of the 'starslice' data array really becomes a benefit. In a normal Cartesian expression of the peak point locations, the locations of two points, unless adjacent, give no information on the likelihood of their being on the same fringe. However as the fringes are known to be positioned around the central point in the angular array, the presence of a fringe peak at a certain distance from the centre of the fringe pattern in 'slice one' suggests there will also be a fringe at the same distance from the centre, or nearly so, at the next angle around. This relationship, though useful, can be used as nothing more than a guide, as fringe distortion can cause significant variation in the radius at which a fringe is detected from one angle to the next.

The first approach used to attempt the linking of fringe peak detections into complete fringe 'objects' was to search around the centre point for points in adjacent fringes of nearly identical distance from the centre point. This was readily achieved with the radial data array mapping, as from a starting point, the algorithm can vary the angle values, looking for points in the next slice which are a similar distance form the centre to the present point. Both directions through the peak pattern must be scanned. Although this search system is good in theory, testing identified numerous problems. The first is how far in or out the next point in a fringe can be and yet still be classified as being part of the present fringe. If only $\pm$ 1 point is permitted, then fringes which lie even for a short distance at an angle of over 45 degrees to the slices will be missed. In practice this is generous as noise disturbances can cause fringe detections to vary randomly about the true fringe peak line on a pixel-by-pixel scale. At the other extreme, if too much change is permitted, the detector can jump from one fringe to the next as demonstrated in Figure 4.10, a situation refered to as 'interfringe jumping'. This is an error condition, but is hard to detect, as with fringes at unknown angles in the image, the shape of an interfringe jumping peak trace can look mathematically, and visually, similar to a good detection following a noise corrupted fringe peak.

This angle-by-angle, varying radius scan, is similar to the method used in the application of image processing to the interactive analysis of interferometric fringes, Funnell [148]. This technique scans fringe peaks looking only over a 180° range of angles to avoid doubling back on itself. However, the radial storage format gives certain advantages, though testing has proved that much less advantage can be taken of this in general images than might seem the case at first. This is because as the fringes tend towards ellipses, as the convection process increase, even with the radial approach, the fringes are closer to being straight lines than circles once the ratio of the maximum to minimum diameter of the elliptical fringe exceeds 2:1. Another problem due to noise in practical general images is that the fringe peak may not be continuous. Allowance therefore has to be made for small gaps in the fringes without causing interfringe jumping to occur.

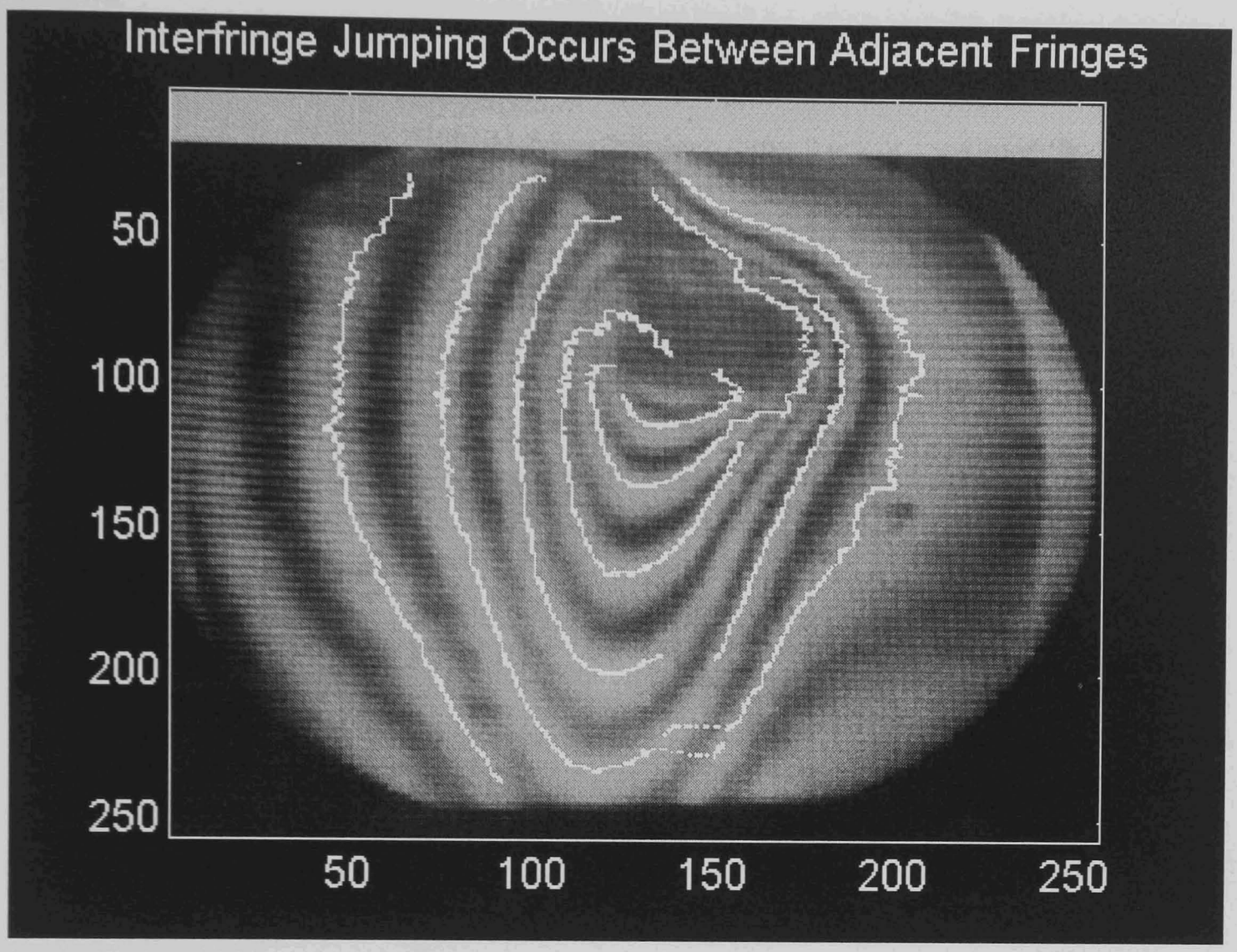


Figure 4.10

4.2.5 Dealing With Errors In Fringes

The basic fringe following procedures proved in practice to have a number of problems in coping with realistic fringes. If the fringe patterns being processed are relatively noise-free, an algorithm that searches for the next fringe point only in very close proximity to the present fringe point is suitable. This is the approach taken in line following algorithms, in simple two-dimensional arrays of line or fringe detection points. However if moderate amounts of random noise as observed in the simulated interferometer convection images are present, either gaps in the fringes occur, or the search must be made sufficiently wide ranging to allow errors in the form of jumps from one fringe to the next to take place. As was noted in the previous section, many of these interfringe jumping errors are hard to detect by examining the shape of the detected fringes. It is better to be strict with the initial linking stage and try to link the gaps than to allow errors to occur. This approach has distinct

advantages. All the points grouped at the first linking stage can be eliminated form further searches and as good fringes do not contain branches, only links to the ends of these fringes from any other fringe sections need to be considered. This greatly reduces the chances for errors, particularly in that errors generated from processing poor fringes do not corrupt good data before it is processed. (points are deleted from the input data to avoid processing the same data twice) This process also removes large amounts of the data points from further consideration (they are no longer potential fringe end points) making further processing much faster.

These decisions gave rise to a three-stage fringe finding algorithm.

Stage 1 : Use the 'Starslice' algorithm to get detection points plotted as ones in an array of radius against angle.

Stage 2 : Collate the points making up the 'good' fringes using the 'slice-string' algorithm with no gaps permitted in fringes and a tight limit on point-to-point radius variations compared to the last pair of fringe points.

Stage 3 : a: Mark fringe ends in the resultant array.

b: Try and match up fringe ends that actually correspond to either side of a gap in a fringe or to a place where, due to a noise spike, a section of the fringe was not detected correctly.

4.3 Fringe Linking Schemes

4.3.1 Fringe End Proximity Linking

Having produced a set of detected fringes that have been linked by a robust linking algorithm, the typical result is a number of fringe segments that apparently to an observer should be connected end-to-end to form several complete fringes and some fringe arcs. A number of approaches to solving this problem are possible and several have been analysed with varying degrees of success.

The first technique tried was to use search 'areas' extended from each fringe end in order to find other fringe ends that might link up, as shown in Figure 4.11. This 'area' as an extension to the fringe ends for matching approach proved unsuccessful primarily because these 'areas' were as likely to overlap the ends of fringes parallel to the other half of the broken fringe as they were to overlap the correct broken end. Although in theory this system should work and is reasonable on a pair of ends in isolation, fairly large 'areas' are required to overcome gaps several slices wide and this approach has no means of coping well with multiple features that interfere with each other, a common occurrence in testing.

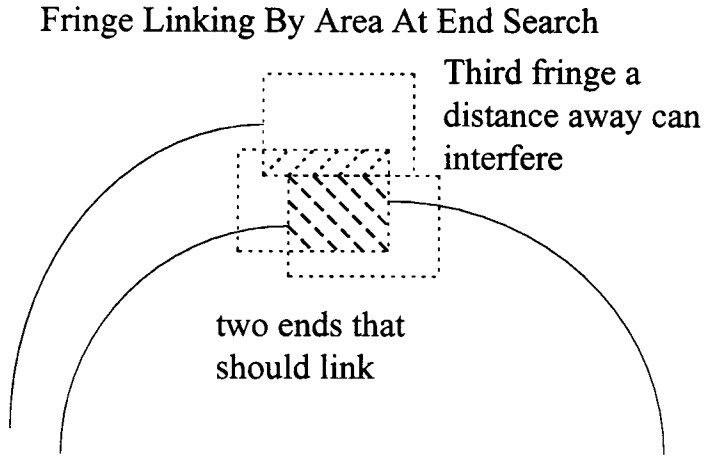


Figure 4.11

4.3.2 Fringe Tangent Linking

Another approach to linking fringe sections is to examine the tangents at the ends of the fringe sections. In theory, if two fringe sections belong to the same fringe in the image, the tangents to the ends of the fringe segments detected so far that lie on the fringe should be nearly parallel. It should therefore be possible to plot these tangents, as in Figure 4.12, and look for cases where they intersect between fringe ends that are relatively close together. Relatively in this context is flexible as the need is mostly to avoid problems from fringe ends 180° apart around the radial slices array having parallel tangents, although they are separated by a considerable lateral distance. In practice, this approach has numerous problems. Fringe ends that should be linked across imperfections in fringes are typically distorted by the algorithm struggling, but succeeding, in detecting the last few points each side of the break as

noise levels increase. This effect often means the fringe ends are curled round to some extent. Due to the radial slice arrangement, an angular deviation proportional to the width of the fringe gap is to be expected. Finally, other parallel fringes may also have loose ends nearly parallel to the first but only slightly further away. The result is that typically if the fringes are averaged far enough to nullify the distortion of the broken ends due to noise, the correct angular inaccuracy in the tangents is at least as large as the probable difference between tangents to the fringe that should be linked and the next fringe. The result is that either no fringe linking occurs, or that the algorithm links ends almost at random depending on the setting of noise suppression parameter limits. This technique was therefore abandoned. Even after multiscalar versions using different numbers of fringe end points to interpolate the tangents from had been tried, no improvement was observed.

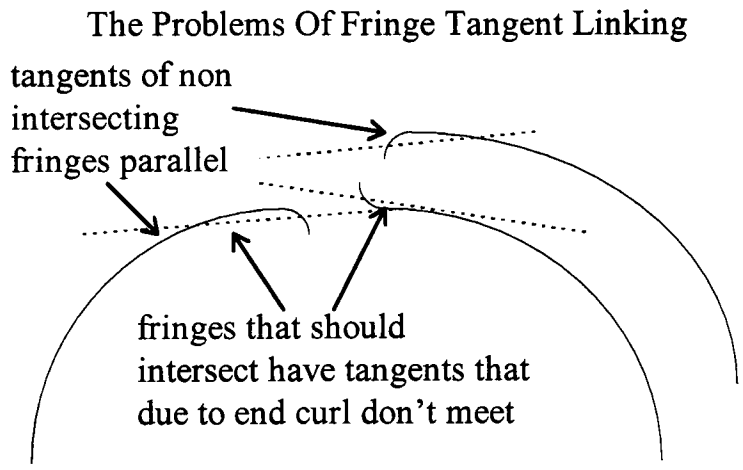


Figure 4.12

4.3.3 Dilate And Thin

An algorithm for the fast derivation of line structures from interferograms, Eichhorn [145] proposed a dilation and thinning approach to fringe linking. Normally using dilation algorithms on fringe peak images has the disadvantage that if fringes get close together they tend to merge into each other faster than they spread across fringe gaps. The solution proposed however was to work from a fringe peak-and-valley detection image using a three-state image, fringe peak, fringe valley, or neither. Fringe peaks and valleys are alternately dilated but a point may only be converted by dilation if it is in the not-yet-assigned state. Therefore as fringe valleys

and peaks alternate with each other in an image, fringe cross-linking is almost excluded, as peak-to-peak links are blocked by valley detections. Likewise peaks prevent dilating valleys from merging into each other. The fringe detector in an algorithm for the fast derivation of line structures from interferograms, Eichhorn [145] however is seriously inadequate to cope with the fringes to be processed in the sample image. The powerful peak detector in the form of the 'starslice' algorithm was therefore combined with this approach.

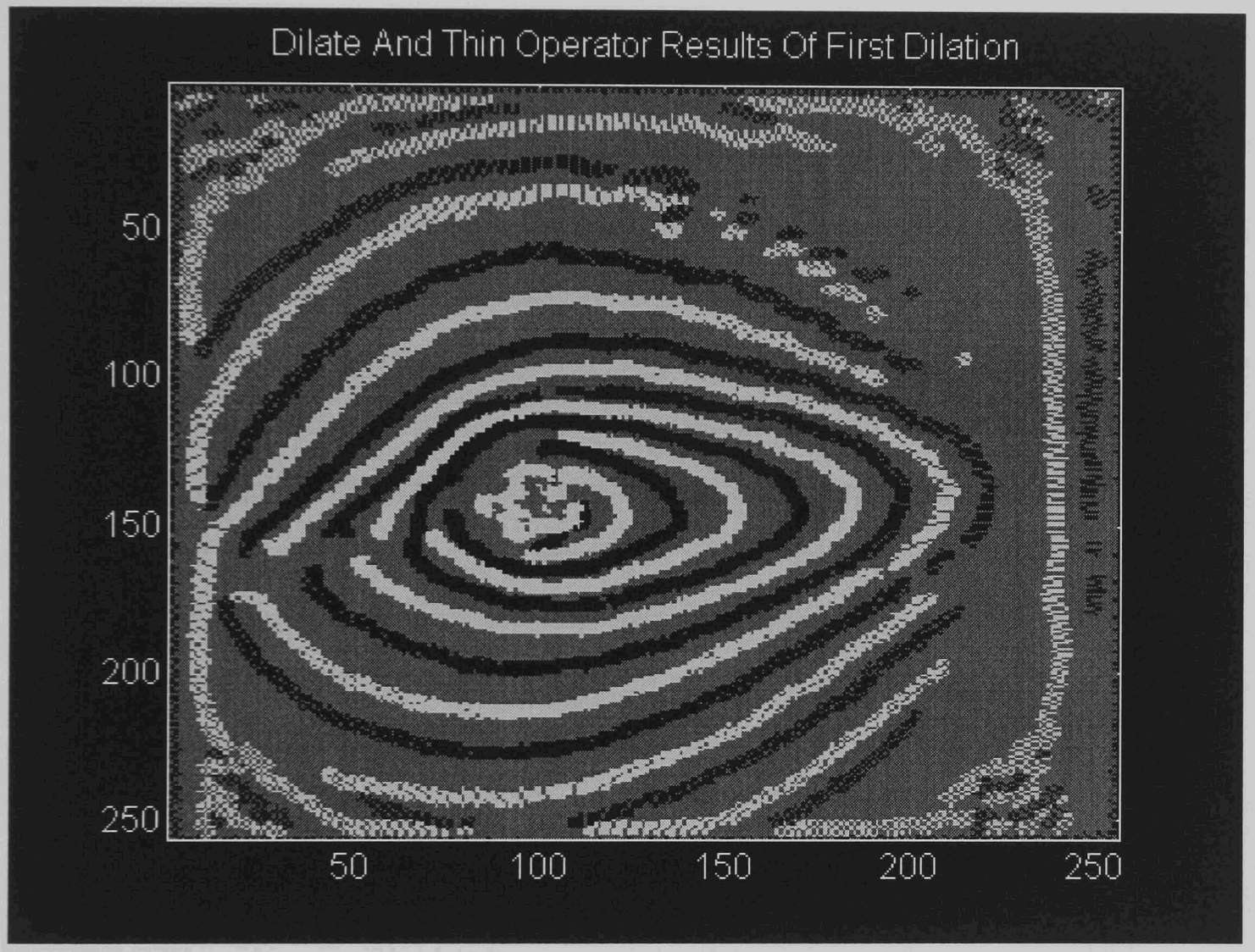


Figure 4.13

In order to achieve valley detection in addition to peak detection, each radial slice is processed twice, once for peaks, and once inverted with the peak detector finding valleys. The two feature types are distinguished by storing them as different values in the results array. Figure 4.13 shows the results of applying this detection scheme on a fringe image, with one dilation stage having been applied to start closing the gaps between the detection points. Two major problems with a simple

dilate-and-thin approach however were soon discovered. The basic algorithm, Eichhorn [145] dilates the fringes until peaks meet valleys, which is acceptable in simple images with evenly spaced fringes. However, the sample images suggest that the microgravity convection images will be uneven leading to the result that one side of a fringe could dilate several steps more than the other. This would result in an offset of the fringe detection result after thinning, which trims both sides evenly. Another problem is that gaps even one pixel wide, that are nearly closed in the dilation stage, expand once more during the thinning stage.

These thinning problems however can be overcome to a large extent by taking advantage of the radial data arrangement. If dilation is carried out treating the radius-versus-angle array as a plain image, but thinning only along the radial slices, then gaps in the fringes do not expand during thinning. Likewise this offers a solution to the fringe centre line displacement/movement problem. As any gaps in the fringes will not now grow along the fringes in a radial only thinning pattern the image can be thinned severely, and this may be done iteratively with dilation stages. This allows large fringe gaps to be closed in several stages while avoiding the need to dilate until the peaks merge into the valleys. This eliminates the migration of the detection locations during the dilate-and-thin stage due to varying fringe widths. Full details of the operation of this operator along with explanative diagrams are presented in the final algorithm description.

4.3.4 Final Development

With the dilate-and-thin stage iterative, the algorithm was improved to get the best out of each of its sub sections. Unnecessary array conversions were removed and the thinning stage was written so that the valleys detections were kept until just before the final linking stage in which they are no longer required.

The fringe dilate-and-thin stages were then integrated with the fringe-linking operators. Finally the whole system was built into an overall control algorithm that supplies the various parameters required by the subsidiary algorithms that are constants for all images but which have been parameters during development. At this stage, further use was made of the tangent fringe linking operator for a final attempt

at linking fringes. This used the much improved input available from the improved algorithms then completed, which brought the interfringe linking problem down to a controllable level. The results however were still unacceptably poor.

The final version of the algorithm is described in Chapter 5. It uses, or improves upon, most of the techniques tested or proposed during the developmental stages.

Chapter 5

Image Processing, The New Algorithm

5.1 The Image Processing System

5.1.1 Introduction

This chapter provides a detailed description of the final version of the image processing algorithm developed to extract a gravity vector, as discussed in Chapter 2, from interferogram fringe images. This algorithm has been developed from both previous work surveyed in Chapter 3 and developmental testing summarised in Chapter 4. It has been demonstrated that the algorithm can provide data extraction from the noisy simulated images generated during the initial experiment conceptual work discussed in Chapter 2.

5.1.2 Overview of the System

The 'roof valley vector' algorithm takes an interferogram fringe image as its input and returns the complete fringes found in the image and the gravity vector it has determined from the average movement of the fringes centres of gravity. The gravity vector result assumes that the best measure of the movement of a fringe due to convection is the motion of its centre of gravity. Convection tends to carry liquid of a certain temperature further from the source of heat in the direction opposite that of the source of gravitational force. As the fringes are equivalent to contours of equal temperature, they therefore show a net movement in the 'upwards' direction. This motion can be extracted regardless of the shape of the resulting fringes by locating the centres of gravity of the fringes.

The algorithm takes as parameters, the image to be processed, the location of the central heating unit in the image, the position in the image of the support wires

that run to it and the fraction of a fringe that is required to be detected for a fringe detection to be considered complete. This last parameter allows the sensitivity of the processing to be varied from for example, detecting only nearly complete fringes, to detecting fringe sections and interpolating where the remainder of the fringes would be. This interpolation assumes the fringes to be complete if the undetected sections arc smoothly across the gaps. This enables a trade-off between the accuracy of the processing and the number of fringes that could be found in an image to be specified. Which aspect is more important depends on whether the operator wishes to use the algorithm for generating high accuracy vectors from data of known quality, or wants a 'best estimate' regardless of the quality of the data.

5.2 Application to the Image

5.2.1 Image Parameters

Two types of parameters are used by the algorithm. The parameters discussed in Section 5.1.2 are taken as parameters for the interferogram fringe image to be processed. This data is mostly experimental set-up dependent and, apart from the fringe fraction to be considered as a minimum for centre of gravity interpolation, is typically constant for all images taken from a specific set of experimental results. The parameters are determined by the physical characteristics of the experimental equipment. Algorithm-dependent constants are also present and form the second group of parameters required by the algorithm. These parameters, for example the number of iterations of algorithm processing stages, are critical. However they have values that are dependent primarily on the operation of the algorithm and not the image being processed. Their values have therefore been selected after testing on the simulated images. The values chosen enable the algorithm to function effectively on any input image, without having to have prior knowledge of the nature of the fringes that the image contains. One of the most important of the constants is the number of image slices radiating from the centre point of the image to be taken by the 'starslice' stage of the algorithm. For optimal results, every point in the input image should be

mapped into one of the slices. In practice, this means that most points near the image centre will be mapped into several different fringes, resulting in data redundancy. If full accuracy is desired, the required number of slices can be calculated, but in practice, testing has shown that utilising better than one-point-in-two from the largest possible radius fringe in the input image provides little benefit. For the 256 by 256 pixel test images therefore, this parameter has been set to 512, giving the best trade-off between speed and accuracy, there being 1020 (just under twice 512) image edge points in such an image. More generally, the choice of a value for the number of angles to be processed is principally limited by the desired processing speed. The required processing time increases approximately linearly with the number of slices processed. The accuracy also increases with the number of slices used but in a less predictable manner, as the gap between pixels at the edge of the image covered by the slices is reduced. The sufficiency of these parameter values is discussed in more detail below.

5.2.2 Algorithm Starting Point

Before commencing fringe location, the image is first smoothed with a 5 X 5 averaging filter to suppress the point noise in the image. Only one pass is made in order to ensure that fringes as small as five pixels in width do not suffer significant corruption to their basic shape at this stage. This operation minimises the effect of sinusoidal noise in the test images. These were entered using an image scanner, resulting in image distortion, mostly in the from of narrow sinusoidal fringe bands due to the imaging technologies in this process. The first processing stage performed by the system is the conversion of the image to an array of slices for processing by the fringe detection operator.

5.3 Image Representation

5.3.1 The Star Slice Operation

Having smoothed out the localised image noise, the image is then copied from its rows and columns format into the distance-angle format of the slices array. This operation uses the 'starslice' operator described in Chapter 4. Figure 5.1 shows the details of the 'starslice' mapping.

'Starslice' Operator Mapping

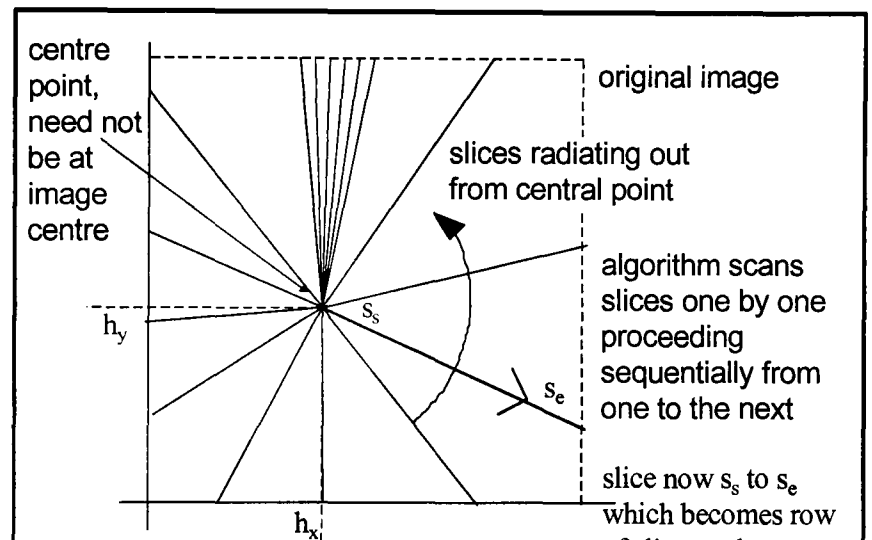


Figure 5.1

Where :

h_x = x co-ordinate position of heating element.

h_y = y co-ordinate position of heating element.

s_s = fringe start at pattern centre.

s_e = fringe end at image edge.

The first step in the 'starslice' operator process is to define the slice array. This requires a row for every angle at which a slice of the fringe image is to be taken, and a column for every 'pixel of radius' from the slice pattern centre to the most distant image corner. The starting point of the slices array is the supplied location of the fringe pattern centre (fluid cell heating element) and the size of the longest slice is calculated. This is graphically illustrated in Figure 5.2.

The maximum distance of any image point from the image centre is:

$$d = \sqrt{\left((c_x - h_x)^2 + (c_y - h_y)^2\right)} \qquad (1)$$

Fringe Centre to Image Corners

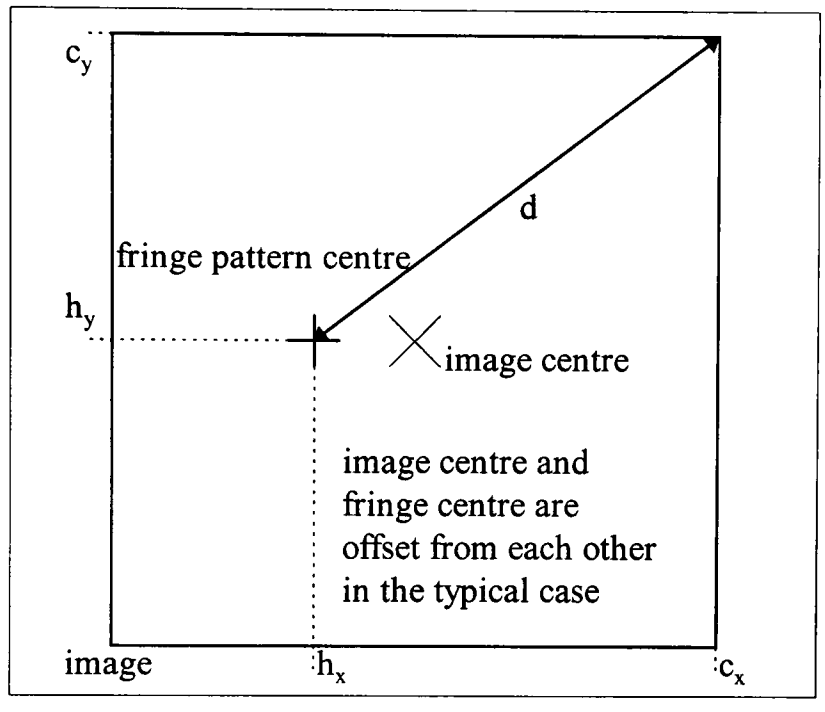


Figure 5.2

Where :

d = distance from centre point to edge of image

h_x = x co-ordinate position of heating element

h_y = y co-ordinate position of heating element

c_x = x co-ordinate position of image point : most distant image corner

c_y = y co-ordinate position of image point : most distant image corner

The maximum gap between points in the input image that are mapped into fringes can then be calculated. To demonstrate the sufficiency of this mapping, an example using parameters typical of the test images used during algorithm development is presented (256 by 256 pixels with processing of 512 image slices). Assuming that the fringe pattern centre point has been specified at pixels x=100 and y=100 (an offset of 28 in each dimension from the image centre, which is typical of the ground simulation data) the distance between pixels mapped from the input image to the slice array can be calculated.

With 512 angles :

$$i = d \tan \frac{360}{s_{tot}} \tag{2}$$

Where :

i = inter-pixel separation

d = distance from centre point to edge of image

s_{tot} = total number of slices

Therefore :

$$i = \sqrt{\left((256-100)^2 + (256-100)^2\right)} \tan \frac{360}{512} = 2.7 pixels \tag{3}$$

Due to the nature of the image, the actual area of the image in which complete fringes can be identified is typically only about half of this radius from the centre point, giving a nearly complete mapping of image points to slices in the central part of the image. This calculation demonstrates that setting the 'angles' value higher would be ineffective as it was previously stated that mapping one point out of every two pixels on the largest fringes has been demonstrated during testing to be sufficient for accurate fringe location.

5.3.2 Mapping Of Fringes

A sample idealised image demonstrates the conversion of a typical fringe pattern from the image space to the distance angle array. Figure 5.3 shows an idealised low gravity convection image in which the gravity vector and the first slice of the array of image slices are vertically downwards. In this example therefore, the fringes move vertically upwards due to convection and outwards in all directions form the centre due to conductive and radiative heat transference (though in general they could move in any direction depending on direction of the gravity vector acting). The corresponding set of fringes are shown converted to 'distance angle

array' format by the 'starslice' mapping in Figure 5.4. Figure 5.5 shows a real sample image of a set of distorted fringes generated during ground simulations and Figure 5.6 is the corresponding set of slices. In practice, the slices are individually converted for processing, and only the detection data is saved for the next stage (as the distance angle conversion of the raw image is of little use).

Gravity Vector and Slices

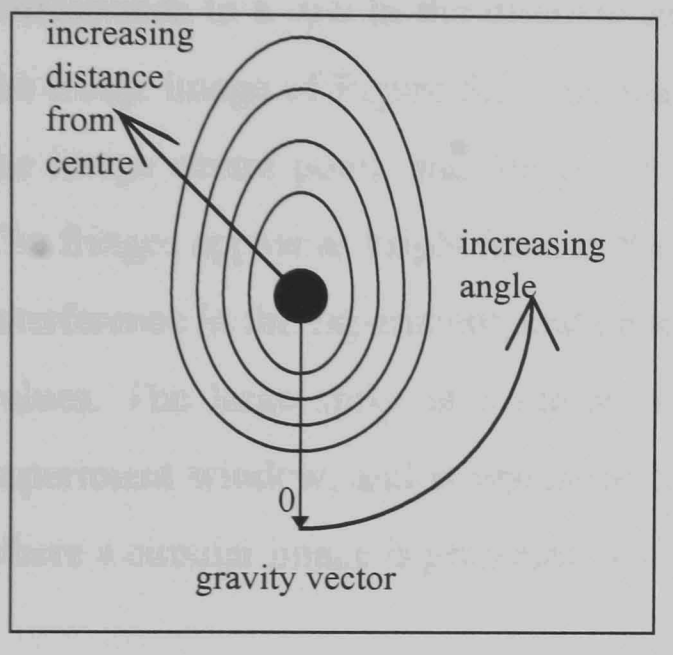


Figure 5.3

Distance Angle Array Format

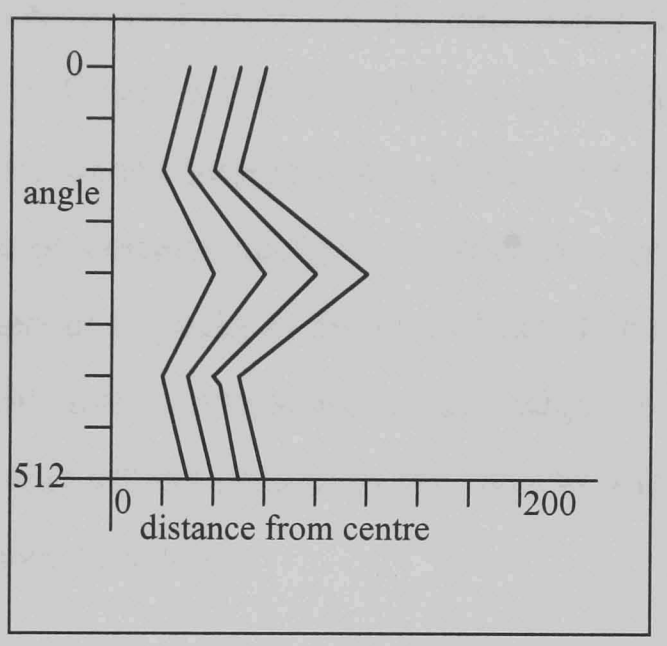


Figure 5.4

Sample Interferogram Image

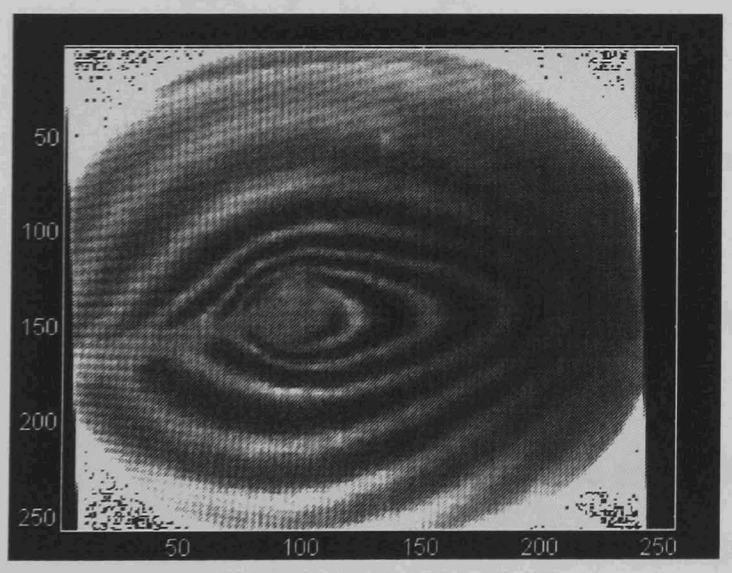


Figure 5.5

Image in Distance Angle Format

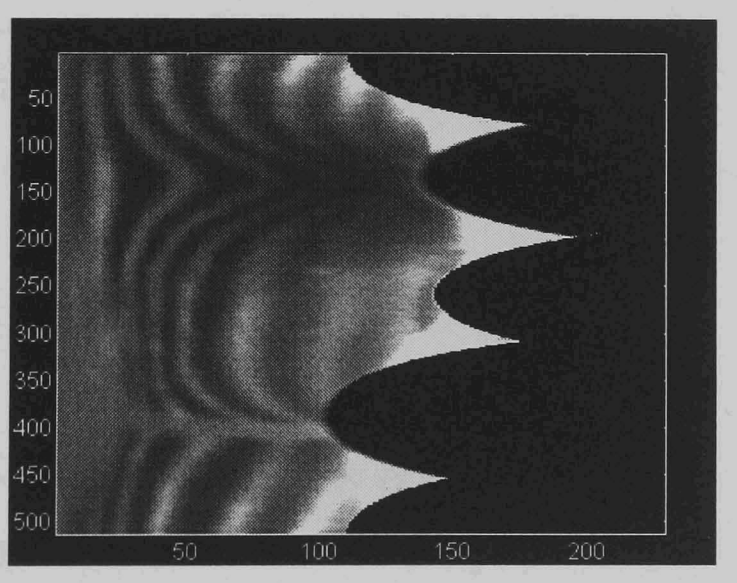


Figure 5.6

5.4 Slice Processing

5.4.1 Slice Contents

Having converted the image into an array of slices which present the fringe data in a more easily processed format, fringe detection is performed on the slices. A sample set of slice data from an angular slice is shown in Figure 5.7. This corresponds to a row in the distance angle array of Figure 5.6, the data being from the fringe image of Figure 5.5. The x-axis in Figure 5.7 is the radius in pixels from the image centre point, and the y-axis is the image intensity at a particular radius. The fringes appear as bright lines in the image, corresponding to constructive optical interference in the experiment, and hence appear as peaks in the plot of the slice data values. The large spike at a radius of 150 units corresponds to the edge of the experiment window, and is typical of the edge effects present in the sample images where a circular image is presented in a square data array.

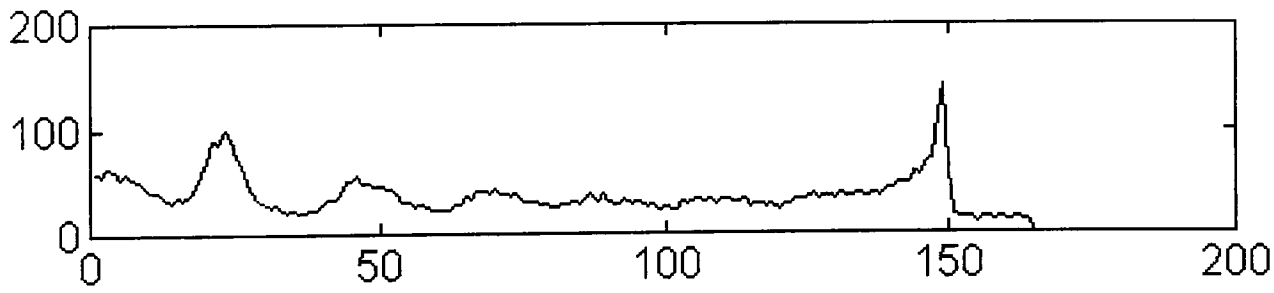


Figure 5.7

5.4.2 Finding Image Intensity Gradient

The array is processed one slice at a time to detect fringe peaks and valleys. A zero-crossing of the image gradient detector is used to find the fringe peaks and troughs. The first stage in this process is to produce a gradient profile of the image slice. The slice contains data points corresponding to image intensity values, and the intensity gradient is therefore achieved by taking the difference between the values of adjacent points. In order to avoid corruption of the data by sharp narrow noise spikes, an averaged gradient value is required. The general smoothing done during

pre-processing helps to minimise, but does not eliminate, this problem. In order to avoid obliterating the fringes entirely, due to smoothing, the average gradient value must be achieved by an operator that does not further distort adjacent fringe points. An operator width of five data points is used to sample the slices, the results being recorded separately. However, the four inter-pixel data value differences which result from processing the four pairs of adjacent data points cannot be simply averaged. This is numerically equivalent to taking the difference between the end points of the array of data points being used to generate the gradient values and dividing by the total distance in pixels between them. This is a problem because this approach ignores most of the data in the image slice, as demonstrated below.

Assume that the slice data is :

$$val(x-2), val(x-1), val(x), val(x+1), val(x+2) \qquad (4)$$

where :

val(y) : is the value of the data point y

x : is the position in the slice for which the gradient is required.

Then for the five data values around the present point in the slice whose gradient is being calculated, the simple average gradient may be expressed as :

$$\frac{\left(val(x+2)-val(x+1)\right)+\left(val(x+1)-val(x)\right)+\left(val(x)-val(x-1)\right)+\left(val(x-1)-val(x-2)\right)}{4}$$

$$\frac{val(x+2)+val(x-2)}{4} \qquad (5)$$

This demonstrates that most of the information content of the image slice is ignored by this approach. In fact this would result in a value with no relationship to the data at the point in the slice it is supposed to represent, clearly making its utilisation as the gradient value for that point invalid.

5.4.3 Gradient Zero Crossing Detection

To overcome the aforementioned problem, namely that averaging the gradient values does not provide a realistic mean result, a more complex method for determining an averaged gradient value is proposed. The four gradient difference values generated by taking the gradient between each of the adjacent pairs of image points are first sorted into numerical order. The average of the two median gradient values is then taken as the gradient value for the point. This approach rejects the highest and lowest gradient values which are the most likely to be due to noise corruption of the image slice, and averages the remaining two values to deal with noisy sections without generating too many erroneous zero crossings. It is still possible that the image value for the slice point being processed will not form part of this final value formula, but the effect of all five image points has been taken into account, providing a more representative average gradient for the point under consideration. In fact if the data at the slice point that is nominally being processed is not used in the final gradient using this scheme, the reason is almost certainly that the averaging operation is correctly suppressing a noise spike at that point. This is because in typical cases, the four gradient values lie on the side of a fringe and are in order, or they lie each side of a smooth fringe peak. In these conditions, the outer pair of gradient values will be the largest and smallest and hence the two which are rejected.

Consider the data points :

$$val(x-2), val(x-1), val(x), val(x+1), val(x+2) \tag{4}$$

The four gradient values from the differences between adjacent points are :

$$\left\{gradient_{((x-1)-(x-2))}, gradient_{((x)-(x-1))}, gradient_{((x+1)-(x))}, gradient_{((x+2)-(x-1))}\right\} \tag{6}$$

These are sorted into numerical order, renaming the values :

$$\{gradient_{lowest}, gradient_{2nd}, gradient_{3rd}, gradient_{highest}\} \tag{7}$$

The average of the two median gradients is then the value used as the gradient at point x in the image slice :

$$gradient_x = \left\{\frac{gradient_{3rd} - gradient_{2nd}}{2}\right\} \tag{8}$$

The result of this operation carried out along the whole slice is a smoothed gradient profile of the image slice. A typical gradient profile result, in this case for the slice data of Figure 5.7, is shown in Figure 5.8. The gradient data point averaging offsets the tendency of the differentiation operation to amplify any noise present in the data.

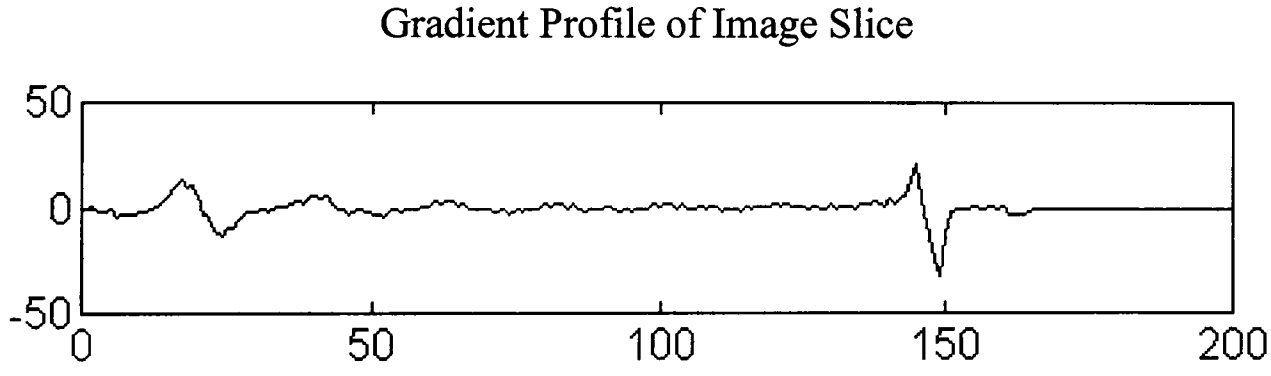


Figure 5.8

5.4.4 Detection Results

Having converted the data points to gradient profiles, the slices are scanned by the zero-crossing detector. This is done for both the positive-to-negative zero crossings, which denote fringe peaks (roofs), and negative-to-positive zero crossings of the gradient that indicate the fringe minima (valleys) respectively. Multiple operator widths are used. For the maximum response, and hence detector sensitivity

to be achieved, the operator applied to the gradient data must be approximately one half fringe wide, although the method functions over a much wider range.

As well as using multiscalar operators, two further levels of smoothing are applied to each slice in turn and the full range of operators are then applied to the results of each of these. This serves to minimise the effects of variations in the image noise level which may otherwise render any one value of image smoothing sub-optimal.

The second width of operator that is used is seven pixels wide and serves as a good example of the formula used for the zero crossing detector:

The fringe detector is implemented by scanning along the fringe looking for zero crossings with an 'n' pixel wide detection operator. First the average gradient of the image intensity slope on each side of the point being tested for the presence of a fringe peak is calculated. The slope length is the first quantity determined in order to accomplish this, and is operator-width dependant.

$$l_{slope} = \frac{w_{operator}}{2} - 1 \qquad (9)$$

Where :

$w_{operator}$ = width of the operator being applied (total fringe points used in peak detection).

l_{slope} = length of average slope section each side of point being processed.

The two slope values can then be calculated for the two supposed fringe sides, one on either side of the operator centre point's position.

$$S_{plus} = x_{(rad-l_{slope})} + x_{(rad-(l_{slope}-1))} + \ldots + x_{rad-1} \qquad (10)$$

$$S_{minus} = x_{(rad+l_{slope})} + x_{(rad+(l_{slope}-1))} + \ldots + x_{rad+1} \qquad (11)$$

Where :

S_{plus} = value of positive slope. (assuming fringe peak search)

S_{minus} = value of negative slope. (assuming fringe peak search)

rad = present position in slice, the radius in pixels from image centre.

x = value of fringe data at a point in the image slice.

This operation assumes that a peak is being detected at the present location, 'rad', and then tests to see if this assumption is correct. If there is a fringe at the present array location, the value of the array at the present position and the value one pixel to one side must be of different signs, or one must be zero and the other non-zero. Also, the gradient over the last l_{slope} slice points will be positive, and that over the next l_{slope} points will be negative. A check is then made to confirm this. If it is confirmed, then a zero crossing has occurred.

$$if \left\{\left\{(x_{rad-1} > 0)\&(x_{rad} \le 0)\right\} \left\{(x_{rad} \ge 0)\&(x_{rad+1} < 0)\right\}\right\} \qquad (12)$$

$$if \left\{(S_{plus} > 3)\&(S_{minus} < -3)\&(S_{plus} < 150)\&(S_{minus} > -150)\right\} \qquad (13)$$

If these two relationships are valid, the present radius in the slice is noted as a fringe peak. A lower limit of 3 on the net slopes is required for the slope to be detected. As the data is stored as 256 grey levels, there exists the possibility that any one or two level fluctuations in an otherwise unvarying image could be called peaks if a lower sensitivity limit is not set. A value of 3 was chosen after testing and has proven to be suitably independent of the input fringe image. The latter 150 limit ignores the level-256 to level-0 jumps that sometimes occur at image edges where the image has been blacked out within parts of the input array. This is required as the fringe images are circular due to the shape of the experimental apparatus, but the

array they are stored in is square to suit computerised storage and manipulation. The value of 150 is used simply as a value greater than one half of 256.

The fringe peak detection process is then repeated for positive-to-negative zero crossings to detect fringe minima, as these separate the fringe peaks. This is done by inverting the equations, replacing positive values with negative values and vice-versa. The minima are not required for the final fringes but are used in the next processing stage to improve algorithm robustness. A typical set of detections for the slice as shown in Figure 5.7 is given in Figure 5.9.

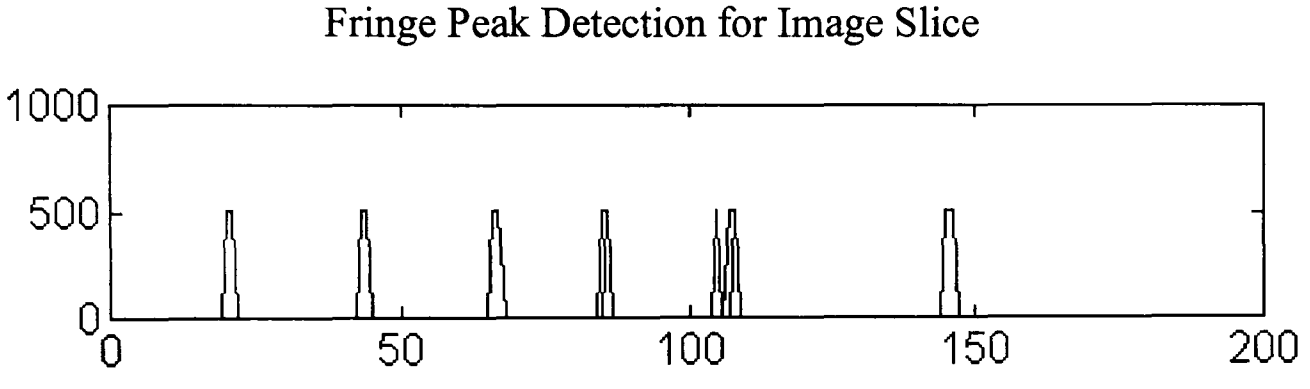


Figure 5.9

5.4.5 Discussion Of Results

Once all the slices have been processed, a raw fringe detection array, containing all the points that the various operator widths have detected in the input image, is available. The resulting array is stored as 'distance along the slices' against 'slice angle'. Although the plot, Figure 5.10, of the array contains structure, this information is not available in a useable format at this stage. This plot, Figure 5.10, is the corresponding 'roof and valley' detection image for the image of Figure 5.5. Roof points are bright (white) and valley points are dark (black), while uncommitted points are grey.

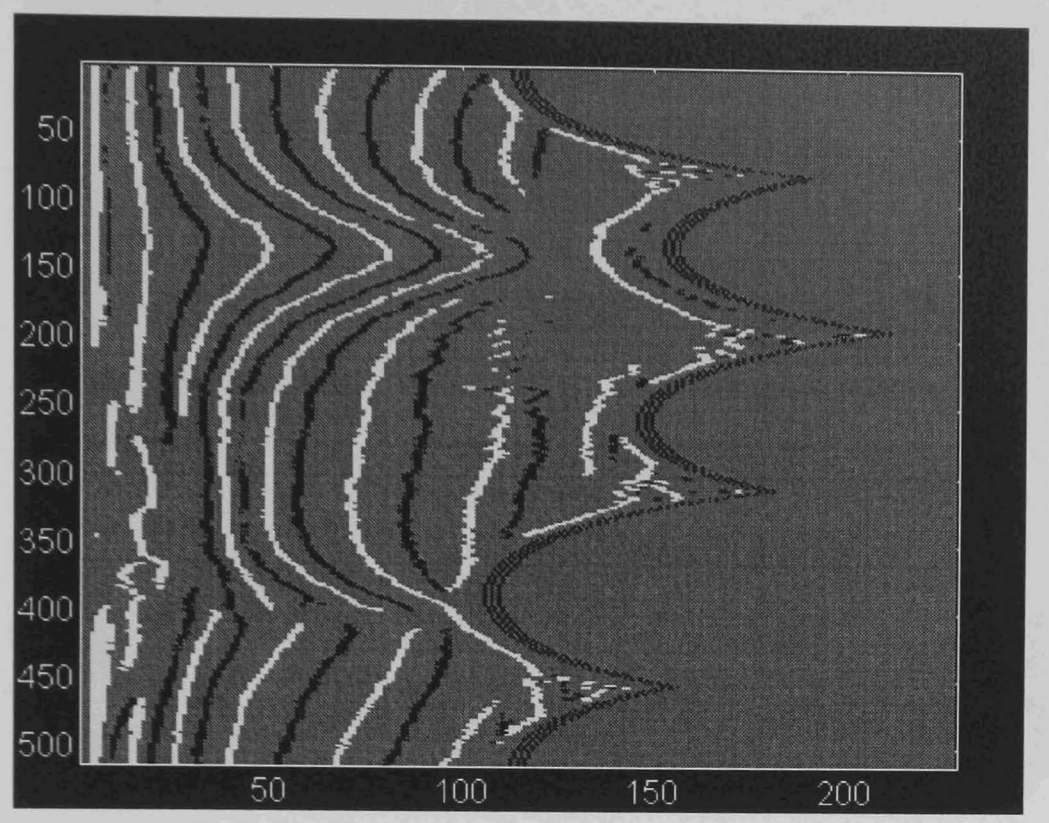


Figure 5.10

At this stage of the detection process, the fringes have been marked. When the data is converted back to a rows-and-columns format, the results, as shown in Figure 5.11, appear to the human operator to indicate that the system has found and identified complete fringes, whereas this is only an illusion. At this stage, the processing system has simply detected all the fringe points but has not identified their relationships to one another. When the array is plotted however, the structure inherent in the data is obvious to a human observer who sees complete fringe detections. The much more difficult task of grouping the detected fringe points into actual fringes that are recognised as such by the processing system is the next stage to be tackled.

Roof and Valley Points Plotted on Fringe Image

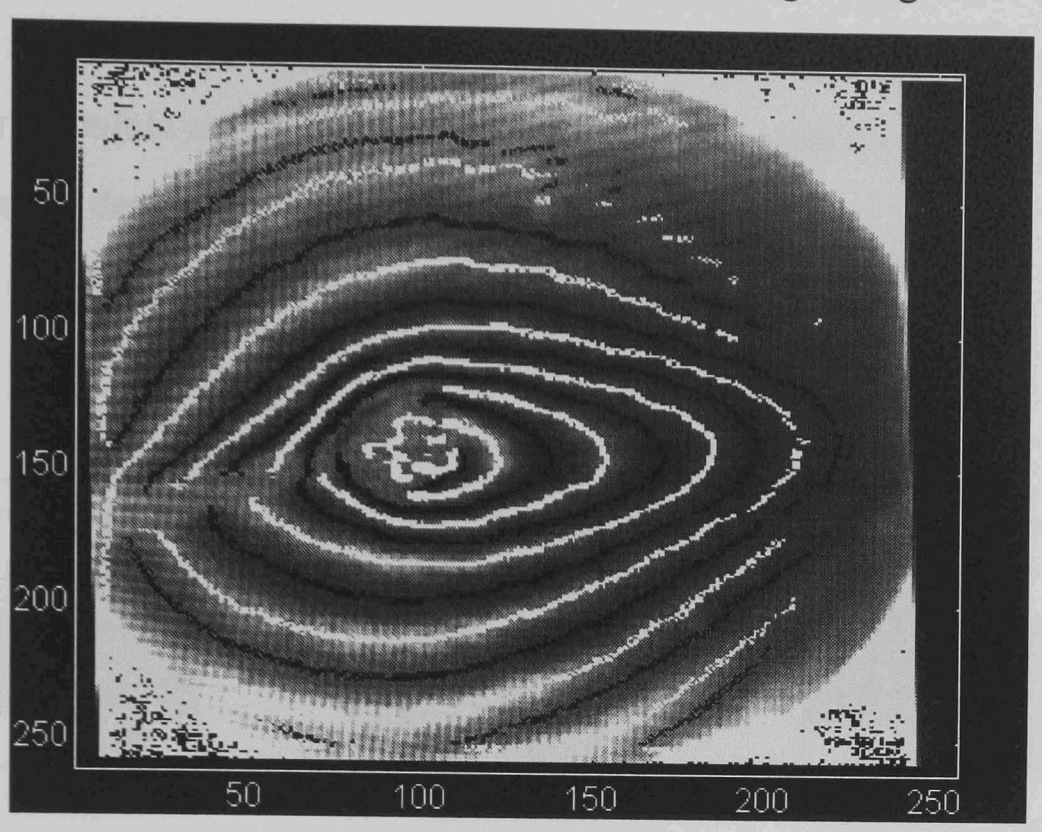


Figure 5.11

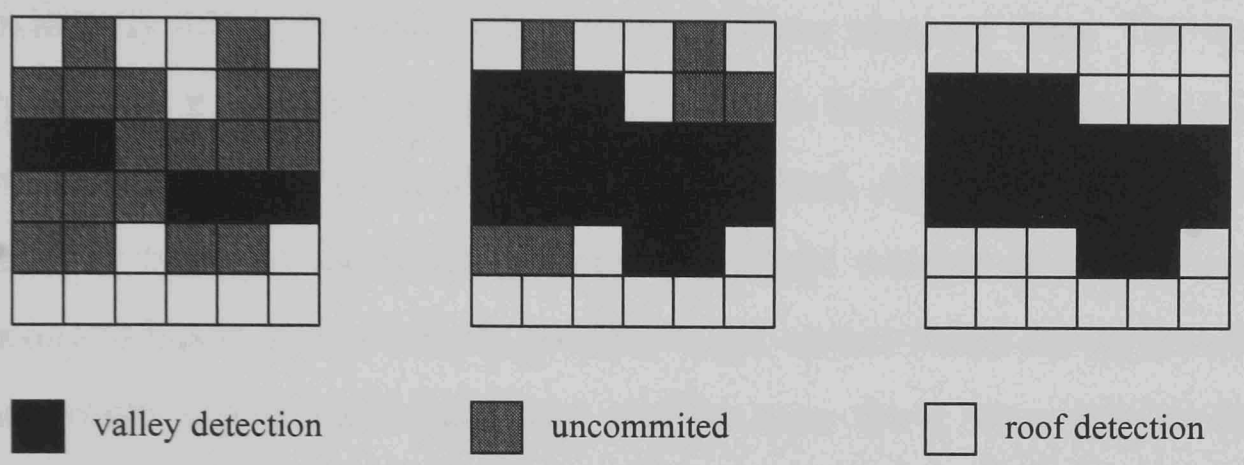
5.5 Raw Data Processing

5.5.1 Detection Dilation

Prior to actually applying the fringe grouping operator, a pre-processing operation to filter the raw detection array is applied. The processing system is designed to retain the distance angle format of the data, which is also used for the remaining processing stages. The filtering operation consists of carrying out dilations of the detected fringe points in order to fill in any small gaps in the detection pattern. These gaps may have been caused by noise in any of a number of stages in the image capture and digitisation process. The valley detections are dilated first, as it is the roofs that are desired, and this order of operations maximises the probability of preventing interfringe linking from occurring. The only type of changes permitted are those of unallocated points, thus preventing the changing of roof points to valley points or valley points to roof points. This enables each type of detection to prevent the dilation of the other. By completing the valleys first, a 'wall of valley detections'

is effectively built between the different roof fringe detections before they have the opportunity to be merged into one another by the dilation process. Dilation of both valley and roof points uses an eight neighbour dilation carried out in the slices angle distance array as demonstrated in Figure 5.12.

Dilation Operator



Dilation operations example. The lefthand image is simulated raw detection results. Note that if the white valley detections were dilated first, the fringe at the top would meet the fringe at the bottom. As it is, the valley dilation prevents this, but still allows the roof detections to link between themselves as in the righthand image.

Figure 5.12

5.5.2 Thinning Of Features

The process of thinning is then carried out along the image slices. The slices may typically contain one to three detections of a fringe peak due to noise even before the dilation stage widens them further. This inevitably may lead to fringes that are several pixels wide by the time that the thinning stage is commenced. These wide fringe detections should by preference be reduced to a single line of fully connected points corresponding to the peak detection. The first operation therefore is to mark all the points on the dilated fringe detections which are inside or outside edges of fringe detections. These are the points to be considered for deletion. The possibility that more than one point may have to be removed from each side of the detection is solved by iteration of the thinning operation. The marked detection edge points are

then deleted if, but only if, they are not required to keep the fringe detection complete.

Care is taken not to just thin to the centre point of the detection that is stored in any particular slice. In this representation of the fringe pattern, two adjacent points along a slice could actually be adjacent pixels along rather than across the fringe in the more typical rows-and-columns representation of the image. This problem typically occurs if the angle between the slice and the fringe has a value of approximately 45° or less during the image-to-slices mapping stage. The reason for multiple fringe points then lying in a single slice is that, in this case, the fringes lie closer to being parallel to the slices than across them. Although Figure 5.12 would typically be taken as representing increasing fringe number along the x-axis and increasing radius in y, it is entirely possible to get correct fringe detections where the fringes would appear such that fringe number runs along the y-axis as shown in Figure 5.12. The fringe radius would similarly be along the x-axis, due to distorted fringes in the original image.

The process is repeated several times in order to get thin fringes. A check for completeness of this operation is not used, as testing has indicated that a set value, around the expected dilated fringe width, is better as a good trade-off between a functional algorithm and perfection. Very noisy fringes could result in detections so wide that such assumptions are not valid, but there comes a point where the results would be so poor that this really does not matter too much. Perfect thinning is not absolutely vital at this stage as a final full thinning process is used at the end of the dilate-and-thin operation in order to ensure that only thin detections are passed to the next stage. An example of thinning the fringes from Figure 5.12, assuming that the points around the window shown are not fringes, and that the slice number is increasing along the x-axis, is shown in Figure 5.13.

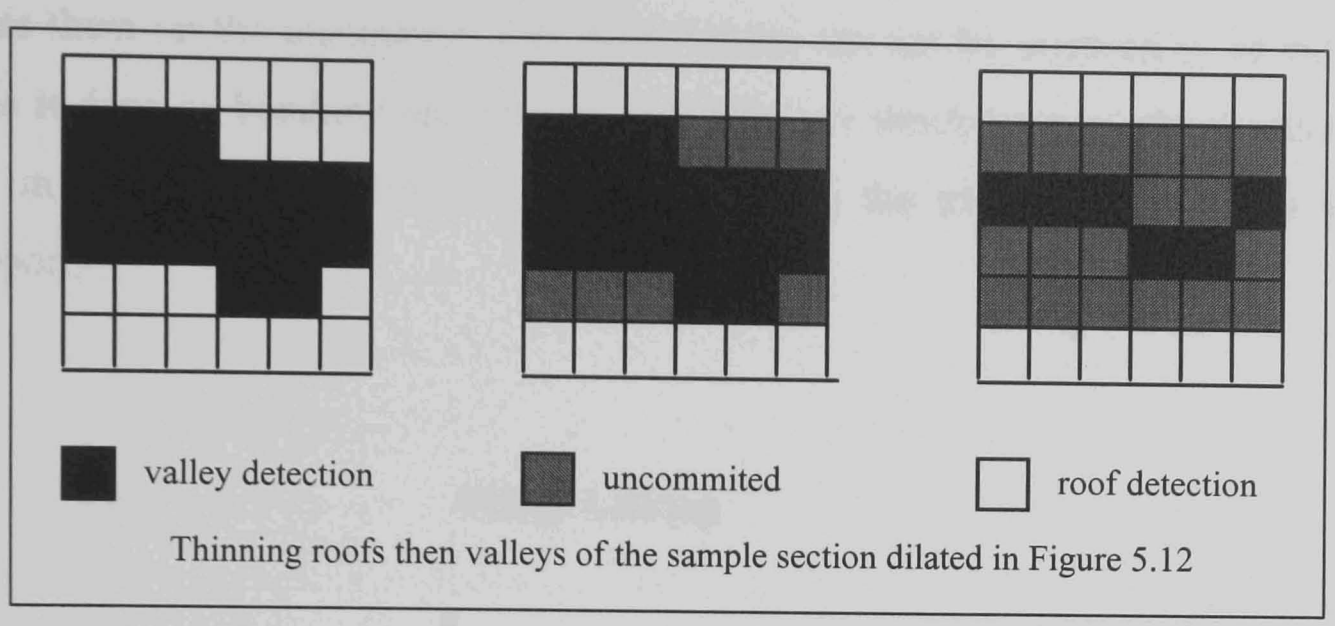


Figure 5.13

5.5.3 Iteration Of Dilate And Thin Algorithm

The thinning process is then repeated for the valley detections, and the whole 'dilate-and-thin' stage is iterated. The result is a set of thinned fringes that have had any small gaps closed by the dilation stage, as dilations are allowed both along the image slices and to adjacent slices. Thinning meanwhile is carried out strictly along individual slices of the image only, to prevent the fringes breaking up where the step in the radius is more than one point. As dilations are kept to a maximum of two pixels at a time between thinnings, operations are still accurately centred on the fringe peaks regardless of the real shape of the fringes. This overcomes the principal problem with the 'dilate until the roof and valley detections meet' approach, discussed in Chapter 4, which although simpler, has the drawback of giving uneven fringe dilation if the fringes are not evenly spaced. As this latter feature is quite common in the sample images available, the robust approach to the problem presented here is considerably more likely to provide a solution without distorting the location of the fringes in the image during processing.

The fringes have by this stage been 'cleaned' and noise suppressed. Image tidying up is then carried out in order to delete data that has been used to assist the fringe detection and thinning stage. The redundant fringe valleys are dropped, and

the heater wire supports are removed by intentionally breaking the fringes which cross them on the assumption that those results can not be assumed to be correct. This is done by blanking the sections of the fringes which lie upon those areas that the input parameters identified as corresponding the physical location of these supports.

5.6 Fringe Linking

5.6.1 Location of Fringe Start Point

With the thinned and tidied roof image completed, the first stage of fringe linking is commenced. A start is made by searching outwards from the centre of the image in a spiral pattern, searching for the first fringe point in the image. This point, once found, is assumed to be on the first fringe, though a backtracking step is available so that if further searching shows the point to be unconnected, it can be discarded without being passed to the next stage as a whole fringe vector. Once a fringe point has been found in the data array, the algorithm starts searching for adjacent fringe points around the centre of the image by checking the adjacent fringe

Spiral Search Pattern

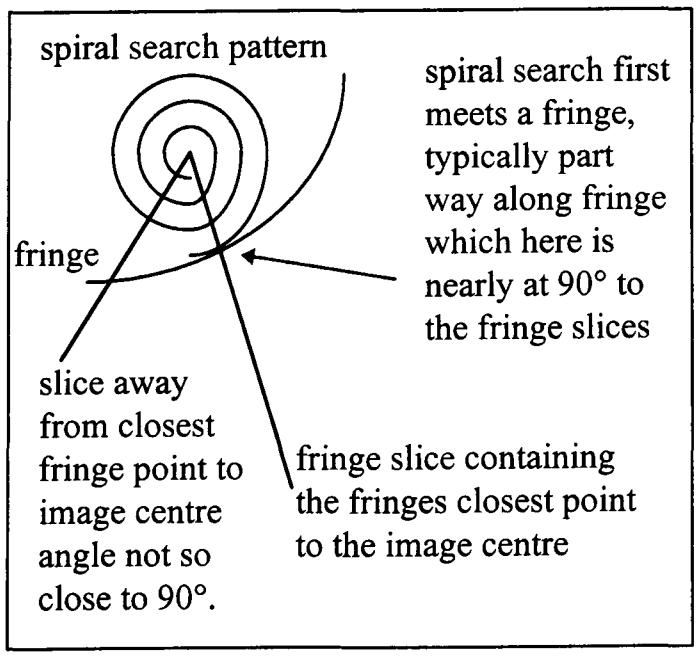


Figure 5.14

slices. As it must be assumed that the first point found is in the centre of a fringe, this scan is repeated in both directions around the image centre point by looking first to increasing the image slice number and then decreasing the slice number. Figure 5.14 shows the spiral search pattern matched against a section of fringe similar to those to be found at a fringes closest approach to the scan starting point. From the sample images, a typical fringe will be much more circular than shown and many points may share the same minimum distance from the pattern centre point. The example shows a considerable difference in the radius of the fringe to the distance from the centre point at which it meets the search for clarity.

5.6.2 Search For Fringe Points

The strategy for the search for other fringe points is based on examining the next angle around the centre in the 'distance against angle array' to see if there is a fringe point at the next angle, at the same or a similar radius. To allow for fringes that cross the slices at shallow angles, and bends in the fringe detections due to noise, more than plus/minus one pixel steps in radius are permitted. Changes in fringe radius value of more than a very few pixels are however permitted only if the fringe has tended to vary in radius by more than plus or minus one unit over the last few angles. This is to prevent the linking operation tracking straight from one fringe over to a point in the next fringe when a gap in the fringe being tracked is encountered. For the first few steps of the algorithm, little or no fringe tracing history is available. It is therefore assumed that the first point found lies on a section of fringe that is 'passing' the centre point, while crossing the slices at approximately a right angle. Because the remaining points in the fringe cannot be closer to the centre of the image, (they would already have been found, due to the central starting point of the fringe starting point scan) this assumption is generally good. If the assumption it is not valid, the point found must lie either at the end of a poor fringe, or on a very distorted fringe. In either case it need not be detected and is deleted as a noise feature.

The amount of radial variation present in the fringe being tracked (assuming sufficiently far into tracking the fringe for a-4 to have a value. Until this point special exceptions are used that limit the fringe variation so that it will be tracked conservatively) is:

$$lastdiff = ad(f_{nos}, a-1) - \left(\frac{ad(f_{nos}, a-2) + ad(f_{nos}, a-3) + ad(f_{nos}, a-4)}{3} \right) \quad (14)$$

Where :

lastdiff = average difference between the radius of last fringe point detection and recent previous fringe points.

ad = angle - distance array containing fringe detection data.

f_{nos} = fringe number.

a = angle or slice number.

For a perfectly circular fringe, lastdiff will equal zero. In order to allow for tracking noisy fringes, the difference in radius between a new point on the fringe being tracked and the last point accepted is limited to lastdiff plus two, and a one slice wide gap in a fringe may be tracked.

The resulting basic search area for the next point on a fringe is shown in Figure 5.15, and an example of the fringe-following operation is shown in Figure 5.16.

Fringe-Linker : Search Area

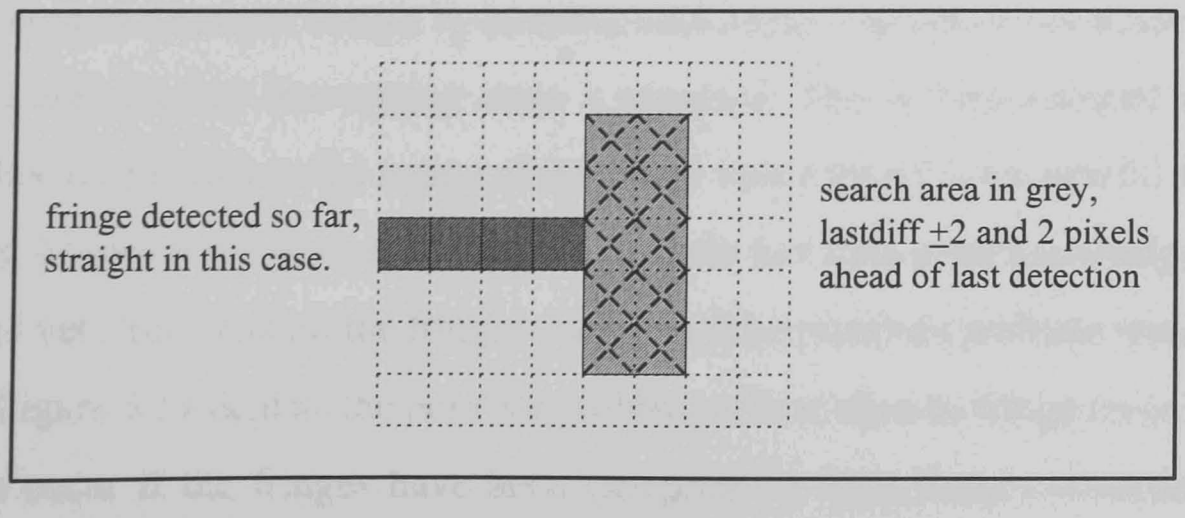


Figure 5.15

Fringe-Linker : Operation Example

This demonstrates how the search for the next point on a fringe is adaptively based on the past history of radius variation of points on that fringe.

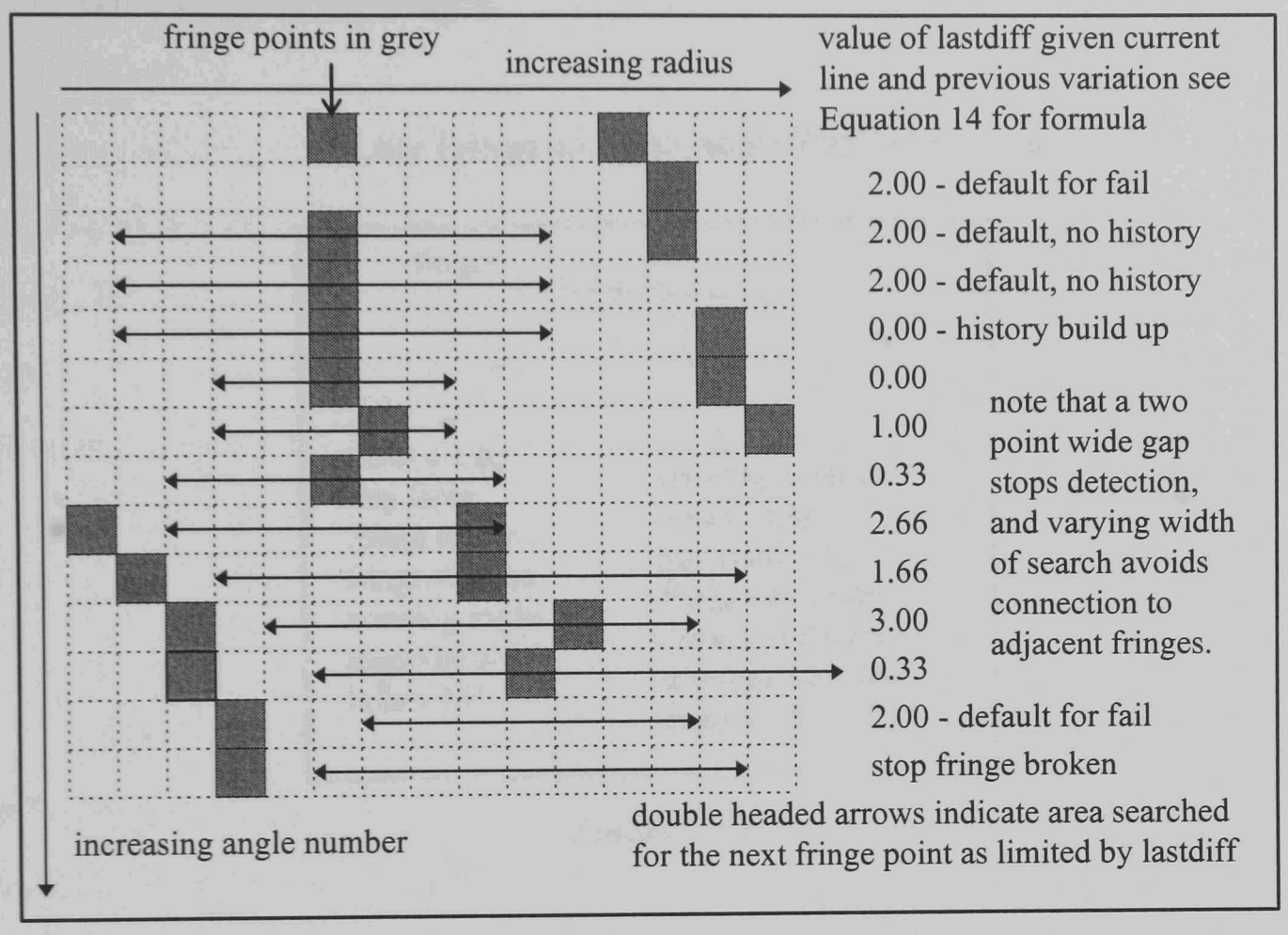


Figure 5.16

The allowance for a threshold using higher-value steps (if but only if the fringe has previously been varying rapidly in radius) allows the algorithm to cope if distorted or highly elongated fringes are present. The detections may step up in distance from the centre by five or more pixels near the image edges. The algorithm is able to adapt to this as the effect increases in magnitude. It also limits the step allowed, thereby avoiding interfringe jumps when closely packed fringes are encountered in the same image, by keeping track of the changes it has made in radial position over the last few tracing steps it accepted. This is demonstrated in Figure 5.16. This adaptability to the nature of the fringe being traced is a powerful feature in allowing images to be processed when the system has little prior knowledge of their form and yet still detect all the fringes without a false positives problem resulting.

Figure 5.17 depicts the possible problem of low slice-to-fringe crossing angle that can occur if the fringes have been elongated by significant convection in the fluid in the experimental chamber. This condition allows the use of a circular

assumption for the fringe shape for the first few points, but at later points on the fringe, demonstrates the case for requiring more flexibility of the algorithm in the general fringe-tracing case as the fringe-to-slice crossing angle becomes less due to the extended shape of the fringes.

Low Fringe-to-Slice Angle Problems

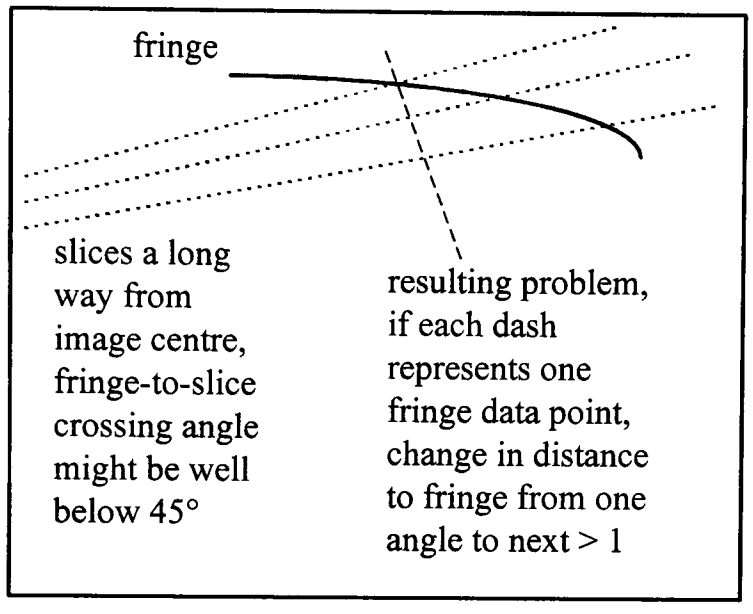


Figure 5.17

5.6.3 Fringe Following

Once the end of the fringe has been found, the algorithm returns to the first point it detected on the fringe and searches along the fringe in the opposite direction. The search can be stopped either because the algorithm has found a fringe that is a complete line around the starting point, or because there are no more fringe points detectable within the range acceptable to the algorithm for the next fringe point. If the fringe is complete after the first fringe-following stage, nothing further is done. Typically, some fringes are broken in at least one section because an image area is distorted, or the fringe overlaps the image boundaries. In this case, a second tracing stage is carried out from the start point to the other end of the fringe section.

Fringe points which have been detected are marked in the 'fringe detection array' so that the algorithm can avoid repeatedly processing the same fringe. These points may be used again to overcome cases where two fringes have blurred into one fringe detection over a small section of their length, and therefore legitimately share

a peak. Previously used points are not used as the start of a fringe detection, ensuring that eventually every detection point will be processed. A result of this approach is that the fringes compiled by the linking stage must be compared, and any similar fringes reduced to a single fringe (otherwise a single noise point can in theory start a second detection of the same fringe). This scan is easily done as most false second detections of fringes due to noise, result in fringes that are over 90% similar, while the sharing of fringe detections due to fringes blurring into each other is usually present for only a small fraction of a fringe. The merging of fringes into each other leads to such distortion of the results if it is not localised, that attempting to detect both fringes independently is not worth undertaking.

5.6.4 Overcoming Data Representation

Although the angle-distance array format is convenient for storing the data for image processing, the raw arrays of results in this format are difficult for a human observer to interpret. This is a problem during testing and is inconvenient for displaying the final results. This problem is solved however by transforming the results to a standard two-dimensional plot of the fringe points merged with the input fringe image for display. The actual numerical results are stored separately in an array of fringes for analysis. This gives a simple visual check that the results are sensible, no major errors having occurred, in the form of the fringe plot and provides the desired array of individual fringe descriptions for centre-of-gravity determination.

5.6.5 Linked Fringe Results

The final output of the fringe linking operation is an array of fringes. These are stored as 'angle of slice' against 'fringe number', with the radius of the fringe detection corresponding to each fringe angle being stored in the array. These results are stored as individual fringes, as this uniquely identifies the detected fringe points associated with each fringe, and they can be saved for further analysis.

To detect the movement of the liquid in the fluid cell in the microgravity experiment, the centres of gravity of each of the fringes are required. As each fringe is stored separately, it is straightforward to calculate the position in the image that each point on each fringe corresponds to, and to find the (1,0) and (0,1) moments of these points to get the co-ordinate location of the fringe centre. In order to minimise errors due to gaps in the detected fringes, a fringe interpolation routine was developed that projects arcs from the two ends of fringe gaps to the gap centres and averages them to get an estimate of where the missing fringe section is. This has been proven by testing to be effective for moderately sized gaps, as the number of accurate points is far greater than the number of interpolated ones. The end result is considerably better than simply setting the points in the gaps at radius zero. The accuracy however inevitably degrades as more estimation is utilised, and the amount of a fringe that can be missing before it must be discarded, must be weighed against getting any data from poor images. An investigation of the stored uninterpolated results can be undertaken manually however if desired, to give an indication of the trustworthiness of the final results.

5.6.6 Assessing Tangent Linking on the Final System Results

Following the success of the final version of the algorithm at extracting fringes from interferogram images, further testing of the tangent-linking approach has been carried out on the improved fringes produced by the final algorithm fringe linking stage. Although no errors are generated as a result of applying the fringe tangent-linking to these images, no further fringe section linking was achieved either. The primary reasons appear to be that the remaining fringe gaps that can be observed in the test images which should be filled are either so distorted by noise that the fringe ends are warped around, or overlap each other as the two fringe ends curl to the opposite sides of a noise disruption. Another common location for breaks is sections of fringe that bend with tighter radii than average. These again are not linked, as the tangents of the resulting ends are at random angles to each other.

The results presented below therefore use the completed algorithm as described above but do not use the tangent-linking algorithm described in Chapter 4.

5.7 Testing Of The Algorithm

5.7.1 Testing On Experiment Proposal Images

Results are presented from testing the algorithm on a series of different sample images. In each case, the only parameters entered are the starting co-ordinates giving the position of the heat source in the image, as due to variations in the test set-up to get different image types, this varies. In some of the images, the angular position of the disturbance due to the support wires for the heating element is utilised, where this has a significant impact on the images. Finally, a parameter called 'noisefract' is used to set the amount of a fringe that the algorithm is required to locate before considering a feature a valid fringe detection. This allows the partial detection of fringe sections that have been separated by the image edges to be ignored, as attempts to interpolate fringe movement become more error prone the smaller the detected fringe sections are.

5.7.2 Testing results

Example results from the testing of the algorithm are presented below. These figures consist of the input fringe image which is 256 pixels on one side. The detected fringes are plotted in white and the calculated gravity vectors are plotted in black. In each case, the notation at the top of the image refers to the parameters passed to the processing algorithm and the gravity vector it returned. The 'starslice' stage has in each case taken 512 slices radiating from the fringe pattern centre starting point, the first slice is vertically downwards and slice number increases in an anti-clockwise direction.

The algorithm (as a Matlab function) is:

function[xvect,yvect] = rfvect10(photo,xstart,ystart,cutang,noisefract)

With the parameters:

xvect = x co-ordinate component of calculated gravity vector

yvect = y co-ordinate component of calculated gravity vector

photo = filename of the file containing the input fringe image

xstart = x co-ordinate position of the fringe pattern centre start point

ystart = y co-ordinate position of the fringe pattern centre start point
(Where the starting point corresponds to the location of the axial heating element in the fluid cell of the interferometer)

cutang = slice angle corresponding to the fringe break
(This corresponds to the position of the heater supports which often distort the fringes. Where the distortion was minimal, it was set at a value greater than the maximum slice angle, which instructs the algorithm to ignore the operator in that specific instance)

noisefract = fraction of a fringe which must be present before the fringe will be used for gravity vector interpolation. All fringe sections that form less than noisefract of a ring have been discarded during processing and are not displayed.

Note that the row and column labels on the images are the Matlab Matrix indices. The starting point xstart,ystart is specified in terms of these indices, rows down the left hand side of the image and columns from left to right. The vector result is meanwhile given in terms of normal Cartesian co-ordinates, x-axis from left to right and y-axis from bottom to top.

Test Image 1

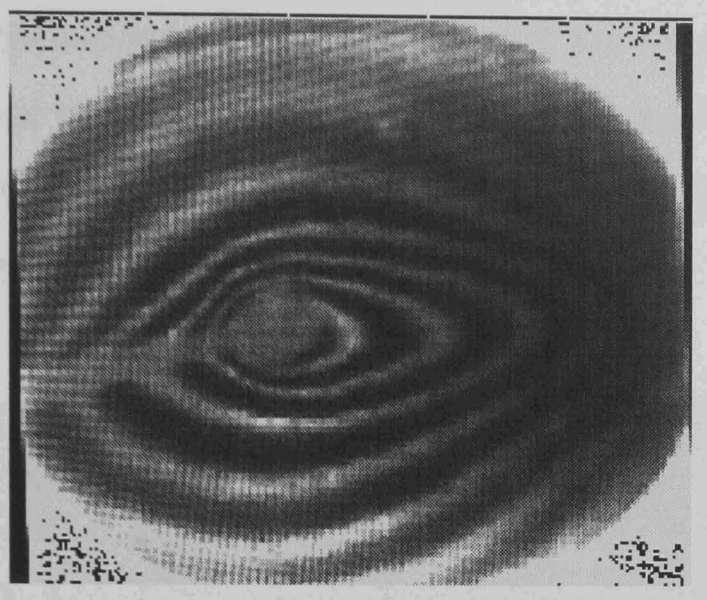


Figure 5.18

This first sample image shown in Figure 5.18, serves as a good demonstration of the principal stages in the image processing algorithm. The results of some of the principal stages of the algorithm are therefore presented for this image before the final results. In the case of the other sample images shown, only the image of the final fringes with the determined gravity vector plotted onto the fringe image is presented. Note that the magnitude of the gravity vector has not been calibrated although it equals the average movement of the fringe centres of gravity relative to the innermost fringe.

Figure 5.19 demonstrates the results of the roof and valley detection stage with the roof detections in white and the valley detections black. The feature at angle 150 (approximately) corresponds to the elongation of the fringes to the right of the fringe image. The fringe breaks at around 400 are the result of distortions due to the heater wire support which can be seen as a grey stripe to the left of the central grey circle which denotes the heater in Figure 5.18.

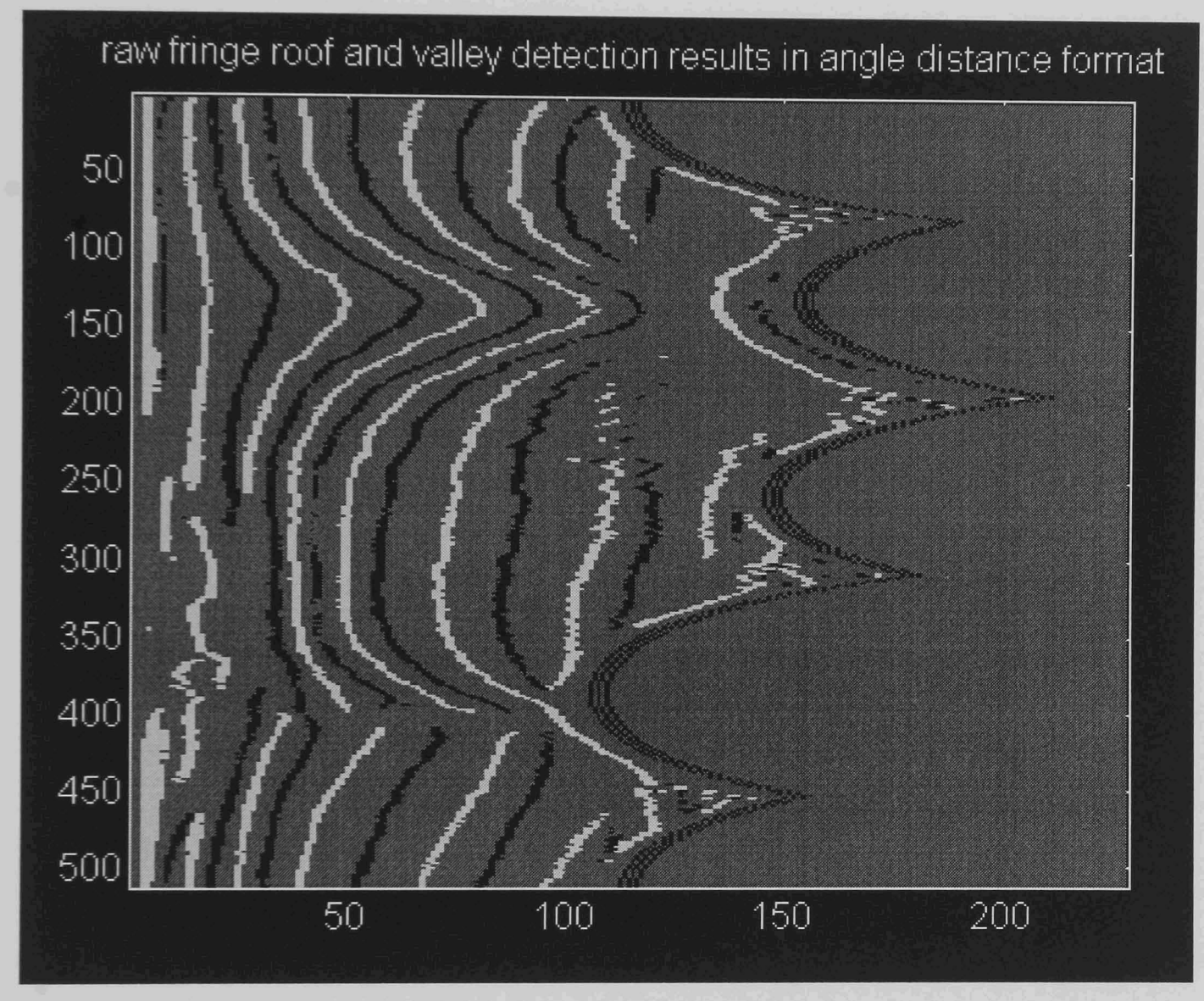


Figure 5.19

Figure 5.20 shows the detected fringe peaks after fringe completion using the iterative 'dilate and thin' stage and deletion of the valley detection points. It is particularly noticeable that the gaps in the fringes around angles 100 to 150 have been neatly filled in and that all the detections have been reduced to a line one pixel wide corresponding to the centre of the detected fringe.

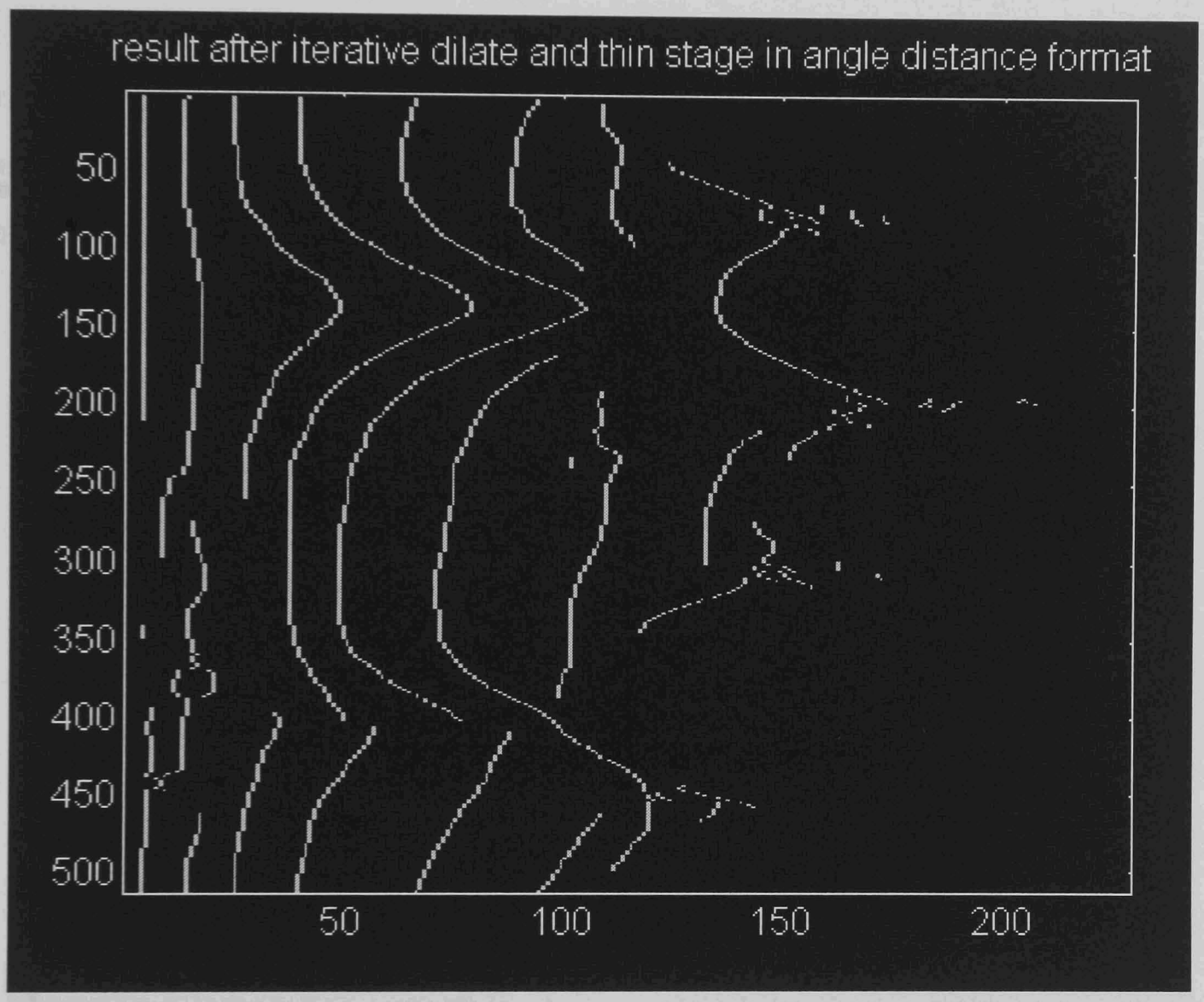


Figure 5.20

Figure 5.21 shows the linked fringes resulting from the application of the fringe finding stage to the tidied but ungrouped fringe points of Figure 5.20. These are shown plotted on the fringe image of 5.18 in black, contrasting with the bright fringe peaks. The very small fractional fringe sections seen in Figure 5.20 have been deleted as obvious noise. The farthest left of the detections crossing the heater support wire shows the use of the 'cutang' parameter (see parameters list at the beginning of this Section), this fringe being the farthest right at angle 400 in Figure 5.20 (as it is the furthest from the centre point). As can be seen by following the fringe around the image, this cut-out has prevented a link between a fringe detection and part of the image edge effects which have been labelled a fringe, from becoming an erroneous fringe detection in the final image.

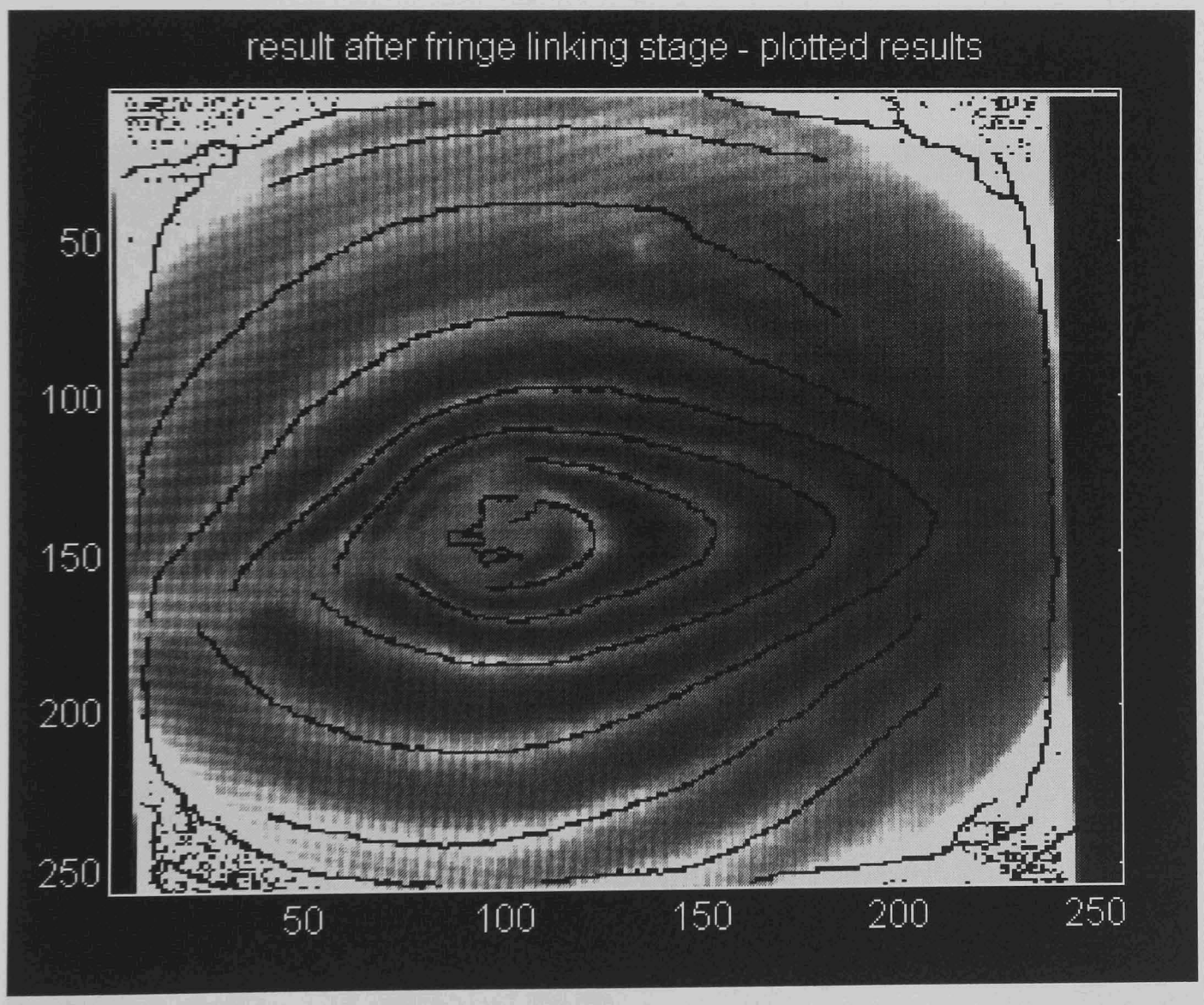


Figure 5.21

Figure 5.22 shows the accepted complete fringe detections. The remaining arcs from Figure 5.21 have been deleted in this case as the fringe completeness requirement was set to accept only fringes over 60% complete. This is low enough to allow for gaps but high enough to delete responses which might correspond to the detection of an image edge or an open arc in the image. The use of fringes which are over 50% complete for centre point estimation ensures that only arcs large enough to have their ends turning back towards each other are used. In practice, arcs which are over half a fringe in length are much more indicative of where the missing fringe section should be given a smooth fringe than is an open arc, where a fairly small change in shape corresponds to a large change in the likely radius of a closed curve including the section found. Shorter arcs could be used for interpolation but the accuracy will inevitably suffer.

Test Image 1

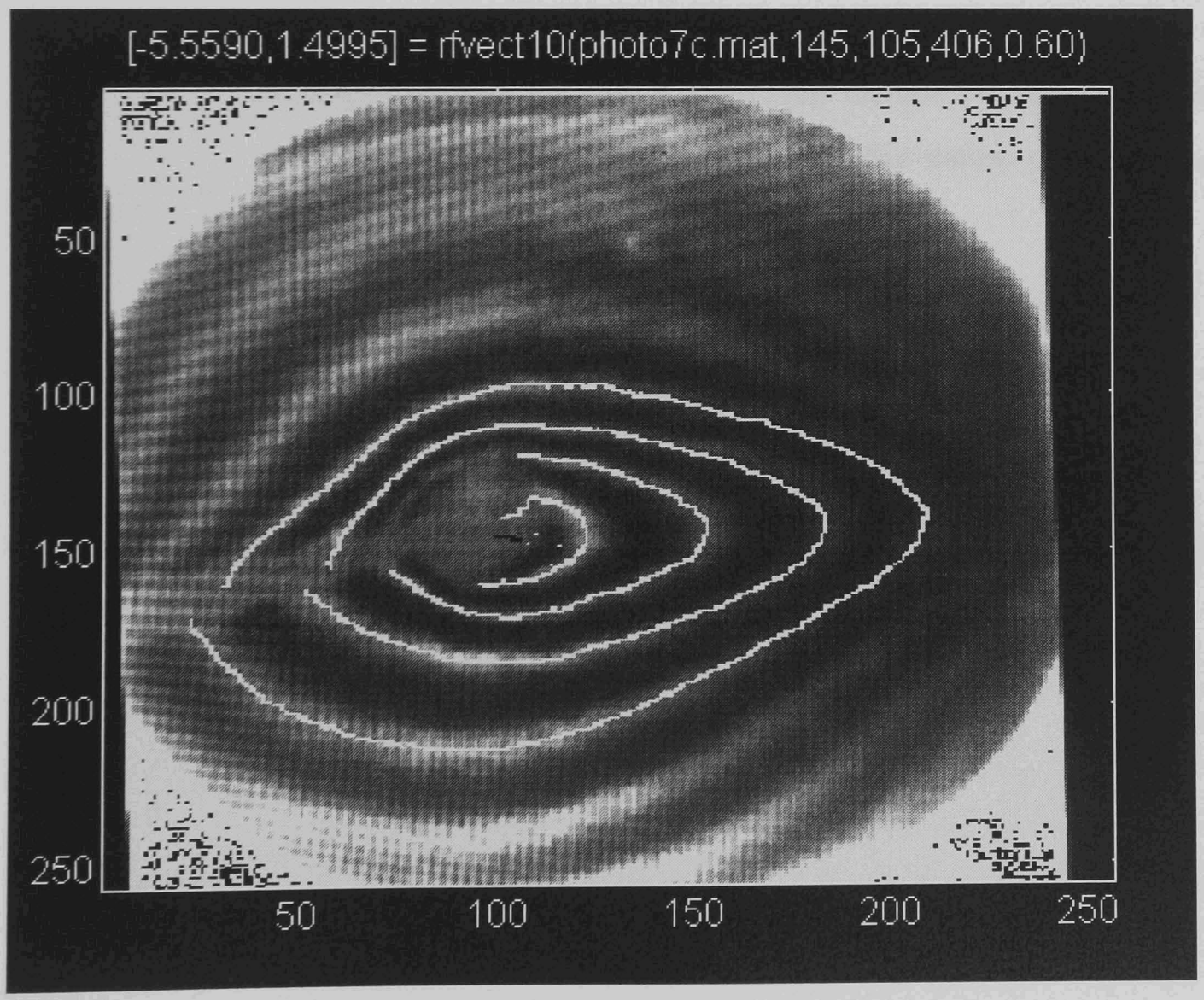


Figure 5.22

Image type : Low gravity convection. This image contains more distortion than anticipated in orbit but is a moderate midpoint between no convection and a typical earth convection pattern.

Starting co-ordinates : Row = 145 : Column = 105

Gravity vector : x component : -5.6 : y component : 1.5 (indicated by the black vector in the centre of the image)

Comments : This image has been used extensively because it contains four complete fringes and one nearly complete one. It has a very distinct heater support wire which as can be seen causes considerable distortion. This image also has very low contrast fringes, the section detected at the right of the image having a difference of approximately 3 grey levels out of 256 between the intensities of adjacent roofs and valleys. Finally, the elongated shape of the fringes in this image ensures that some sub-optimal fringe-to-slice crossing angles are seen. This demonstrates the successful application of the algorithm in conditions where the radial slices cannot be claimed to be taking prior knowledge of the concentric nature of the fringes overly in thier favour.

Test Image 2

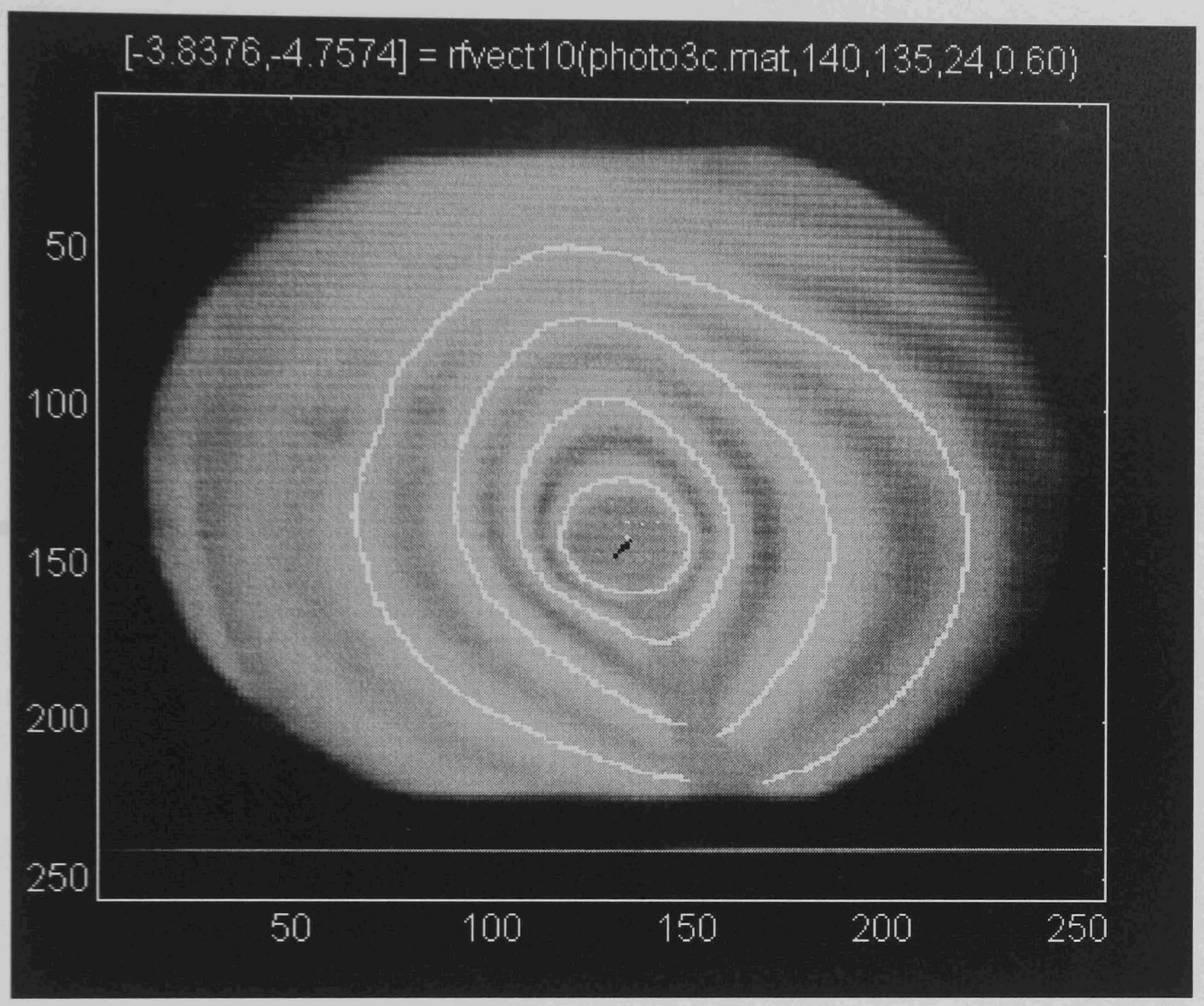


Figure 5.23

Image type : Simulated microgravity convection pattern.

Starting co-ordinates : Row = 140 : Column = 135

Gravity vector : x component : -3.8 : y component : -4.8

Comments : The grey line from the centre of the image towards the bottom of the image (x-axis) is the heater support. This image, containing four tidy and complete fringes and produces good fringe detections. The peaks of the partial fringe arcs in the image have been detected but the data has been disregarded as they are insufficiently complete for good estimation of the location of the centres of gravity of the feature on which they lie. The vector result corresponds to a gravitational force acting towards the bottom left corner of the image. It can be noted

that the outer fringes appear to be distorted to the top right, and this corresponds to a convective drift in that direction as the warmer medium rises.

Test Image 3

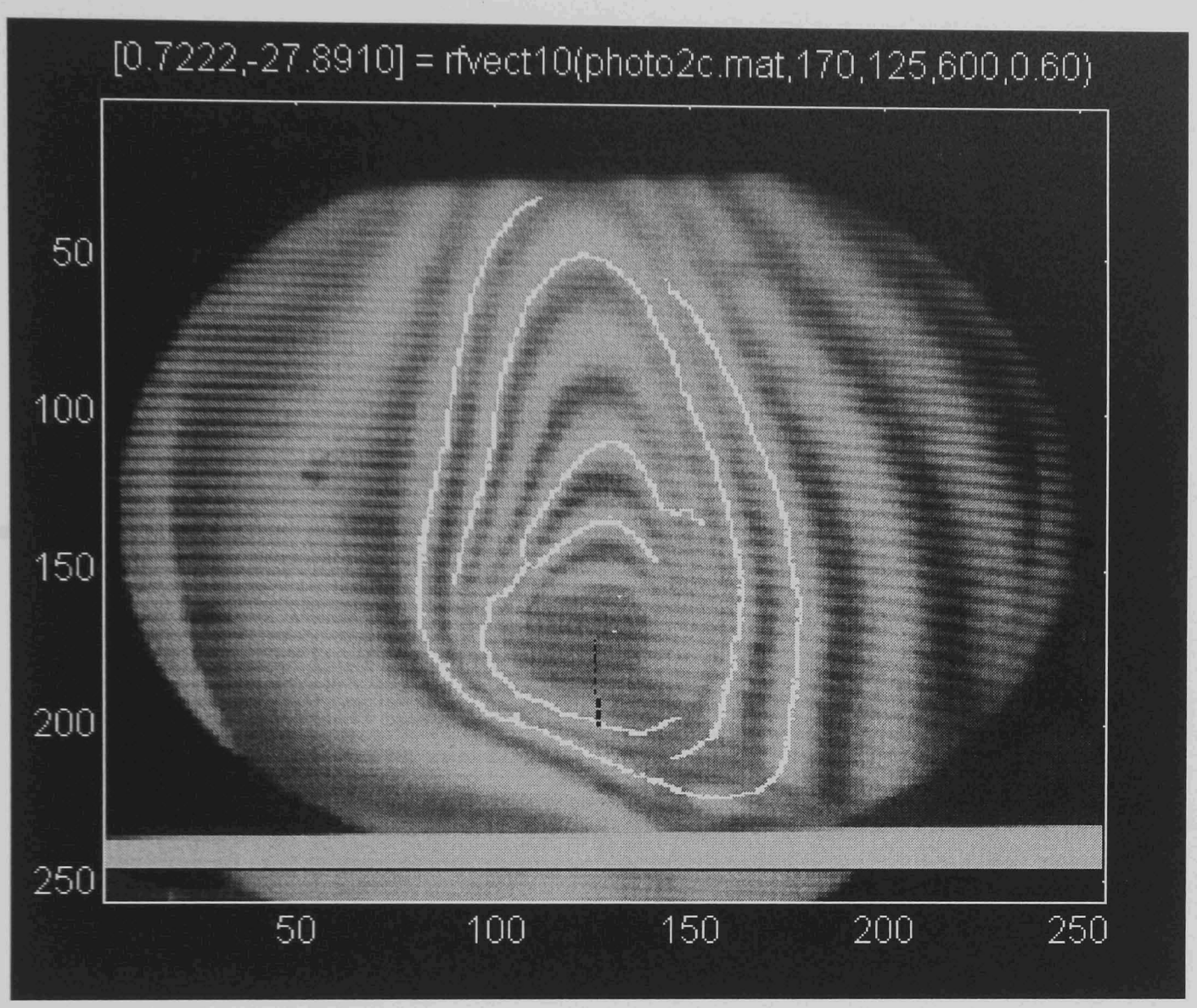


Figure 5.24

Image type : Distorted convection image with gravity..

Starting co-ordinates : Row = 170 : Column = 125

Gravity vector : x component : 0.7 : y component : -27.9

Comments : This image contains fringes that, while still showing an obvious movement towards the upper edge of the image, are quite distorted. It is much more apparent in this case that attempts to match analytic shapes to the fringes would have problems, especially given the somewhat sizeable gaps in the fringes. The heater support wire cut-out has not been used although the two grey streaks cutting across the fringes are obviously distortions. The algorithm has coped in this case. For general usage however, the 'cutang' parameter is either used or

not used for a particular physical arrangement of experimental equipment.

Test Image 4

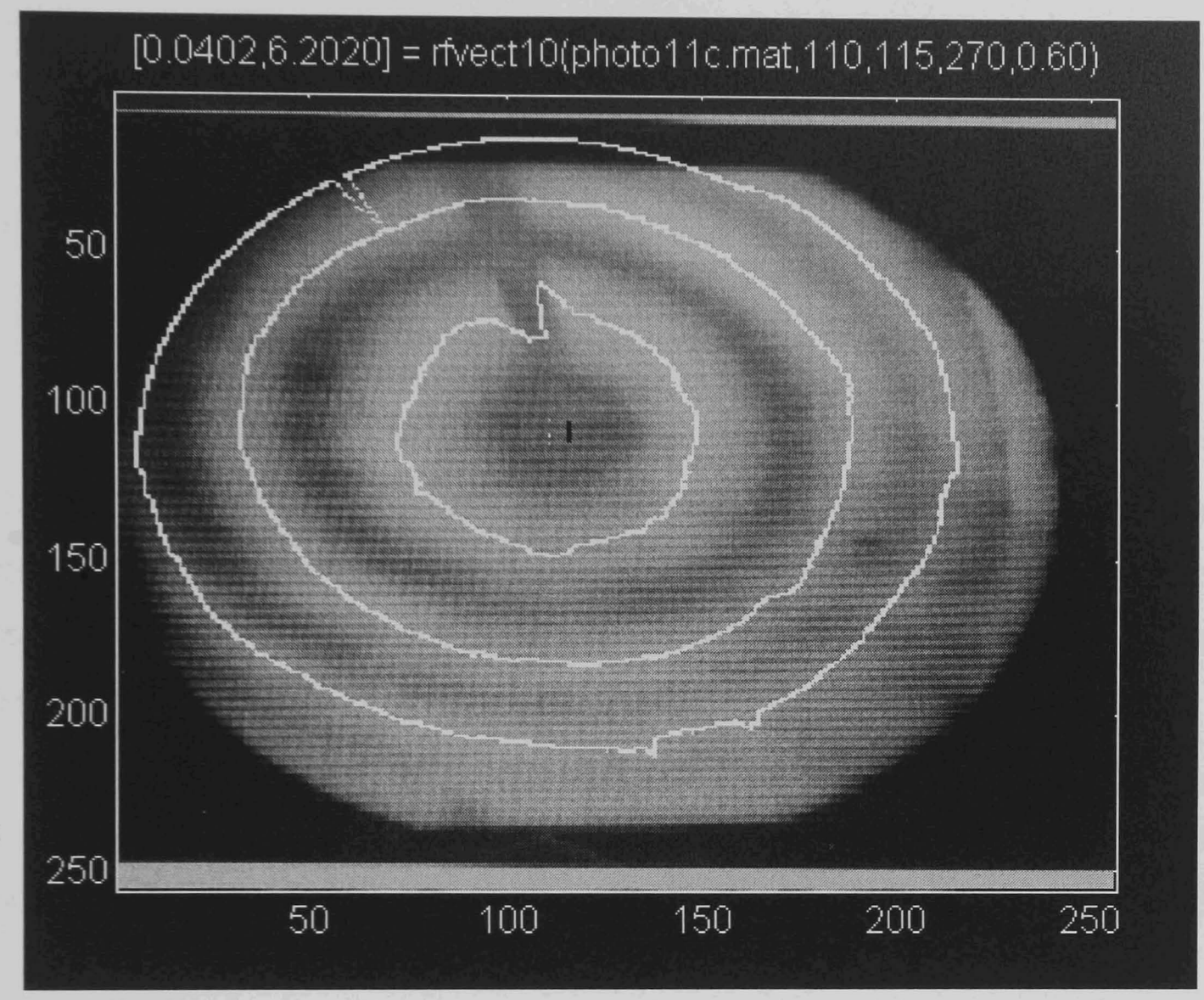


Figure 5.25

Image type : Simulated microgravity with virtually no gravity effects.

Starting co-ordinates : Row = 110 : Column = 115

Gravity vector : x component : 0.0 : y component : 6.2

Comments : This image is very similar to the test image in Figure 5.23, except that the fringes are much wider. This demonstrates the algorithm's ability to process with these wide, but quite uneven width, fringes. It is notable that the first interfringe gap is considerably wider than the second. This feature would make fringe detection by the fitting of sine curves to the fringe slices, (another possible fringe detection technique) considerably more error prone, as such techniques assume that the basic fringe shape is a smooth curve. The upper left of the outer

fringe plot in this case is interpolated by the gap filling algorithm and demonstrates how the missing section of a fringe is easily interpolated where a little over half of the fringe has been detected directly, if the fringe is regular in shape.

5.7.3 Conclusions

The results are generally satisfactory and superior to more typical edge detector results on similar images. Improvements over existing algorithms are noted in noise rejection, accurate localisation of the fringes and sensitivity of the detector in finding very low contrast fringes correctly, particularly in images with varying background intensities.

It should be noted that the variation in the settings of the 'start' and 'cutang' parameters between these images is due to the use of different experimental arrangements to simulate a variety of conditions. This is also the cause of the considerable variation in the amount of fringe distortion observed in the images.

Chapter 6

Safe Design And Construction Of The Flight Experiment

6.1 Overview of Payload Tasks

6.1.1 Payload Design

This chapter addresses the principal payload design issues, covering the effort involved in attempting to acquire real, as opposed to simulated, microgravity convection data. This was achieved through the flight of an experimental system in low earth orbit aboard NASA's Space Transport System (STS), or 'Space Shuttle' as it is more commonly known.

The first objective to be achieved in preparing the payload, as opposed to the NASA safety documentation which is discussed below, was the experiment design. The payload was designed for maximum safety and reliability, while enduring the rigours of the space launch and flight environment. This chapter will briefly address the major factors affecting the design of the payload, the required flight operations, the approaches used to overcome the problems, and most importantly why they were used.

6.1.2 Payload Safety Approval

In parallel with the design stages of the experiment, the project was required to progress through the formal NASA safety approval process. The safety documentation written as a result of this process, to demonstrate to NASA that the payload was safe to fly, is discussed in the latter part of this chapter. This was an iterative process, beginning at the conceptual stage of the experiment and expanding in detail as the payload design progressed. This not only allowed NASA to follow the development of the payload and identify safety concerns at an early stage, but

also provided feedback to the experimenters on the suitability of design changes made and indicated further requirements. This approach was beneficial as it identified potential problems before progress in the payload design process had advanced such that redesign would have become necessary. An introduction to the safety requirements is discussed first and then a more detailed description of the documentation stages involved and their effect on the project is presented.

6.2 Flight Plan

6.2.1 Experiment Operations Requirements

In order to acquire suitable data for analysis from the flight experiment, a continuous stream of images was preferred. However, this desire had to be balanced with the acquisition of data over as long a period of time as possible, so as to maximise the chances of recording changes in the images over any timescale. The experiment was also required to be operable with the available control inputs and had to complete its operations within the available video tape supplies and battery power reserves of the payload. To minimise disturbances from crew and equipment activity on the shuttle orbiter, it was also desirable that the experiment operate during crew sleep periods.

6.2.2 Orbital Operations Plan

The orbital operations plan is the sequence of experimental operations selected as the best compromise for acquiring experimental data within the constraints imposed upon the experiment. The experiment was operated twice during the flight, in each case activated by one of the two available NASA relay inputs to the payload, the first relay having to be allocated to overall payload power supply control. In order to maximise the balance between continuity of images and duration the first run was a short duration, (30 minutes) of continuous recording of images. During the second experiment however the recorder was cycled on for a twenty

second burst every two minutes to provide recording for a three hour experiment with only a further thirty minutes of tape. The detailed operating schedule including timing of the cell heater, which is not operated until ten minutes into the first experiment in order to record some quiescent space data, is shown in Figure 6.1 (Experimental Run #1) and Figure 6.2 (Experimental Run #2).

Experiment #1 Schedule

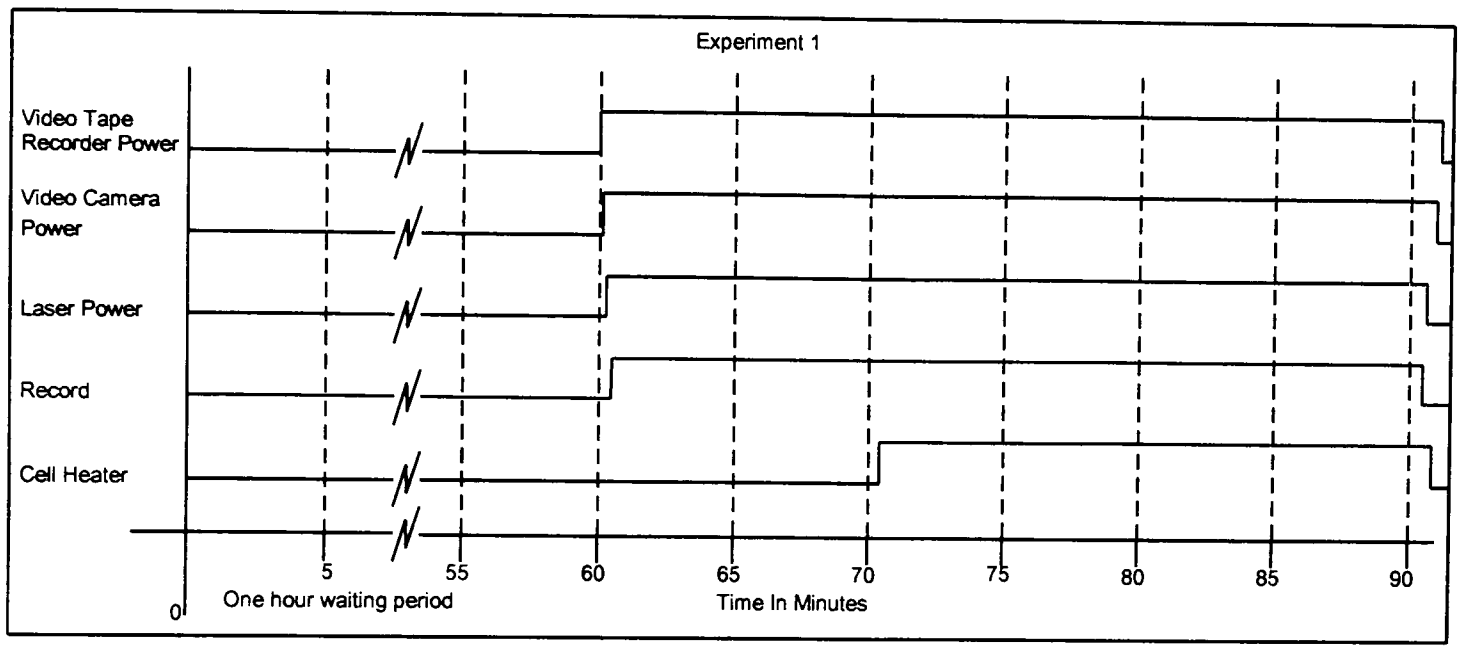


Figure 6.1

Experiment #2 Schedule

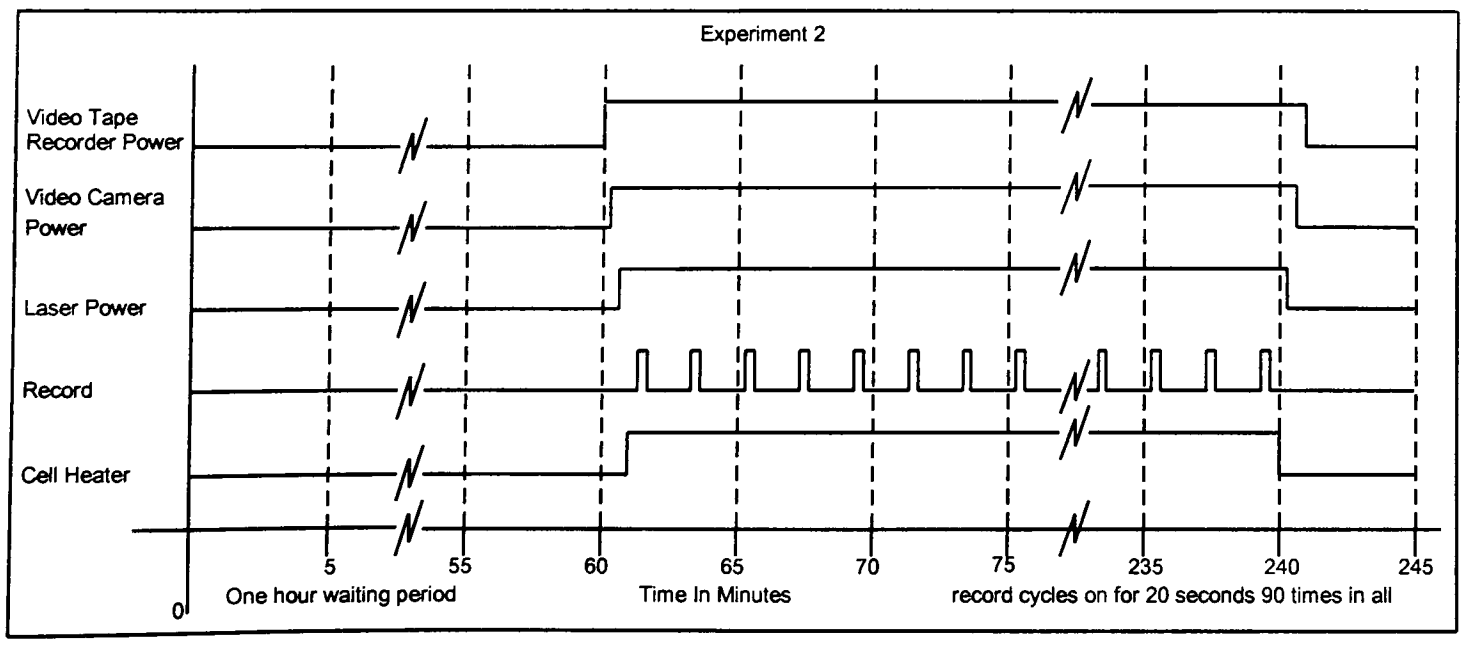


Figure 6.2

Apart from the main experimental operations, there were also data logging operations to be scheduled, and the operation of the environmental control system. In order to ensure that the experiment did not cool down too much between launch and the first available astronaut activation time, the main power supplies were activated using NASA's baroswitch option. This is a pressure sensitive switch that closes during ascent, and it was used as a means of commencing the operation of the environmental control and data logging systems. This was convenient, as the baroswitch operates the highest current handling of the three NASA relays. Therefore experimental requirements for early environmental control activation, and NASA's need to have total control of power supplies, could be accomplished using only one of the three available relay inputs. The environmental control system and data logging operated from baroswitch activation until the astronauts shut the payload down prior to re-entry. As a safety precaution, had low battery voltages been detected prior to re-entry, the heating units would have been shut down to avoid draining the batteries completely.

6.3 Operating Environment Of The Experiment

6.3.1 Thermal Environment

The first consideration in the design of the payload, after the overall primary safety considerations, was the environment within which the experimental package was required to operate, and the demands this placed on the payload. Some of the design decisions influenced by these environmental and operational requirements are examined and the design solutions and processes used to overcome them are summarised.

The Earth orbit environment within which the STS flies can be very harsh. The temperature on the outside of the GAS canister in which the experiment was housed can reach as low as -160 °C or as high as +100 °C, depending on whether the cargo bay of the Shuttle is facing deep space or pointing at the sun. Insulation blankets on the GAS canister reduce the extremes to a more manageable -100°C to

+40 °C in the worst case. The mass of the canister also gives it considerable thermal inertia. The minimum temperature of -100°C does not cause serious problems because the insulation on the GAS canister means that relatively little power is required to achieve a more acceptable minimum temperature. Despite the insulation, and the fact that the extreme temperatures are only generated during unusual pointing attitudes of the Shuttle Orbiter, the experiment was required to be designed to cope with the worst case possibilities to ensure that it was safe. The design of the thermal control system which was designed to meet the requirements of keeping the payload warm in the cold case, while not overheating in the worst case hot scenario during a total thermal runaway failure of all energy dissipation inhibits is discussed in Section 6.4.4. This latter scenario is that required to be analysed by NASA for the Safety Data Package, (Section 6.4). This is a typical example of how safety is maximised in the GAS programme while maintaining simplicity, in this case by avoiding a full reliability analysis by assuming that all the inhibits fail however unrealistic that might be in practice.

6.3.2 Mechanical Vibration And Shock Loads

The Shuttle launch operation causes considerable vibration during firing of the orbiter engines and solid rocket boosters. This has resulted in a design specification from NASA, [151], requiring that GAS payloads survive vibration loadings of ten times Earth's gravity in each of the three co-ordinate axis simultaneously. The payload's main structure is required to stay mechanically intact and must not deform such that it impacts the sides of the GAS canister.

Further to the requirements associated with vibration loadings, the payload structure is required to withstand the crash landing design load of the Shuttle Orbiter on return to Earth. This is defined, [151] as 10g's in the vertical axis, and has a required safety factor of 1.5 if verified by static testing or 2.0 if verified by analysis of the structure without actual testing.

From an experimental success point of view, not only must the experiment survive this environment, but it must still be functional afterwards in order to perform its operations once it reaches orbit. The interferometer presented a particular

problem as it must be initially set-up to an accuracy of a fraction of the wavelength of light. It was required to resist deformation between being placed in the canister and arriving in orbit.

6.3.3 Materials Selection

For a space experiment NASA must approve all the materials that are to be contained in the enclosed environment of the GAS canister. The potential risk of ground personnel being exposed to dangerous substances during integration with the NASA carrier hardware was also required to be addressed. Materials that could cause contamination, corrosion or pose fire damage risks to the flight equipment during experiment operations or failures either had to be avoided or their safe containment assured. This placed particular limitations on the use of flammable materials, or any material with a tendency to out-gas harmful substances, both common concerns when using many standard materials.

6.3.4 Reliability Requirements

Reliability is a common concern for experimental equipment, though typically experiments are carried out in an environment where the experimenter can gain access to perform repairs should something fail. This is not the case when the experiment is in a sealed canister over 100 miles above the Earth however. Reliability was therefore a vital consideration, it would have been disastrous if a failure had occurred that could have been foreseen and prevented by better design. The last experimenter access to the payload was approximately two months before flight and there was no means of monitoring the behaviour of the payload in real time during flight, or attaining access to its systems to resolve problems. It was vital therefore that critical items were designed for maximum reliability and that where possible they were redundant or otherwise protected against failures.

6.3.5 Control And Communication

There is no communication capability with GAS experiments during flight. Only three NASA relay inputs are provided, one of which is required for experiment power control. This meant that all the required experimental operations had to be selectable by the remaining two inputs, or based on a pre-set sequence from a relay change event. Similarly all the results generated by the payload were required to be stored onboard, for recovery and analysis after the payload had returned to Earth. This required that simple operations of the GAS control relays be turned into complex sequences of experiment commands by a primary experiment control circuit. This area was particularly influenced by reliability considerations.

6.4 Payload Design

6.4.1 System Overview

The payload design process was significantly more complex than that of a typical Earth-based project due to the requirements of the Space Shuttle launch, flight and Earth-return environment, as discussed above. The most important factor in this respect was the safe functioning of the payload throughout the flight. Figure 6.3 shows the block diagram of the experiment as designed, the systems being subdivided into blocks of circuitry, each performing a single task and each unit being removable for testing.

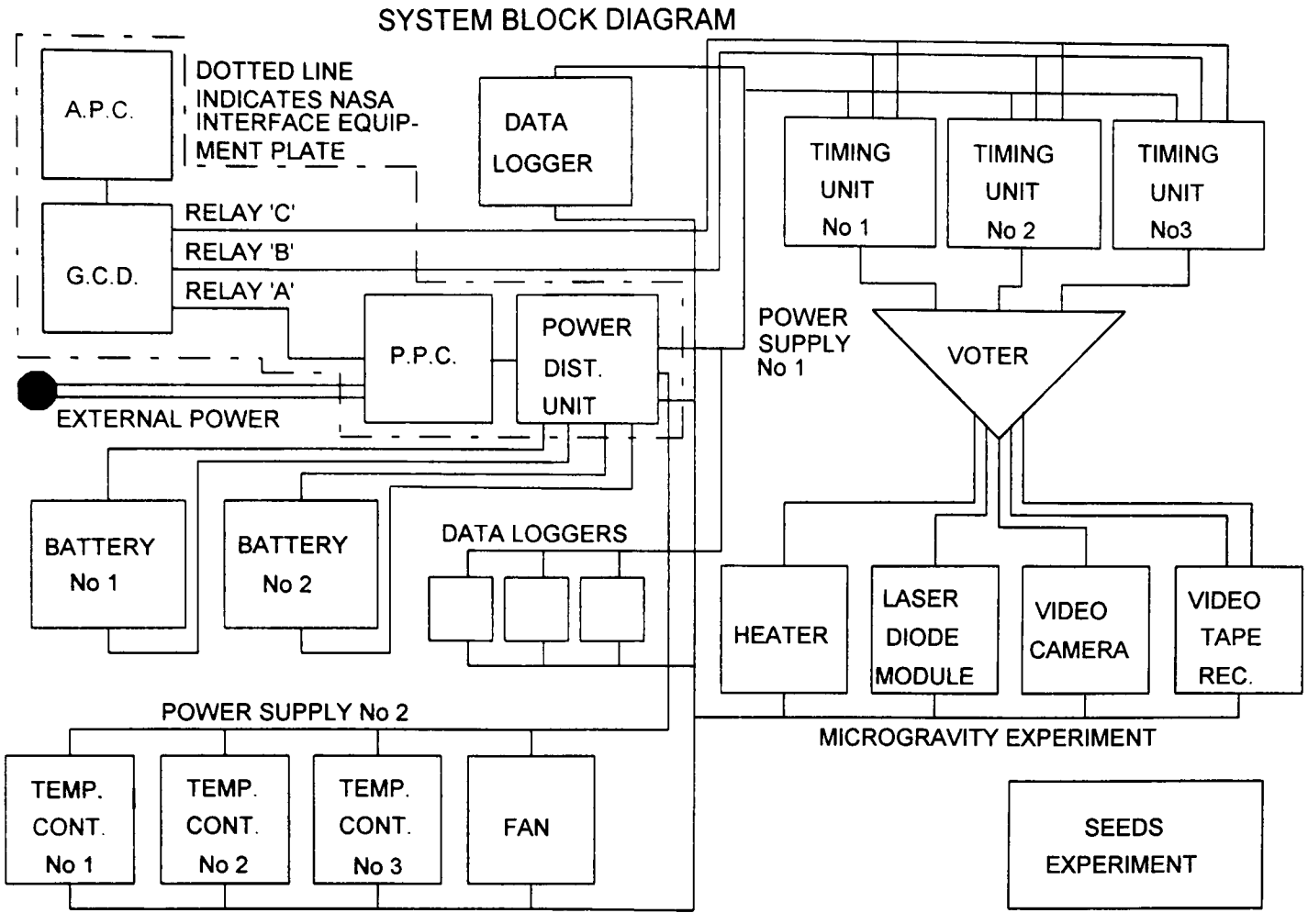


Figure 6.3

A.P.C. : Autonomous Payload Controller

In the orbiter crew compartment, receives commands from the astronauts.

G.C.D. : GAS Control Decoder

Interface between NASA GAS command system and the experimenter's hardware. This is located on the interface equipment plate in the bottom of the GAS canister.

P.P.C. : Payload Power Contactor

Main power control unit. It has a fault shutdown input and also controls GCD relay 'A' which acts as the power distribution unit controlling power during flight and switching between flight and external supplies during ground test.

6.4.2 Mechanical Structure

The main structure had to meet important design criteria. It was required to survive the Shuttle launch at 10g's random vibration in each axis and a crash landing load of 10g's with a safety factor of 2.0. It was also required to meet the experiment mounting requirements, provide easy access to the experiment subsections, and be within the specified weight limit.

In order to meet the launch environment requirements, the material selected to construct the payload chassis had to be chosen from NASA's approved list of stress corrosion cracking resistant materials to ensure that the repeated strains from vibration did not lead to eventual fatigue failure. Considering past usage of aluminium in GAS experiments, which indicated its suitability, acceptability to NASA and its high strength-to-weight ratio, aluminium was selected as the most suitable material for the main structure. An alloy equivalent to American 6000 series aluminium was selected as this is approved by NASA as stress corrosion cracking resistant. The alloy also had high strength thereby maximising the strength-to-weight ratio and structural safety factors.

A 'shelves and struts' style structure was chosen for the payload as this was considered to be the optimum design to meet the equipment mounting requirements. This structure is also easy to assemble and dissemble into sections as 'I' type struts were used between individual shelves. The 'I' struts were designed to give multiple attachment points between the different components, thereby avoiding weak links at the joints and spreading the loads on the plates while keeping the struts themselves no larger than required. Easy access to bolt heads for assembly and disassembly of the payload was also provided. Figure 6.4 shows the completed structural assembly.

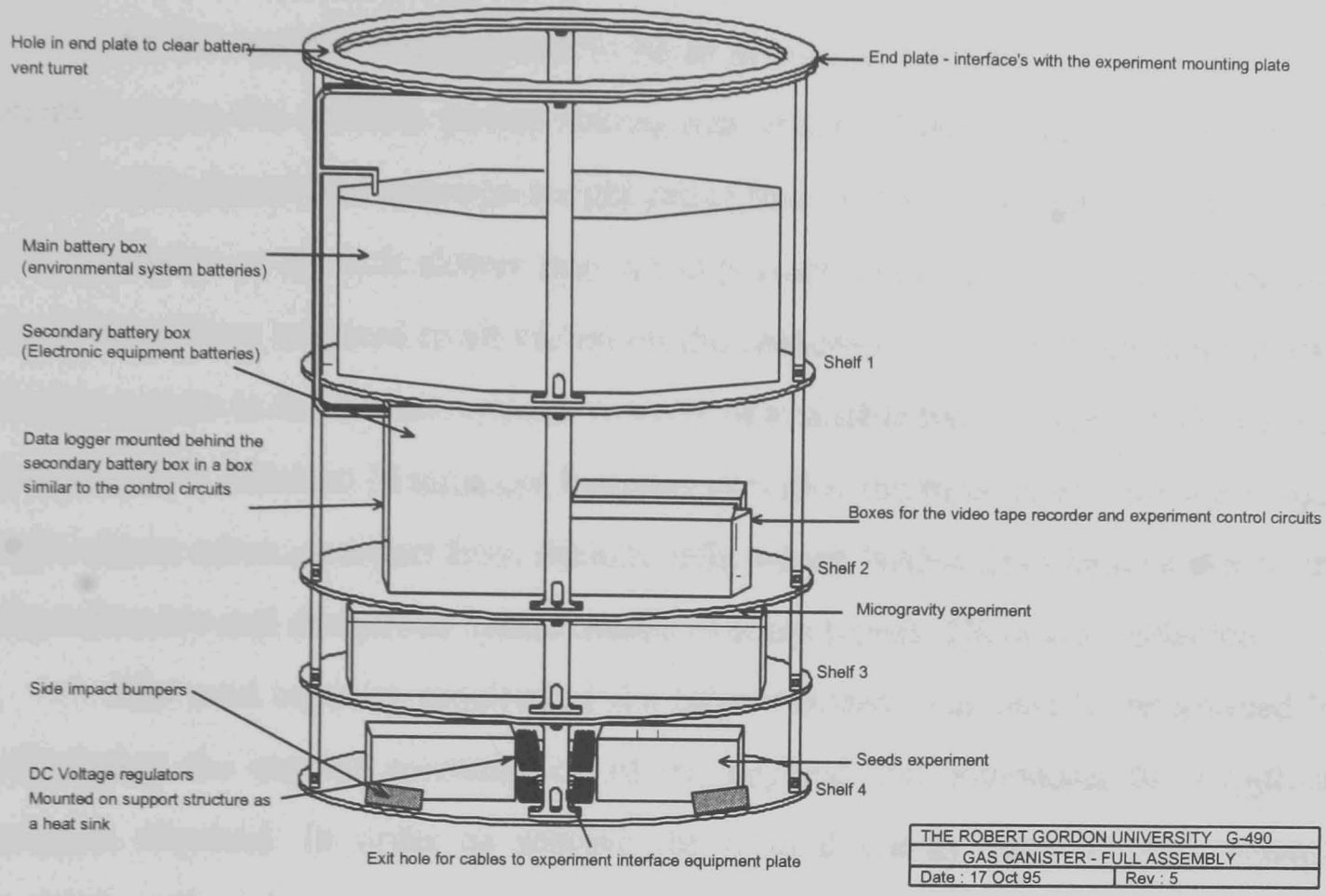


Figure 6.4

To demonstrate to NASA that the structure met their safety requirements, a structural analysis was carried out [152]. This covered both the ultimate strength, compared to the structural loadings specified and a natural frequency calculation to demonstrate that the natural frequency was high enough to avoid inducing resonance due to vibrations generated by the carrier system.

6.4.3 Power Supply System

The payload power supply system was required to provide all the electrical power needs of the payload during the mission. There is no external power available to GAS payloads, so both the electronics and the environmental control system had to be supplied with power. Further restrictions existed because of NASA's safety requirements, requiring the system to be fail-safe and be adequately protected against

potential faults. The design also had to take into consideration operation within safe limits in the microgravity environment of Earth orbit.

As the system was required to be as simple as possible and only needed to operate once, the obvious power source was primary battery cells. These provide much better power and space-to-weight ratios than rechargeable batteries. They also self discharge at a much slower rate, an important consideration for a system that could have been required to sit sealed on the launchpad for over three months after the last access to the battery system. A study of available battery types [159] showed that Duracell Alkaline Manganese batteries provided the most power for the weight and volume taken up (apart from lithium cells, which NASA have banned due to the unpredictable and dangerous failure modes of some types). These were selected.

The total capacity required of the battery system was readily determined by calculating the current consumption of the payload and estimating the length of mission required. In order to achieve the desired energy capacity and voltage, multiple cells strings were used in series, blocking diodes being utilised to prevent the charging of weak strings by stronger ones. Individual string fusing was utilised to overcome the fact that the overall battery peak output current exceeded the rating for a fuse that guaranteed protection of individual strings from short circuits. A further overall fuse was used to limit the risk of a 'smart' short failing to blow the individual fuses until the total current considerably exceeded nominal maximum power. A further complication was that NASA required that, for safety reasons, the batteries did not completely discharge during the mission assuming nominal experiment operation. This required low voltage cut-outs to be fitted to the heating system so that on a cold mission, the heaters would have disengaged (having operated long enough for experiment success purposes) while there was still sufficient power to keep the fan running until power shutdown by NASA on return to Earth.

6.4.4 Environmental Control System

The Environmental Control System provided and distributed heat around the payload in order to maintain the payload electronics at a nominal 20°C to ensure their continued function during the experiment. The system consisted of three

heating strings comprised of power resistors controlled by precision thermostats. These were bolted onto the experiment shelves in order to take advantage of the thermal conductivity of the aluminium to dissipate the heat. The system also had a stirrer fan to even out the temperature distribution around the payload by providing forced air circulation in the absence of convective cooling.

The principal problem with the environmental control system was balancing the requirements of keeping the payload warm in the worst cold case, for experimental success purposes, and the NASA requirement to keep the maximum temperature reached in the worst case failure of the payload under control for safety reasons. NASA take the worst case of the Shuttle Orbiter positioned continuously in the hot thermal attitude, with all thermal inhibits failed so the heating system and all power dissipating units in the experiment operate at full power until the batteries are depleted. This is unrealistic but is based on the policy that if everything is acceptable in this scenario, NASA does not have to concern itself with the reliability of any of the payloads over-temperature inhibits or control systems while assessing the safety of the payload. In order to confirm that the payload conformed to the requirements an 'Energy Containment Analysis' (See Appendix 4) was performed and the results submitted to NASA. To simulate the results of a total failure of the payloads thermal inhibits occurring, the system was modelled using a computer mathematics package (Matlab) to implement a time-stepping thermal simulation. The simulation, see Appendix 4, treats each of the major thermal masses in the payload independently and therefore gives a detailed estimate of the result of such a failure. This simulation was run under the worst case conditions, namely, the Shuttle Orbiter cargo bay facing the sun and all inhibits failed with the conclusion that the maximum temperature that could be attained by the payload was within the safe operating temperature range of all it components. The thermal simulation programme was designed so that parameters such as the power of the heating system and the thermal sink temperatures could be altered. This enabled many different simulations to be run in different conditions during development, assisting general payload thermal design in addition to proving safety in the worst case. The thermal regulation of the system by the thermostats was also modelled, but with the temperature set at several hundred degrees to simulate failure for worst-case results. This enabled a single

parameter change to switch the system to providing a test of a particular set-up in the cold case to check if it met the experiment success requirements while remaining safe. This simulation system was an invaluable tool in trying out different combinations of conditions to find a system that satisfied both NASA safety and the experiment success requirements.

6.4.5 Experiment Control Circuit

The Experiment Control Circuit was required to transform the NASA relay signals used for command inputs into the two experiment operating sequences as described in Section 6.2. The basic options available for an approach to developing the control system were a commercial sequencing system, a microcontroller, a custom controller or a microprocessor system implementing the operations in software.

Typical commercial sequencing systems have three major drawbacks. The majority are mains operated, bulky and expensive. However they are reliable and easy to use. This was the first and easiest option to discard as bulk and voltage requirements were both guaranteed to be outside acceptable limits.

A microcontroller or microprocessor system has the advantage of being simple to set up and programme. However the flexibility offered by software was not required, as no feedback was required from the equipment under control and the operation sequence is deterministic. Other problems are complexity and the resulting power consumption of a typical microprocessor system. Although these are fairly reasonable by normal standards, when maximum reliability in a hostile environment is considered, minimum component count is a priority. Likewise minimal power consumption is essential.

The final option, that of a custom hardware system using programmable logic devices has the drawback that the circuit has to be custom designed in hardware. However this approach has advantages. One of the most important advantages over a microprocessor system that no software is present to obscure undetected errors. This solution also requires minimal power, as no endless looping need occur in the idle condition, a principal problem from a power consumption viewpoint with a typical

microprocessor system. This was the option selected for development, aided by the availability of programmable logic chips to keep the chip count low. The operation sequence was contained in a pre-programmed EPROM.

The final controller design consisted of three triply-modularly-redundant control circuits. Each of these comprised a microprocessor crystal with a multistage divider circuit. These outputs then drove PLD's each containing a long divider chain to further reduce the clock speed and finally produce a set of outputs that formed the addresses for the EPROM containing the operating sequence for the payload programmed as a pattern of bits. An end stop prevented address wrap around. Experiment selection circuits generated outputs enabling Experiment #1 in response to NASA GCD Relay 'B' and Experiment #2 in response to Relay 'C'. The circuit diagrams are in Appendix 1.

6.4.6 Microgravity Experiment Block

The microgravity experiment itself was the central part of the payload. The basic requirement was to implement a Michelson interferometer in the space available on one of the experiment shelves. This interferometer had to be rugged enough to survive the launch of the payload with the primary optical arrangement intact, while still having reasonable mass and being as simple to construct as possible. Figure 6.5 shows the mechanical arrangement of the interferometer.

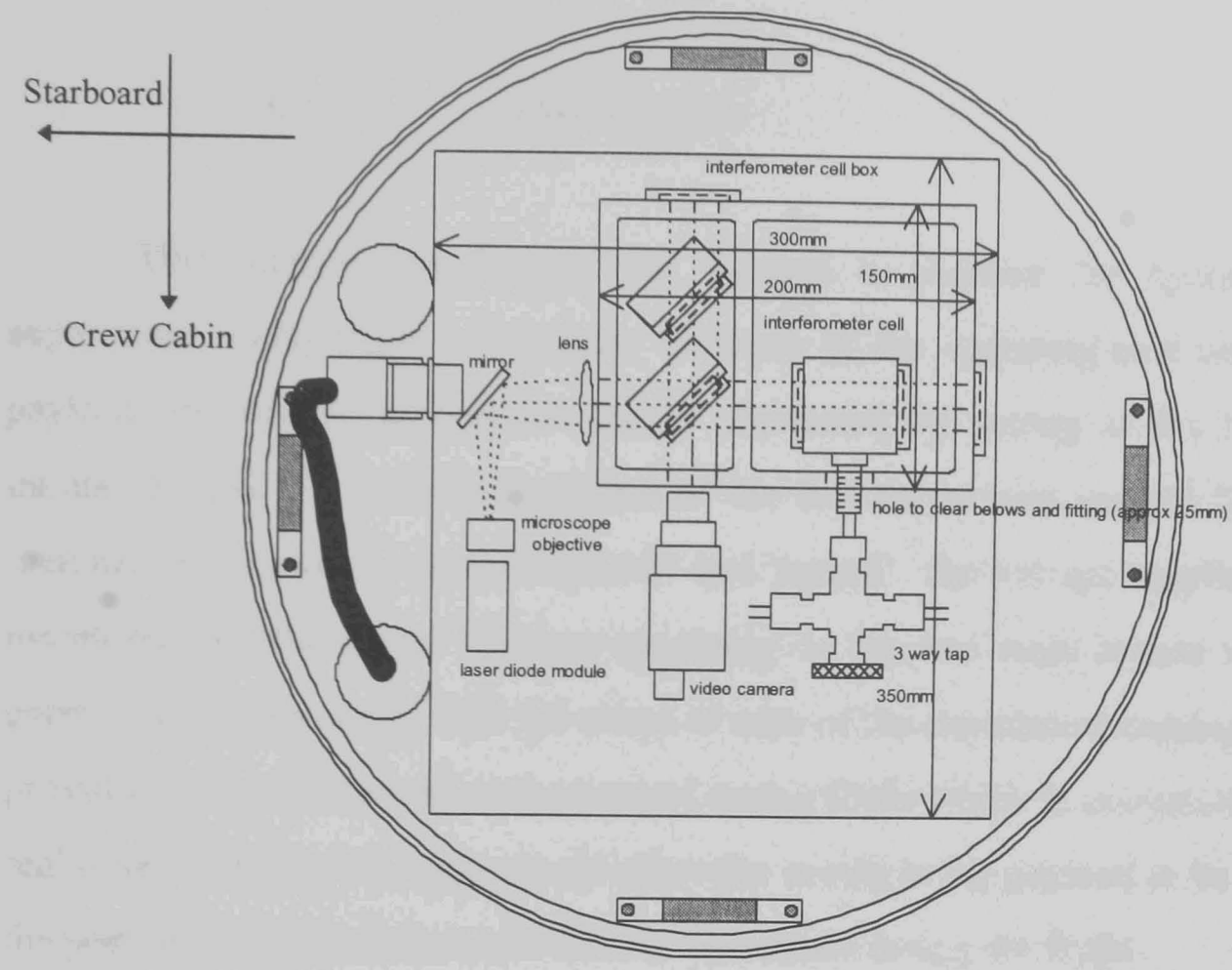


Figure 6.5

In order to provide sufficient stiffness to the main optical beam paths from the beam-splitter to the two mirrors of the interferometer, the main block of the interferometer was milled from a solid block of aluminium. The fringe patterns were generated with the experiment cell containing the heated fluid in one 'arm' of the interferometer and the undistorted reference beam in the second 'arm'. This unit was attached to the shelf by means of a three-point mounting, and the auxiliary interferometer components were arranged around it. The accuracy of the positioning of the laser module and camera was important but not as critical as the stability of the primary optical components in the milled cell block. The first auxiliary unit, the laser source, consisted of a laser diode module and beam spreading arrangement that provided a wide collimated beam of laser light. The video camera was mounted in line with the output of the unit, and a three way tap (which allowed for vacuum induced filling of the experiment cell with the silicone oil experimental fluid and its sealing for flight) was provided. Adjustment systems based on three-point mounting

units were provided for all of the optical components to allow fine adjustment of the system after assembly.

6.4.7 Data Loggers

The primary data logger was required to monitor the operation of the experiment during flight and provide feedback on the operating environment of the payload. In order to accomplish this, it monitored the setting of the NASA relay inputs 'B' and 'C', and the condition of the five experiment control lines, 'laser', 'cell heater', 'camera', 'recorder power' and 'record'. The voltage supplies were also monitored and the actual voltages appearing on the two main busses were logged every five minutes along with the status of each of the experiment heating units. This provided a record of events that occurred during flight which in conjunction with the real time clock on the logger would allow the events in the payload to be matched to the postflight NASA data on the Shuttle operations during the flight.

A number of miniature data loggers were positioned in the payload to provide a temperature profile record for the payload during the flight and to log data on the conditions in the seeds experiment. These loggers were split into two groups, each intended to carry out a specific task. Small commercial single channel data loggers were selected for this task, being simple to use, small and having low current requirements.

The seeds box contained one temperature logger and one humidity logger which served to monitor the seeds environment so that the seeds experimenters had a complete record of the environmental conditions in the seeds box. The seeds experiment basically consisted of a compartmentalised box. The box was configured as a single airspace so that humidity and temperature were the same throughout as far as possible.

The main temperature logging system was required in order to monitor the actual temperature on the four main experiment shelves in order to confirm the actual temperature profile of the mission. This was not only to prove the correct operation of the environmental control system, but also to allow a check to be made that all the systems were actually operating within their design temperatures. In addition, the

flight results were to be compared with the thermal model of the experiment after the flight.

6.4.8 Payload Modularity

The payload system was designed to be modular, with each of the main sections being a module that could have been relatively easily removed from the overall assembly. There were two main reasons for this. Firstly it was necessary to provide facilities for the easy disassembly of the completed payload both for inspection by NASA and for experimenter access during testing. The second major advantage of modularity was that, should a failure have been detected during ground testing of the payload, it would have been straight forward to switch the faulty module for a replacement before flight. In the case of the environmental battery box and the interferometer unit, the modules were in effect complete shelves. The wiring between the units was grouped in wiring looms so that it could be unplugged and removed from the payload enabling access.

6.5 NASA Documentation And Liaison

6.5.1 Safety Documents

This section undertakes to explain the safety requirements NASA enforced for the experimental equipment that formed the flight experiment, along with the NASA safety review process which was followed in parallel with the design and construction of the payload. As safety is so central to the space flight, this process was a considerable task in the preparing of the flight experiment. The primary result was the preparation of the 'Payload Accommodations Requirements' (PAR) [153] and the 'Safety Data Package' (SDP) [154] documents for approval by NASA.

6.5.2 NASA Safety Requirements

The GAS programme is unusual amongst the NASA space programmes in that NASA are not themselves primarily concerned about the success or otherwise of experiments in achieving their flight objectives. NASA's only concern is that the experiment should be safe and that it poses no hazards to the STS system. This includes the Shuttle Orbiter itself, as well as their ground facilities, GAS canister and associated interface electronics, and most importantly their personnel, both astronauts and ground staff. It is also a requirement that the payload does not interfere with any other payloads on the Shuttle through for example excessive EMI emissions.

The first line of defence against erroneous payloads is the GAS canister itself. This has been designed and approved by NASA to contain a maximum weight payload even if catastrophic failure of the mechanical structure occurs, thereby ensuring that even in the worst case the Shuttle Orbiter is protected from the payload. NASA however prefer that this is not required, as it could at the least lead to the destruction of the interface equipment plate components mounted directly below the experiment structure, if not damage to the canister. Measures must therefore be taken and detailed in the SDP to ensure for all foreseeable failures that, either the experiment will not fail, or it will contain failures safely without requiring the GAS canister to back it up. The experimenter is therefore required to prove that the design is safe and that the primary structure is strong enough to withstand the launch and landing loads, and not release objects that could damage the canister. This process is primarily carried out by document reviews, and NASA inspecting the hardware when it is delivered to Kennedy Space Centre for integration. The hardware is considered acceptable if it is as described in the accepted final version of the Safety Data Package.

6.5.3 Introduction to Safety Documents

Safety approval of the payload for flight is given by NASA on the basis of a number of documents submitted to them. The first document required, the 'Payload

Accommodation Requirements' (PAR) document, established the basic design of the payload, the flight plan, the astronaut interaction with the experiment and a preliminary hazard analysis. With this document completed and baselined, work progressed to the 'Safety Data Package' (SDP). The SDP is nominally submitted in three stages, Preliminary, Final and Phase III. In practice, the process becomes iterative as NASA accept that the experimenter has reached each stage. The role of this document is to provide a description of the payload such that it can be assessed for safety without NASA engineers requiring to visually inspect the flight hardware prior to final integration.

6.6 Payload Accommodations Requirements

6.6.1 Synopsis

The PAR document was submitted to NASA at an early stage in the experimental design. It basically explained to NASA the objectives of the experiment. It contained a brief description of the proposed experiment and its operating schedule, and the resources required.

6.6.2 Defining The Payload

The details of the control of the experiment by NASA were specified in the PAR. The NASA control sequence was a list of the six required operations of the three GAS control relays by the astronauts. NASA also required to know in the PAR about any known hazards in the payload which were obvious from the nature of the proposed experimental package. At the PAR stage, this consisted of a requirement to carry out an initial failure mode hazard analysis. In many cases, these hazard possibilities disappeared at a later date (for example several electrical hazards were shown in the PAR for this experiment which later did not result in explicit hazard reports, as the danger had been avoided by following the NASA-supplied guidelines for the wiring of the payload).

This stage resulted in a basic definition of the payload, giving NASA an overview of the experiment and the resources that were going to be required in order for it to achieve its objectives. The PAR also indicated any obvious areas of concern from a safety perspective. Finally the PAR document agreed between the experimenter and NASA what would constitute the flight experiment. This covered both the details of the experiment, and its utilisation of GAS programme facilities, and confirmed acceptances of the basic foundations of the flight while the experimenters design work was still at an early stage and changes could have been be made relatively easily.

6.7 Safety Data Package

6.7.1 Synopsis

The 'Safety Data Package' or SDP [154] is the primary safety document which was regularly reviewed by NASA during the design stages to give clearance for flight. This document described the construction of the payload, its subsystems and the nature of the potential hazards posed by the experimental equipment. The SDP demonstrated through hazard analyses that suitable measures had been taken to prevent potential hazards becoming real. The final Phase Three Safety Data Package formed the basis of an internal NASA safety approval request from the GAS programme to the STS programme for permission to fly the payload aboard the Space Shuttle.

6.7.2 Flight Description

The first section of the SDP document replicated the format of the PAR, updated to include relevant design revisions. Having set the scene, the SDP moved on to describe the payload in detail starting with the overall mechanical structure. This posed the first of many problems for the payload as it was required to be strong enough to support the 90Kg payload in the Shuttle launch and return environment.

This mechanical description was basically a detailed description of the structure as introduced in Section 6.4.2.

Having dealt with the principal structural components the SDP then set out the configuration of the experiment and addressed the individual payload subsections. The various parts of the actual experimental equipment were housed in boxes mounted on the shelves. This compartmentalisation served several purposes, in that the individual units could be removed and replaced, and they could all be developed and tested independently. The side-impact bumpers which provided lateral support for the payload were detailed, along with their mounting at the base of the payload. Discussion of the general boxes for the electronics and the stirrer fan and airflow control systems were also given. These were discussed in less detail as they were not safety critical (failures here would not release large items capable of causing damage to the canister, and electrical failures are addressed as part of the battery system rather than with the units it served). The battery boxes received considerable attention as there are more restrictions on them. NASA required the boxes to be vented due to the potential for gas accumulation in the worst-case scenario of battery failure.

Having dealt with the actual experimental components, their arrangement in the payload was presented. This demonstrated how the individual components were assembled into a complete system. The assorted extras, and components that completed the payload and dealt with integrating it into an operating whole, completed the physical descriptions.

Attention then turned to the details of the power subsystem, including the routing of the power control to the NASA interface relays thereby giving NASA final control of power supply to the payload for safety, in order to allow emergency shutdown should this have been required. This section was critical from a safety point of view, as the electrical system controlled the dissipation of the stored chemical energy in the experiments battery pack. The safeguards on this system were required to prevent dangerous electrical discharges form occurring and to ensure the safe shutdown of the system in the event of failures. This was achieved by designing for the worst-case scenario and protecting the system with fuses and capacity de-

ratings so as to ensure that components operated within safe limits at all times. Experiment control was also covered in this section.

The battery system and its stored energy were then discussed. Finally the remaining items were summarised and details relating to materials, (referencing to the materials listing which was a separate document [155]), the interferometer cell, environmental control and the data logging of the actual experiment conditions during flight were given.

6.7.3 Safety Assessment And Verification

This section contained an assessment of the potential hazards posed by the payload, a description of how these were controlled, and hazard report forms which set the data out in a compact and standardised format.

The first matter addressed was the energy containment analysis summary, see Appendix 4. This considered the worst case situation of the Shuttle facing the sun and the battery energy being dissipated into the payload causing it to become hotter than intended. NASA again required examination of worst, and therefore simplest case conditions on the basis that if the payload could be proved safe under the worst conditions it could theoretically face, then it would be safe in any practically realisable failure situation. The other hazards assessed and explained were the failure of the mechanical structure, the failure of the batteries, as they could release potentially reactive products if they were to fail, and the optical system as it contained a laser.

6.7.4 Ground Operations Description

This section dealt with operations that only occurred during ground operations. These consisted of the integration tests and the test equipment required to carry them out. Firstly the test set was described as this is electrically interfaced to the payload and, although not a flight item, was still required to meet NASA safety regulations as it was used during integration tests at NASA. The procedures themselves were then detailed as NASA needed to know of any procedures that

could have posed hazards due to operations different to those carried out during flight. The two potential hazards posed by ground operations were electrical shock to personnel and optical radiation from the laser. Electrical shock risks were prevented by correct insulation and the following of regulations requiring that the live half of connectors have socket contacts. Measures taken to isolate the laser included a physical barrier as well as warnings and technical operating procedures [156] to ensure that the laser was not operated with the protective barrier out of position. Having detailed the solutions to the potential hazards, the associated hazard reports were then presented.

6.7.5 Summary of the Safety Review Process

With the safety documentation playing such a central role in the project, the preparation of the safety documents was started as early as possible in the project. At regular intervals, the progress made was reviewed by teams of NASA engineers and each iteration of the Safety Data Package became both more detailed and closer to NASA's requirements for final safety approval. The progress of the safety documents was reviewed with NASA after each new document submission through telephone conferences held with the NASA engineering team to assess the document, its deficiencies and suggested changes to either the document or payload. This process was also used to discuss questions raised as a result of the development of the payload with NASA directly, and to clarify the interpretation of the requirements and documentation where required. This latter was required both for clear understanding and to overcome situations where NASA's ongoing evolution of the Space Transport System led to changes in the requirements on GAS payloads during the period of development of the experiment.

Chapter 7

Results

7.1 Types Of Results

7.1.1 Images to be Processed

Fringe images for analysis by the processing system developed and described in this thesis have been acquired from several sources. The research and development of the image processing system discussed in Chapters 3 through 5 was based upon interferogram fringe images generated during the conceptual development of the space flight payload. Further results have been produced using the flight equipment both during pre-flight testing and the space experiment. Due to differences between the equipment arrangements used there are notable differences between the concept testing and payload generated images. Although the format, as expected, is similar they appear quite different to a casual inspection. This is demonstrated by Figures 7.1 and 7.2 which illustrate comparable sections of a 'concept testing' image and a 'ground test of the flight equipment' image, respectively. A summary of the source of each of these groups of images is discussed below along with explanations of the differences.

Concept Test : Image Section

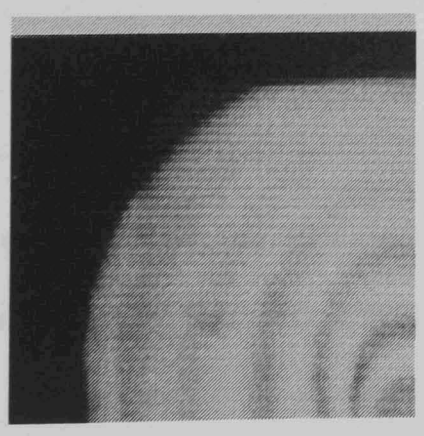


Figure 7.1

Flight Equipment Ground Test : Image Section

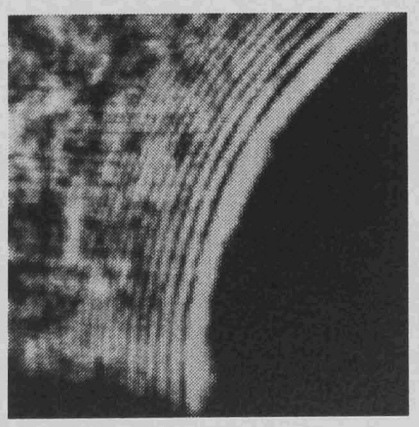


Figure 7.2

7.1.2 Summary Of Testing And Experimentation

The initial ground test images were generated during the conceptual development of the experiment using a 'hot resistor in air' to form convection patterns. The patterns observed in these images were quite varied, but all have the form of a series of light and dark interferogram fringes. In most of the images these are arrayed concentrically about the heating element. Some of the images though are of equipment arrangements that produced insufficient heating to generate concentric fringe patterns. As a result only fringes generated as a result of attributes of the experimental equipment were observed. These images therefore contained no convection data, although the fringes in them were used for development testing of the system. No results from these images are therefore presented.

The results take the format of concentric fringes, as thermal conduction in the fluid cell produces a temperature gradient from the heating element to the surrounding cell walls on which the thermal gradient distortions produced by convection are superimposed. As discussed previously the fringes are then formed on the temperature contours. Four of the concept testing images were presented in Chapter 5 as part of the demonstration and explanation of the workings of the processing system. The remaining five images for which sufficient heating was applied for concentric fringes to be generated are presented below. Images from the pre-flight ground testing of the payload interferometer are then presented. These were taken in the lab with a steady one Earth's gravity force acting.

The principal visual differences between the payload generated images and the concept testing images is that the camera in the flight equipment imaged only a part of the fringe pattern, but the contrast was much improved. It is also notable that all the images have the centre of heating in the same location as no realignments were carried out. Further unlike the concept testing images multiple images in sequence have been taken of a single experiment operation. Finally the flight results are presented with notes on processing used to overcome interference in these images.

7.2 Experiment Concept Tests

7.2.1 Description of Tests

These results are from testing the image processing system on five of the test images generated during conceptual development of the microgravity experiment that were not used in Chapter 5 for processing system demonstration purposes. The system in each case is searching for reasonably complete fringes in the form of concentric rings about the heating element, which in each case is positioned near the centre of the image. A 'gravity vector' is returned based on the average displacement of the fringes furthest from the centre with respect to those closer to the centre. The fringes furthest from the centre denote previous conditions as heat takes time to 'flow' outwards from the heating element. The outermost fringes are therefore the last to be affected by changes in the convective flows in the cell caused by variations in the gravitational field acting on the experiment.

A variety of gravity conditions are simulated in these images. The results therefore demonstrate varied convection patterns and include a variety of image features that cause less than optimal fringe detection. These difficulties are discussed with the features which produce them.

As before, the algorithm (a Matlab function) is:

function[xvect,yvect] = rfvect10(photo,xstart,ystart,cutang,noisefract)

With the parameters:

xvect = x co-ordinate component of calculated gravity vector

yvect = y co-ordinate component of calculated gravity vector

photo = filename of the file containing the input fringe image

xstart = x co-ordinate position of the fringe pattern centre start point

ystart = y co-ordinate position of the fringe pattern centre start point
(Where the starting point corresponds to the location of the axial heating element in the fluid cell of the interferometer.)

cutang = slice angle corresponding to the fringe break
(This corresponds to the position of the heater supports which often distort the fringes. Where the distortion was minimal, it was set at a value greater than the maximum angle, which instructs the algorithm to ignore the operator in that specific instance.)

noisefract = fraction of a fringe which must be present before the fringe will be used for gravity vector interpolation. All fringe sections that form less than noisefract of a ring have been discarded during processing and are not displayed.

7.2.2 Concept Test Image Results.

Test Image 5

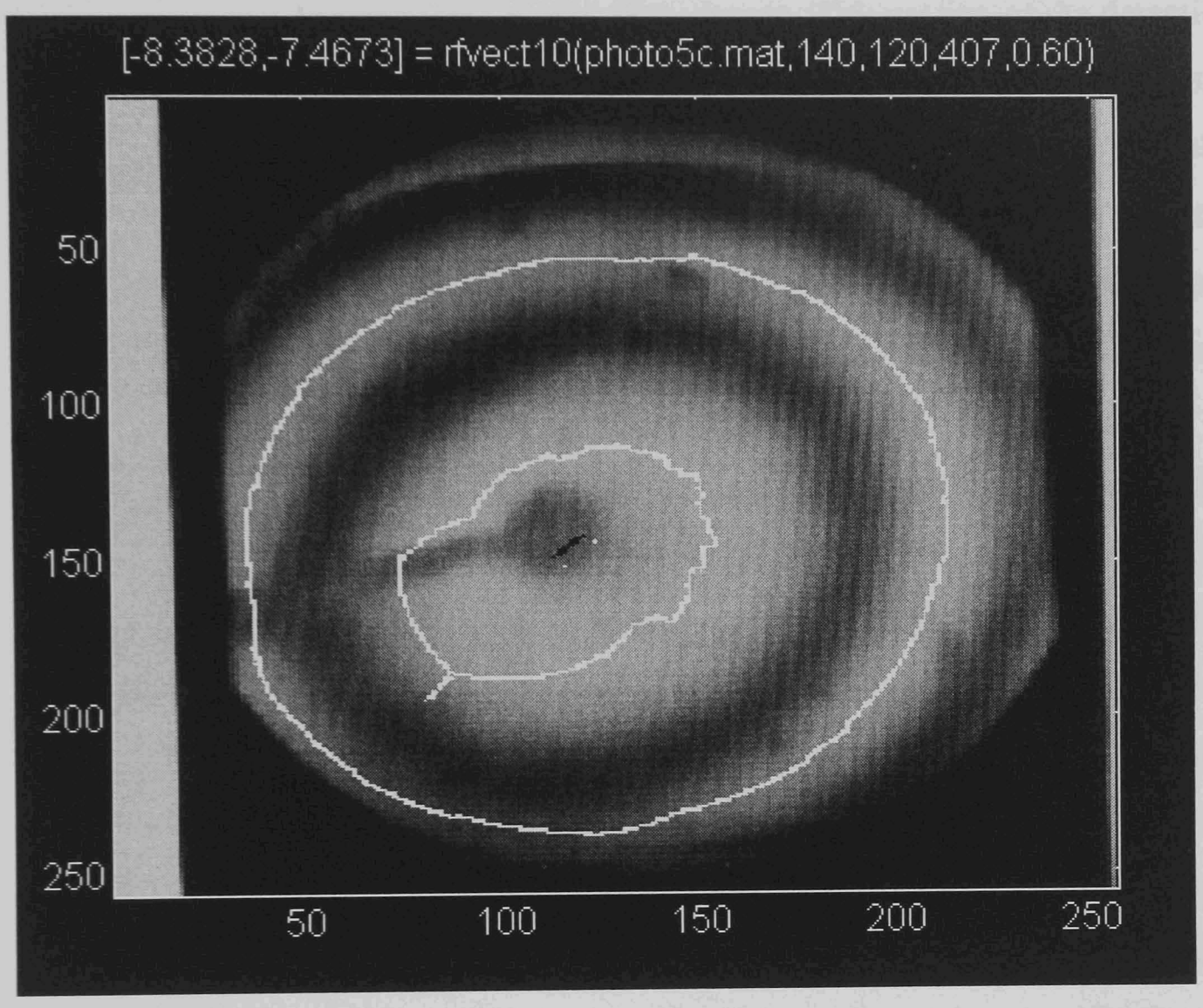


Figure 7.3

Image type : Simulated low gravity convection

Starting co-ordinates : Row = 140 : Column = 120

Gravity vector : x component : -8.4 : y component : -7.5

Comments : This image demonstrates the operation of the algorithm on wide fringes. The bottom left quadrant of the inner fringe has been interpolated to complete the fringe across a gap in its detection by the system. The result though is not good, as can be seen by the fact that the interpolated section is displaced from the fringe centre line. The interpolated result however is more accurate than calculating the gravity vector based on

assuming the missing fringe section has a radius of zero. The cause of the problem is the section of the inner fringe just above the horizontal 'grey band' positioned toward the left of the image. The 'grey band' feature is the heater support which has obscured a slice of the image. This section of the detection is obviously following a different path to the rest of the fringe detection. As it is on a section of fringe which is partially obscured by the heater support the fringe seen by the algorithm is partially obscured resulting in a narrower and displaced fringe being erroneously detected by the algorithm. This result is considered as being due to the poor choice of starting co-ordinates that have resulted in the slices not lying parallel to the heater support wire. The distortion has therefore resulting in fringe corruption on each side of the image section obscured by the heater support, as seen by the image slices taken by the 'star algorithm' stage of the processing system. This is demonstrated to be the case by Figure 7.4 in which the system starting point has been defined in line with the area obscured by the heater support wire. This adjustment takes advantage of the fact that the starting point co-ordinates are parameters set to overcome problems such as this caused by the physical arrangement of the heater in a particular experimental arrangement. As a result the slices taken by the 'Star' algorithm in Figure 7.4 are parallel to the disturbance rather than meeting it at a shallow angle as they did before, with some being partially fringe and partially corrupted by the supports shadow. Thus the erroneous processing of corrupted fringes is avoided.

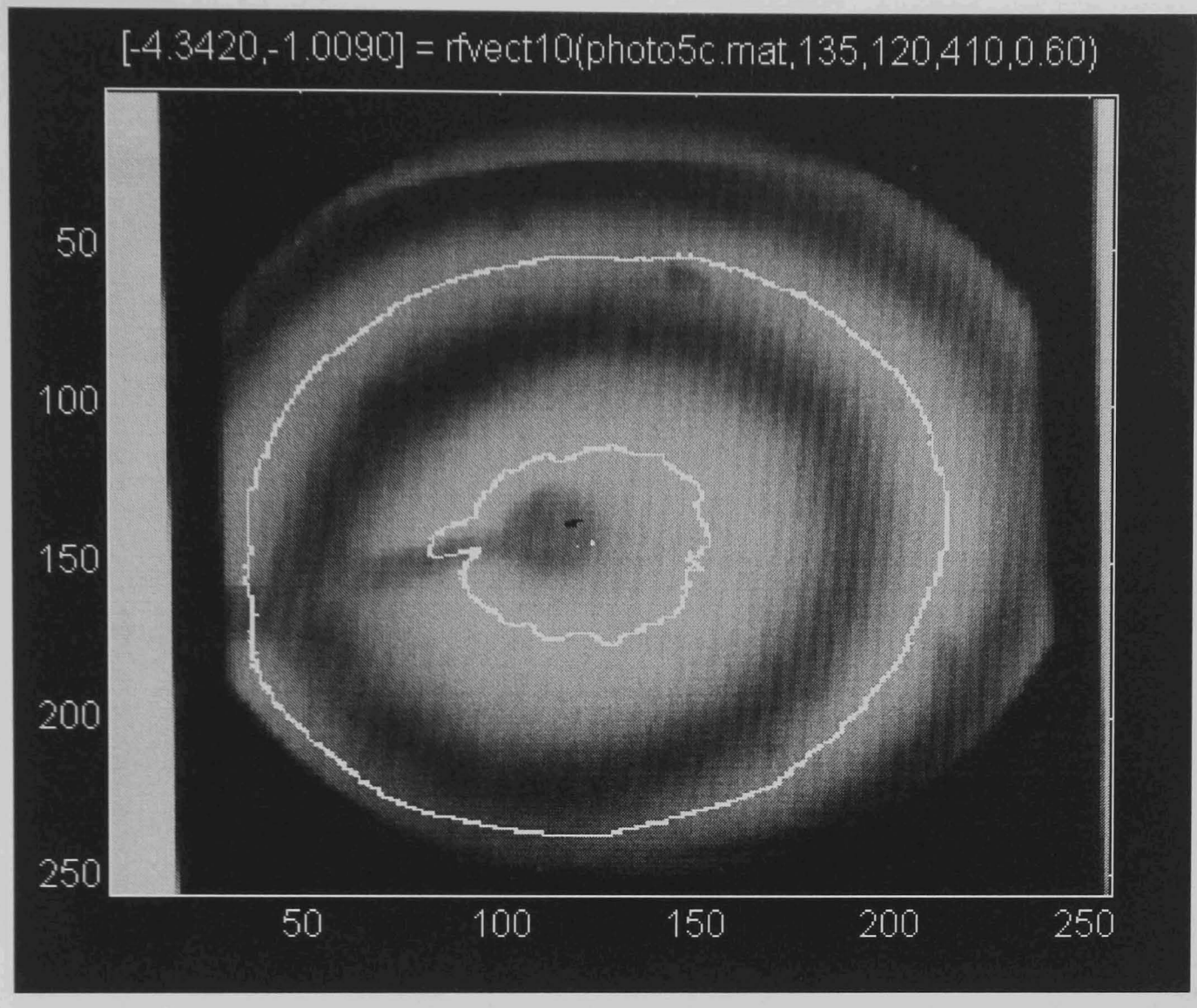


Figure 7.4

Image type : Simulated low gravity convection

Starting co-ordinates : Row = 140 : Column = 120

Gravity vector : x component : -4.3 : y component : -1.0

Comments : Improved image that resulted from changing the starting row point by 5 as discussed above. This aligned the slice scan better with the physical arrangement of the experimental equipment used to generate the image. Although it took two attempts to get good results for a single image if more images had been generated with this experimental arrangement the same parameter set would be good for all of them. This operator assistance is therefore not significant for using the system as it is intended, processing large numbers of images generated by a single set of experimental apparatus.

Test Image 6

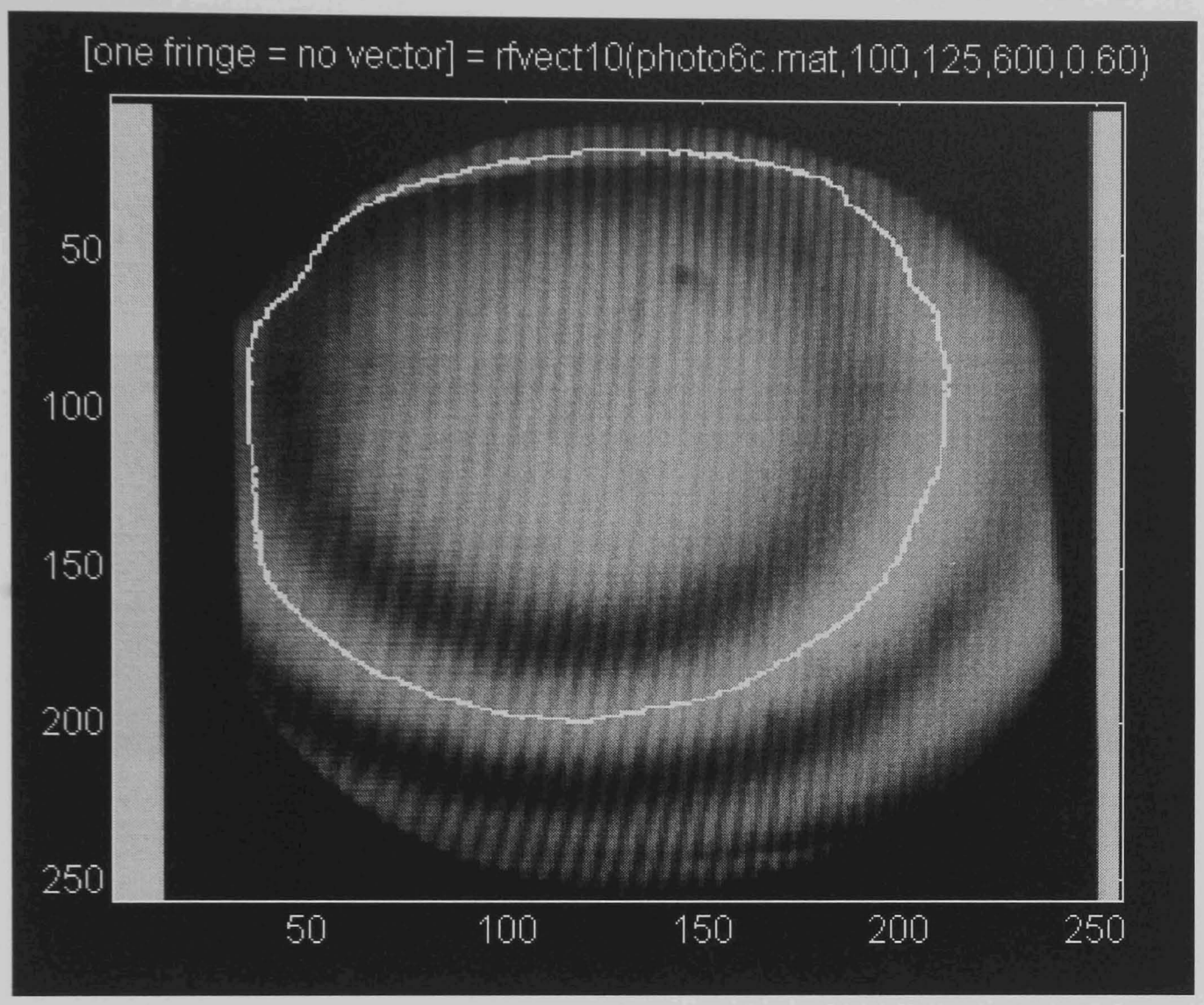


Figure 7.5

Image type : Simulated microgravity.

Starting co-ordinates : Row = 100 : Column = 125

Gravity vector : x component : N/A : y component : N/A

Comments : Only one even reasonably complete fringe is visible in this image so no gravity vector can be established. However this image shows a particularly high degree of rippling from image capture effects. The result is therefore an effective demonstration of the operation of the system in adverse conditions where existing operators detect so many of the 'noise' fringes as to be useless. This is demonstrated by a comparison with two of the well known edge finding operators in Figure 7.6 and Figure 7.7. The results from these operator

have obviously been affected by the noise bands and although the feature edges are indicated the edge localisation results are poor.

Sobel Edge Detector

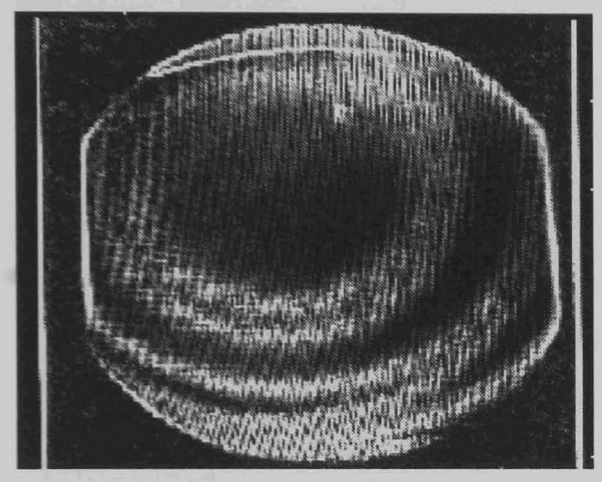


Figure 7.6

Robert's Cross Edge Detector

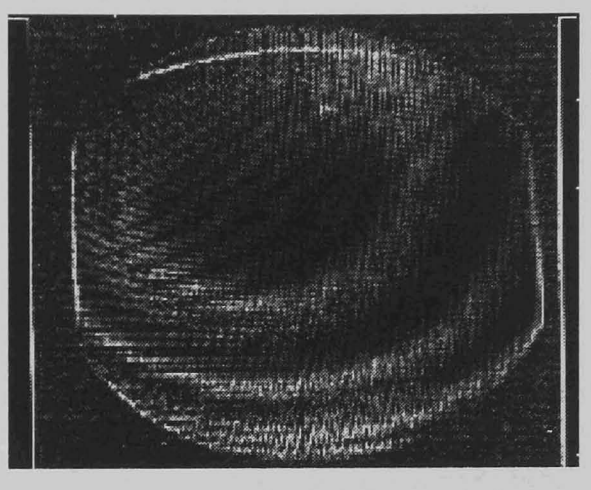


Figure 7.7

Test Image 7

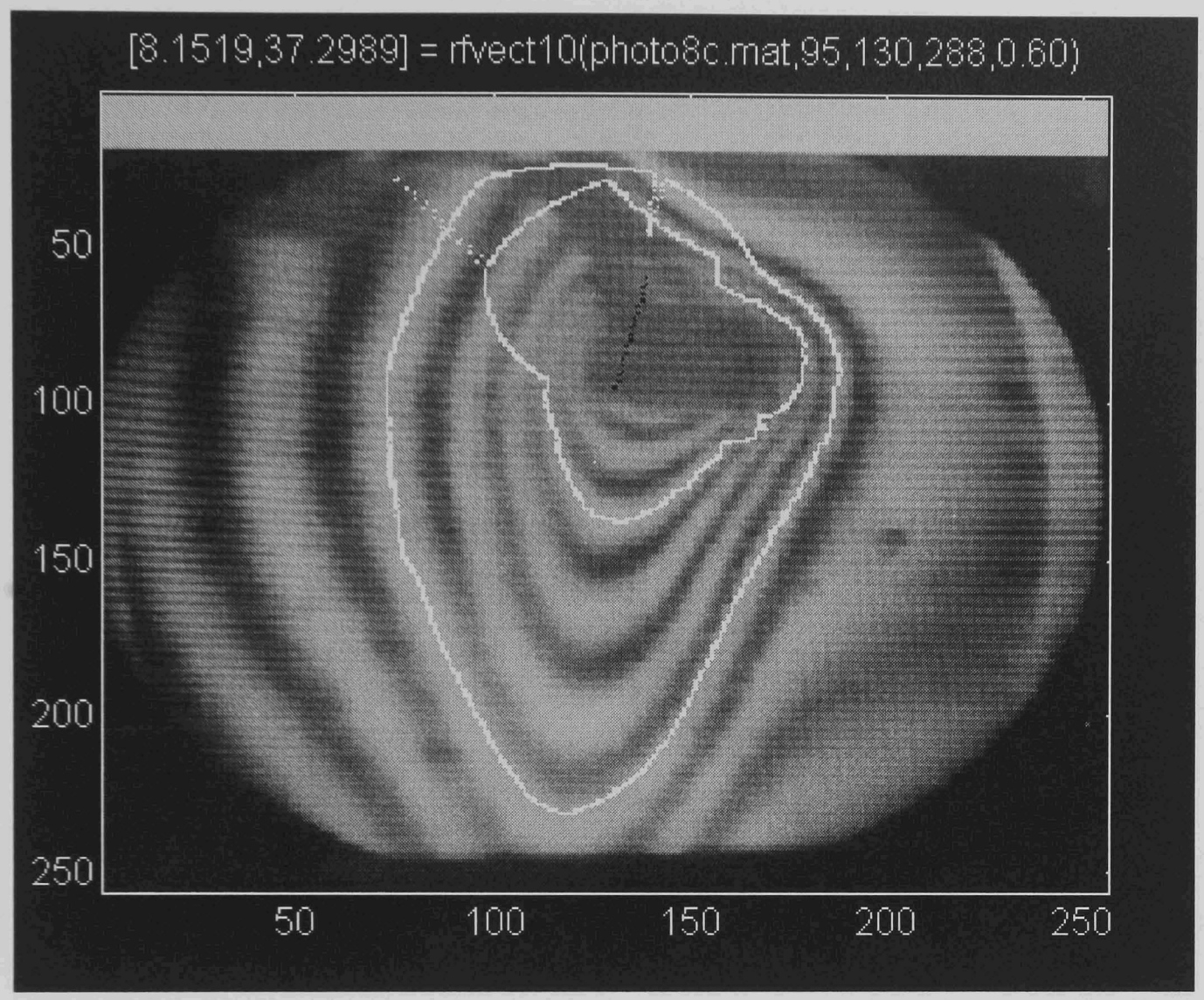


Figure 7.8

Image type : Simulated low gravity convection

Starting co-ordinates : Row = 95 : Column = 130

Gravity vector : x component : 8.2 : y component : 37.3

Comments : An example of highly elongated fringes. This image which simulates a low gravity environment is more distorted than microgravity results are anticipated to become. Despite the distortion in the upper section of the image filling of the gaps in the fringes by the interpolation system has resulted in a sensible gravity vector from the image. The principal source of the error is the short vertical line on the inner fringe, which corresponds to an interfringe jump having occurred at one of the distorted ends of the inner fringe detection.

Test Image 8

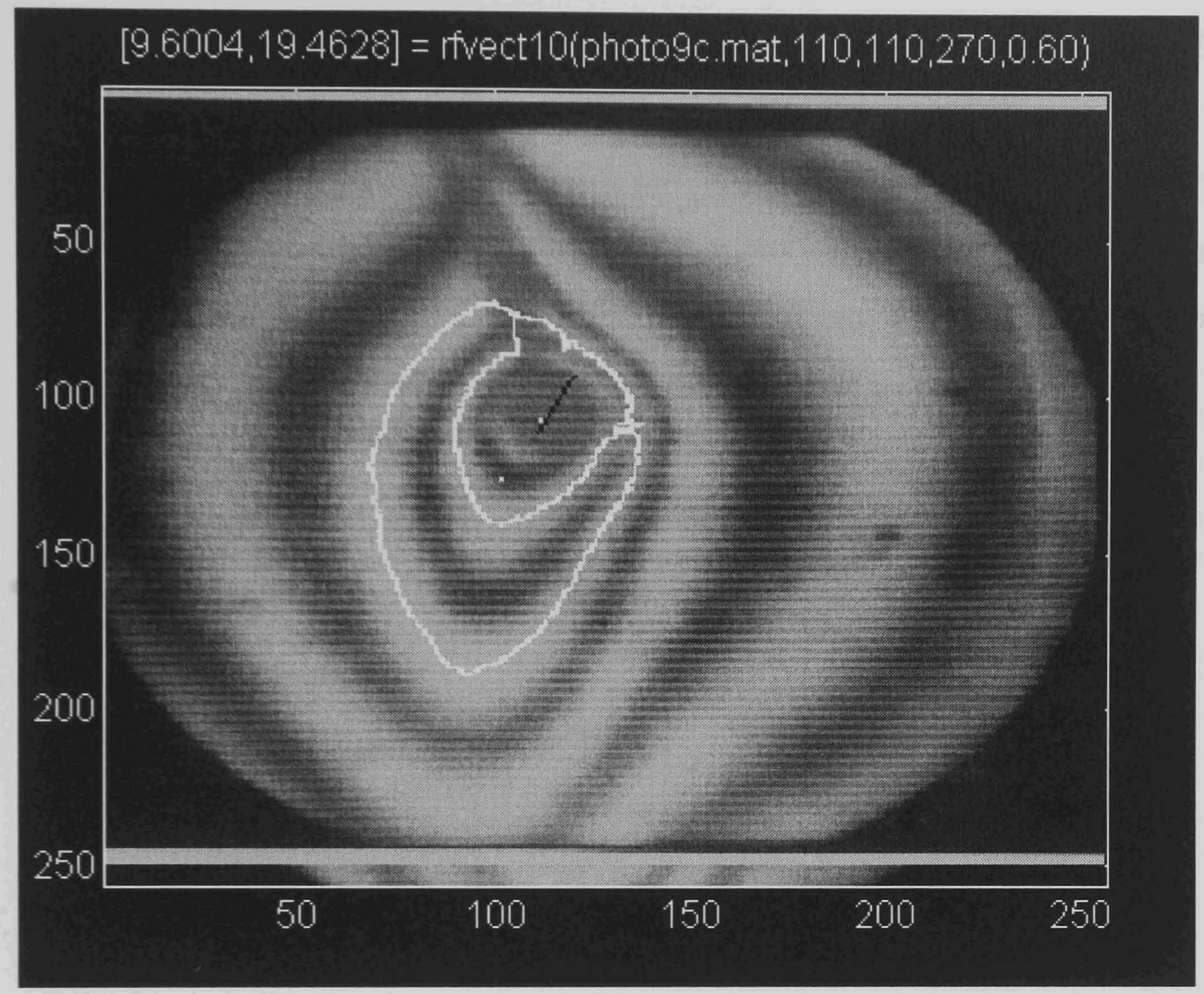


Figure 7.9

Image type : Simulated low gravity convection

Starting co-ordinates : Row = 110 : Column = 110

Gravity vector : x component : 9.6 : y component : 19.4

Comments : Another image with fringes of varying widths. The notable points are the elongated fringe that has been detected and the use of a second tracing of the same fringe rather than interpolation where the two fringes have blurred into one another close to the centre point. The varying fringe widths in this image are something which would have made its processing using a curve fitting operator either impossible or very computation intensive. The different fringes would have fitted to different curves and have a cross section that does not

conform to a sinusoidal model. Although it may seem the algorithm should have found the third fringe, the contact that this fringe makes with the image edge effects around row 240, column 100, probably caused a large jump in the radius of the detected fringe. This would have led to a second fringe break opposite that due to the heater support wire and two fringe detections of about 45% of the fringe which testing has shown to be of little use in the general case.

Test Image 9

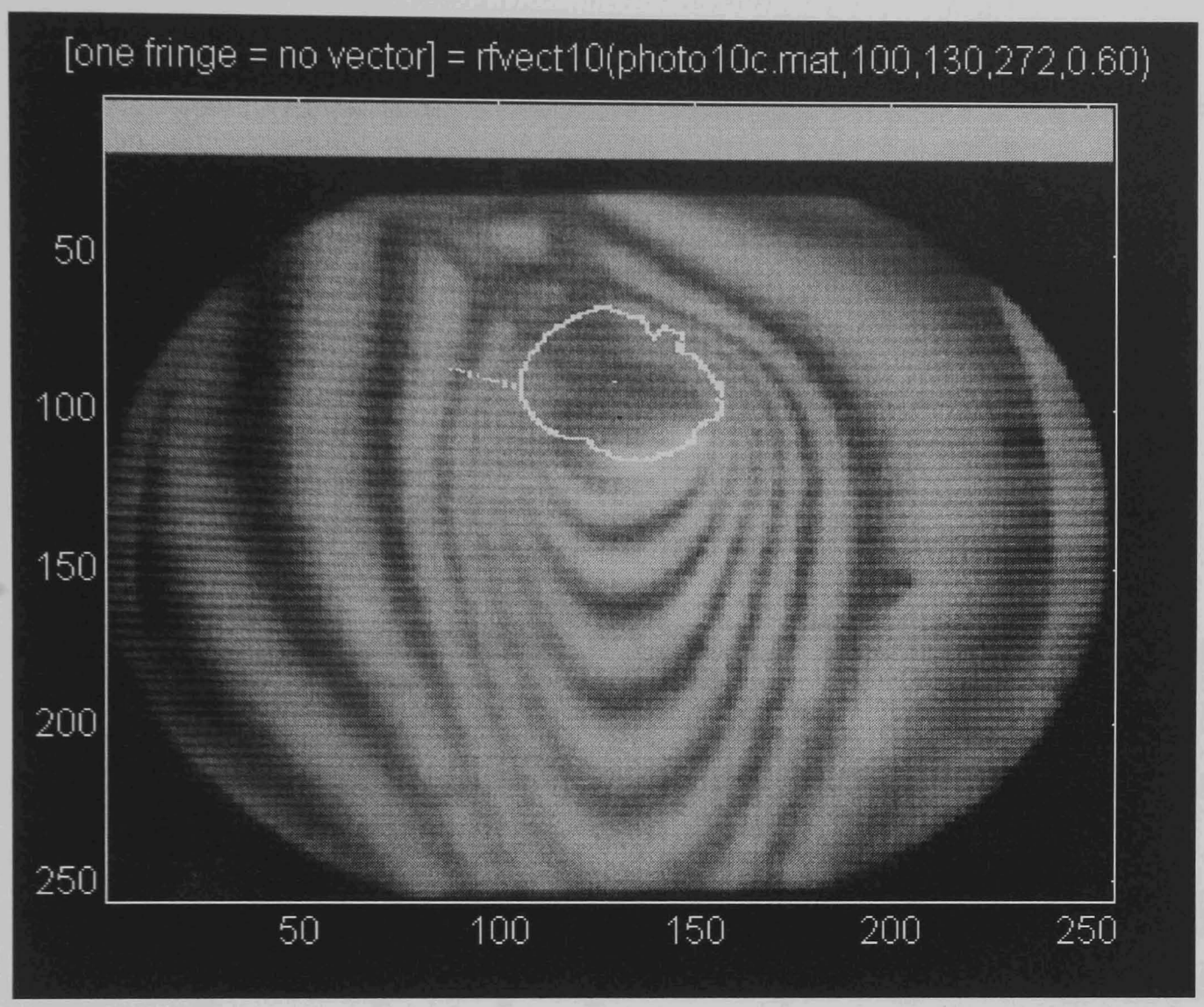


Figure 7.10

Image type : Moderate gravity convection.

Starting co-ordinates : Row = 100 : Column = 130

Gravity vector : x component : N/A : y component : N/A

Comments : Failure of the algorithm. Although a lot of fringe sections were detected in this image most of them were small sections as the five fringes seen below the heater location blur into a single fringe above it. This is due to the large amounts of convection occurring. This resulted in the mixing together of the fringes and combined with image edge overlap and two heater supports the varied fringe distortions have broken the fringes up too much for the system to achieve automatic fringe recognition. Even if the small sections that are complete were

used for interpolation their shapes give no indication of the much sharper bend of the fringes about the upper side of the heating element, hence mostly erroneous results would be produced. In order to overcome problems such as these a much more 'intelligent' processing system is required.

7.3 Ground Test Of Experiment

7.3.1 Test Conditions

This section presents a selection of results from processing images generated on the ground with the payload equipment prior to flight. These results cover the testing of both existing image processing algorithms and the system which has been developed.

Two sections of a screen image have been processed both with the system developed using the concept testing images and using a selection of edge detection operators from the literature. The utility of further operators from the literature which have not been implemented and tested to produce direct comparisons is discussed in Section 7.3.3. Due to the improved contrast present in this image, compared to the concept testing images, the existing operators perform satisfactory edge detection in this image unlike their poor performance on the concept testing images, as was demonstrated in Figures 7.6 and 7.7. These operators are however still of minimal assistance in actually extracting the fringes as features from the images.

Several minor alterations were made to the parameters and pre-processing utilised by the proposed processing system in order to process the images generated by the flight hardware. The principal differences were the addition of control over the angles at which slices were processed, (as these image sections are only fractions of the area surrounding the heating element) and the suspension of the initial overall smoothing, (as the contrast is clearly better and this assists in the detection of narrow fringes). These changes were intended to avoid unnecessary processing and to adapt the system to the physical arrangement of the flight equipment, as opposed to the

concept testing system. The function call to the system is therefore considerably different. The processing meanwhile remains as developed previously and the flexibility of the system is demonstrated by its performance on these images with no significant alterations having been made from the system used to process the concept images. The revised algorithm is listed in Appendix 2.

The processing system algorithm (as a Matlab function) is:

function[xvect,yvect] =

rfptarc3(photo,xstart,ystart,cutang,noisefract,xmax,ymax,sang,eang,angmul,tacc)

With the parameters:

xvect = x co-ordinate component of calculated gravity vector.

yvect = y co-ordinate component of calculated gravity vector.

photo = filename of the file containing the input fringe image.

xstart = x co-ordinate position of the fringe pattern centre start point.

ystart = y co-ordinate position of the fringe pattern centre start point.
(Where the starting point corresponds to the location of the axial heating element in the fluid cell of the interferometer.)

cutang = slice angle corresponding to the fringe break.
(This corresponds to the position of the heater supports which often distort the fringes. Where the distortion was minimal, it was set at a value greater than the maximum angle, which instructs the algorithm to ignore the operator in that specific instance.)

noisefract = fraction of a fringe which must be present before the fringe will be used for gravity vector interpolation. All fringe sections that form less than noisefract of a ring have been discarded during processing and are not displayed.

xmax = maximum rows, required by Matlab.

ymax = maximum columns, required by Matlab.

sang = starting angle for slice processing. Specified to avoid processing a lot of irrelevant data.

eang = end angle.

angmul = angle multiple, tacc times angmul equals the number of angles required for all 360 degrees to be processed.

tacc = total angles to be processed.

7.3.2 Results for Ground Test Images

Two sections of a single image are used. These were separated by a black area due to masking of the image by the heating element support wire. The principal notable feature of the fringes in these images is that they are more regular than those in the concept testing images and the contrast is much greater.

For each of the two fringe quadrants, the fringe image, results with image processing operators from the literature, raw fringe detections by the processing system and completed fringes are presented. The systems raw detection images give a much better comparison of the capability of the fringe detector stage when applied to these images compared with the existing operators than does the final processing system result. In the final system results images the fringe grouping and feature identification stage has eliminated most of the noise and erroneous detections. This final stage demonstrates the principal advantage of the system in its ability to return cleaned up images in which the spurious detections of image noise have been identified and suppressed and the detected fringes have been recognised as distinct objects.

Fringes Section 1

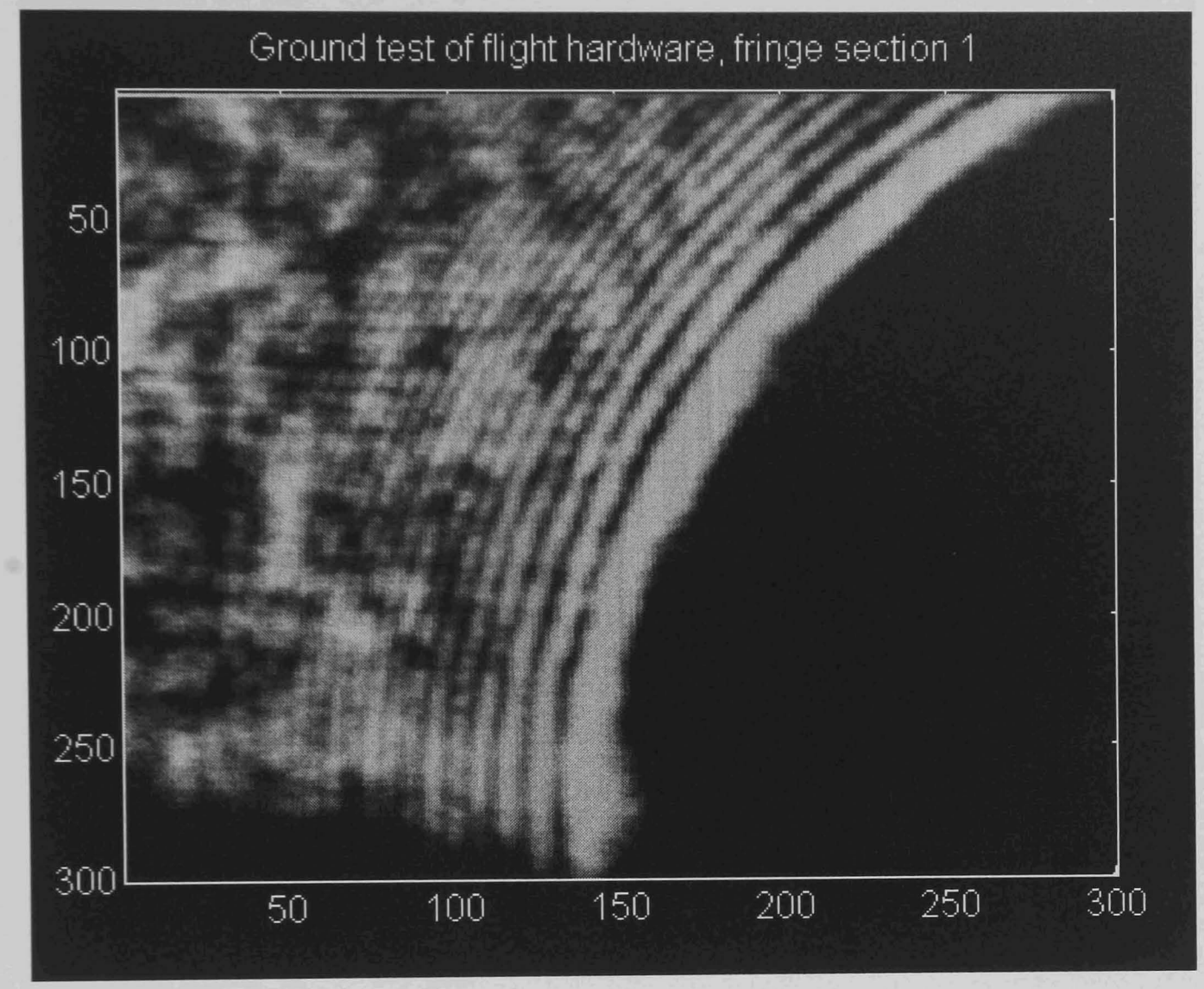


Figure 7.11

Figure 7.11 illustrates the first ground testing fringe section. The upper edge of this section is the limit of the video camera CCD while at the lower edge blanking due to the heating element support is visible. Over a dozen fringes can be made out by eye but only the first three are really complete and distinct. The heating element itself is in this case somewhat to the right of the bottom right corner of the image section presented.

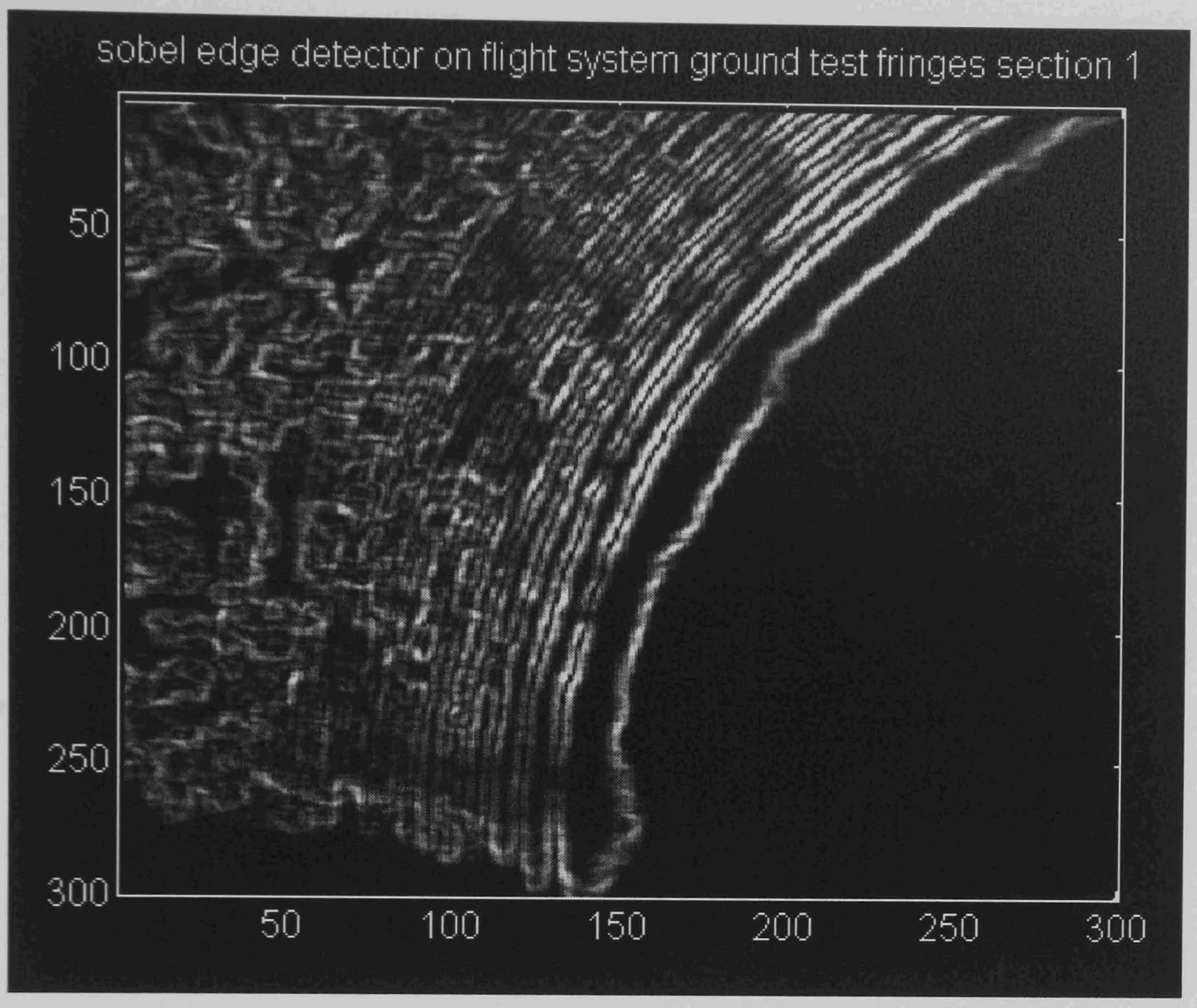


Figure 7.12

Image type : Earth's gravity pre-flight test of flight hardware.

Comments : This image, Figure 7.12, demonstrates the result of applying the standard Sobel edge detector to the image of Figure 7.11. These high contrast fringes are readily detected, however the detections tend to be on the 'dark' side of the fringe edges and are rather indistinct in darker areas of the image. The width of the detections could be fairly readily overcome if the detection points could be grouped. However an edge following algorithm would have problems following the rather fuzzy edges and would be confused by the basically random mass of fringe detections in the left hand section of the image. Work during the development of the new system demonstrated that attempting to identify the quality of fringes by their curvature over small distances is problematic at best. Further this operator has not localised the detections very well. As commented the maximum operator response

is biased toward the darker section of the fringe edge, demonstrating that the part of the intensity gradient denoting the fringe side that is detected is somewhat random. This is rather poor as although the edges are detected it generally has to be accepted in any further processing that detections represent either the midpoint of the fringe sides or their extremes and invalid assumptions may result where this is not the case.

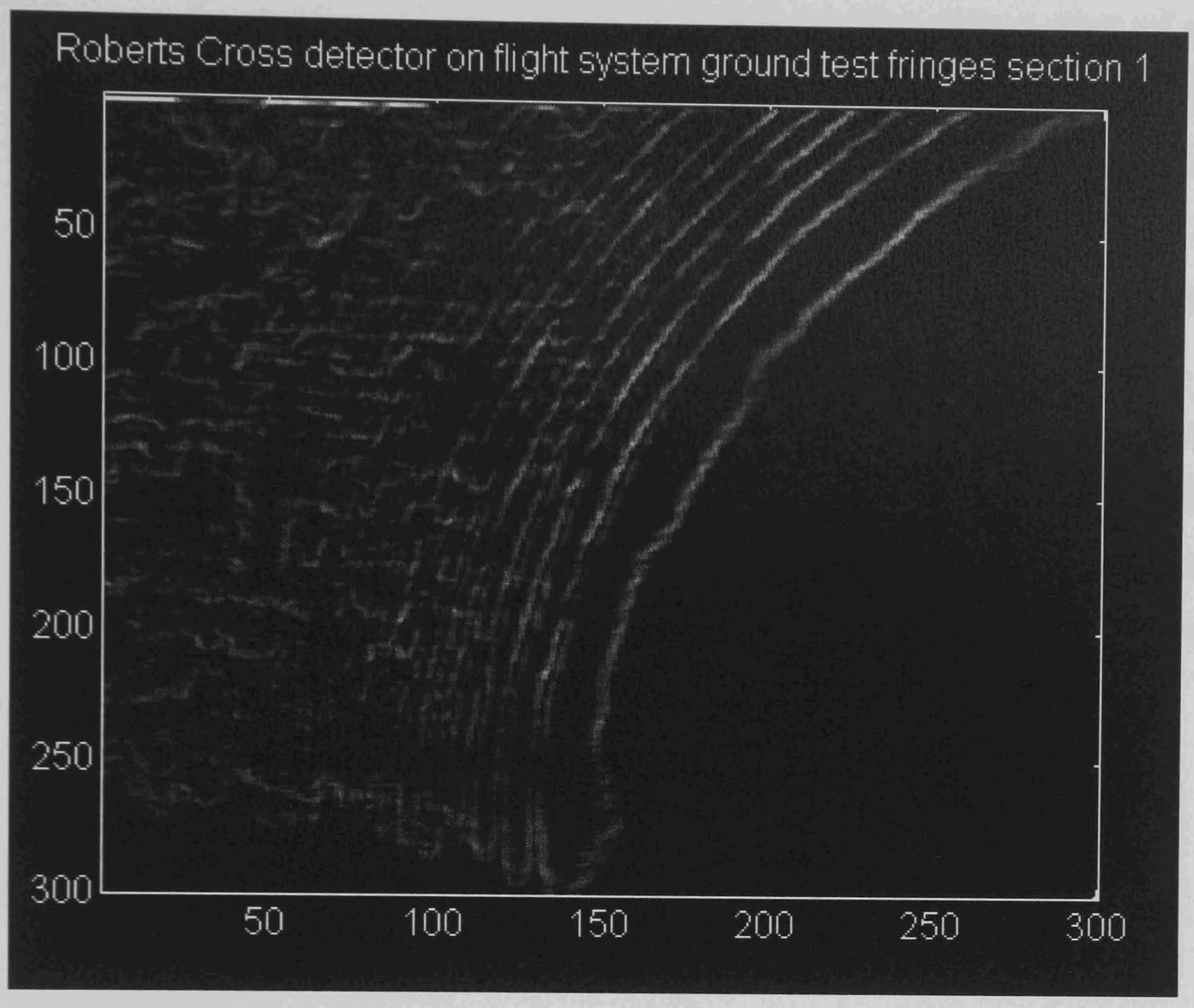


Figure 7.13

Image type : Earth's gravity pre-flight test of flight hardware.

Comments : Result of application of the Robert's Cross operator, Figure 7.13. The result is very similar to that for the Sobel operator, Figure 7.12 as could be expected. The fringe edges are however less well defined and the Sobel result is clearly superior. A further serious problem with both this operator and the Sobel operator is that the output at each point is a subjective value, high for an edge and low for not an edge. Although these operators amplify the edges a thresholding operation is required to achieve definitive fringe detection. This will inevitably require operator intervention if the fringe edge points are to be reliably extracted from the results array. Testing carried out on the Hough transform demonstrated that a pre-determined, or fraction of some image parameter threshold, is not reliable without checking. Even if this difficulty could be overcome the problem remains that for any

thresholding of even moderate complexity the darker areas of the image will cause breaks in the detections as the weaker returns will be deleted. This may be limitable using an adaptive thresholding technique but the resulting thresholding map will have to be calculated using some form of large averaging mask as an analytic plane will not describe the threshold pattern required.

Overall these images demonstrate that for edges of good contrast these well known operators are very good at the task they were designed for. They however provide little assistance in actually identifying and extracting whole fringes from a general input image without considerable post-processing and operator assistance in interpreting the results.

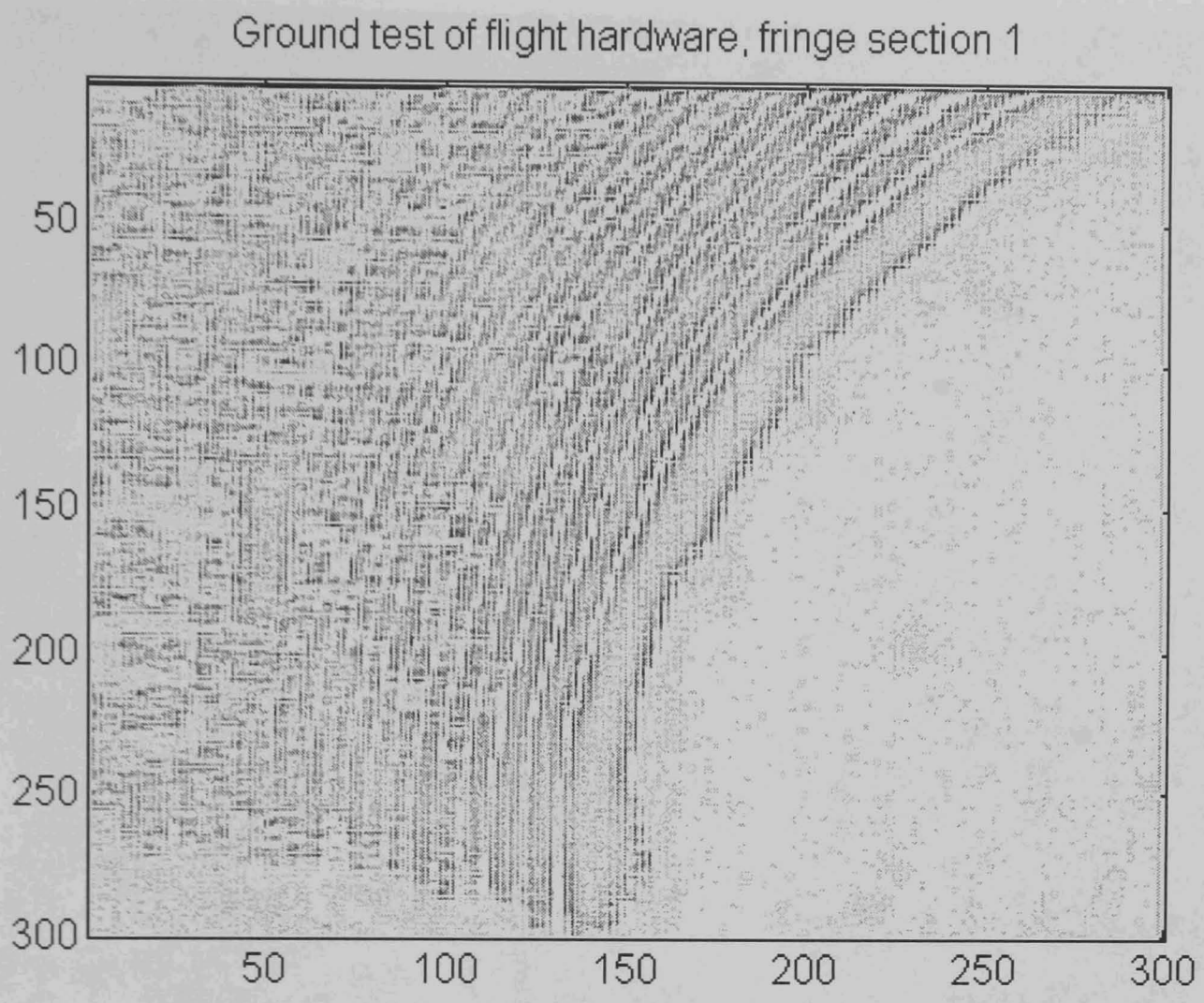


Figure 7.14

Image type : Earth's gravity pre-flight test of flight hardware.

Comments : Figure 7.14, Commercial edge enhancement algorithm applied to the image of Figure 7.11. This is a negative image to highlight the result which is best described as poor. Although the edges have been highlighted to some extent they are not particularly distinct and the result is therefore of minimal use for anything other than operator interpretation which was easier in the original image.

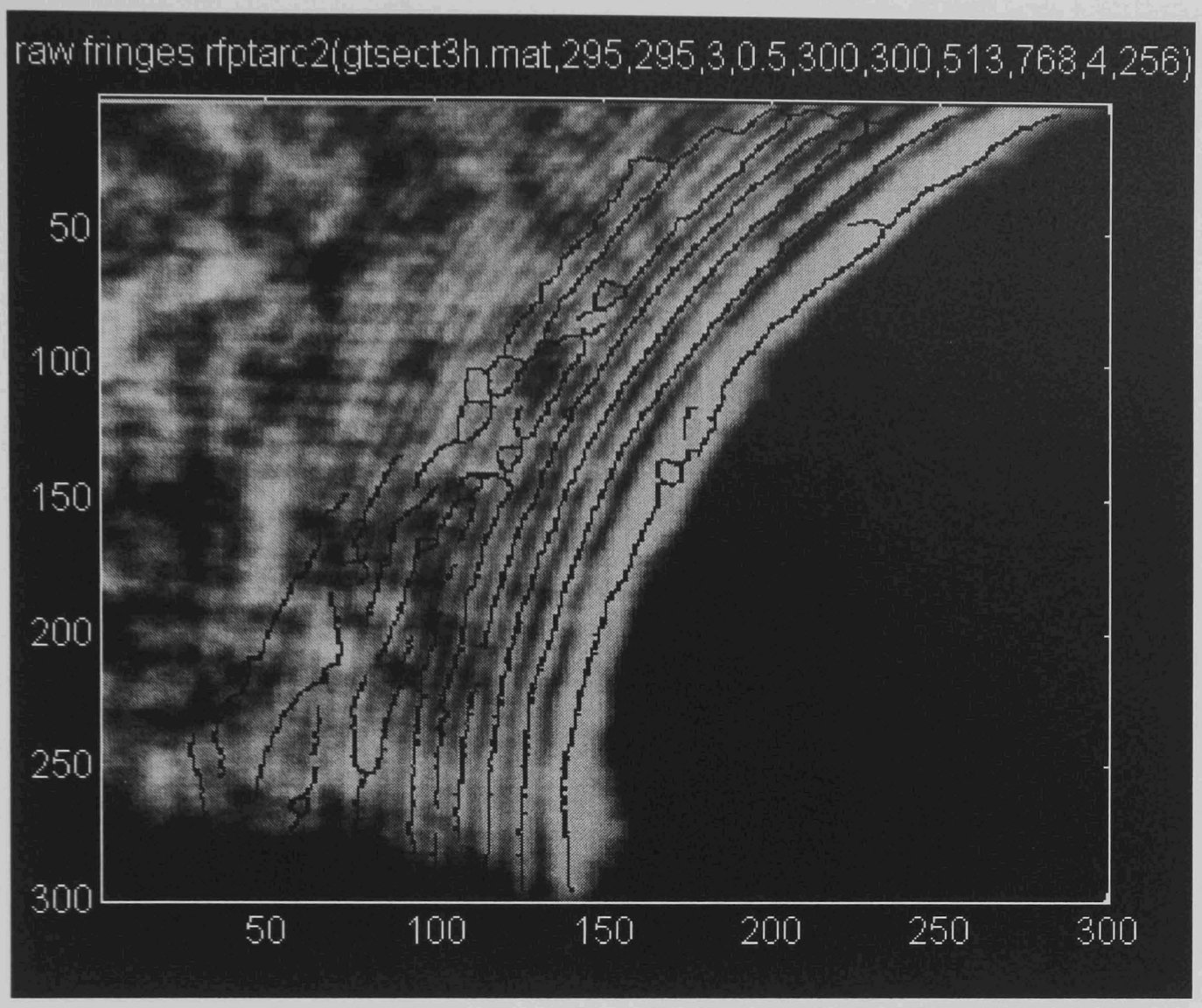


Figure 7.15

Image type : Earth's gravity pre-flight test of flight hardware.

Starting co-ordinates : Row = 295 : Column = 295

Comments : Fringe detection result from the proposed image processing system. This image, Figure 7.15 is the fringes as detected by the 'fringe roof detection' stage with only the smoothing and tidying of the 'dilate-and-thin' stage of the system having been applied. This compares directly with the results obtained using the Sobel and Robert's Cross detectors, although this is a fringe peak as opposed to edge detector. The image corruption due to noise is fairly comparable to the previous operators, however there are two important differences. The majority of the pure random noise in the upper left quadrant of the image has been ignored by the system, the distorted lines that are observed being poor detection of the corrupted outer fringes. The edge detecting algorithms

meanwhile faithfully trace the boundaries of the light and dark sections of the noise just as successfully as they do the fringe sides. Principally however the results returned from the new system are stored by the system in a format that is much more convenient for further processing and the detections shown are firm 'there is a fringe here' results. This as noted is not true of the edge detectors which only returned a probability result that was not thresholded despite the results appearing clear to an observer. To assist further processing the detection data is stored in its own array of data points and not merged with the image, though it is displayed on the fringe image above to demonstrate the results in an easily comprehended format.

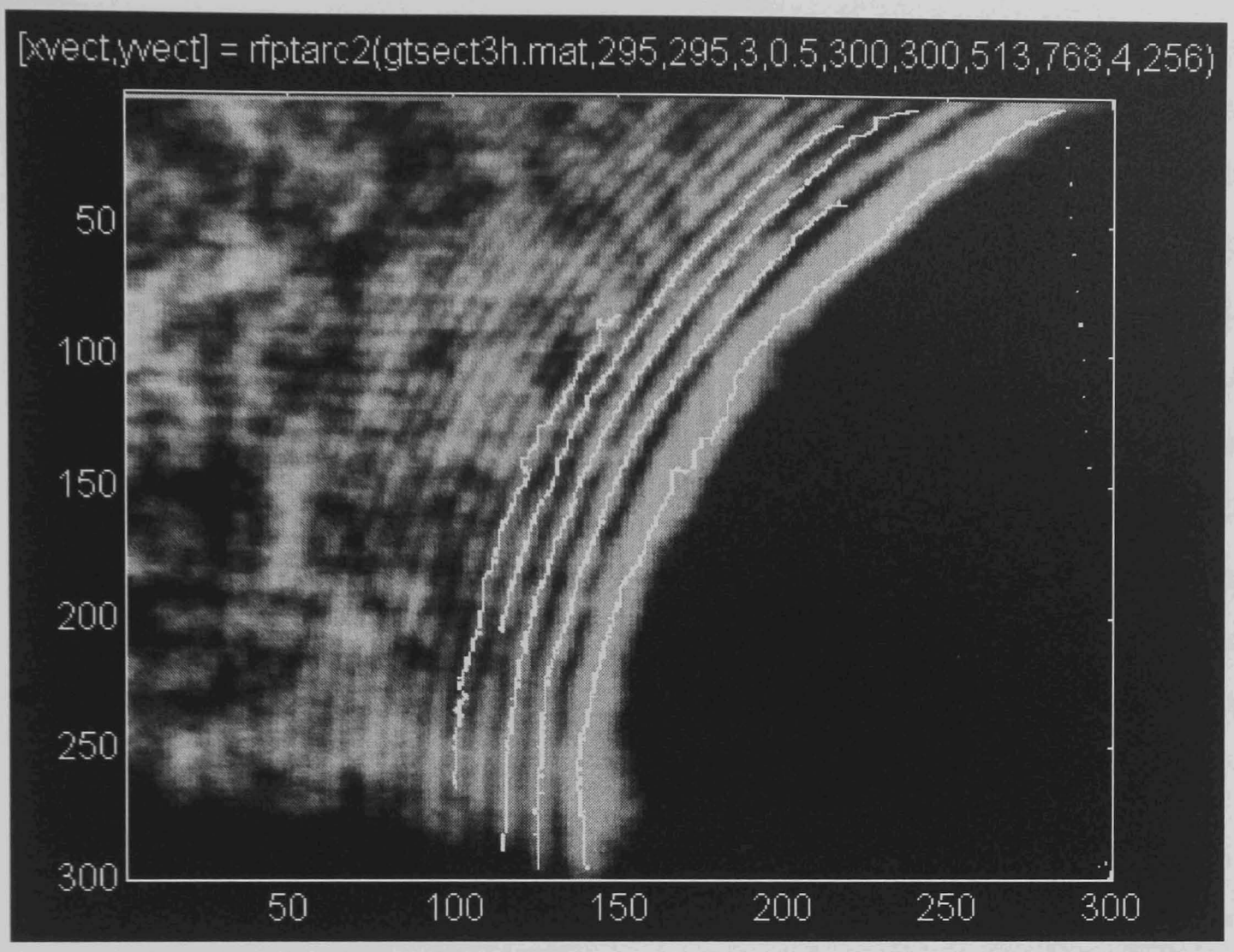


Figure 7.16

Image type : Earth's gravity pre-flight test of flight hardware.

Starting co-ordinates : Row = 295 : Column = 295

Comments : Result from the new image processing system. This is using a somewhat modified version of the algorithm that has an expanded list of input parameters as given in Section 7.3.1. The principal difference between this result and those for the experiment concept tests images being that only partial rather than mostly complete fringe are present. Although the five fringe sections present in Figure 7.16 have been detected, separated from the noise and were correctly grouped as fringes one through five, interpolating a gravity vector from this image has proven impossible. As discussed previously it is in theory possible to match the sections detected to closed curves but in practice the amplification of noise is such that the result is effectively meaningless. Testing of several interpolation schemes matching

the detected arcs to circles and comparing the positions of the centres of gravity of the common sections of the fringes have been assessed. In each case however the error uncertainties are too high to be able to rely on the results having any real meaning. The fringe sections in this image equate to a lot less of the detectable fringes than either the previous image, Figure 7.15, or the edge detectors. This is due to noise suppression deleting the short fringe detections including the upper parts of some of the fringes along with the noise. This makes the result of minor discontinuities in the fringes much more obvious than does the basic image which simply showed the fringe sections both sides of a detection gap.

Fringes Section 2

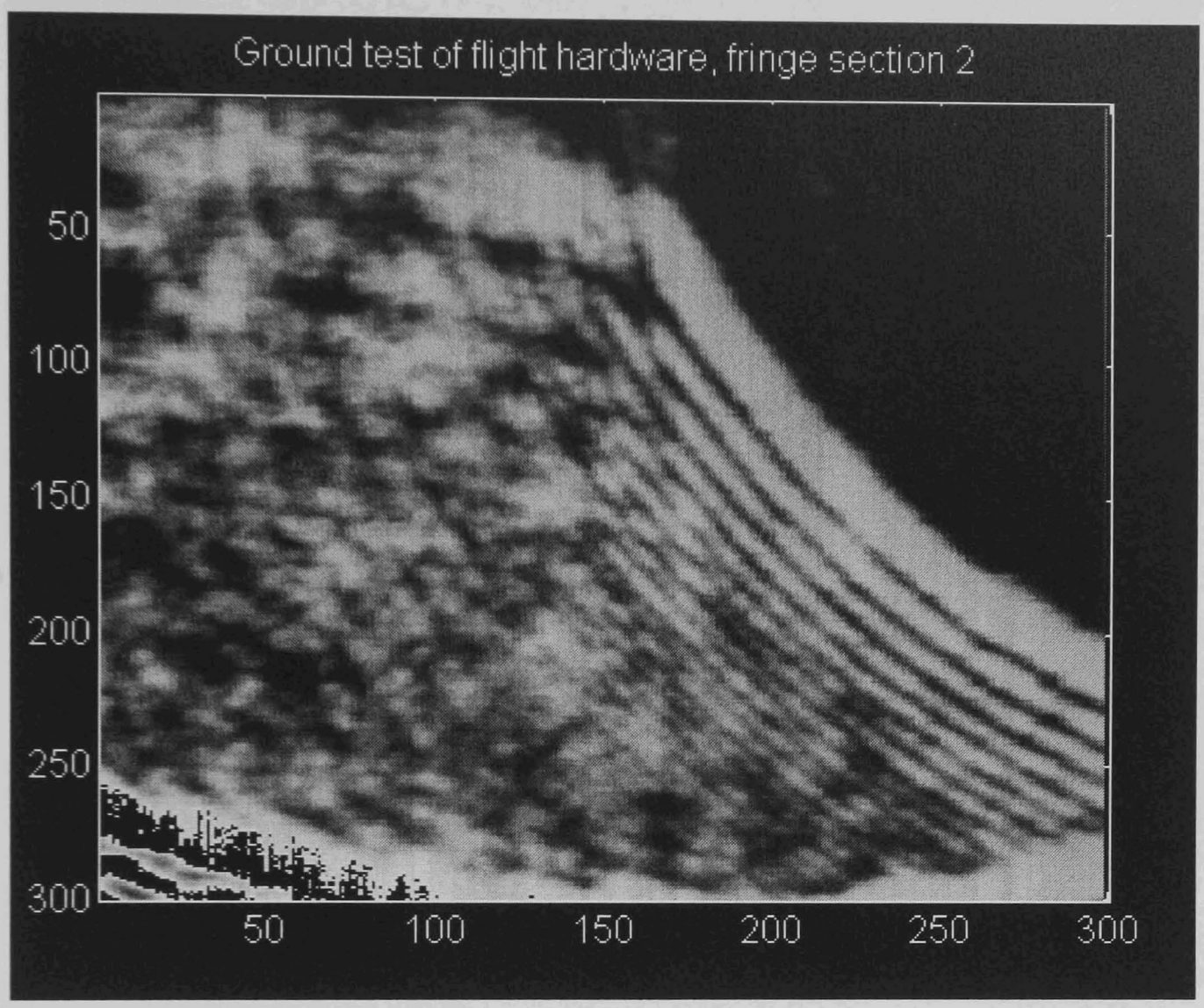


Figure 7.17

The second section of the ground test generated image. This is the section from below the first part, the other side of the area of the fringes masked by the heating element support wire. The heating element is slightly above and to the right of the upper right-hand corner of this image.

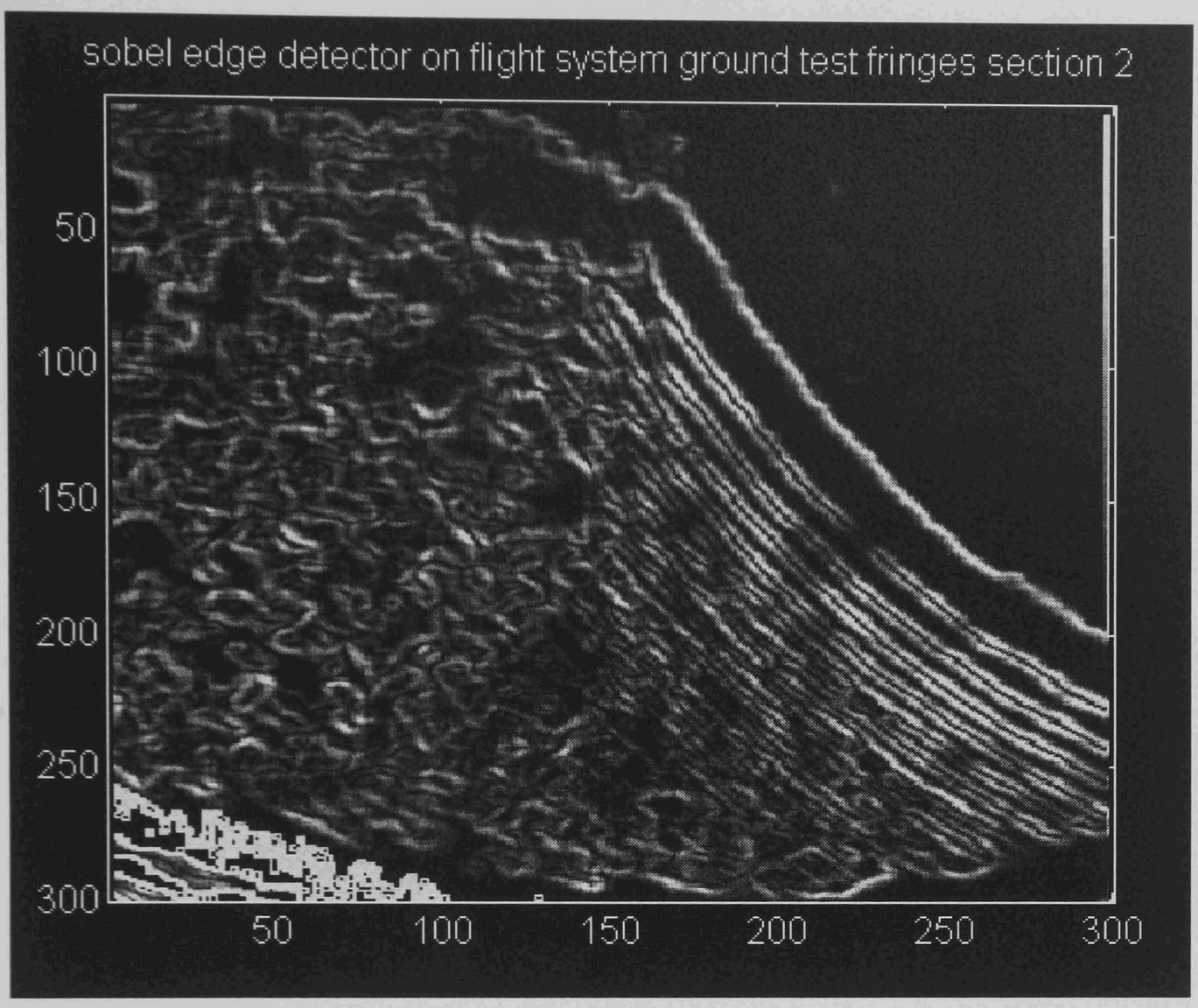


Figure 7.18

Image Type : Earth's gravity pre-flight test of flight hardware.

Comments : Sobel edge detector on the second images section. Again the results are reasonable. However the noise detections are again hard to distinguish from the fringe detections without further intelligent interpretation.

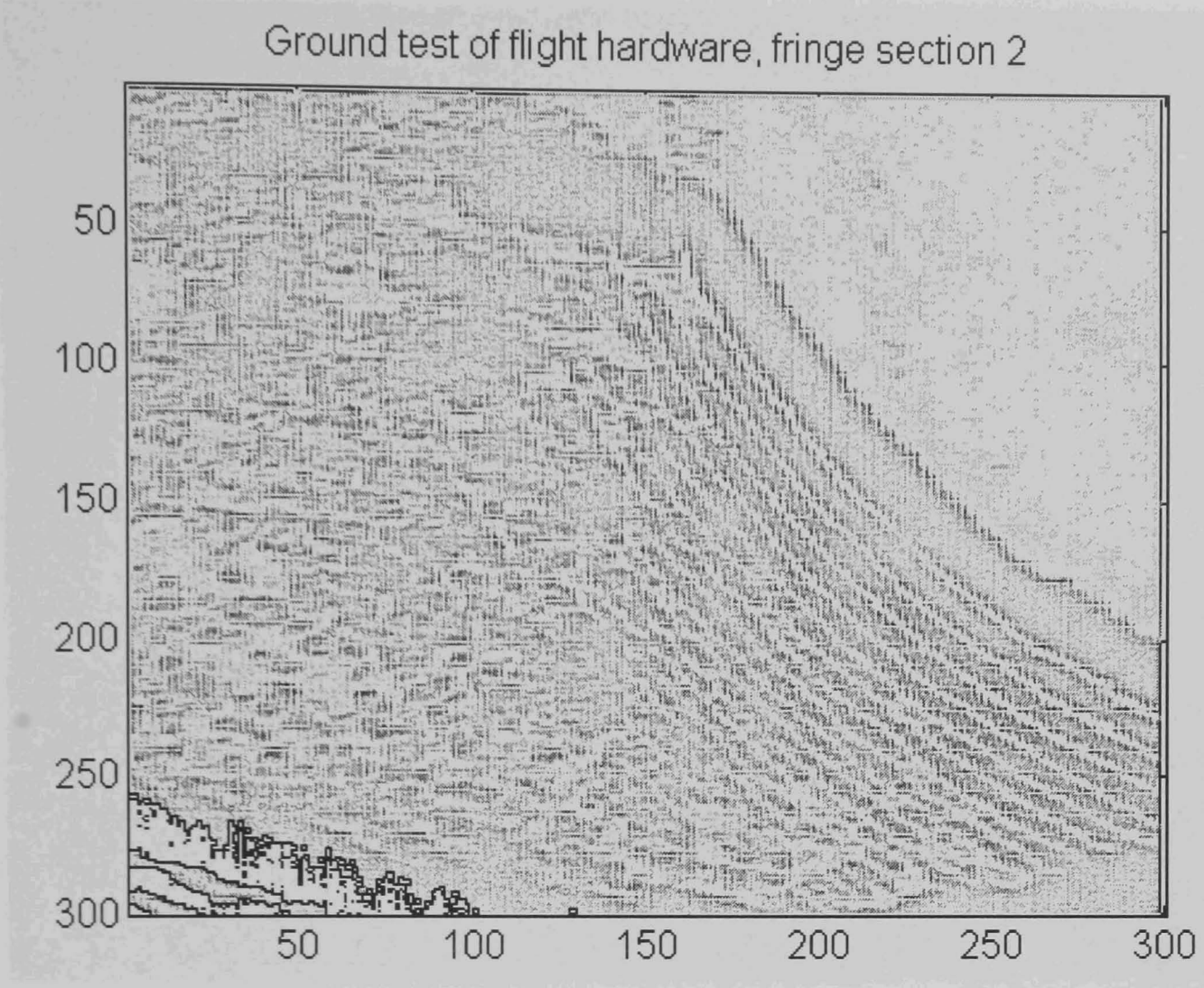


Figure 7.19

Image Type : Earth's gravity pre-flight test of flight hardware.

Comments : Commercial edge enhancement algorithm applied to the section fringe section, Figure 7.17. This is once more a poor result.

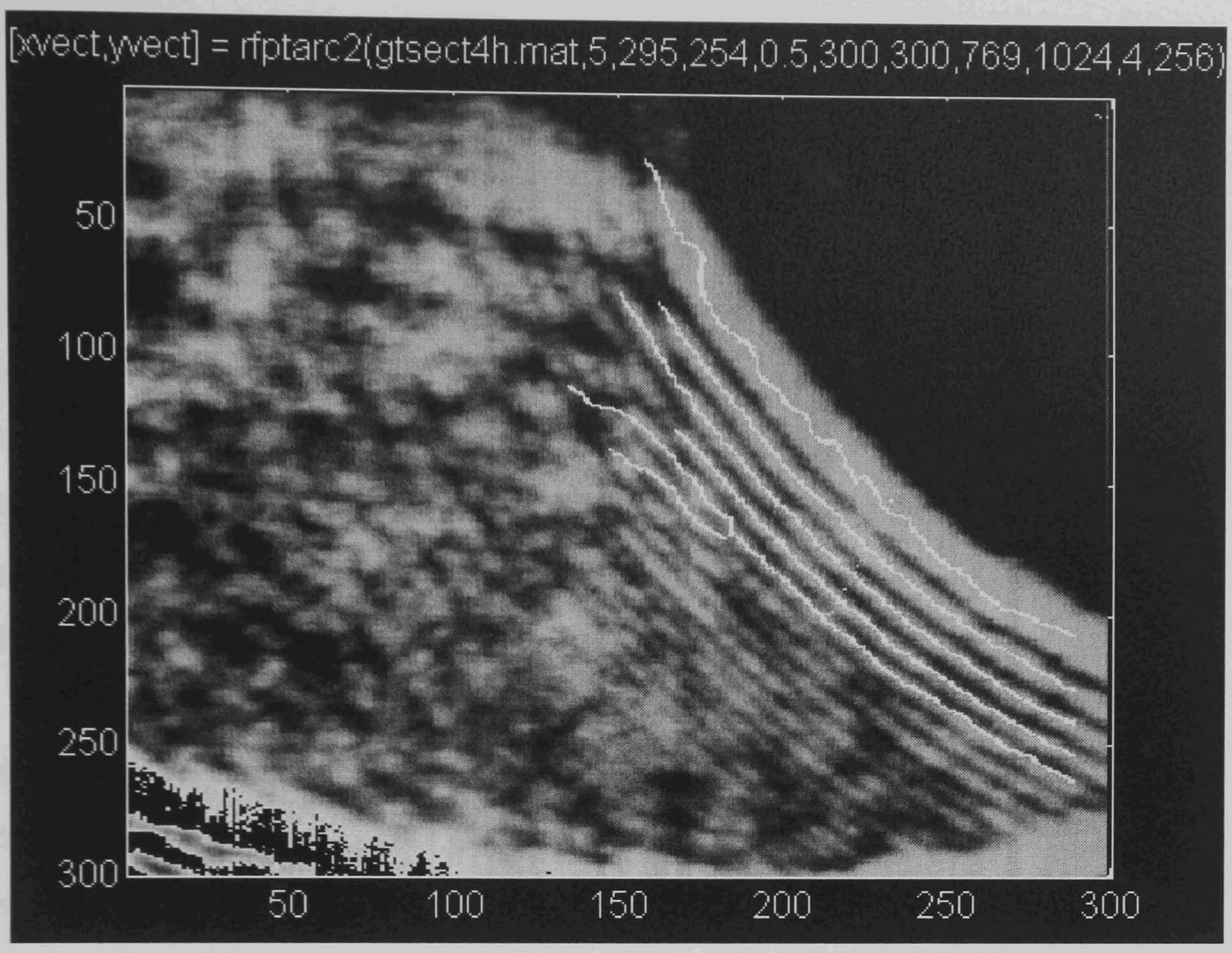


Figure 7.20

Image Type : Earth's gravity pre-flight test of flight hardware.

Starting co-ordinates : Row = 5 : Column = 295

Comments : Figure 7.20, result for the image processing system on the second fringe section. Despite the fact that the final fringe has been detected twice this is superior to both the standard and commercial detector results. The noise rejection is good, the results are firm fringe detections and they are complete. This last is most applicable when compared with Figure 7.18 where one section of the first fringe detection is very faint.

7.3.3 Conclusions On Existing Operators

It is noted that the Sobel and Robert's Cross operators are not the best direct comparisons with the general processing system that has been proposed. They are however easy to implement and test on the same images unlike the majority of operators presented in the literature where considerable detailed work would be required to expand the mathematical descriptions of systems presented into implemented algorithms. However by reviewing the results of the simpler operators and results presented in the literature the general applicability of other operators can be inferred.

The most relevant operators are the 'roof and valley' detectors, 'robust edge' detectors and varied transform and filter based edge, and fringe detection, operators that have been presented in the literature. A summary of the literature review of this area of the literature was presented in Chapter 3, Section 3.1.5. Particularly it is reasonable to interpolate that the composite edge detector, Ghosal [121], and the sub-pixel accuracy detector using local energy minimisation, Kisworo [122] would perform very well on the high contrast images generated by the flight equipment. These operators are considerably more advanced than the edge detection operators tested and overcome one of the principal problems of the earlier mask based operators in that they provide a edge or no edge output rather than a edginess response which must be thresholded. However a line tracing algorithm would still be required to process the output. As the testing of operators such as the tangent fringe linker has demonstrated identifying the difference between errors due to gaps in the fringes and corruption of complete fringes by noise using such operators is difficult. It is therefore considered unlikely that complete fringes could be reliably extracted from the images by this approach as isolated features without operator assistance. For a set of similar images though image composition dependant parameters could be determined and an efficient processing system based on existing work in the published literature could be anticipated to operate satisfactorily. The robustness of such a system given general images of unknown quality is doubtful however.

7.3.4 Conclusions on the Proposed Operator

The proposed interferogram image processing operator has performed well on the test images. Some minor problems have been observed but where complete fringe sections were present in the images they have been reliably detected and identified as fringe features. Similarly the noise present in the test images has been identified as such and rejected. This is despite the considerable variation in the widths, shapes and intensities of the fringes observed in the images. The following Section 7.4 demonstrates the operation of the proposed algorithm on flight results generated aboard the Space Shuttle Endeavour.

7.4 Flight Operations In Microgravity

7.4.1 Flight Operations

The flight images are digitised frames from the video tape recorded by the flight experiment. These images were recorded during the flight by the system under the control of the experiment control unit. Analysis of the payload data logging systems after flight has shown that the system operated as planned, (see Section 6.2, flight plan) and that the thermal environment in the payload was within the equipment specifications throughout.

7.4.2 The Flight Video Tape

Prior to presenting the results of testing the image processing system on the flight results, a summary of the data collected by the flight experiment is presented. This overview gives an indication of the difficulties encountered in processing the flight data, from observing the quality. This section also comprises a summary of the results of the test of the novel microgravity measurement system, this being the research objective of the flight experiment.

The results on the flight video tape can be divided principally into four groups of image frames. These correspond to four phases of the experiments operations as detailed in the flight plan (see Section 6.2). These sections are :

1 • The ten minute period of quiescent data from the first experiment, during which no changes were observed. See Figure 7.21

2 • The thirty second period after the heating element was activated in which bright fringes are clearly observed as the experiment cell goes through its initial 'warming up' phase. Images from this phase are used in Section 7.4.3 to demonstrate the operation of the image processing research on fringe images generated in space. See Figures 7.22 to 7.26.

3 • The remaining nineteen minutes of the first experiment. See Figures 7.27 to Figure 7.31.

4 • The second experiment. As the heating element was activated prior the camera, the initial 'warming up' phase seen during the first experiment was not observed. However the total duration of the second experiment was greater and it therefore provides a better indication of the long-term performance. During this period, a second very faint fringe can be observed in the upper region of the image frame slowly expanding as time goes by. Frames from this period are shown in Figures 7.32 to 7.35

Multiple sets of fringe are seen due to the laser diode module which provided the illumination producing multiple output beams. The effect of this phenomena is minimal as the apparatus was adjusted prior to flight to

use only one of the sets of fringes generated. The other fringe sets are generally outside the area of the CCD screen.

In each image the vertical image axis corresponds to the towards/away from the Earth axis, and the horizontal axis equates to the fore/aft axis of the Space Shuttle Orbiter. Hence vertical displacements equate to variations in gravitational field and similarly the principal atmospheric drag component relates to horizontal disturbances relative to the image plane.

The darkening of the upper two thirds of the flight image frames is suspected to be due to the slight displacement of the interferometer components due to vibration during launch. As the atmospheric re-entry could have further disturbed the apparatus, however, a definitive answer as to the cause of this phenomena is not possible.

A series of representative image frames from the flight tape are now presented. These are in chronological order but concentrate on the most dynamic periods in the flight data. These are full frames. The image processing results which follow (Section 7.4.3) show only the lower third of a frame.

Frames from the Flight Video Tape

Experiment #1

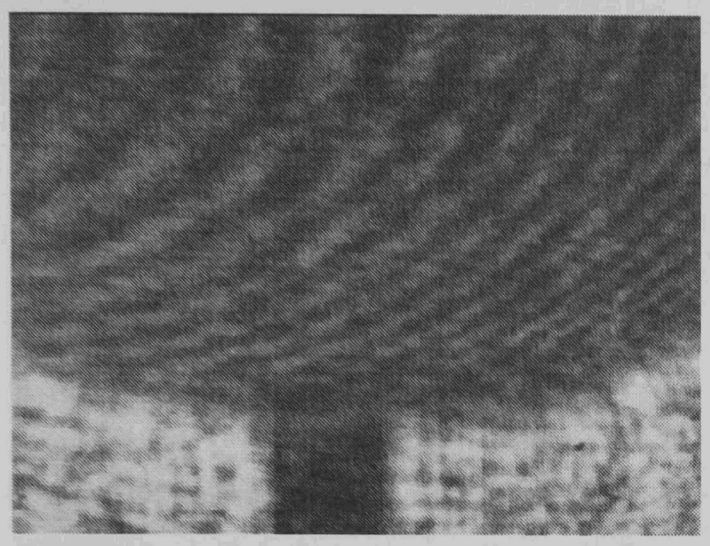


Figure 7.21
Quiescent Experiment #1

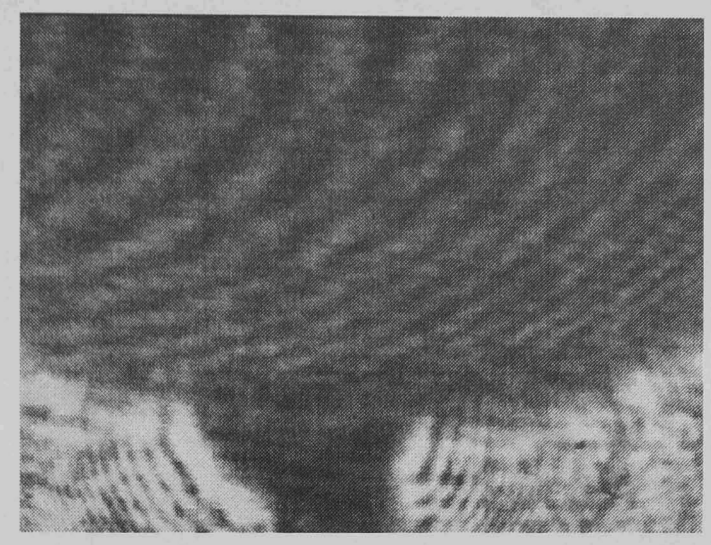


Figure 7.22
Heating commences Experiment #1

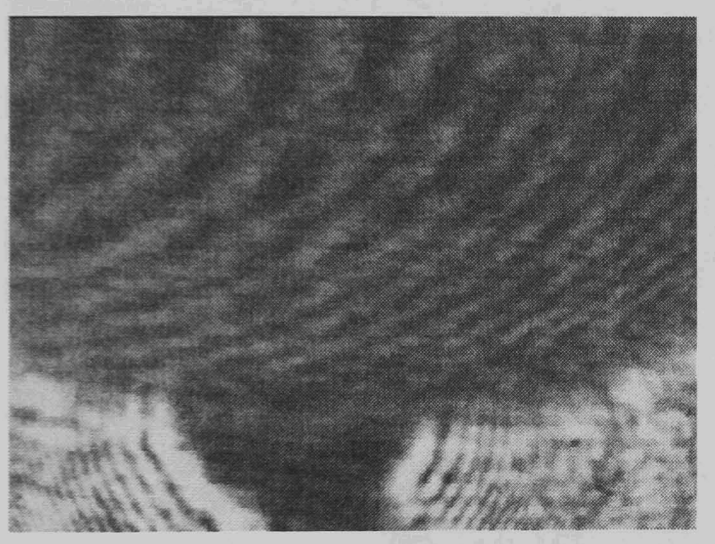


Figure 7.23

Heating continues

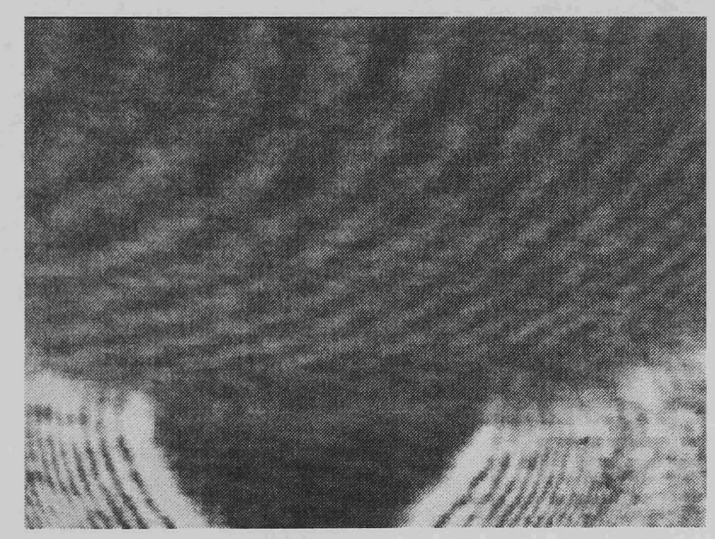


Figure 7.24

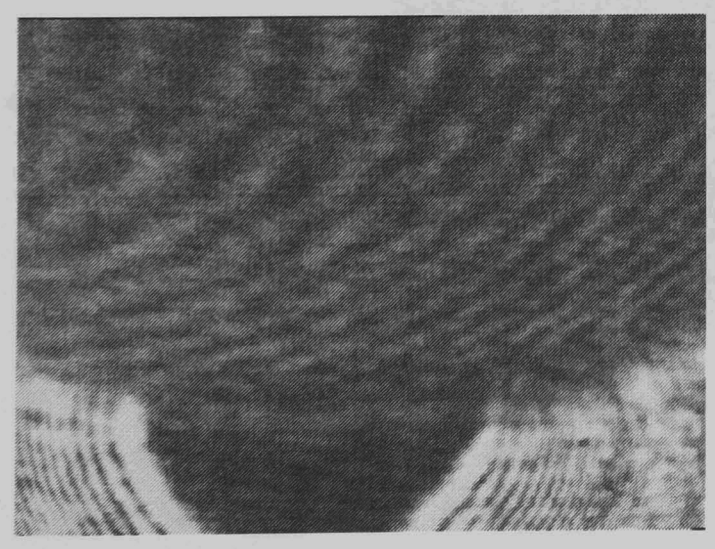


Figure 7.25

Heating continues

Figure 7.26

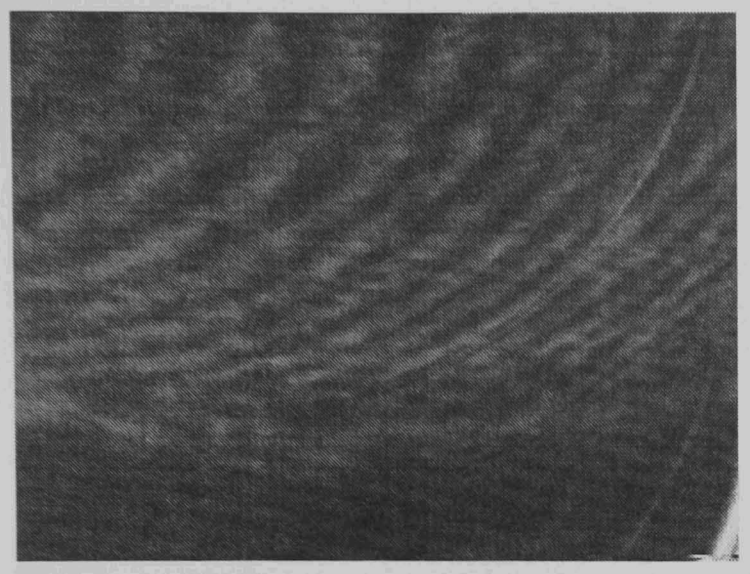


Figure 7.27 Heating continues, beyond 1 minute Figure 7.28

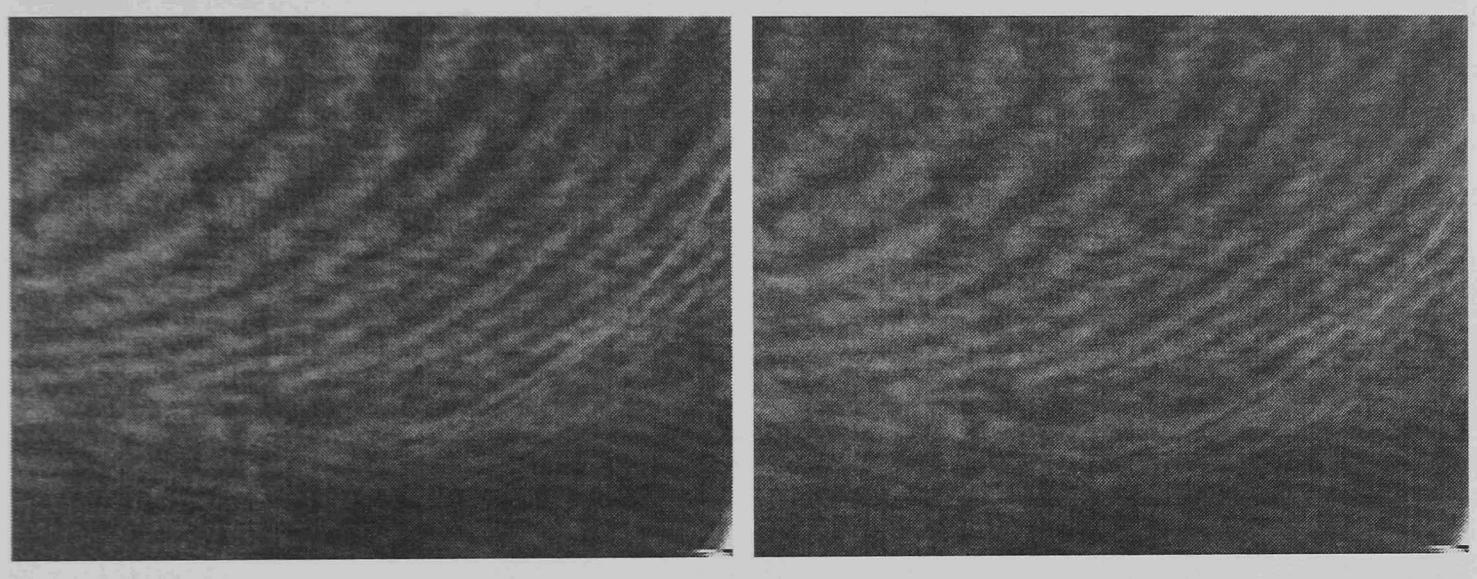


Figure 7.29 Heating continues Figure 7.30

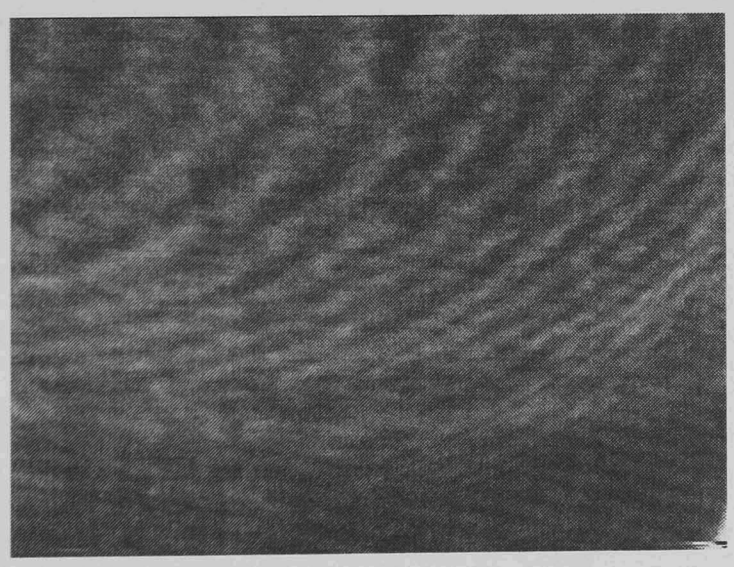


Figure 7.31 Heating continues

Experiment #2

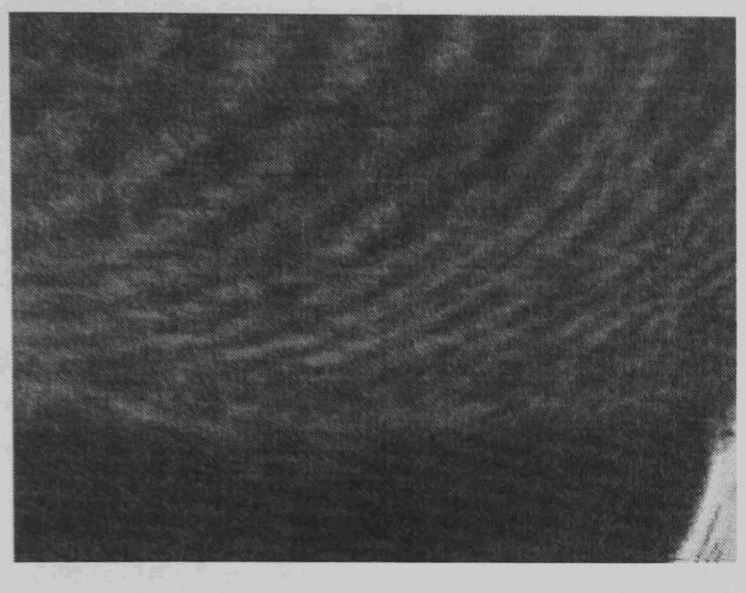


Figure 7.32
Start of second experiment

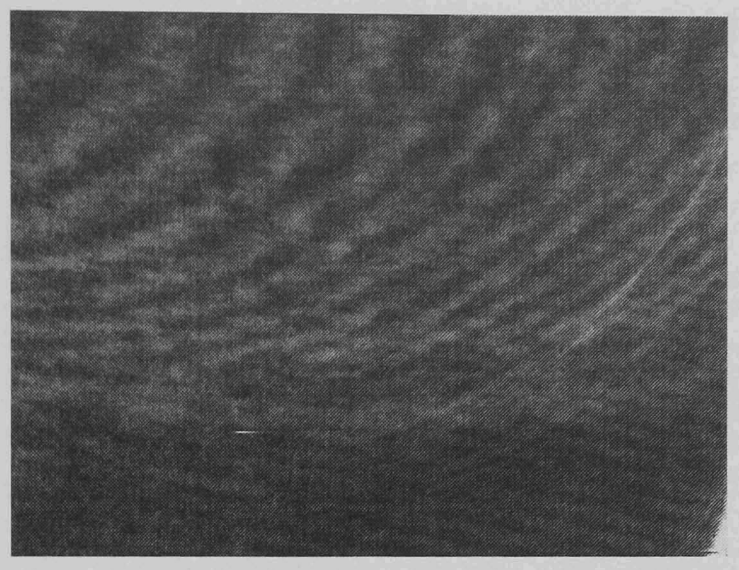


Figure 7.33
Heating continues

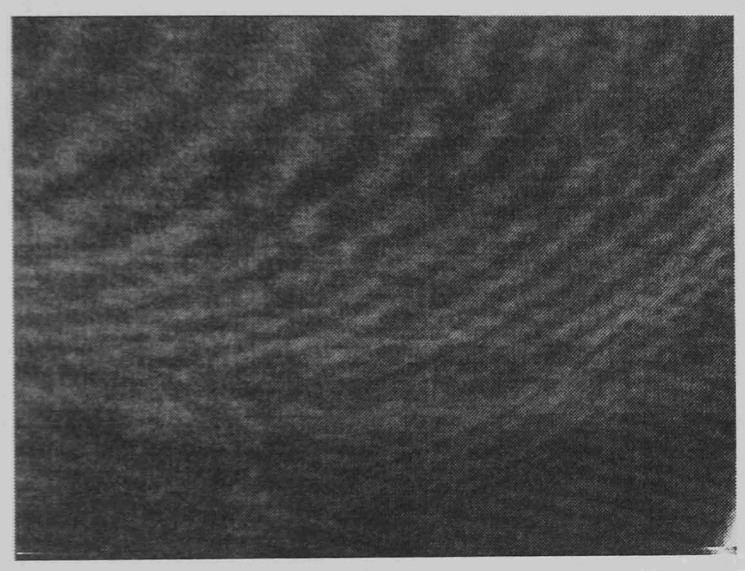


Figure 7.34

Heating continues

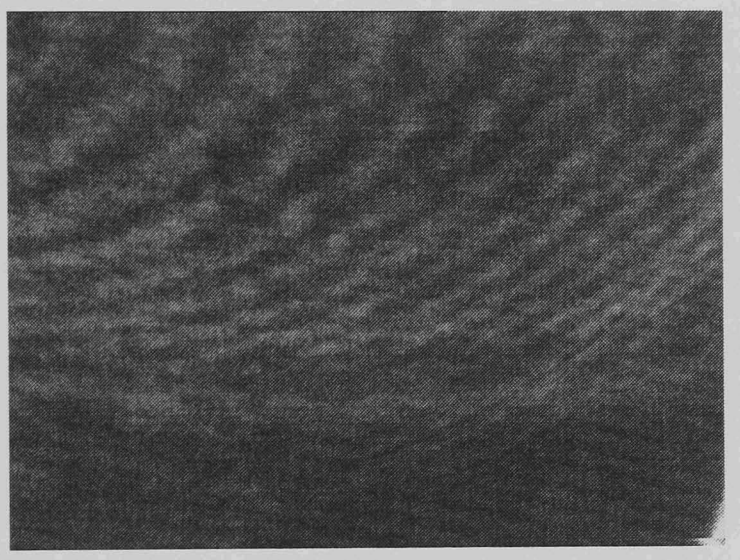


Figure 7.35

7.4.3 Image Processing of The Flight Results

A sequence of frames from the flight results is presented. These were taken in quick succession from the video tape and are from the period just after the heating element was activated during the first experimental operation. In each case, a 'raw' fringe image which corresponds to approximately the lower third of the video frame is presented. Only the lower section is shown as this is where the fringes were observed. Two sections of this fringe image are then shown with the fringes that were detected and identified by the proposed image processing system highlighted. The heating element is approximately central on the upper edge of the 'raw' fringe images. The gap between the right-hand and left-hand fringe sections is due to the obscuring effect of the heating element support wires.

The 'limited number slices' version of the processing system introduced in Section 7.3.1 was used to process the flight results. This is the same system as was used for processing the ground test data. Each of the left-hand image sections was processed by the same slice pattern, and similarly each right-hand section was processed by the same slice pattern. This highlights the flexibility of the system, sections of the images having been taken simply to avoid unnecessary processing and to present the resulting detections in a standard format. The fact that each set of three sections could be processed by the same operator confirms that the assumption made, viz. that multiple images from a single experimental arrangement could use the same processing, is valid. Each subsection is 150 rows by 200 columns.

Overall Flight Image : Frame 1

Figure 7.36

Left-hand Fringe Section #1

Figure 7.37

Right-hand Fringe Section #1

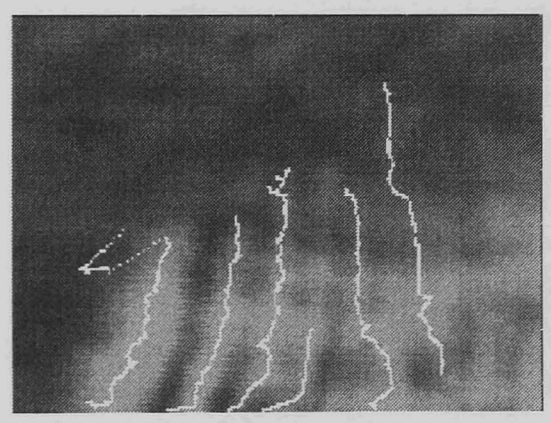


Figure 7.38

Refering to Section 7.3.1.

Processing the left-hand section

[x,y] = rfptarc3('flight1sec1',100,198,642,0.2,150,200,641,896,4,256);

Processing the right-hand section

[x,y] = rfptarc3('flight1sec2',120,2,383,0.2,150,200,129,384,4,256);

Overall Flight Image : Frame 2

Figure 7.39

Left-hand Fringe Section #2

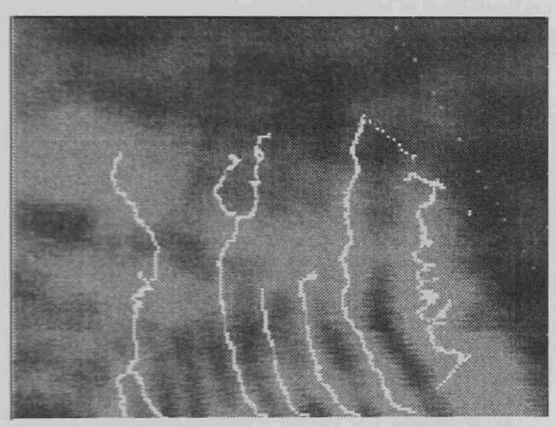


Figure 7.40

Right-hand Fringe Section #2

Figure 7.41

Refering to Section 7.3.1.

Processing the left-hand section

[x,y] = rfptarc3('flight2sec1',100,198,642,0.2,150,200,641,896,4,256);

Processing the right-hand section

[x,y] = rfptarc3('flight2sec2',120,2,383,0.2,150,200,129,384,4,256);

Overall Flight Image : Frame 3

Figure 7.42

Left-hand Fringe Section #3

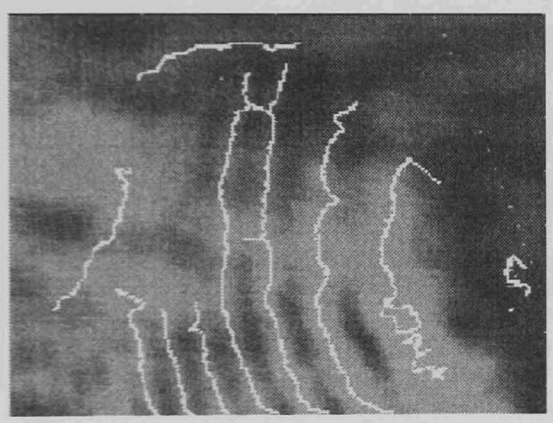


Figure 7.43

Right-hand Fringe Section #3

Figure 7.44

Refering to Section 7.3.1.

Processing the left-hand section

[x,y] = rfptarc3('flight3sec1',100,198,642,0.2,150,200,641,896,4,256);

Processing the right-hand section

[x,y] = rfptarc3('flight3sec2',120,2,383,0.2,150,200,129,384,4,256);

7.4.4 Summary of Processing the Flight Results

The three image sections, Figure 7.36, Figure 7.39 and Figure 7.42 represent three snapshots in time in quick succession. They are therefore similar but it is notable that from frame to frame, the fringes do become clearer in the later frames. This is reflected in the better fringe detection achieved in the later frames. Although the detected fringes are moderately noisy due to the fact that short fringe sections have been retained, the indistinct fringes near the image edges have been ignored and in each case the clear fringes have been correctly detected.

Overall, the proposed system has been successfully demonstrated on a varied selection of images. In each case it has performed well and generally outperformed existing techniques. Finally it has been operated successfully on a selection of the space flight experiment video image frames that it was developed to process. Gravity vectors for the flight frames have not been generated as only small fractions of fringes were observed. The capability to extract the fringes, and gravity vectors therefrom, if improved images could be obtained has however been demonstrated.

Chapter 8

Conclusions

8.1 Summary Of The Research

8.1.1 Project Overview

A review of existing image processing techniques has been undertaken and a selection of methods have been tested on the simulated microgravity convection images, obtained during testing. Techniques for extracting data from these images have been developed using image processing techniques. A space flight experiment has been constructed and flown as a payload aboard the Space Shuttle, and has obtained microgravity convection results for analysis.

8.1.2 Summary of Achievements

A review of the existing literature has been undertaken. The techniques presented in the literature have either been implemented and tested directly, on simulated interferogram images, or have been assessed based on the reports of their capabilities and knowledge of the performance of similar approaches. Of particular relevance were, the automatic detection of ventricular cavities by Friedland [114], and the, dilation and thinning based approach to the grouping of fringe point detections used by Eichhorn [145]. Techniques from these sources were utilised in the development of the image processing system for gravity vector extraction. In the latter case, further refinement was carried out resulting in the elimination of a limitation of the reported technique. This review demonstrated that existing image processing techniques were not suitable for the fully automated extraction of fringes, as identified objects, from interferogram images containing concentric grey-scale fringes.

A processing system has been developed from the image processing research performed that carries out complete fringe extraction from the simulated microgravity interferogram images. The development of this system was based on images produced during the conceptual development and simulation of the microgravity experiment proposal. The system, as desired, is capable of extracting fringes from a wide variety of images. It performs this task with a minimum of assistance from the operator, correctly rejects noise, and returns fringes which are not just 'detected points', but identified 'fringe objects'. This expands the scope and simplicity of further processing of the images for data extraction. The extraction of a 'gravity vector' from the movement of the fringes was achieved where complete, or nearly complete, closed fringes were present. This constitutes an example of the suitability of the 'fringe object' results for the application of further analysis of the images. The resulting vector agrees with the visible distortion of the fringe images due to natural convection. These features are demonstrated in the results presented in Chapters 5 and 7, utilising the concept testing images.

Initially images were obtained from two sources, namely from the initial tests performed to demonstrate the concept of using convection for measuring microgravity and later from testing of the flight payload in Earth's gravity.

In parallel with the research on image processing, a flight experiment has been designed, constructed and flown on the Space Shuttle Endeavour (flight STS-77 19th to 29th May 1996). This successfully acquired flight microgravity data and the image processing system has been tested on these results as discussed in Chapter 7. The operability of the algorithm on the concept testing, ground test and flight images, with no alterations to the processing system itself demonstrates the flexibility of the system. This demonstrates the resulting system fulfilment of the requirement that it enable the automatic processing of many similar frames without operator assistance.

It was proposed that the visual monitoring and analysis of interferogram fringe patterns could be used as a sensitive method of measuring fluctuations in microgravity. Concept testing proved that this system could be made functional. The construction and flight of the space payload has demonstrated the operation of a convection current generation and imaging system in space. An image processing

system has been developed that is capable of automatic analysis of fringe images and the extraction therefrom of a gravity vector.

Due in part to differences in performance between the flight equipment and the results on which the processing system was developed, the extraction of the entire set of gravity data from the flight images remains incomplete at this time. The systems required for a complete gravity measurement facility have however been successfully demonstrated. This result is taken to be a further step in the advancement of knowledge on the processing of fringe images and the investigation of methods of obtaining improved gravity maps.

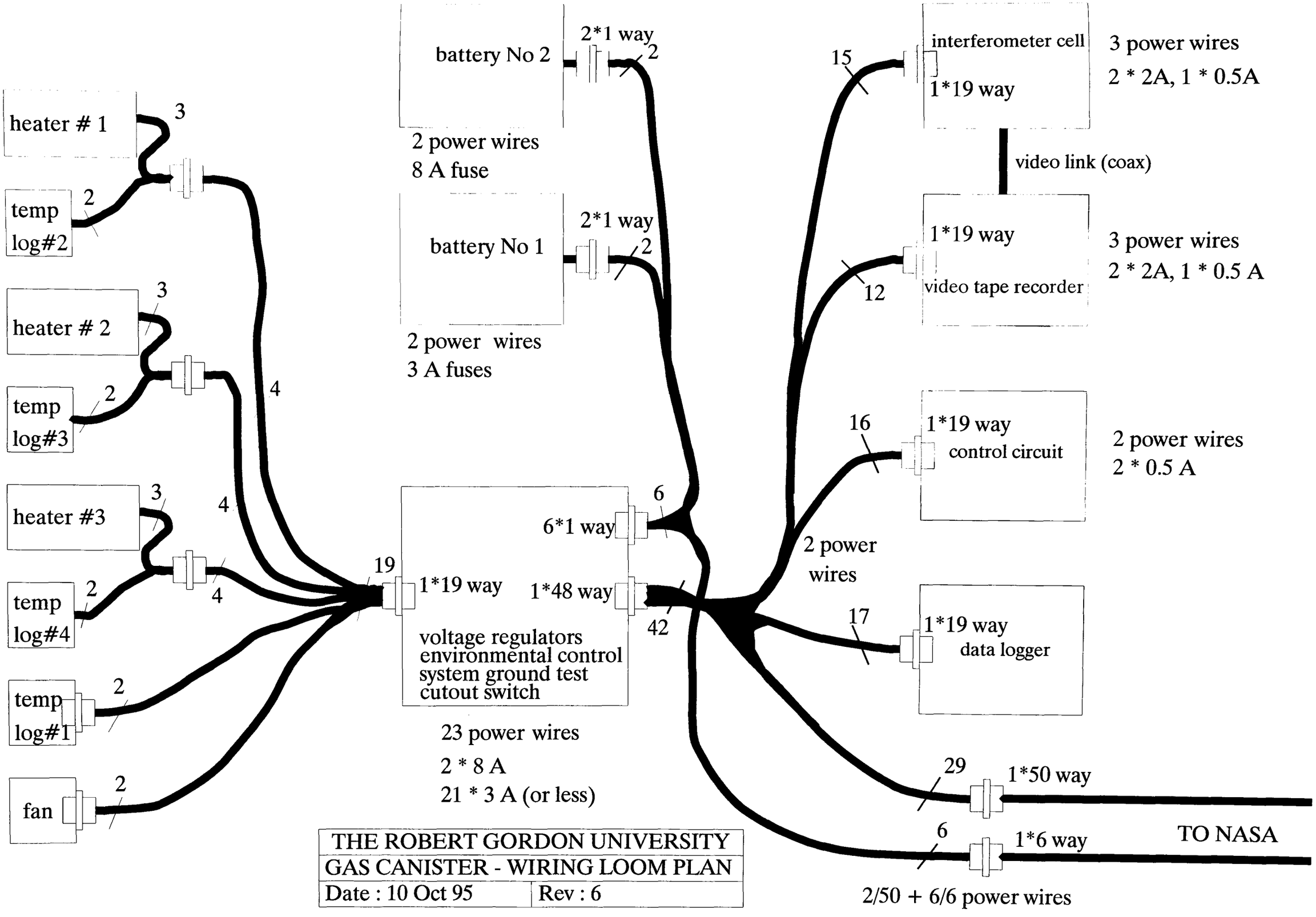
8.2 Further Work

Further flight data remains to be analysed. The principal aim of this assessment would be to investigate the possibility that the fringe images recovered do contain data on gravity fluctuations experienced by the payload during its operation in orbit. It was postulated that the detection of these fluctuations would enable gravity mapping to be carried out. The image processing system developed from the research should be of assistance in this task. However an image enhancement system tailored to the flight results will be required as a pre-processing stage due to the quantity of noise disturbances seen in the flight data.

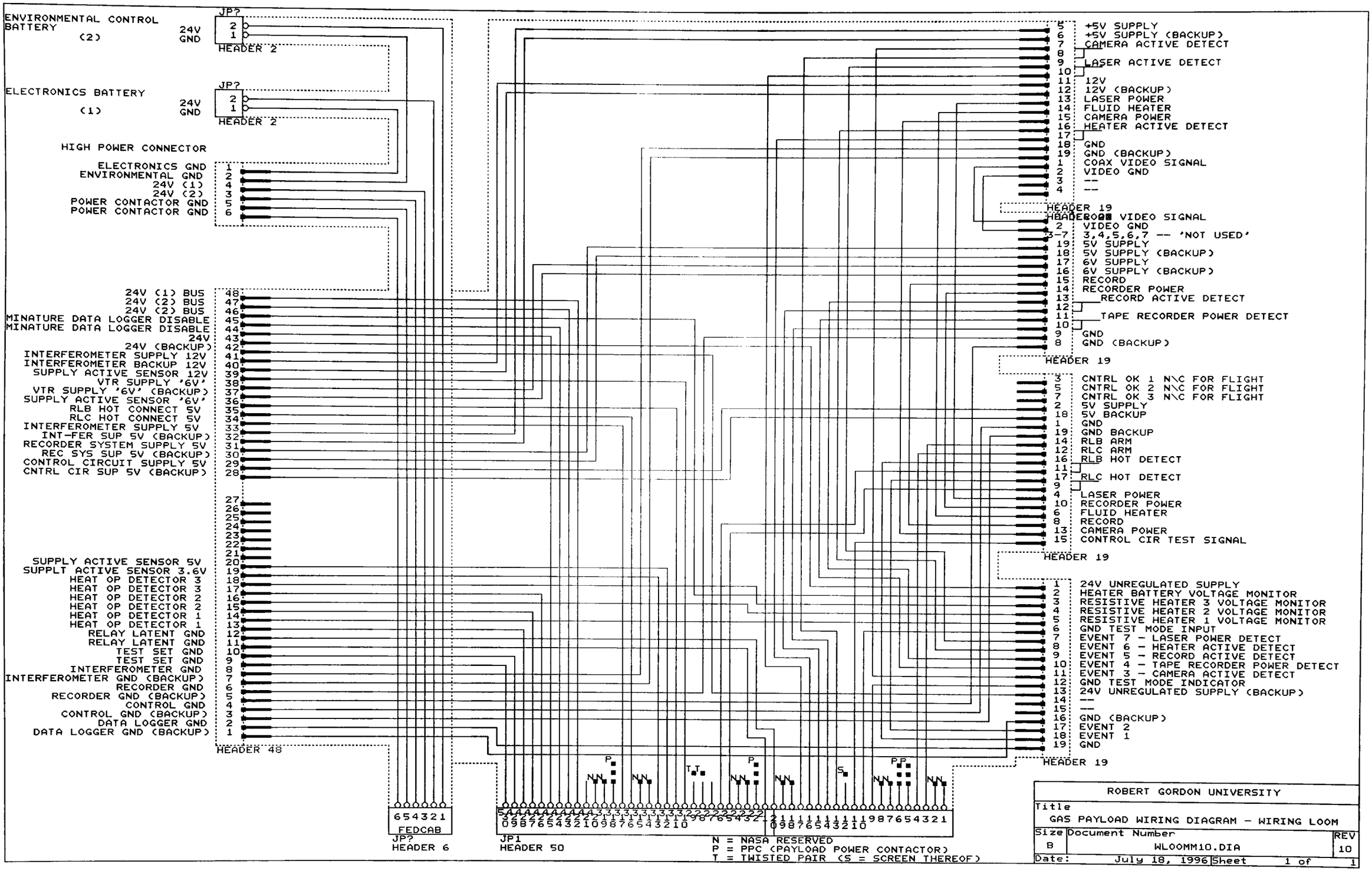
Further to the extraction of a gravity vector from the images, it remains necessary to carry out further work in order to calibrate this method of gravity measurement in a controlled environment. This technique should also be applicable to normal, and increased gravity conditions. In these cases however, accelerometers may be more applicable. This awaits further investigation to establish experimental results and conclusions.

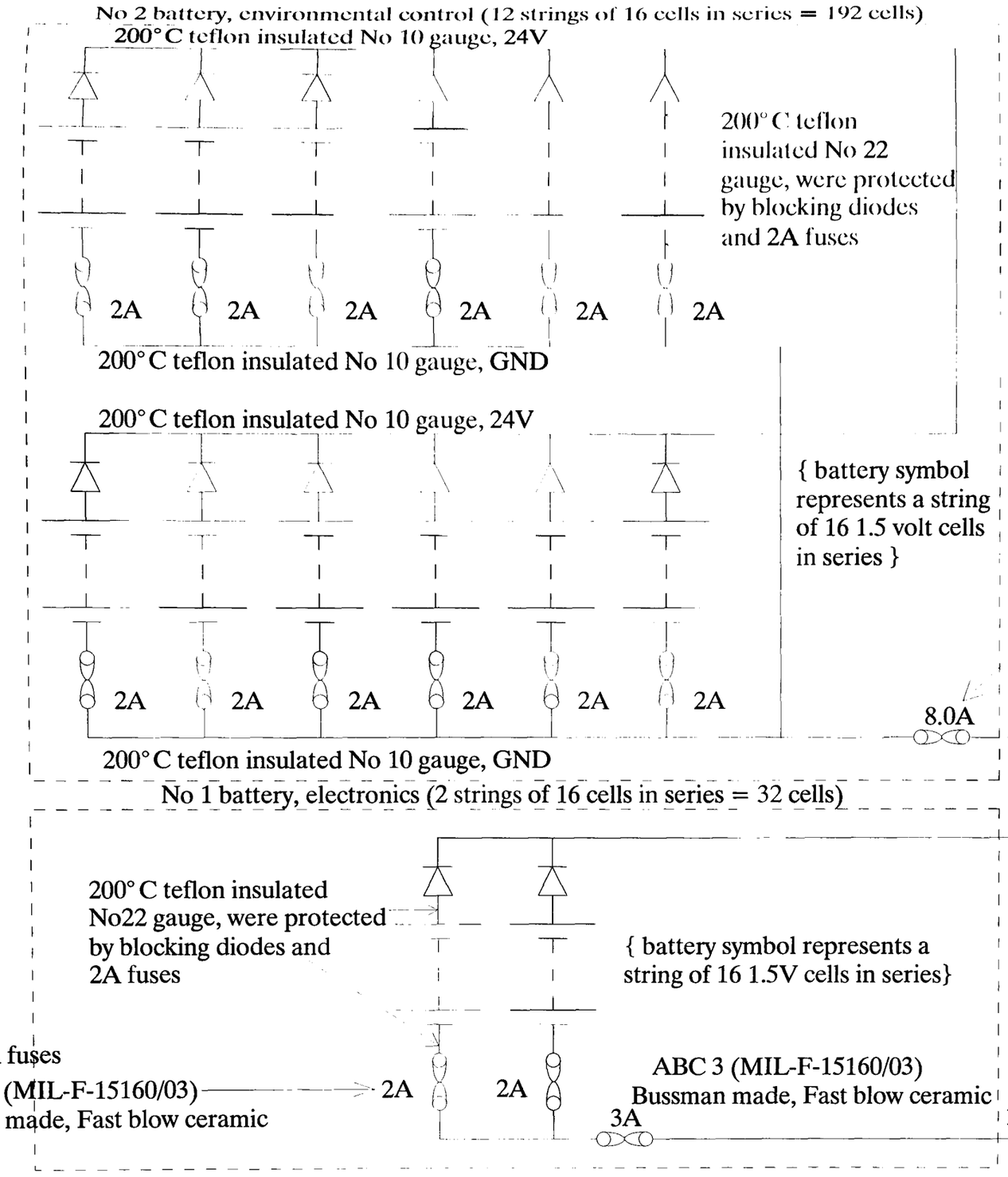
Appendix 1

Circuit Diagrams



THE ROBERT GORDON UNIVERSITY	
GAS CANISTER - WIRING LOOM PLAN	
Date : 10 Oct 95	Rev : 6


ENVIRONMENTAL CONTROL BATTERY (2) 24V GND
JP?
HEADER 2
ELECTRONICS BATTERY (1) 24V GND
JP?
HEADER 2
HIGH POWER CONNECTOR
ELECTRONICS GND 1
ENVIRONMENTAL GND 2
24V (1) 4
24V (2) 3
POWER CONTACTOR GND 5
POWER CONTACTOR GND 6
24V (1) BUS 48
24V (2) BUS 47
24V (2) BUS 46
MINATURE DATA LOGGER DISABLE 45
MINATURE DATA LOGGER DISABLE 44
24V 43
24V (BACKUP) 42
INTERFEROMETER SUPPLY 12V 41
INTERFEROMETER BACKUP 12V 40
SUPPLY ACTIVE SENSOR 12V 39
VTR SUPPLY '6V' 38
VTR SUPPLY '6V' (BACKUP) 37
SUPPLY ACTIVE SENSOR '6V' 36
RLB HOT CONNECT 5V 35
RLC HOT CONNECT 5V 34
INTERFEROMETER SUPPLY 5V 33
INT-FER SUP 5V (BACKUP) 32
RECORDER SYSTEM SUPPLY 5V 31
REC SYS SUP 5V (BACKUP) 30
CONTROL CIRCUIT SUPPLY 5V 29
CNTRL CIR SUP 5V (BACKUP) 28
27 26 25 24 23 22 21
SUPPLY ACTIVE SENSOR 5V 20
SUPPLT ACTIVE SENSOR 3.6V 19
HEAT OP DETECTOR 3 18
HEAT OP DETECTOR 3 17
HEAT OP DETECTOR 2 16
HEAT OP DETECTOR 2 15
HEAT OP DETECTOR 1 14
HEAT OP DETECTOR 1 13
RELAY LATENT GND 12
RELAY LATENT GND 11
TEST SET GND 10
TEST SET GND 9
INTERFEROMETER GND 8
INTERFEROMETER GND (BACKUP) 7
RECORDER GND 6
RECORDER GND (BACKUP) 5
CONTROL GND 4
CONTROL GND (BACKUP) 3
DATA LOGGER GND 2
DATA LOGGER GND (BACKUP) 1
HEADER 48
654321
FEDCAB
JP?
HEADER 6
JP1
HEADER 50
5 +5V SUPPLY
6 +5V SUPPLY (BACKUP)
7 CAMERA ACTIVE DETECT
8
9 LASER ACTIVE DETECT
10
11 12V
12 12V (BACKUP)
13 LASER POWER
14 FLUID HEATER
15 CAMERA POWER
16 HEATER ACTIVE DETECT
17
18 GND
19 GND (BACKUP)
1 COAX VIDEO SIGNAL
2 VIDEO GND
3 --
4 --
HEADER 19
HEADER 8mm VIDEO SIGNAL
2 VIDEO GND
3-7 3,4,5,6,7 -- 'NOT USED'
19 5V SUPPLY
18 5V SUPPLY (BACKUP)
17 6V SUPPLY
16 6V SUPPLY (BACKUP)
15 RECORD
14 RECORDER POWER
13 RECORD ACTIVE DETECT
12
11 TAPE RECORDER POWER DETECT
10
9 GND
8 GND (BACKUP)
HEADER 19
3 CNTRL OK 1 N\C FOR FLIGHT
5 CNTRL OK 2 N\C FOR FLIGHT
7 CNTRL OK 3 N\C FOR FLIGHT
2 5V SUPPLY
18 5V BACKUP
1 GND
19 GND BACKUP
14 RLB ARM
12 RLC ARM
16 RLB HOT DETECT
11
17 RLC HOT DETECT
9
4 LASER POWER
10 RECORDER POWER
6 FLUID HEATER
8 RECORD
13 CAMERA POWER
15 CONTROL CIR TEST SIGNAL
HEADER 19
1 24V UNREGULATED SUPPLY
2 HEATER BATTERY VOLTAGE MONITOR
3 RESISTIVE HEATER 3 VOLTAGE MONITOR
4 RESISTIVE HEATER 2 VOLTAGE MONITOR
5 RESISTIVE HEATER 1 VOLTAGE MONITOR
6 GND TEST MODE INPUT
7 EVENT 7 - LASER POWER DETECT
8 EVENT 6 - HEATER ACTIVE DETECT
9 EVENT 5 - RECORD ACTIVE DETECT
10 EVENT 4 - TAPE RECORDER POWER DETECT
11 EVENT 3 - CAMERA ACTIVE DETECT
12 GND TEST MODE INDICATOR
13 24V UNREGULATED SUPPLY (BACKUP)
14 --
15 --
16 GND (BACKUP)
17 EVENT 2
18 EVENT 1
19 GND
HEADER 19
N = NASA RESERVED
P = PPC (PAYLOAD POWER CONTACTOR)
T = TWISTED PAIR (S = SCREEN THEREOF)
ROBERT GORDON UNIVERSITY
Title
GAS PAYLOAD WIRING DIAGRAM - WIRING LOOM
Size B
Document Number WLOOMM10.DIA
REV 10
Date: July 18, 1996
Sheet 1 of 1



24V

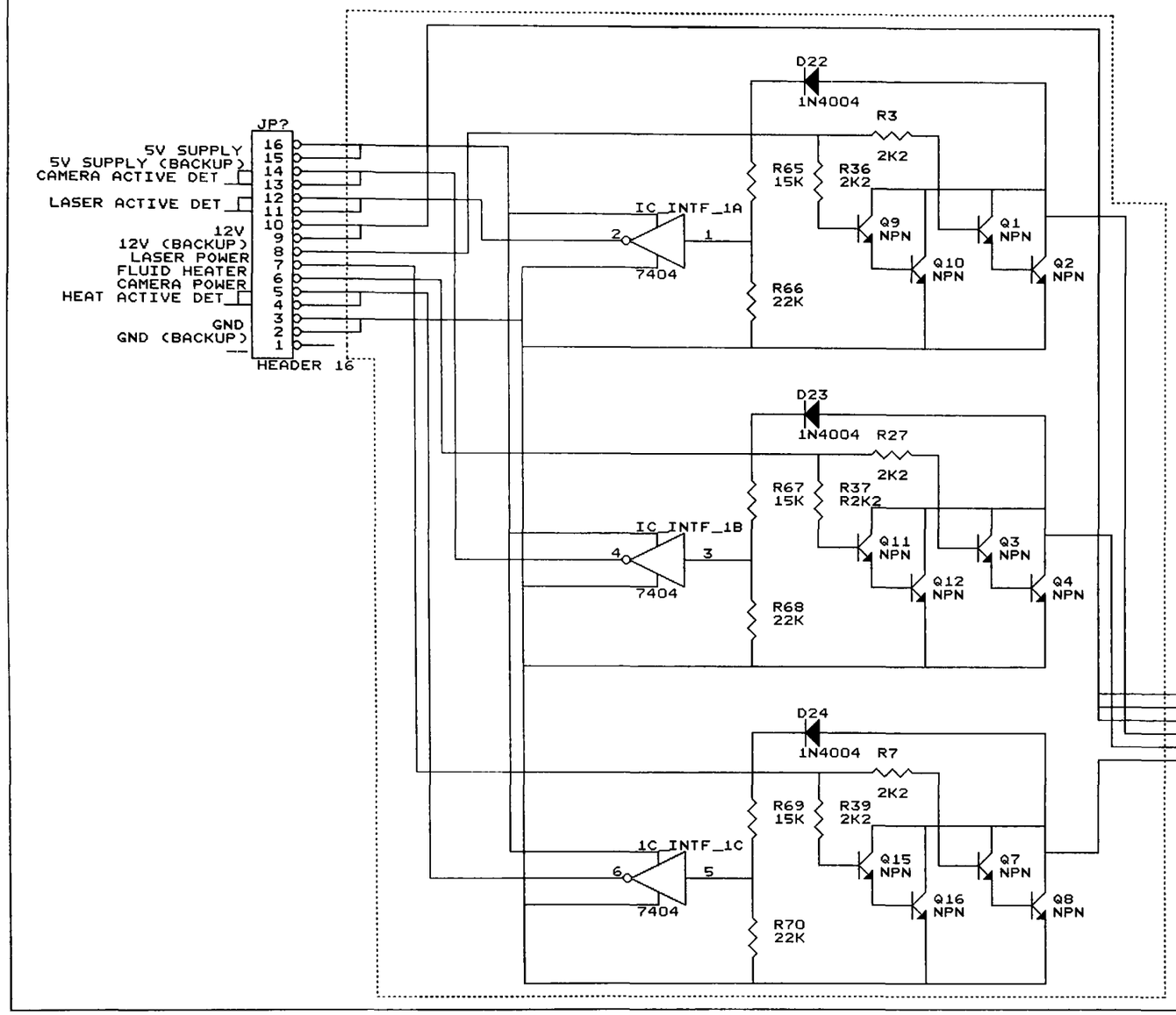
10 gauge wire environmental control system supply 200° C teflon insulated

ABC 3 (MIL-F-15160/03)
Bussman made, Fast blow ceramic

Notes :-

1. Supplies formed from 224 cells (No1 = 32, No2 = 192) giving 24V DC nominal.
2. Cells are 1.5V, 18Ah 'D' type alkaline manganese 'primary' types.
3. Fuses are placed as close as possible to the battery negative, within the battery box.

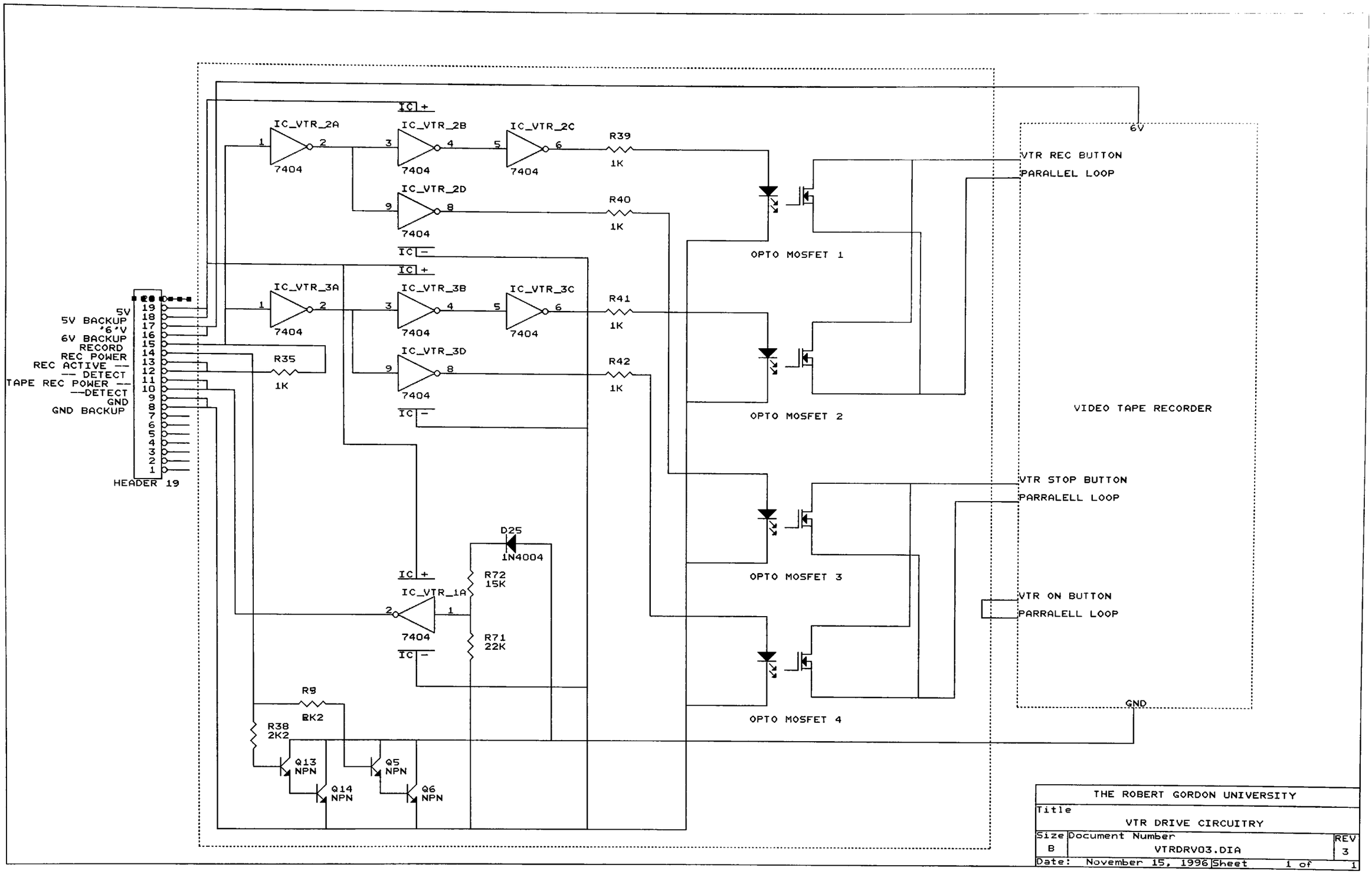
THE ROBERT GORDON UNIVERSITY G-490	
GAS CANISTER - BATTERY ARRANGEMENT	
Date : 19 Dec 95	Rev : 3

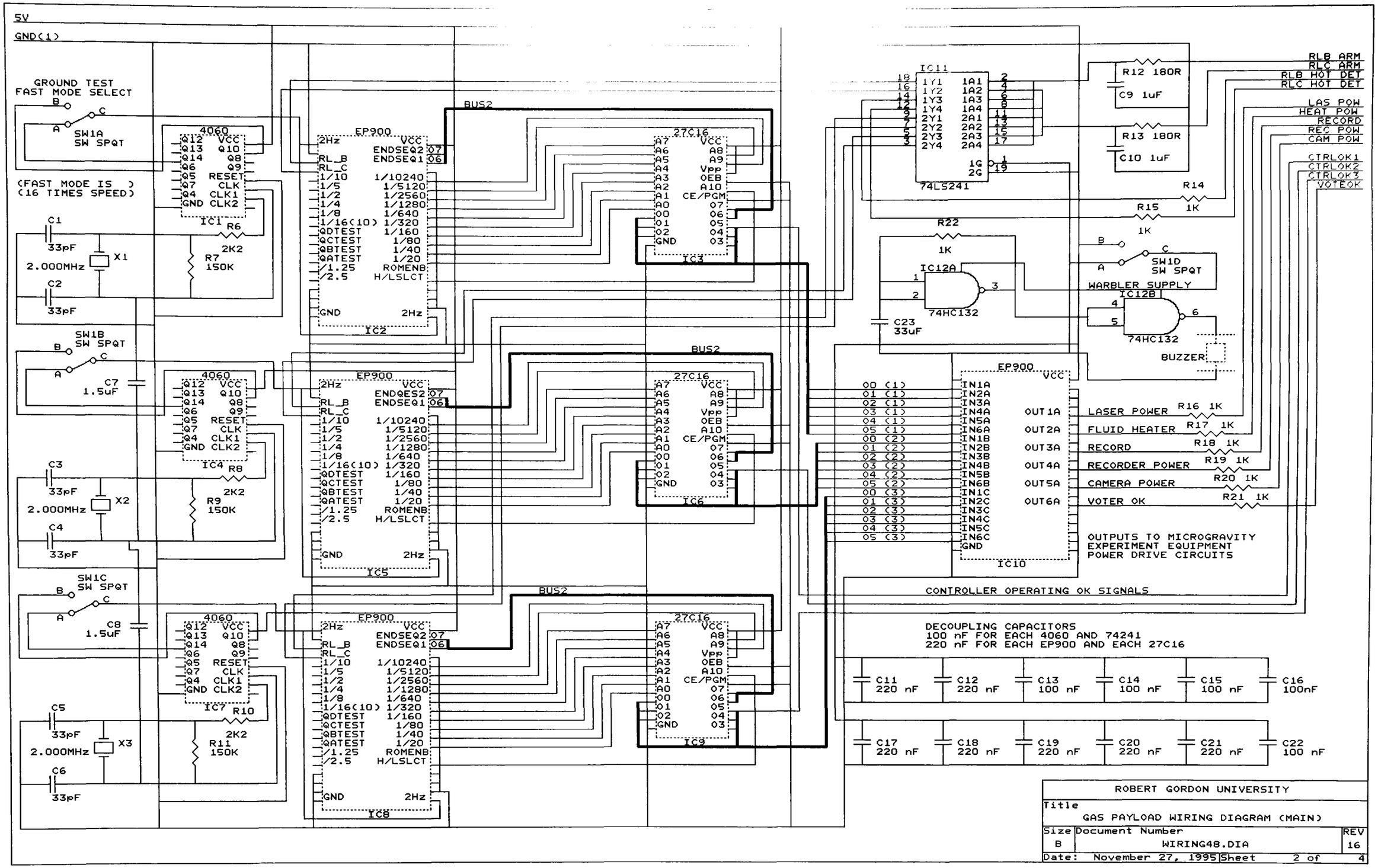


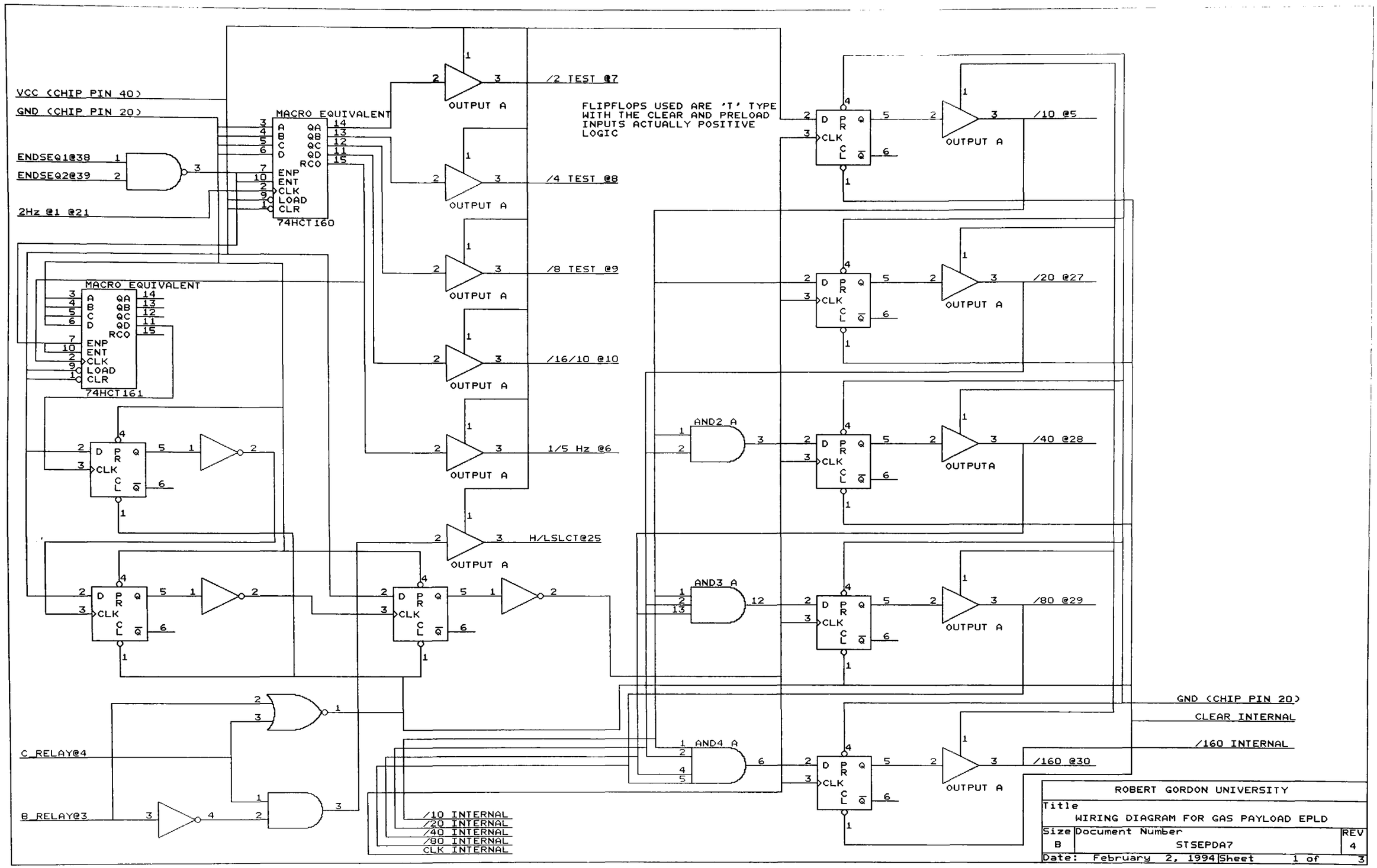
Q1,3,5,7,9,11,13,15 = BC547 TO_92 PINOUT B (AVAILABLE RGU STORES)
Q2,4,6,8,10,12,14,16 = TIP 41A TOP_66 PINOUT J (AVAILABLE RGU STORES)

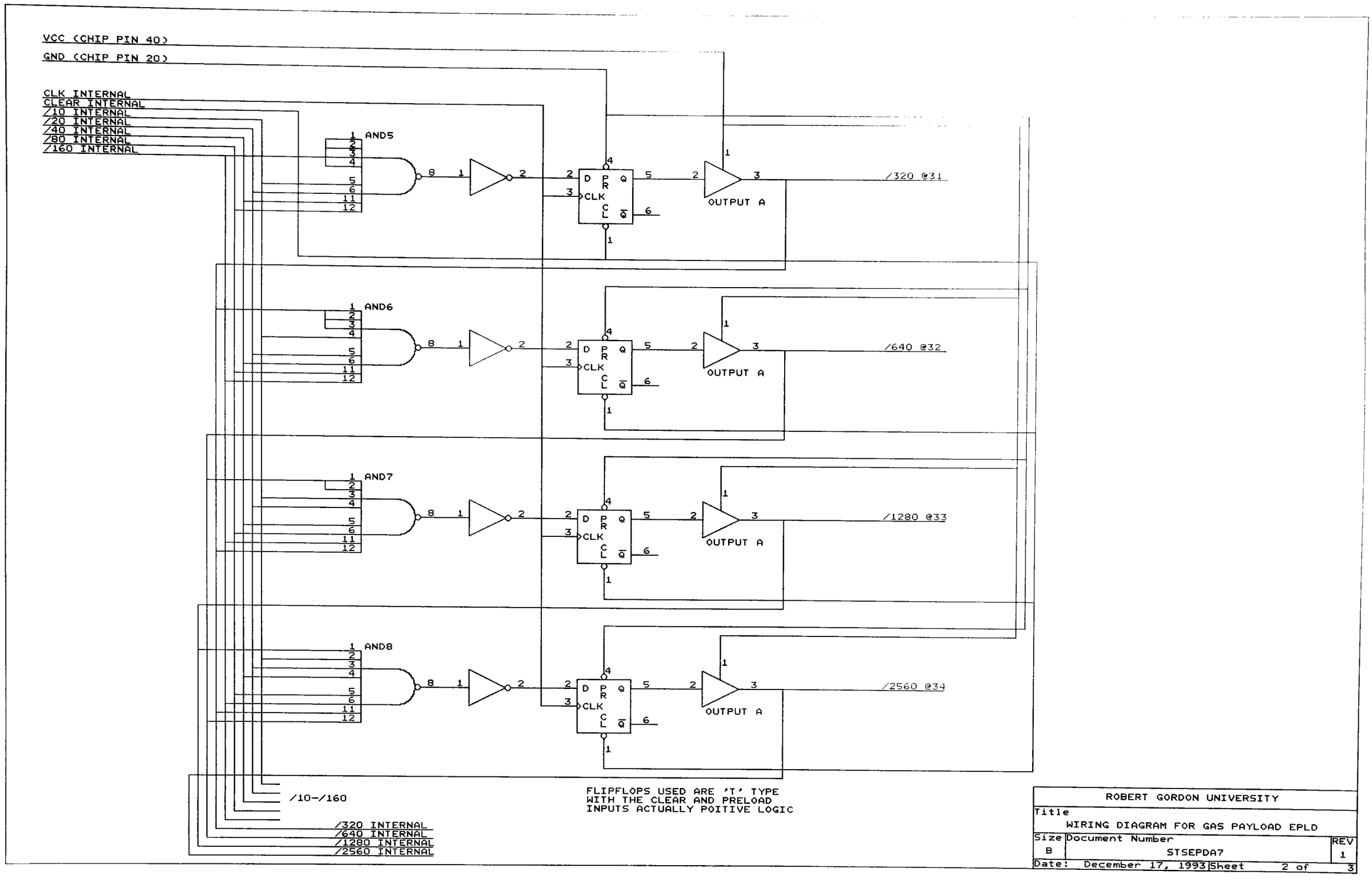
ALL RESISTORS ARE 1/8 W ORDINARY TYPES
DIODES ARE 1N4004 OR EQUIVALENT SIGNAL DIODES

ROBERT GORDON UNIVERSITY		
Title: GAS PAYLOAD WIRING DIAGRAM (SUB BOARDS)		
Size: B	Document Number: WLMBDK06.DIA	REV: 6
Date: July 18, 1996	Sheet 1 of 1	


IC +
IC_VTR_2A
IC_VTR_2B
IC_VTR_2C
7404
R39
1K
IC_VTR_2D
R40
OPTO MOSFET 1
IC -
IC_VTR_3A
IC_VTR_3B
IC_VTR_3C
R41
IC_VTR_3D
R42
OPTO MOSFET 2
R35
5V
5V BACKUP
'6'V
6V BACKUP
RECORD
REC POWER
REC ACTIVE
DETECT
TAPE REC POWER
DETECT
GND
GND BACKUP
HEADER 19
6V
VTR REC BUTTON
PARALLEL LOOP
VIDEO TAPE RECORDER
VTR STOP BUTTON
PARRALELL LOOP
OPTO MOSFET 3
VTR ON BUTTON
PARRALELL LOOP
OPTO MOSFET 4
GND
D25
1N4004
R72
15K
IC_VTR_1A
R71
22K
R9
8K2
R38
2K2
Q13
NPN
Q14
NPN
Q5
NPN
Q6
NPN
THE ROBERT GORDON UNIVERSITY
Title
VTR DRIVE CIRCUITRY
Size
B
Document Number
VTRDRV03.DIA
REV
3
Date: November 15, 1996
Sheet 1 of 1


5V
GND(1)
GROUND TEST
FAST MODE SELECT
SW1A
SW SPQT
(FAST MODE IS)
(16 TIMES SPEED)
SW1B
SW SPQT
SW1C
SW SPQT
4060
IC1
IC4
IC7
EP900
IC2
IC5
IC8
27C16
IC3
IC6
IC9
2.000MHz
X1
X2
X3
C1
C2
C3
C4
C5
C6
C7
C8
33pF
1.5uF
R6
R7
R8
R9
R10
R11
2K2
150K
BUS2
74LS241
IC11
IC12A
IC12B
74HC132
R12 180R
R13 180R
C9 1uF
C10 1uF
R14
R15
R22
1K
C23
33uF
WARBLER SUPPLY
BUZZER
RLB ARM
RLC ARM
RLB HOT DET
RLC HOT DET
LAS POW
HEAT POW
RECORD
REC POW
CAM POW
CTRLOK1
CTRLOK2
CTRLOK3
VOTEOK
IC10
LASER POWER
FLUID HEATER
RECORD
RECORDER POWER
CAMERA POWER
VOTER OK
R16 1K
R17 1K
R18 1K
R19 1K
R20 1K
R21 1K
OUTPUTS TO MICROGRAVITY
EXPERIMENT EQUIPMENT
POWER DRIVE CIRCUITS
CONTROLLER OPERATING OK SIGNALS
DECOUPLING CAPACITORS
100 nF FOR EACH 4060 AND 74241
220 nF FOR EACH EP900 AND EACH 27C16
C11 220 nF
C12 220 nF
C13 100 nF
C14 100 nF
C15 100 nF
C16 100nF
C17 220 nF
C18 220 nF
C19 220 nF
C20 220 nF
C21 220 nF
C22 100 nF
ROBERT GORDON UNIVERSITY
Title
GAS PAYLOAD WIRING DIAGRAM (MAIN)
Size B
Document Number WIRING48.DIA
REV 16
Date: November 27, 1995
Sheet 2 of 4


VCC (CHIP PIN 40)
GND (CHIP PIN 20)
ENDSEQ1@38
ENDSEQ2@39
2Hz @1 @21
MACRO EQUIVALENT
74HCT160
MACRO EQUIVALENT
74HCT161
OUTPUT A
/2 TEST @7
/4 TEST @8
/8 TEST @9
/16/10 @10
1/5 Hz @6
H/LSLCT@25
FLIPFLOPS USED ARE 'T' TYPE WITH THE CLEAR AND PRELOAD INPUTS ACTUALLY POSITIVE LOGIC
/10 @5
/20 @27
/40 @28
/80 @29
/160 @30
AND2 A
AND3 A
AND4 A
OUTPUTA
C_RELAY@4
B_RELAY@3
/10 INTERNAL
/20 INTERNAL
/40 INTERNAL
/80 INTERNAL
CLK INTERNAL
GND (CHIP PIN 20)
CLEAR INTERNAL
/160 INTERNAL
ROBERT GORDON UNIVERSITY
Title
WIRING DIAGRAM FOR GAS PAYLOAD EPLD
Size B
Document Number STSEPDA7
REV 4
Date: February 2, 1994
Sheet 1 of 3



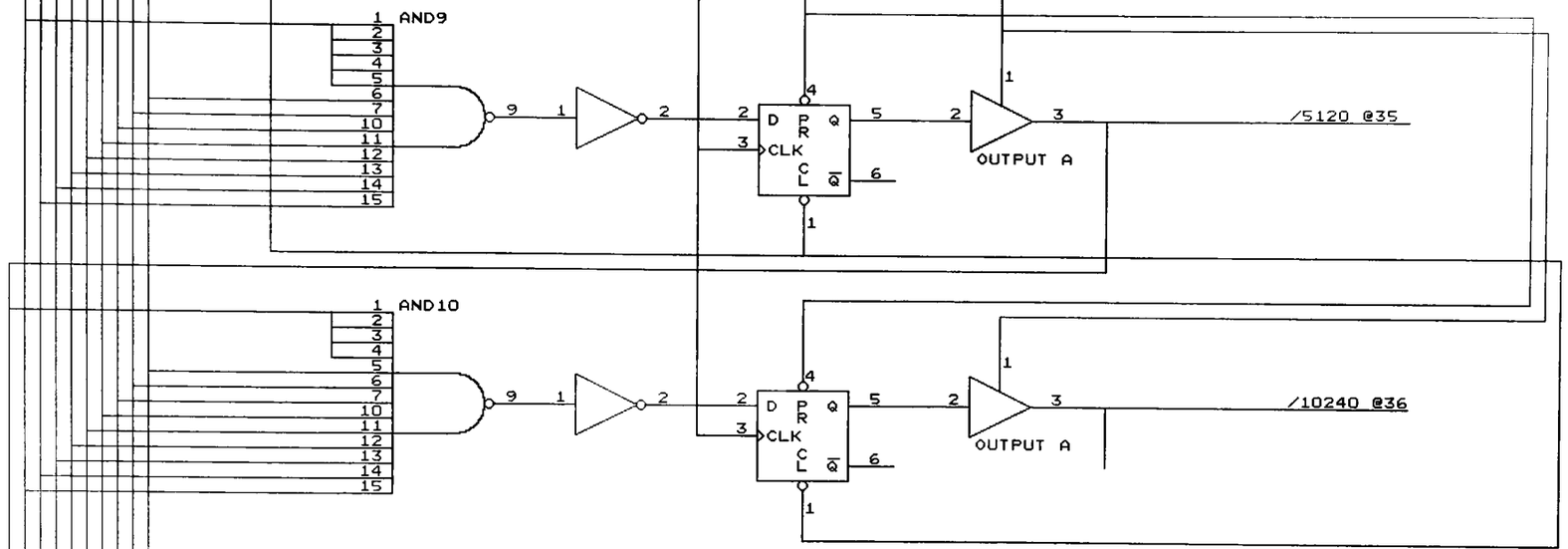
ROBERT GORDON UNIVERSITY		
Title: WIRING DIAGRAM FOR GAS PAYLOAD EPLD		
Size: B	Document Number: STSEPDA7	REV: 1
Date: December 17, 1993	Sheet 2 of 3	

VCC (CHIP PIN 40)

GND (CHIP PIN 20)

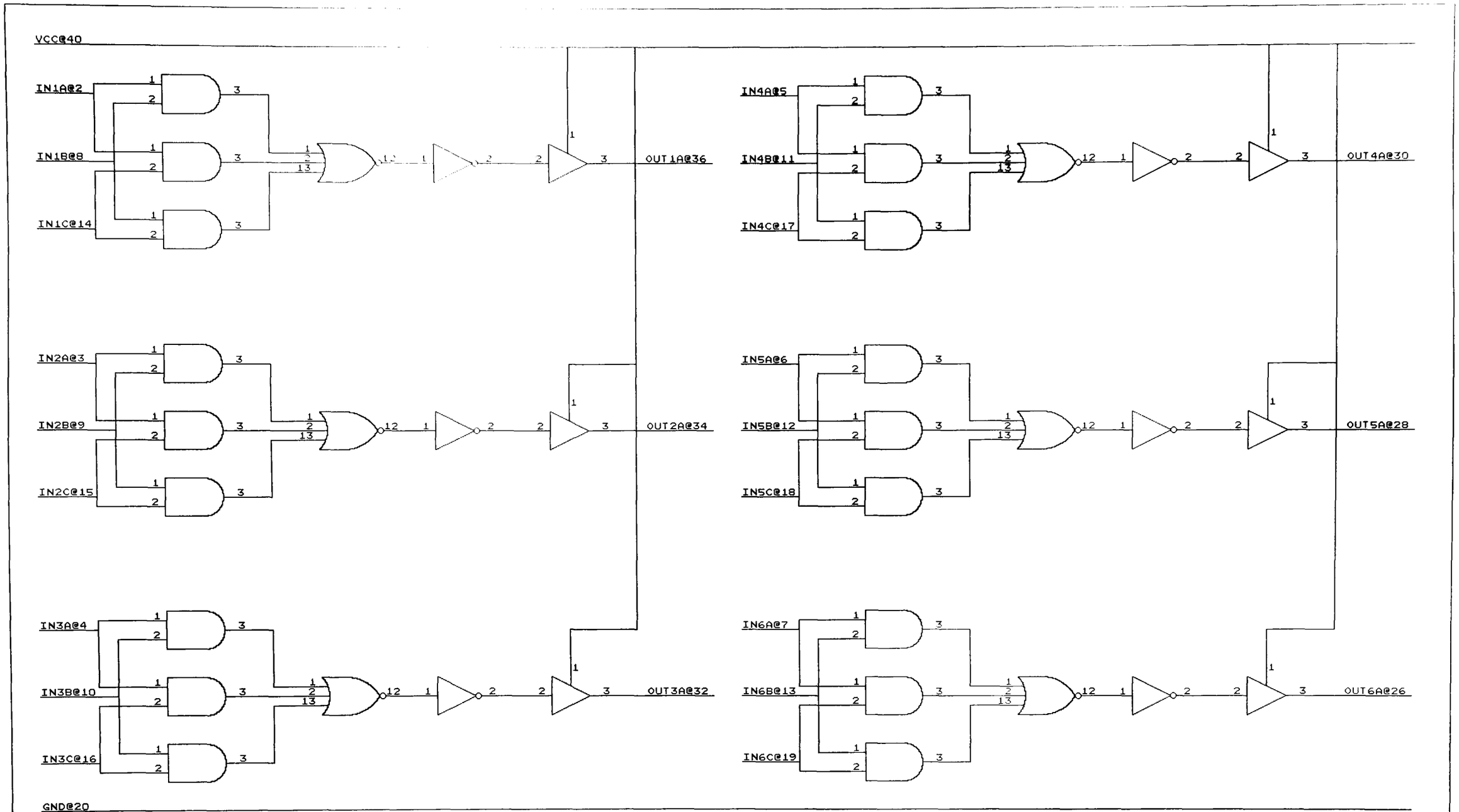
CLEAR INTERNAL
CLK INTERNAL

/10 INTERNAL
/20 INTERNAL
/40 INTERNAL
/80 INTERNAL
/160 INTERNAL
/320 INTERNAL
/640 INTERNAL
/1280 INTERNAL
/2560 INTERNAL



FLIPFLOPS USED ARE 'T' TYPE
WITH THE CLEAR AND PRELOAD
INPUTS ACTUALLY POSITIVE LOGIC

ROBERT GORDON UNIVERSITY		
Title: WIRING DIAGRAM FOR GAS PAYLOAD EPLD		
Size: B	Document Number: STSEPDA7	REV: 1
Date: December 17, 1993	Sheet 3 of 3	



ROBERT GORDON UNIVERSITY		
Title: GAS PAYLOAD VOTER EPLD		
Size: B	Document Number: STVOTEA5.DIA	REV: 4
Date: March 18, 1994	Sheet 1 of 1	

24V (1)

24V

5V

REG4
V-REG
VI ADJ VO

C29
0.1uF

R48
240R

R49
820R

C30
1uF

IC13

VCC

GND

A8-A15

AD0-AD7

68HC11
MAIN BOARD
(THIS IS A
STANDARD BOARD)

AS

R/WBAR

E

AAAAA
DDDDD
CCCCC
Ø1234

HEAT1
HEAT2
HEAT3
MHEAT
EPXVCC

R52 8K2
R53 8K2
R54 8K2
R55 8K2
R56 8K2

R57 1K6
R58 1K6
R59 1K6
R60 1K6
R61 1K6

IC14
74LS373
D0 D1 D2 D3 D4 D5 D6 D7
Q0 Q1 Q2 Q3 Q4 Q5 Q6 Q7
OC
G

IC15
2764
A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 A10 A11 A12
O0 O1 O2 O3 O4 O5 O6 O7
CE
OE
PGM
VPP

IC16
X28C256
A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 A10 A11 A12 A13 A14
O0 O1 O2 O3 O4 O5 O6 O7
CEBAR
OEBAR
WEBAR

IC17
74LS374
D0 D1 D2 D3 D4 D5 D6 D7
Q0 Q1 Q2 Q3 Q4 Q5 Q6 Q7
OC
CLK

R63
4K7

TESTIN

IC18A
7410
A14
A15
A13

IC19C
7400
A15

IC18C
7410
A15
A15
A15

IC18B
7410

IC19A
7400

IC19B
7400

IC20D
7400

A14

A15

IC19D
7400

C31
470nF

IC20A
7400

IC20C
7400

D19
DIODE

R64
4K7

GNDTEST

R62
560R

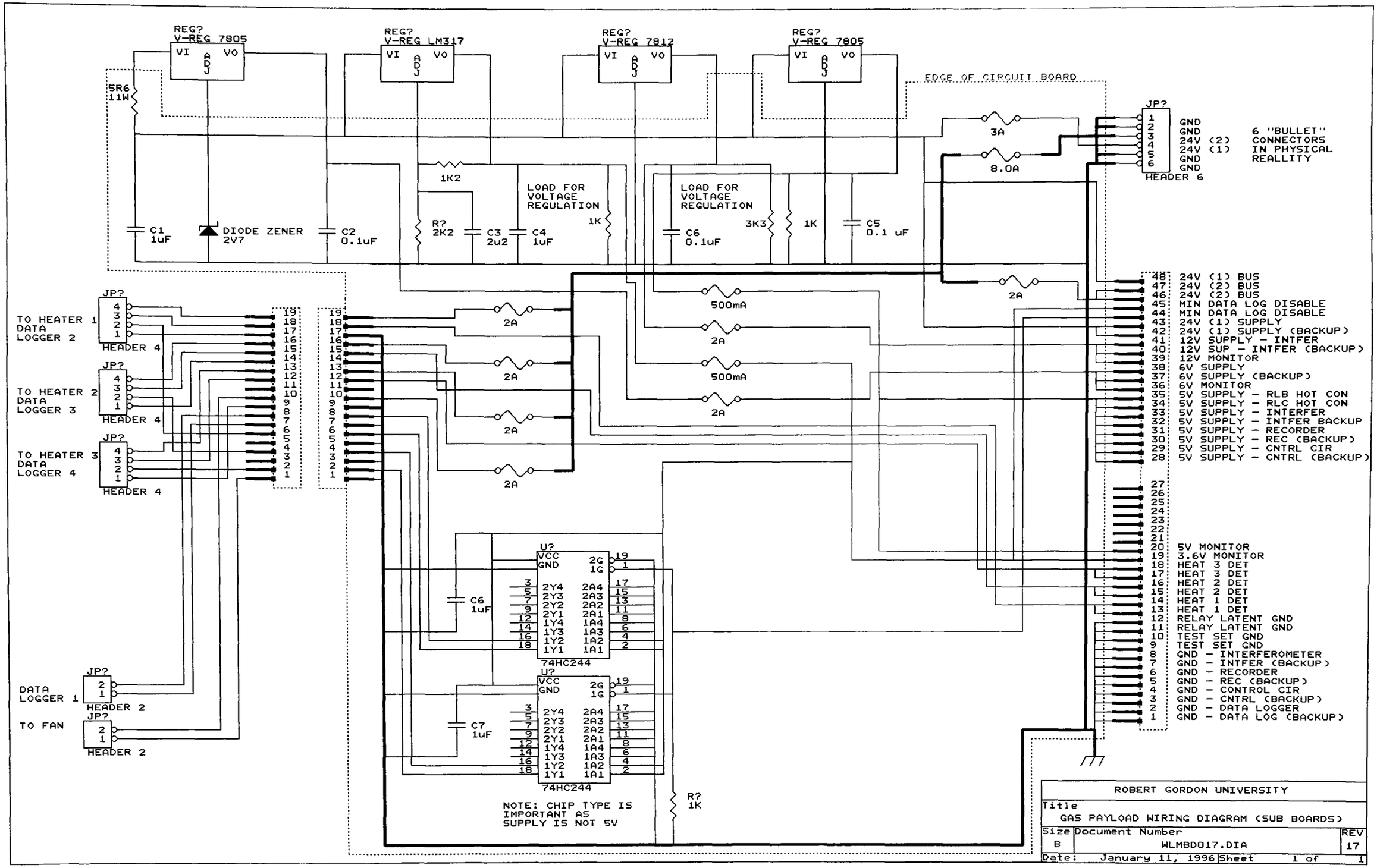
D20
DIODE

D21
DIODE

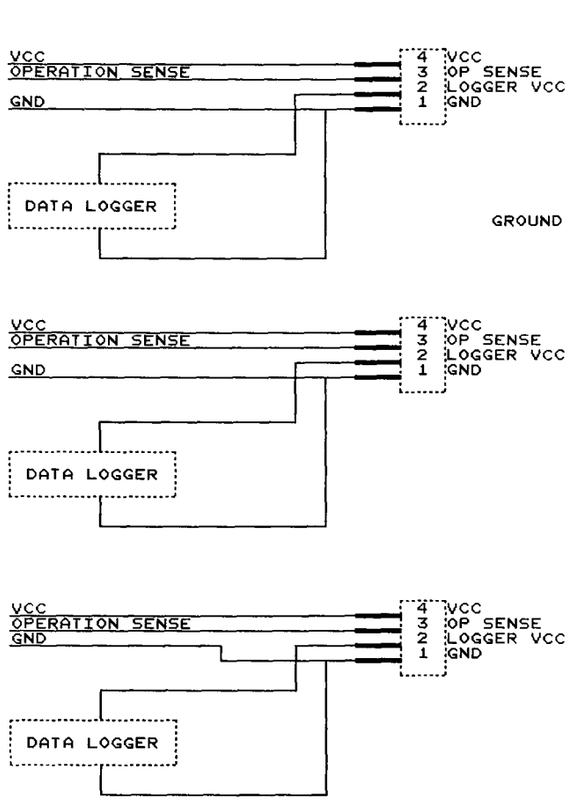
GND

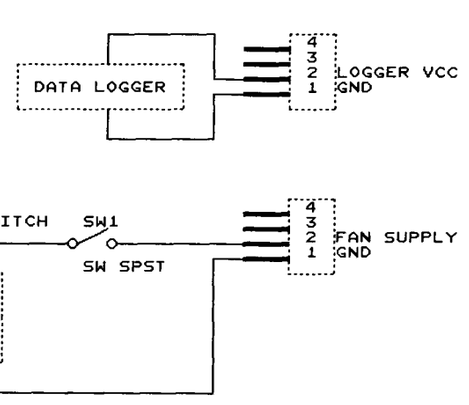
GND

ROBERT GORDON UNIVERSITY	
Title	GAS PAYLOAD WIRING DIAGRAM (MAIN)
Size: B	Document Number: WIRING47.dia — REV: 7
Date: November 3, 1994	Sheet 4 of 4

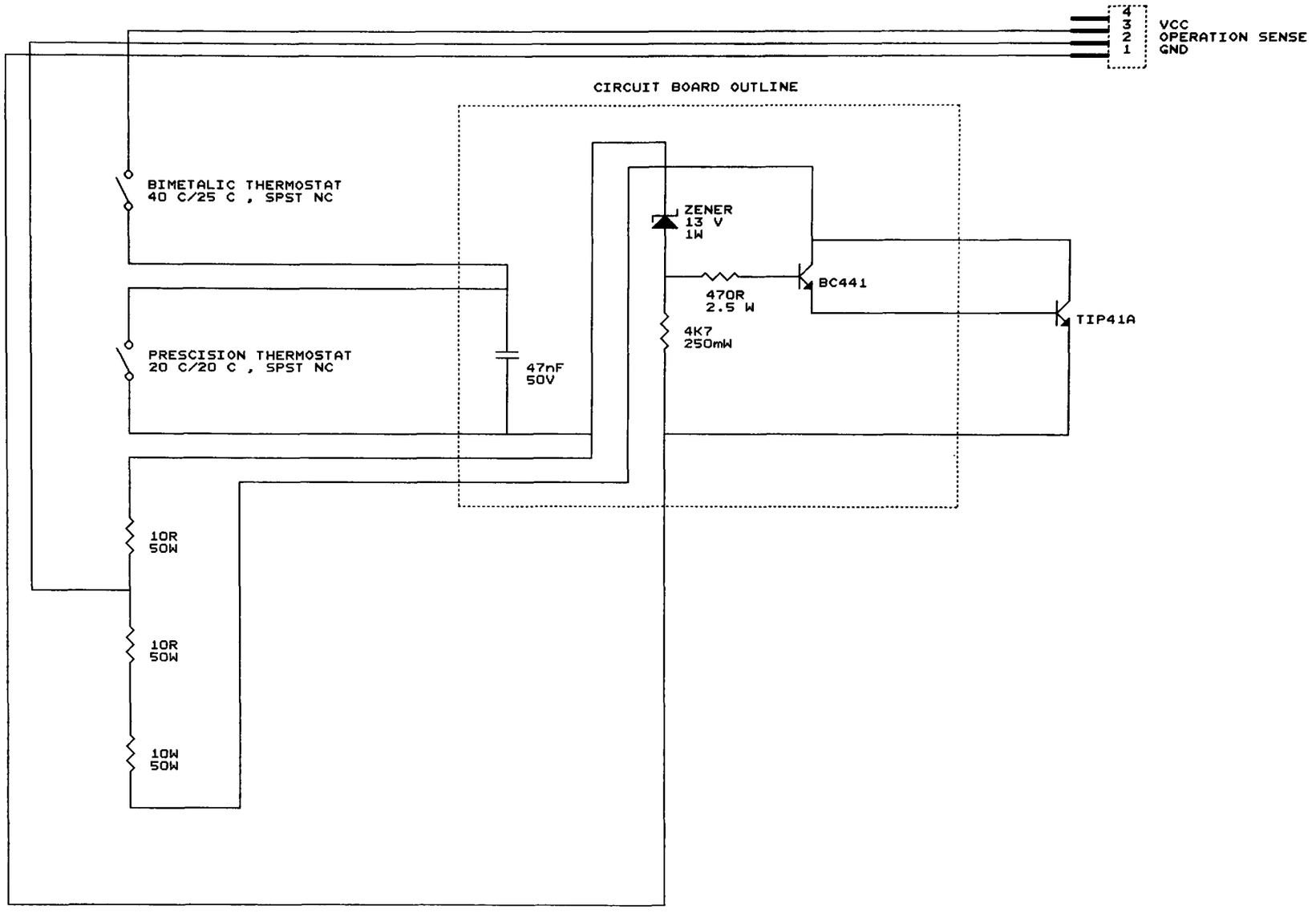

REG?
V-REG 7805
REG?
V-REG LM317
REG?
V-REG 7812
REG?
V-REG 7805
VI
ADJ
VO
EDGE OF CIRCUIT BOARD
5R6
11W
1K2
LOAD FOR
VOLTAGE
REGULATION
LOAD FOR
VOLTAGE
REGULATION
C1
1uF
DIODE ZENER
2V7
C2
0.1uF
R?
2K2
C3
2u2
C4
1uF
1K
C6
0.1uF
3K3
1K
C5
0.1 uF
3A
8.0A
2A
500mA
JP?
GND
GND
24V (2)
24V (1)
GND
GND
HEADER 6
6 "BULLET"
CONNECTORS
IN PHYSICAL
REALLITY
TO HEATER 1
DATA
LOGGER 2
TO HEATER 2
DATA
LOGGER 3
TO HEATER 3
DATA
LOGGER 4
HEADER 4
DATA
LOGGER 1
TO FAN
HEADER 2
U?
VCC
GND
2G
1G
2Y4
2Y3
2Y2
2Y1
1Y4
1Y3
1Y2
1Y1
2A4
2A3
2A2
2A1
1A4
1A3
1A2
1A1
74HC244
C6
1uF
C7
1uF
NOTE: CHIP TYPE IS
IMPORTANT AS
SUPPLY IS NOT 5V
R?
1K
48 24V (1) BUS
47 24V (2) BUS
46 24V (2) BUS
45 MIN DATA LOG DISABLE
44 MIN DATA LOG DISABLE
43 24V (1) SUPPLY
42 24V (1) SUPPLY (BACKUP)
41 12V SUPPLY - INTFER
40 12V SUP - INTFER (BACKUP)
39 12V MONITOR
38 6V SUPPLY
37 6V SUPPLY (BACKUP)
36 6V MONITOR
35 5V SUPPLY - RLB HOT CON
34 5V SUPPLY - RLC HOT CON
33 5V SUPPLY - INTERFER
32 5V SUPPLY - INTFER BACKUP
31 5V SUPPLY - RECORDER
30 5V SUPPLY - REC (BACKUP)
29 5V SUPPLY - CNTRL CIR
28 5V SUPPLY - CNTRL (BACKUP)
27
26
25
24
23
22
21
20 5V MONITOR
19 3.6V MONITOR
18 HEAT 3 DET
17 HEAT 3 DET
16 HEAT 2 DET
15 HEAT 2 DET
14 HEAT 1 DET
13 HEAT 1 DET
12 RELAY LATENT GND
11 RELAY LATENT GND
10 TEST SET GND
9 TEST SET GND
8 GND - INTERFEROMETER
7 GND - INTFER (BACKUP)
6 GND - RECORDER
5 GND - REC (BACKUP)
4 GND - CONTROL CIR
3 GND - CNTRL (BACKUP)
2 GND - DATA LOGGER
1 GND - DATA LOG (BACKUP)
ROBERT GORDON UNIVERSITY
Title
GAS PAYLOAD WIRING DIAGRAM (SUB BOARDS)
Size
B
Document Number
WLMBD017.DIA
REV
17
Date: January 11, 1996
Sheet 1 of 1

SEE LOWTHV03.DIA
FOR THE LOW VOLTAGE CUT OUT AND
HEATER CIRCUITRY WIRE HERE IN ORDER





THE ROBERT GORDON UNIVERSITY		
Title: SMALL CIRCUITS, HEATERS, LOGGER FAN		
Size: B	Document Number: SMALLC03.DIA	REV: 3
Date: January 11, 1996	Sheet 1 of 1	



THE ROBERT GORDON UNIVERSITY		
Title: LOW VOLTAGE CUT OUT FOR THERMAL CONTROL		
Size: B	Document Number: LOWTHV04.DIA	REV: 4
Date: December 8, 1995	Sheet 1 of 1	

Appendix 2

Matlab Files

Main Algorithm as used for processing the concept testing images.

```
function[xvect,yvect] = rfvect10(photo,xstart,ystart,cutang,noisefract)
%
% function[xvect,yvect] =
rfvect10(photo,xstart,ystart,cutang,noisefract)
%
% roof valley vector creating operator.
% This uses the dilate and thin, fringe noise overcoming system. It is
% applied using the starslice pattern and linked into fringes using the
% advanced adaptive fringe linking developed for starslice and its
% derivatives. Fringe linking using lnktstxx.m is then carried out to
% try and overcome any remaining breaks. This uses the multiscalar
% version of the fringe end tangent plotting fringe linker. Finally the
% remaining fringes are thresholded for length to minimise rubbish
% due to image edges being detected and the presence of noise or
% small fringe segments. the results are all saved to disk along with
% many Intermediate results.
%
% to G:\imageres\
%
% rfvect10.dry    diary file
% rfvt_raw.mat  srvdts9c.m  raw sliregch.m fringe roof and valley
%                           detection results
% rfvt_det.mat  srvdts9c.m  result after iterative dilate and thin stage
% rfvt_frg.mat  srvdts9c.m  result after fringe linking stage - fringes
% rfvt_res.mat  srvdts9c.m  result after fringe linking stage - plotted
%                           results
% rftv_sim.mat              fringes after deletion of overly similar
%                           fringes
% rftv_unf.mat              fringes just before gap filling algorithm
%                           but after noise fringe suppression
% rftv_rfg.mat              final fringe results after noise
%                           suppression
% rfvt_frs.mat  'int16'     all final fringes plotted onto photo image
% rfvt_vct.mat  'float'     final gravity vector
% rfvt_cnt.mat  'float'     centres of fringes used for gravity vector
%                           calculation
% rfvt_frv.mat  'int16'   final fringes with centres and vector plotted
%                           onto the image as well
%

diary h:\imageres\rfvect10.dry

% constants
passnos     = 2;    % number of dilate and thin stages
acc         = 512;  % number of angles algorithm operates at
maxgap      = 1;    % maximum permitted fringe gap in fringe
                    % linking stage, reduced from2 due to errors
                    % with image photo10c.mat, noted good fringes
                    % don't have any gaps at all.
thinsp      = 5;    % operations of thinning stages per pass.
complete    = 0.5;  % fraction of fringe to be present before it
                    % will be completed by gap filler and used for
                    % vector calculations
maxsimilarity = 0.8; % maximum fringe similarity permitted before
                    % overly similar fringes are deleted leaving
                    % only the first detected fringe

cd h:\matlab\matforh

resfringe =
srvdts9c(photo,passnos,acc,maxgap,thinsp,xstart,ystart,cutang);

% srvdts9c.m algorithm, saves most results to disk
% srvdtis9.m version returns the detected fringes array.

sz1       = size(resfringe);
fringetot  = sz1(1,1);
fringelgth = sz1(1,2);

% call fringe linking algorithm
% helps to fill in gaps in the fringe results caused by noise points

% no tangent linking
% [resfringe,enddata] =
lnkwca05(resfringe,xstart,ystart,acc,fringetot,cutang);
%
% but carry out ultra similar fringe suppression, as point reuse
% allowed get rid of fringes over maxsimilarity similar

resfringe = delsim01(resfringe,maxsimilarity);

cd H:\imageres

fid   = fopen('rftv_sim.mat','w');
count = fwrite(fid,resfringe,'char');

cd h:\matlab\matforh

% now scan fringe and discard if just noise
% set noise level as 25% of angles

noiselevel = acc * noisefract;

for fringenos = 1:fringetot
 pointsf   = 0;

 for scanpos = 1:acc
  if resfringe(fringenos,scanpos) > 0
   pointsf = pointsf + 1;
  end
 end

 if pointsf < noiselevel    % discard fringe
  for scanpos = 1:acc
   resfringe(fringenos,scanpos) = 0;
  end
 end
end

% having thrown out the noise plot the remaining fringes one at
% time and find their centres
% also calculate a vector through these centres on the assumption
% though it may be invalid that the first fringe is the innermost.

centres   = zeros(fringetot,2);
centrenos = 0;

cd h:\imageres

fid   = fopen('rftv_unf.mat','w');
count = fwrite(fid,resfringe,'char');

cd h:\matlab\matforh

resfringe = fillfg03(resfringe,complete); % fill fringe that are >
            complete % complete

for fringenos = 1:fringetot
 if max(max(resfringe(fringenos:fringenos,1:acc))) > 0
  res =
pltarc04(zeros(256),resfringe(fringenos:fringenos,1:acc),xstart,ystart);
  [xord,yord] = moment02(res);
  centrenos = centrenos + 1;
  centres(centrenos,1) = xord;
```

```
   centres(centrenos,2) = yord;
  end
end

% vector
xvect = 0;
yvect = 0;
vector = [xvect,yvect];

if centrenos > 1

  for vectornos = 2:centrenos
    xvect = xvect + (centres(1,1) - centres(vectornos,1));
    yvect = yvect - (centres(1,2) - centres(vectornos,2));
  end

  xvect = xvect / (centrenos - 1);
  yvect = yvect / (centrenos - 1);
  vector = [xvect,yvect];
end

% store matalb results to disk for checkups after - final fringes array

cd h:\imageres

fid   = fopen('rftv_rfg.mat','w');
count = fwrite(fid,resfringe,'char');

cd h:\bitmap

fid = fopen(photo,'r');
pht = fread(fid,[256,256],'char');

cd h:\matlab\matforh

res = pltarc04(pht,resfringe,xstart,ystart);

cd h:\imageres

fid   = fopen('rfvt_frs.mat','w');
count = fwrite(fid,res,'int16');
fid   = fopen('rfvt_vct.mat','w');
count = fwrite(fid,vector,'float');
fid   = fopen('rfvt_cnt.mat','w');
count = fwrite(fid,centres,'float');

sz1   = size(res);
maxrow = sz1(1,1);
maxcol = sz1(1,2);

for icentre = 1:centrenos
  if (centres(icentre,2) > 0.5) & (centres(icentre,2) < (maxrow + 0.5))
& (centres(icentre,1) > 0.5) & (centres(icentre,1) < (maxcol + 0.5))
    res(centres(icentre,2),centres(icentre,1)) = 256;
  end
end

for pvector = 1:20
  if (xstart - (vector(1,2)*pvector/20) > 0.5) & (xstart -
(vector(1,2)*pvector/20) < (maxrow + 0.5)) & (ystart +
(vector(1,1)*pvector/20) > 0.5) & (ystart + (vector(1,1)*pvector/20) <
(maxcol + 0.5))
    res(xstart - (vector(1,2)*pvector/20),ystart +
(vector(1,1)*pvector/20)) = 0;
  end
end

fid   = fopen('rfvt_frv.mat','w');
count = fwrite(fid,res,'int16');

cd h:\matlab\matforh

diary off      % stop diary for g:\imageres\rfvect10.dry

fclose('all')  % close all files, avoids data loss if computer crashes
later

%
% created  : 05/09/95
%  latest  : 22/09/95
% adaption : 22/01/96 fringe reusuing version allows multiple fringes
%                to use a set of fringe points, no tangent linking stage.
%                with trap to delete multiple detections of basically
%                identical fringes
%     latest : 23/01/96
% adaption : 09/02/96 tidied version uses int16 storage to avoid 256
%                to 0 conversions where undesirable, vector creator no
%                longer uses start point, produces fringes, plus centres
%                plus gravity vector final all data image
%     latest : 12/02/96
%         by : M.J. Steedman
%     status : Development version
%

-------------------------------------------------------------------------------


function[fringe] =
srvdts9c(photo,passnos,acc,maxgap,thinsp,xstart,ystart,cutang)
%
% function[fringe] =
srvdts9c(photo,passnos,acc,maxgap,thinsp,xstart,ystart,cutang)
%
% Slice - roof - valley - dilate - thin - iterative - subroutine.
%
% photo   = photograph   = 'xxxxxxxx.xxx' : typical = 'photo7c.mat'
% passnos = dilate/thin iterations      = x    : typical = 4
% acc     = angles to operate at        = x    : typical = 512
% maxgap  = gap allowed by linker       = x    : typical = 2
% thinsp  = thins per dilation          = x    : typical = 5
% xstart  = x coordinate start          = xxx  : typical = 128
% ystart  = y coordinate start          = xxx  : typical = 128
% cutang  = cutangle for heater wire    = xxx  : typical = 256
%

%
% This version c allows a fringe to be used multiple times. A fringe
% point is only allowed to start a fringe once but by setting used
% fringe points to another but non zero value they can be crossed
% again by later fringe tracing for use only with rfvect07 without
% tangent linking stage.
%
% using heater wire cut out step as well.
% built on the srvdth11.m algorithm but with iterative dilation and
% thinning. This should minimise interfringe linking while allowing
% more dilation stages, and still accomplish results as the thinning
% used in this algorithm is 'along slices' only so angular spread is
% maintained. Also reworking thinning stage to work with the roof
% and valley image so that the valleys are retained for future dilations.
%
% fringe extraction test using the dilate and thin approach working
% from a detection of both the roofs and valleys in a fringe image
%
% slice, roof, valley, dilate, thin, algorithm
% operates slireg, roof and valley finder on image, followed by
% frvalc01.m which converts this to a roof only image. rvtoad01.m is
% used to generate the image in angle-radius form so the peak
% thinning averaging routine can be applied along image slices. The
% data is then returned to the plain format for the dilation stage. The
% results of this being thinned with passnos + 1 applications of
% thin013.m
%
% version 6 uses group along a slice down to 1 rather than thinning
% algorithm to control the dilation. This 'thins' the dilated fringe
% pattern at right angles to the fringes but does not trim back any
% fringe length like a normal  thinning algorithm does, hence no loss
% of detections

cd h:\bitmap

fid  = fopen(photo,'r');
pht7 = fread(fid,[256,256],'char');

cd h:\matlab\matforh

%xstart   = 80;
%ystart   = 140;
%cutang   = 315;

%acc      = 256;
sz1      = size(pht7);
rmax     = sz1(1,1);
cmax     = sz1(1,2);
fringetot = 50;

rfval = sliregch(pht7,xstart,ystart,acc);
```

```
% having got detarray a clean up can now be carried out to remove
% multiple detections of the same fringe, it can be safely assumed
% that any sequences of points in the detect array are actually all
% multiple detections of one fringe

cd h:\imageres

% store matalb results to disk for checkups after - raw detections roof
valley
fid   = fopen('rfvt_raw.mat','w');
count = fwrite(fid,rfval,'char');

cd h:\matlab\matforh

detsize   = size(rfval);
detlength = detsize(1,2);

for passes = 1:passnos

 % dilation first, valleys then roofs

 detarray = rfval;

 for a = 1:acc
  for b = 2:(detlength - 1)
   if (a == 1)
    a1 = acc;
   else
    a1 = a - 1;
   end
   if (a == acc)
    a2 = 1;
   else
    a2 = a + 1;
   end

   if detarray(a,b) == 0;  % case valley, expand
    if rfval(a1,b-1) == 1
     rfval(a1,b-1) = 0;
    end
    if rfval(a,b-1) == 1
     rfval(a,b-1) = 0;
    end
    if rfval(a2,b-1) == 1
     rfval(a2,b-1) = 0;
    end
    if rfval(a1,b) == 1
     rfval(a1,b) = 0;
    end
    if rfval(a2,b) == 1
     rfval(a2,b) = 0;
    end
    if rfval(a1,b+1) == 1
     rfval(a1,b+1) = 0;
    end
    if rfval(a,b+1) == 1
     rfval(a,b+1) = 0;
    end
    if rfval(a2,b+1) == 1
     rfval(a2,b+1) = 0;
    end
   end
  end
 end
end
if detarray(a,b) == 2;  % case roof, expand
 if rfval(a1,b-1) == 1
  rfval(a1,b-1) = 2;
 end
 if rfval(a,b-1) == 1
  rfval(a,b-1) = 2;
 end
 if rfval(a2,b-1) == 1
  rfval(a2,b-1) = 2;
 end
 if rfval(a1,b) == 1
  rfval(a1,b) = 2;
 end
 if rfval(a2,b) == 1
  rfval(a2,b) = 2;
 end
 if rfval(a1,b+1) == 1
  rfval(a1,b+1) = 2;
 end
 if rfval(a,b+1) == 1
  rfval(a,b+1) = 2;
 end
 if rfval(a2,b+1) == 1
  rfval(a2,b+1) = 2;
 end
end
end


if passes > 1      % ie hugely wide detections are under control
                   % second dilation stage to close larger gaps.

 detarray = rfval;

 for a = 1:acc
  for b = 2:(detlength - 1)
   if (a == 1)
    a1 = acc;
   else
    a1 = a - 1;
   end
   if (a == acc)
    a2 = 1;
   else
    a2 = a + 1;
   end

   if detarray(a,b) == 0;  % case valley, expand
    if rfval(a1,b-1) == 1
     rfval(a1,b-1) = 0;
    end
    if rfval(a,b-1) == 1
     rfval(a,b-1) = 0;
    end
    if rfval(a2,b-1) == 1
     rfval(a2,b-1) = 0;
    end
    if rfval(a1,b) == 1
     rfval(a1,b) = 0;
    end
    if rfval(a2,b) == 1
     rfval(a2,b) = 0;
    end
    if rfval(a1,b+1) == 1
     rfval(a1,b+1) = 0;
    end
    if rfval(a,b+1) == 1
     rfval(a,b+1) = 0;
    end
    if rfval(a2,b+1) == 1
     rfval(a2,b+1) = 0;
    end
   end
  if detarray(a,b) == 2;  % case roof, expand
   if rfval(a1,b-1) == 1
    rfval(a1,b-1) = 2;
   end
   if rfval(a,b-1) == 1
    rfval(a,b-1) = 2;
   end
   if rfval(a2,b-1) == 1
    rfval(a2,b-1) = 2;
   end
   if rfval(a1,b) == 1
    rfval(a1,b) = 2;
   end
   if rfval(a2,b) == 1
    rfval(a2,b) = 2;
   end
   if rfval(a1,b+1) == 1
    rfval(a1,b+1) = 2;
   end
   if rfval(a,b+1) == 1
    rfval(a,b+1) = 2;
   end
   if rfval(a2,b+1) == 1
    rfval(a2,b+1) = 2;
   end
  end
 end
end
```

```
% detarray = thinroof(detarray,acc,distmax);
% detarray = thinvall(detarray,acc,distmax);

for thin = 1:thinsp
  rfval = thnrof04(rfval,acc,detlength);
end

for thin = 1:thinsp
  rfval = thnval04(rfval,acc,detlength);
end

end

% first convert results to the right format

detarray = rfvalc01(rfval);

% ensure thinned properly
detarray = thnroofo(detarray,acc,detlength - 20);

cd h:\imageres

% store matalb results to disk for checkups after - rooves only image
fid   = fopen('rfvt_det.mat','w');
count = fwrite(fid,detarray,'char');

cd h:\matlab\matforh

% now operate fringe linking algorithm
% after first removing the heater wires

detarray = imtidy01(detarray,maxgap,cutang);

% now use detarray to extract arcs directly from detected points
% aim is to extract whole features in one algorithm rather than
% marking edge points with starslice and then trying to get something
% from that data at a later stage.

% first arrays to hold fringe point sequences

% fringetot = 5; make this a parameter

fringe = zeros(fringetot,acc);
finingesfound = 0;
lastfringe = 0;
fringenos = 0;
distmax   = detlength - 10; % because of old arraysize of distmax + 10
noisefract   = 0.02;

% for fringenos = 1:fringetot

while lastfringe == 0
 fringenos = fringenos + 1;
 if fringenos == fringetot
  lastfringe = 1;
 end

 rad     = 5;    % resets ready for next fringe (was 7)
 angle   = 0;
 firsthit = 0;

 while((rad < (distmax - 2)) & (firsthit == 0))
  rad = rad + 1;
  angle = 0;
  while((angle < acc) & (firsthit == 0))
   angle = angle + 1;
   if (detarray(angle,rad) > 0) & (detarray(angle,rad) < 10)
    fringe(fringenos,angle) = rad;
    detarray(angle,rad) = 10;  % different but non zero value
                               % records used, so not for fringe
                               % starts but may be reused for
                               % tracing through
    firsthit = 1;
    fringesfound = fringesfound + 1;
   end
  end
 end

% trap case of not found enough fringes but have exhausted array
% i.e. asked to find more than there was to find.

 if (firsthit == 0)
  lastfringe = 1;
 end

 firstrad    = rad;
 nodetectsfor = 0;
 stoppedat   = 0;          % position at which forward sweep
                           % was stopped by fringe end, used
                           % as an alternative stop for the
                           % reverse sweep so that it does not
                           % overwrite a fringe that has
                           % already been found.

 for wrapnang = angle + 1:angle + acc
  nang = wrapnang;
  if nang > acc
   nang = nang - acc;
  end
  lastdiff = 2;                % default radius difference
                               % between last pair of detects
  if (nang > 4)
   if ((fringe(fringenos,nang-1) > 0) & (fringe(fringenos,nang-2) > 0)
& (fringe(fringenos,nang-3) > 0) & (fringe(fringenos,nang-4) > 0))
    lastdiff = fringe(fringenos,nang-1) - ((fringe(fringenos,nang-2) +
fringe(fringenos,nang-3) + fringe(fringenos,nang-4)) / 3);
   end
  elseif (nang > 3)
   if ((fringe(fringenos,nang-1) > 0) & (fringe(fringenos,nang-2) > 0)
& (fringe(fringenos,nang-3) > 0))
    lastdiff = fringe(fringenos,nang-1) - ((fringe(fringenos,nang-2) +
fringe(fringenos,nang-3)) / 2);
   end
  elseif nang > 2
   if ((fringe(fringenos,nang-1) > 0) & (fringe(fringenos,nang-2) >
0))
    lastdiff = fringe(fringenos,nang-1) - fringe(fringenos,nang-2);
   end
  end
  if lastdiff < 0
   lastdiff = 0 - lastdiff;
  end
  if rad < 9
   rad = 9;
  end
  if nodetectsfor < (maxgap + 1)
   if detarray(nang,rad) > 0
    detarray(nang,rad) = 10;
    fringe(fringenos,nang) = rad;
    nodetectsfor = 0;
   elseif detarray(nang,rad + 1) > 0
    detarray(nang,rad + 1) = 10;
    fringe(fringenos,nang) = rad + 1;
    nodetectsfor = 0;
    if rad < (distmax - 2)
     rad = rad + 1;
    end
   elseif detarray(nang,rad - 1) > 0
    detarray(nang,rad - 1) = 10;
    nodetectsfor = 0;
    fringe(fringenos,nang) = rad - 1;
    if rad > 7
     rad = rad - 1;
    end
   elseif (detarray(nang,rad + 2) > 0) & (lastdiff > -1)
    detarray(nang,rad + 2) = 10;
    nodetectsfor = 0;
    fringe(fringenos,nang) = rad + 2;
    if rad < (distmax - 3)
     rad = rad + 2;
    end
   elseif (detarray(nang,rad - 2) > 0) & (lastdiff > -1)
    detarray(nang,rad - 2) = 10;
    nodetectsfor = 0;
    fringe(fringenos,nang) = rad - 2;
    if rad > 8
     rad = rad - 2;
    end
   elseif (detarray(nang,rad + 3) > 0) & (lastdiff > 0)
    detarray(nang,rad + 3) = 10;
    nodetectsfor = 0;
    fringe(fringenos,nang) = rad + 3;
    if rad < (distmax - 4)
     rad = rad + 3;
    end
```

```
elseif (detarray(nang,rad - 3) > 0) & (lastdiff > 0)
 detarray(nang,rad - 3) = 10;
 nodetectsfor = 0;
 fringe(fringenos,nang) = rad - 3;
 if rad > 9
  rad = rad - 3;
 end
elseif (detarray(nang,rad + 4) > 0) & (lastdiff > 1)
 detarray(nang,rad + 4) = 10;
 nodetectsfor = 0;
 fringe(fringenos,nang) = rad + 4;
 if rad < (distmax - 5)
  rad = rad + 4;
 end
elseif (detarray(nang,rad - 4) > 0) & (lastdiff > 1)
 detarray(nang,rad - 4) = 10;
 nodetectsfor = 0;
 fringe(fringenos,nang) = rad - 4;
 if rad > 10
  rad = rad - 4;
 end
elseif ((detarray(nang,rad + 5) > 0) & (lastdiff > 2))
 detarray(nang,rad + 5) = 10;
 nodetectsfor = 0;
 fringe(fringenos,nang) = rad + 5;
 if rad < (distmax - 6)
  rad = rad + 5;
 end
elseif (detarray(nang,rad - 5) > 0) & (lastdiff > 2)
 detarray(nang,rad - 5) = 10;
 nodetectsfor = 0;
 fringe(fringenos,nang) = rad - 5;
 if rad > 11
  rad = rad - 5;
 end
elseif ((detarray(nang,rad + 6) > 0) & (lastdiff > 3))
 detarray(nang,rad + 6) = 10;
 nodetectsfor = 0;
 fringe(fringenos,nang) = rad + 6;
 if rad < (distmax - 7)
  rad = rad + 6;
 end
elseif (detarray(nang,rad - 6) > 0) & (lastdiff > 3)
 detarray(nang,rad - 6) = 10;
 nodetectsfor = 0;
 fringe(fringenos,nang) = rad - 6;
 if rad > 12
  rad = rad - 6;
 end
elseif ((detarray(nang,rad + 7) > 0) & (lastdiff > 4))  % 4
 detarray(nang,rad + 7) = 10;
 nodetectsfor = 0;
 fringe(fringenos,nang) = rad + 7;
 if rad < (distmax - 8)
  rad = rad + 7;
 end
elseif (detarray(nang,rad - 7) > 0) & (lastdiff > 4)
 detarray(nang,rad - 7) = 10;
 nodetectsfor = 0;
 fringe(fringenos,nang) = rad - 7;
 if rad > 13
  rad = rad - 7;
 end
elseif ((detarray(nang,rad + 8) > 0) & (lastdiff > 5))  % 5
 detarray(nang,rad + 8) = 10;
 nodetectsfor = 0;
 fringe(fringenos,nang) = rad + 8;
 if rad < (distmax - 9)
  rad = rad + 8;
 end
elseif (detarray(nang,rad - 8) > 0) & (lastdiff > 5)
 detarray(nang,rad - 8) = 10;
 nodetectsfor = 0;
 fringe(fringenos,nang) = rad - 8;
 if rad > 14
  rad = rad - 8;
 end
else
 nodetectsfor = nodetectsfor + 1;
end
else
 if stoppedat == 0
  stoppedat = wrapnang - (maxgap + 1);
  end
 end
end

rad        = firstrad;
nodetectsfor = 0;

for wrapnang = angle - 1:-1:angle - acc
 nang = wrapnang;
 if nang < 1
  nang = nang + acc;
 end
 lastdiff = 2;                % default radius difference
                              % between last pair of detects
 if (nang < acc - 3)
  if ((fringe(fringenos,nang+1) > 0) & (fringe(fringenos,nang+2) >
0) & (fringe(fringenos,nang+3) > 0) & (fringe(fringenos,nang+4) > 0))
   lastdiff = fringe(fringenos,nang+1) - ((fringe(fringenos,nang+2)
+ fringe(fringenos,nang+3) + fringe(fringenos,nang+4)) / 3);
  end
 elseif (nang < acc - 2)
  if ((fringe(fringenos,nang+1) > 0) & (fringe(fringenos,nang+2) >
0) & (fringe(fringenos,nang+3) > 0))
   lastdiff = fringe(fringenos,nang+1) - ((fringe(fringenos,nang+2)
+ fringe(fringenos,nang+3)) / 2);
  end
 elseif nang < acc - 1
  if ((fringe(fringenos,nang+1) > 0) & (fringe(fringenos,nang+2) >
0))
   lastdiff = fringe(fringenos,nang+1) - fringe(fringenos,nang+2);
  end
 end
 if lastdiff < 0
  lastdiff = 0 - lastdiff;
 end
 if rad < 9
  rad = 9;
 end
 if (nodetectsfor < (maxgap + 1)) & (wrapnang > (stoppedat - acc))
  if detarray(nang,rad) > 0
   detarray(nang,rad) = 10;
  nodetectsfor = 0;
  fringe(fringenos,nang) = rad;
 elseif detarray(nang,rad + 1) > 0
  detarray(nang,rad + 1) = 10;
  nodetectsfor = 0;
  fringe(fringenos,nang) = rad + 1;
  if rad < (distmax - 2)
   rad = rad + 1;
  end
 elseif detarray(nang,rad - 1) > 0
  detarray(nang,rad - 1) = 10;
  nodetectsfor = 0;
  fringe(fringenos,nang) = rad - 1;
  if rad > 7
   rad = rad - 1;
  end
 elseif (detarray(nang,rad + 2) > 0) & (lastdiff > -1)
  detarray(nang,rad + 2) = 10;
  nodetectsfor = 0;
  fringe(fringenos,nang) = rad + 2;
  if rad < (distmax - 3)
   rad = rad + 2;
  end
 elseif (detarray(nang,rad - 2) > 0) & (lastdiff > -1)
  detarray(nang,rad - 2) = 10;
  nodetectsfor = 0;
  fringe(fringenos,nang) = rad - 2;
  if rad > 8
   rad = rad - 2;
  end
 elseif (detarray(nang,rad + 3) > 0) & (lastdiff > 0)
  detarray(nang,rad + 3) = 10;
  nodetectsfor = 0;
  fringe(fringenos,nang) = rad + 3;
  if rad < (distmax - 4)
   rad = rad + 3;
  end
 elseif (detarray(nang,rad - 3) > 0) & (lastdiff > 0)
  detarray(nang,rad - 3) = 10;
  nodetectsfor = 0;
  fringe(fringenos,nang) = rad - 3;
  if rad > 9
```

```
    rad = rad - 3;
   end
  elseif (detarray(nang,rad + 4) > 0) & (lastdiff > 1)
   detarray(nang,rad + 4) = 10;
   nodetectsfor = 0;
   fringe(fringenos,nang) = rad + 4;
   if rad < (distmax - 5)
    rad = rad + 4;
   end
  elseif (detarray(nang,rad - 4) > 0) & (lastdiff > 1)
   detarray(nang,rad - 4) = 10;
   nodetectsfor = 0;
   fringe(fringenos,nang) = rad - 4;
   if rad > 10
    rad = rad - 4;
   end
  elseif ((detarray(nang,rad + 5) > 0) & (lastdiff > 2))
   detarray(nang,rad + 5) = 10;
   nodetectsfor = 0;
   fringe(fringenos,nang) = rad + 5;
   if rad < (distmax - 6)
    rad = rad + 5;
   end
  elseif (detarray(nang,rad - 5) > 0) & (lastdiff > 2)
   detarray(nang,rad - 5) = 10;
   nodetectsfor = 0;
   fringe(fringenos,nang) = rad - 5;
   if rad > 11
    rad = rad - 5;
   end
  elseif ((detarray(nang,rad + 6) > 0) & (lastdiff > 3))
   detarray(nang,rad + 6) = 10;
   nodetectsfor = 0;
   fringe(fringenos,nang) = rad + 6;
   if rad < (distmax - 7)
    rad = rad + 6;
   end
  elseif (detarray(nang,rad - 6) > 0) & (lastdiff > 3)
   detarray(nang,rad - 6) = 10;
   nodetectsfor = 0;
   fringe(fringenos,nang) = rad - 6;
   if rad > 12
    rad = rad - 6;
   end
  elseif ((detarray(nang,rad + 7) > 0) & (lastdiff > 4))  % 4
   detarray(nang,rad + 7) = 10;
   nodetectsfor = 0;
   fringe(fringenos,nang) = rad + 7;
   if rad < (distmax - 8)
    rad = rad + 7;
   end
  elseif (detarray(nang,rad - 7) > 0) & (lastdiff > 4)
   detarray(nang,rad - 7) = 10;
   nodetectsfor = 0;
   fringe(fringenos,nang) = rad - 7;
   if rad > 13
    rad = rad - 7;
   end
  elseif ((detarray(nang,rad + 8) > 0) & (lastdiff > 5))  % 5
   detarray(nang,rad + 8) = 10;
   nodetectsfor = 0;
   fringe(fringenos,nang) = rad + 8;
   if rad < (distmax - 9)
    rad = rad + 8;
   end
  elseif (detarray(nang,rad - 8) > 0) & (lastdiff > 5)
   detarray(nang,rad - 8) = 10;
   nodetectsfor = 0;
   fringe(fringenos,nang) = rad - 8;
   if rad > 14
    rad = rad - 8;
   end
  else
   nodetectsfor = nodetectsfor + 1;
  end
 end
end

% now scan fringe and discard if just noise
% set niose level as 15% of angles

 noiselevel = acc * noisefract;
 pointsf    = 0;
 for scanpos = 1:acc
  if fringe(fringenos,scanpos) > 0
   pointsf = pointsf + 1;
  end
 end

 if pointsf < noiselevel   % discard fringe
  for scanpos = 1:acc
   fringe(fringenos,scanpos) = 0;
  end
  fringenos = fringenos - 1;
 end

end                        % now increment fringetot

clear detarray;            % used so clear to save memory

% array of fringes now aquired so plot and save the results

res = pltarc04(pht7,fringe,xstart,ystart);

cd h:\imageres

% store matalb results to disk for checkups after - fringes array
fid   = fopen('rftv_frg.mat','w');
count = fwrite(fid,fringe,'char');
fid   = fopen('rfvt_res.mat','w');
count = fwrite(fid,res,'char');

cd h:\matlab\matforh

fclose('all')

%
% adapted : 22/01/96  multi fringe using
%  created : 05/09/95
%    latest : 15/09/95
%        by : M.J. Steedman
%   status : Development version
%
-------------------------------------------------------------------------------



function[result,fringe,detections] = sliregch(image,initcr,initcc,acc)
%
% function[result,fringe,detections] = sliregch(image,initcr,initcc,acc)
%
% slireg07 and on, development
% set out to improve the performance of the algorithm for fringes near
% the limits of the image by offsetting the slices by a distance into the
% image this gives some zeros at the centre so the operators can go
% much closer to the image centre than before and also the outside
% overrun is extended
%
% slireg array of ones valleys are set to zeros and roofs to 2, format
% is a matrix rows by cols. rather than the dist and angle arrangement
%
% slice-store, save detections to allow direct line rather than peak
% detection
% slistl13 - tighten up lastdiff constraint, this proves that angle
% limiter is too constraining and also very time consuming to use.
% checking detailed results from slistl12 shows that in good fringes
% the change in radius from one radius to the next is only +/- 2 from
% the last and the largest hole in a fringe is one point missed and that
% is very rare. slistl12.m - as slistl09 (not11) adds maxgap parameter,
% equal to the maximum permitted run of zeros
% between points to be detected as in the same fringe - but no angle
% limit slistl09.m - as slistr08 but with extra blurring of image slices
% in order to reduce the effect of high frequency noise on the operator
% slistl08.m - as slistr07.m but multiscalar detectors turned on
% widths are 13, 19, 25, 31, 37 : basic slistr07 only runs the 13 wide
% detector
%
% new starslice algorithm based on the idea of the star algorithm but
% using a completely new feature finding method involving gradients,
% this version detects zero crossings of the image gradient that look
% like non local maxima ( i.e. it filters out noise spurs )
% in this case points where the gradient goes form +ve to -ve without
% bouncing back across the zero point right alongside. the algorithm
% takes the average of the slope between four sequential pairs of
```

```
% points as the gradient, it then scans this vector of locally averaged
% gradients. In this scanning pass it sums the five gradients before
% the present point and if the result is greater than +3 assumes that
% the image has been increasing intensity up to this point along this
% vector. The five points on from the present scan point are likewise
% summed and assumed to represent a drop if their sum is below -3.
%A fringe ( a `bright` ridge in the image ) is declared to have been
% found if both the above tests are true for a centre point which has a
% gradient value of > -2 and < 2
%
% this tooled up version ignores points with plus or minus slopes of
% over 150 as these generally only occur if the image edge has been
% overlapped giving a false edge as the scan arrays are pre-set at zero
% beyond these points. It also checks that rad-1 > 0 and rad+1 < 0 to
% ensure that the point in question is actually a place where the
% gradient crosses zero and not a dip below gradient +2 just before a
% crossing, something that has been giving several false detects up to
% now.
%
% success of the new algorithm as described above
% this has been tested on photo5.mat and over half the vectors detect
% the main fringe in the image with the only spurious detects being in
% the noise at the edges of the picture well outside the main image
% area.
%
% this version reduces the number of multiple detects and is perfect
% on the vertically downwards slice which has been used as the one
% for which graphs of the intermediate stages are extracted to see
% what is actually going on. The problem is though that it does lose
% some of the real limit detects on the very poor fringes that are
% heavily disrupted by the high frequency noise that this image
% contains from being a screen photograph
%
% latest stars111 version without the gradient at radius rad >-2 and <2
% detector, this reduces the safety net but much decreases the chance
% that a sharp swing of the gradient through zero will cause the
% crossing point to be missed by this safeguard. now contains a
% sensible progress indicator output, no `junk` output no dummy
% parameter requirements.
%
% star algorithm for finding the centre of gravity of shapes in a grey
% level image
% see Paper No 36 for basics of the star algorithm
% The general idea is that you scan out along a number of equally
% radially spaced points from an initial start point until you detect an
% image feature by a suitable thresholded function becoming true.
% This is for use in grey level images with the intention that the `stop`
% points for each angle will highlight the desired image feature. In
% order to improve the boundary detection the algorithm having
% found all the stop points for the initial centre now works out their
% centre of gravity and then repeats using this point as the new start
% point, and so it goes on until the centre of gravity of the stop points
% `matches` the start point.
%
% stars112.m : higher speed version has been hacked to streamline
% execution pre computation used where possible to speed operations.
%
% stars113.m distmax now corrected, more 'junk' removal
% stars114.m multiscalar version
%

sz1   = size(image);
rmax  = sz1(1,1);
cmax  = sz1(1,2);
xy    = zeros(64,2);
cr    = initcr;
cc    = initcc;
oldcr = 0;
oldcc = 0;

%result = ones(rmax,cmax);

% presmoothing

image = blurim01(image);

% calculates the maximum distance from the start point specified to
% an image corner and sets this as the longest required slice length.

if cr > rmax / 2
 rmaxsquare = cr*cr;
else
 rmaxsquare = (rmax - cr) * (rmax - cr);
end
if cc > cmax / 2
 cmaxsquare = cc*cc;
else
 cmaxsquare = (cmax - cc) * (cmax - cc);
end
distmax = sqrt(rmaxsquare + cmaxsquare) + 40;  % +20 * 2 due to
need extra
                                % zeros required if processing
                                % is to reach the image edge
                                % and centre due to the
                                % width of the operators.
result = ones(acc,distmax);

% now slice length is known create array to hold detected points.

detarray = zeros(acc,distmax+10); % + 10 to avoid out of range errors
                                % from the point linking algorithm

gdsph  = 2;                     % constant used for results visibility

for angle = 1:acc

 theta = 2*pi*angle/(acc);
 slsp  = zeros(distmax,1);      % slicespace holds the slice of the image
                                % saves repeated complex references to
                                % the main image.
 % first copy image information into slicespace array the 20 offset on
 % slsp gives the 20 zeros in the image centre to allow the operators
 % to start scanning before the image points are reached therby
 % allowing them to detect fringes much closer to the start point

 for rad = 1:distmax - 30
  if ((cr+(rad*cos(theta)) < rmax) & (cr+(rad*cos(theta)) > 1))
   if ((cc+(rad*sin(theta)) < cmax) & (cc+(rad*sin(theta)) > 1))
    slsp(rad + 20,1) =
image(cr+(rad*cos(theta)),cc+(rad*sin(theta)));
   else
    slsp(rad + 20,1) = 0;
   end
  else
   slsp(rad + 20,1) = 0;
  end
 end

 % pre - smooth the present image slice in order to ease the effects
 % of high frequency noise
 % uses the vertical over 5 point smoother blur_line_vertical
 % two applications brings in some effect from a nine pixel region

 slsp  = burlnv01(slsp);
 slsp2 = burlnv01(slsp);   % incremental bluring
 slsp3 = burlnv01(slsp2);


 % now calculate the gradient at each point along the present radius in
 % preparation for finding the zero crossing points.

 gradsave = zeros(distmax,1);

 for rad = 3:(distmax - 2)
  gradtemp = [(slsp(rad+2)-slsp(rad+1)),(slsp(rad+1)-
slsp(rad)),(slsp(rad)-slsp(rad-1)),(slsp(rad-1)-slsp(rad-2))];
  for aa = 1:3
   for bb = 1:3
    if gradtemp(bb) > gradtemp(bb + 1)
     temp = gradtemp(bb);
     gradtemp(bb) = gradtemp(bb + 1);
     gradtemp(bb + 1) = temp;
    end
   end
  end
  gradsave(rad) = (gradtemp(3) + gradtemp(2)) / 2;
 end

 gradsave2 = zeros(distmax,1);

 for rad = 3:(distmax - 2)
  gradtemp = [(slsp(rad+2)-slsp(rad+1)),(slsp(rad+1)-
slsp(rad)),(slsp(rad)-slsp(rad-1)),(slsp(rad-1)-slsp(rad-2))];
  for aa = 1:3
   for bb = 1:3
    if gradtemp(bb) > gradtemp(bb + 1)
```

```
      temp = gradtemp(bb);
      gradtemp(bb) = gradtemp(bb + 1);
      gradtemp(bb + 1) = temp;
     end
    end
   end
   gradsave2(rad) = (gradtemp(3) + gradtemp(2)) / 2;
end

gradsave3 = zeros(distmax,1);

for rad = 3:(distmax - 2)
  gradtemp = [(slsp(rad+2)-slsp(rad+1)),(slsp(rad+1)-
slsp(rad)),(slsp(rad)-slsp(rad-1)),(slsp(rad-1)-slsp(rad-2))];
  for aa = 1:3
   for bb = 1:3
    if gradtemp(bb) > gradtemp(bb + 1)
     temp = gradtemp(bb);
     gradtemp(bb) = gradtemp(bb + 1);
     gradtemp(bb + 1) = temp;
    end
   end
  end
  gradsave3(rad) = (gradtemp(3) + gradtemp(2)) / 2;
end


% now have found somethign if the gradient = zero and the five
% gradient values before it total over +5 end the five after total
% below -5; version 12 on use 3 scales +3 to -3, +6 to -6 and +9 to -9

a    = gradsave;
b    = gradsave2;
c    = gradsave3;

result = slirsub1(result,a,distmax,angle,gdsph);
result = slirsub1(result,b,distmax,angle,gdsph);
result = slirsub1(result,c,distmax,angle,gdsph);

% all done now go to next angle

end

result = result(1:acc,21:distmax);  % 20 offset to get rid fo the 20
                                    % values 'inside' the centre of the
                                    % image that were there simply for
                                    % algorithm operating requirements

%
%       created : 09/06/95
% slireg  latest : 15/09/95 : this version improvements on slireg06.m
%                             - all angle rad version
% starsl created : 9/8/94
%         latest : 3/2/95
% slistl created : 16/02/95
%         latest : 23/03/95
%            by : M.J. Steedman
%         status : Development version
%

-------------------------------------------------------------------------------



function[result] = blurim01(image)
%
% image bluring function averages each point in an image with its 24
% nieghbours
%

sz1   = size(image);
rmax  = sz1(1,1);
cmax  = sz1(1,2);
result = zeros(rmax,cmax);

for a = 3:rmax - 2
 for b = 3:cmax - 2
  result(a,b) = image(a,b);
  result(a,b) = result(a,b) + image(a-2,b-2) + image(a-2,b-1);
  result(a,b) = result(a,b) + image(a-2,b) + image(a-2,b+1);
  result(a,b) = result(a,b) + image(a-2,b+2) + image(a-1,b-2);
  result(a,b) = result(a,b) + image(a-1,b-1) + image(a-1,b);
  result(a,b) = result(a,b) + image(a-1,b+1) + image(a-1,b+2);
  result(a,b) = result(a,b) + image(a,b-2) + image(a,b-1);
  result(a,b) = result(a,b) + image(a,b+1) + image(a,b+2);
  result(a,b) = result(a,b) + image(a+1,b-2) + image(a+1,b-1);
  result(a,b) = result(a,b) + image(a+1,b) + image(a+1,b+1);
  result(a,b) = result(a,b) + image(a+1,b+2) + image(a+2,b-2);
  result(a,b) = result(a,b) + image(a+2,b-1) + image(a+2,b);
  result(a,b) = result(a,b) + image(a+2,b+1) + image(a+2,b+2);
 end
end

result = result / 25;

%
% created : 4/8/94
%    by : M.J. Steedman
% status : functional
%

-------------------------------------------------------------------------------



function[result] = burlnv01(image)
%
% function[result] = burlnv01(image)
%
% image bluring function averages each point in a vertical line
% with its 4 nieghbours
%

sz1   = size(image);
rmax  = sz1(1,1);
cmax  = sz1(1,2);
result = zeros(rmax,cmax);

for a = 3:rmax - 2
 for b = 1:cmax
  result(a,b) = image(a,b);
  result(a,b) = result(a,b) + image(a-2,b) + image(a-1,b);
  result(a,b) = result(a,b) + image(a+1,b) + image(a+2,b);
 end
end

result = result / 5;

%
% created : 07/03/95
%    by : M.J. Steedman
% status : functional
%

-------------------------------------------------------------------------------



function[result] = slirsub1(result,a,distmax,angle,gdsph)
%
% function[result] = slirsub1(result,a,distmax,angle,gdsph)
%

 for rad = 3:(distmax - 2)              % first scale 5 wide

  plusslope  = a(rad-2)+a(rad-1);
  minusslope = a(rad+1)+a(rad+2);
  if (((a(rad-1)>0)&(a(rad)<0.001))|((a(rad)>-0.001)&(a(rad+1)<0)))
                                % zero crossing occurs !!
   if ((plusslope > 3) & (minusslope < - 3))  % >5  <-5
    if ((plusslope < 150) & (minusslope > -150))
                                % threshold out image edge
                                % effects
     result(angle,rad) = gdsph;
    end
   end
  end
 if (((a(rad-1)<0.001)&(a(rad)>0))|((a(rad)<0)&(a(rad+1)>-0.001)))
                                % zero crossing occurs !!
                                % minus to plus direction = valley
   if ((plusslope < -3) & (minusslope < 3))  % >5  <-5
    if ((plusslope > - 150) & (minusslope < 150))
                                % threshold out image edge
                                % effects
     result(angle,rad) = 0;
    end
   end
  end
```

```
end

for rad = 4:(distmax - 3)                % second scale 7 wide

 plusslope  = a(rad-3)+a(rad-2)+a(rad-1);
 minusslope = a(rad+1)+a(rad+2)+a(rad+3);
 if (((a(rad-1)>0)&(a(rad)<0.001))|((a(rad)>-0.001)&(a(rad+1)<0)))
                                % zero crossing occurs !!
  if ((plusslope > 3) & (minusslope < - 3))   % >5  <-5
   if ((plusslope < 150) & (minusslope > -150))
                                % threshold out image edge
                                % effects
    result(angle,rad) = gdsph;
   end
  end
 end
if (((a(rad-1)<0.001)&(a(rad)>0))|((a(rad)<0)&(a(rad+1)>-0.001)))
                                % zero crossing occurs !!
                                % minus to plus direction = valley
  if ((plusslope < -3) & (minusslope < 3))   % >5  <-5
   if ((plusslope > - 150) & (minusslope < 150))
                                % threshold out image edge
                                % effects
    result(angle,rad) = 0;
   end
  end
 end
end

for rad = 5:(distmax - 4)                % third scale 9 wide

 plusslope  = a(rad-4)+a(rad-3)+a(rad-2)+a(rad-1);
 minusslope = a(rad+1)+a(rad+2)+a(rad+3)+a(rad+4);
 if (((a(rad-1)>0)&(a(rad)<0.001))|((a(rad)>-0.001)&(a(rad+1)<0)))
                                % zero crossing occurs !!
  if ((plusslope > 3) & (minusslope < - 3))   % >5  <-5
   if ((plusslope < 150) & (minusslope > -150))
                                % threshold out image edge
                                % effects
    result(angle,rad) = gdsph;
   end
  end
 if (((a(rad-1)<0.001)&(a(rad)>0))|((a(rad)<0)&(a(rad+1)>-0.001)))
                                % zero crossing occurs !!
                                % minus to plus direction = valley
  if ((plusslope < -3) & (minusslope < 3))   % >5  <-5
   if ((plusslope > - 150) & (minusslope < 150))
                                % threshold out image edge
                                % effects
    result(angle,rad) = 0;
   end
  end
 end
end


 for rad = 7:(distmax - 6)               % fourth scale 13 wide

  plusslope  = a(rad-6)+a(rad-5)+a(rad-4)+a(rad-3)+a(rad-2)+a(rad-
1);
  minusslope = a(rad+1)+a(rad+2)+a(rad+3)+a(rad+4)+a(rad+5)+a
(rad+6);
  if (((a(rad-1)>0)&(a(rad)<0.001))|((a(rad)>-0.001)&(a(rad+1)<0)))
                                % zero crossing occurs !!
                                % plus to minus direction = peak
   if ((plusslope > 3) & (minusslope < - 3))   % >5  <-5
    if ((plusslope < 150) & (minusslope > -150))
                                % threshold out image edge
                                % effects
     result(angle,rad) = gdsph;
    end
   end
  end
 if (((a(rad-1)<0.001)&(a(rad)>0))|((a(rad)<0)&(a(rad+1)>-0.001)))
                                % zero crossing occurs !!
                                % minus to plus direction = valley
  if ((plusslope < -3) & (minusslope < 3))   % >5  <-5
   if ((plusslope > - 150) & (minusslope < 150))
                                % threshold out image edge
                                % effects

    result(angle,rad) = 0;
   end
```

```
   end
  end
 end


 for rad = 10:(distmax - 9)              % fifth scale 19 wide

  plusslope  = a(rad-9)+a(rad-8)+a(rad-7)+a(rad-6)+a(rad-5)+a(rad-
4)+a(rad-3)+a(rad-2)+a(rad-1);
  minusslope = a(rad+1)+a(rad+2)+a(rad+3)+a(rad+4)+a(rad+5)+
a(rad+6)+a(rad+7)+a(rad+8)+a(rad+9);
  if (((a(rad-1)>0)&(a(rad)<0.001))|((a(rad)>-0.001)&(a(rad+1)<0)))
                                % zero crossing occurs !!
                                % plus to minus direction = peak
   if ((plusslope > 3) & (minusslope < - 3))   % >5  <-5
    if ((plusslope < 150) & (minusslope > -150))
                                % threshold out image edge
                                % effects
     result(angle,rad) = gdsph;
    end
   end
  end
 if (((a(rad-1)<0.001)&(a(rad)>0))|((a(rad)<0)&(a(rad+1)>-0.001)))
                                % zero crossing occurs !!
                                % minus to plus direction = valley
  if ((plusslope < - 3) & (minusslope < - 3))   % >5  <-5
   if ((plusslope > -150) & (minusslope < 150))
                                % threshold out image edge
                                % effects
    result(angle,rad) = 0;
   end
  end
 end
end

 for rad = 13:(distmax - 12)             % sixth scale 25 wide

  plusslope  = a(rad-12)+a(rad-11)+a(rad-10)+a(rad-9)+a(rad-
8)+a(rad-7)+a(rad-6)+a(rad-5)+a(rad-4)+a(rad-3)+a(rad-2)+a(rad-1);
  minusslope = a(rad+1)+a(rad+2)+a(rad+3)+a(rad+4)+a(rad+5)+
a(rad+6)+a(rad+7)+a(rad+8)+a(rad+9)+a(rad+10)+a(rad+11)+a(rad+1
2);
  if (((a(rad-1)>0)&(a(rad)<0.001))|((a(rad)>-0.001)&(a(rad+1)<0)))
                                % zero crossing occurs !!
                                % plus to minus direction = peak
   if ((plusslope > 3) & (minusslope < - 3))   % >5  <-5
    if ((plusslope < 150) & (minusslope > -150))
                                % threshold out image edge
                                % effects
     result(angle,rad) = gdsph;
    end
   end
  end
 if (((a(rad-1)<0.001)&(a(rad)>0))|((a(rad)<0)&(a(rad+1)>-0.001)))
                                % zero crossing occurs !!
                                % minus to plus direction = valley
  if ((plusslope < -3) & (minusslope > 3))   % >5  <-5
   if ((plusslope > -150) & (minusslope < 150))
                                % threshold out image edge
                                % effects
    result(angle,rad) = 0;
   end
  end
 end
end

 for rad = 16:(distmax - 15)             % seventh scale 31 wide

  plusslope  = a(rad-15)+a(rad-14)+a(rad-13)+a(rad-12)+a(rad-
11)+a(rad-10)+a(rad-9)+a(rad-8)+a(rad-7)+a(rad-6)+a(rad-5)+a(rad-
4)+a(rad-3)+a(rad-2)+a(rad-1);
  minusslope = a(rad+1)+a(rad+2)+a(rad+3)+a(rad+4)+a(rad+5)+
a(rad+6)+a(rad+7)+a(rad+8)+a(rad+9)+a(rad+10)+a(rad+11)+a(rad+1
2)+a(rad+13)+a(rad+14)+a(rad+15);
  if (((a(rad-1)>0)&(a(rad)<0.001))|((a(rad)>-0.001)&(a(rad+1)<0)))
                                % zero crossing occurs !!
                                % plus to minus direction = peak
   if ((plusslope > 3) & (minusslope < - 3))   % >5  <-5
    if ((plusslope < 150) & (minusslope > -150))
                                % threshold out image edge
                                % effects
     result(angle,rad) = gdsph;
    end
```

```
  end
 end
if (((a(rad-1)<0.001)&(a(rad)>0))|((a(rad)<0)&(a(rad+1)>-0.001)))
                                % zero crossing occurs !!
                                % minus to plus direction = valley
   if ((plusslope < -3) & (minusslope > 3))   % >5  <-5
    if ((plusslope > -150) & (minusslope < 150))
                                % threshold out image edge
                                % effects
      result(angle,rad) = 0;
     end
    end
   end
  end
 end

 for rad = 19:(distmax - 18)              % eighth scale 37 wide

   plusslope  = a(rad-18)+a(rad-17)+a(rad-16)+a(rad-15)+a(rad-
14)+a(rad-13)+a(rad-12)+a(rad-11)+a(rad-10)+a(rad-9)+a(rad-
8)+a(rad-7)+a(rad-6)+a(rad-5)+a(rad-4)+a(rad-3)+a(rad-2)+a(rad-1);
   minusslope = a(rad+1)+a(rad+2)+a(rad+3)+a(rad+4)+a(rad+5)+
a(rad+6)+a(rad+7)+a(rad+8)+a(rad+9)+a(rad+10)+a(rad+11)+a(rad+1
2)+a(rad+13)+a(rad+14)+a(rad+15)+a(rad+16)+a(rad+17)+a(rad+18);
   if (((a(rad-1)>0)&(a(rad)<0.001))|((a(rad)>-0.001)&(a(rad+1)<0)))
                                % zero crossing occurs !!
                                % plus to minus direction = peak
    if ((plusslope > 3) & (minusslope < - 3))   % >5  <-5
     if ((plusslope < 150) & (minusslope > -150))
                                % threshold out image edge
                                % effects
      result(angle,rad) = gdsph;
     end
    end
   end
 if (((a(rad-1)<0.001)&(a(rad)>0))|((a(rad)<0)&(a(rad+1)>-0.001)))
                                % zero crossing occurs !!
                                % minus to plus direction = valley
   if ((plusslope < -3) & (minusslope > 3))   % >5  <-5
    if ((plusslope > -150) & (minusslope < 150))
                                % threshold out image edge
                                % effects
      result(angle,rad) = 0;
     end
    end
   end
  end


%
% created : 15/09/95
%  latest : 15/09/95
%      by : M.J. Steedman
%  status : Development version
%

-------------------------------------------------------------------------


function[detarray] = thnrof04(detarray,acc,distmax)
%
% function[detarray] = thnrof04(detarray,acc,distmax)
%
%

% mark edge points

for angle = 1:acc
 for pos = 2:(distmax - 1)
  if detarray(angle,pos) == 2
   if ((detarray(angle,pos - 1) < 2) & (detarray(angle,pos + 1) > 1)) |
((detarray(angle,pos + 1) < 2) & (detarray(angle,pos - 1) > 1))
    detarray(angle,pos) = 3;
   end
  end
 end
end

% delete points on edges which are not required for continuity

for angle = 2:acc - 1
 for pos = 2:(distmax - 1)
  if detarray(angle,pos) == 3
   if (detarray(angle - 1,pos + 1) < 2) & (detarray(angle,pos + 1) < 2)
          (detarray(angle + 1,pos + 1) < 2)
    if (detarray(angle - 1,pos - 1) > 1) & (detarray(angle,pos - 1) > 1)
          (detarray(angle + 1,pos - 1) > 1)
     detarray(angle,pos) = 1;
    end
    if (detarray(angle - 1,pos) < 2) & (detarray(angle + 1,pos) < 2)
     detarray(angle,pos) = 1;
    end
    if (detarray(angle - 1,pos) < 2) & (detarray(angle - 1,pos - 1) < 2)
          (detarray(angle + 1,pos - 1) > 1)
     detarray(angle,pos) = 1;
    end
    if (detarray(angle + 1,pos) < 2) & (detarray(angle + 1,pos - 1) <
          2) & (detarray(angle - 1,pos - 1) > 1)
     detarray(angle,pos) = 1;
    end
   end
   if (detarray(angle - 1,pos - 1) < 2) & (detarray(angle,pos - 1) < 2)
          & (detarray(angle + 1,pos - 1) < 2)
    if (detarray(angle - 1,pos + 1) > 1) & (detarray(angle,pos + 1) >
          1) & (detarray(angle + 1,pos + 1) > 1)
     detarray(angle,pos) = 1;
    end
    if (detarray(angle - 1,pos) < 2) & (detarray(angle + 1,pos) < 2)
     detarray(angle,pos) = 1;
    end
    if (detarray(angle - 1,pos) < 2) & (detarray(angle - 1,pos + 1) <
          2) & (detarray(angle + 1,pos + 1) > 1)
     detarray(angle,pos) = 1;
    end
    if (detarray(angle + 1,pos) < 2) & (detarray(angle + 1,pos + 1) <
          2) & (detarray(angle - 1,pos + 1) > 1)
     detarray(angle,pos) = 1;
    end
   end
    if (detarray(angle - 1,pos) > 1) & (detarray(angle - 1,pos - 1) > 1)
& (detarray(angle,pos + 1) < 2) & (detarray(angle,pos - 1) > 1) &
(detarray(angle + 1,pos + 1) < 2) & (detarray(angle + 1,pos) < 2)
     detarray(angle,pos) = 1;
    end
    if (detarray(angle + 1,pos) > 1) & (detarray(angle + 1,pos - 1) > 1)
& (detarray(angle,pos + 1) < 2) & (detarray(angle,pos - 1) > 1) &
(detarray(angle - 1,pos + 1) < 2) & (detarray(angle - 1,pos) < 2)
     detarray(angle,pos) = 1;
    end
   end
  end
 end
end

% two missing points, 1 and acc

angle = 1;
for pos = 2:(distmax - 1)
 if detarray(angle,pos) == 3
  if (detarray(acc,pos + 1) < 2) & (detarray(angle,pos + 1) < 2) &
          (detarray(angle + 1,pos + 1) < 2)
   if (detarray(acc,pos - 1) > 1) & (detarray(angle,pos - 1) > 1) &
          (detarray(angle + 1,pos - 1) > 1)
    detarray(angle,pos) = 1;
   end
   if (detarray(acc,pos) < 2) & (detarray(angle + 1,pos) < 2)
    detarray(angle,pos) = 1;
   end
   if (detarray(acc,pos) < 2) & (detarray(acc,pos - 1) < 2) &
          (detarray(angle + 1,pos - 1) > 1)
    detarray(angle,pos) = 1;
   end
   if (detarray(angle + 1,pos) < 2) & (detarray(angle + 1,pos - 1) < 2)
          & (detarray(acc,pos - 1) > 1)
    detarray(angle,pos) = 1;
   end
  end
  if (detarray(acc,pos - 1) < 2) & (detarray(angle,pos - 1) < 2) &
          (detarray(angle + 1,pos - 1) < 2)
   if (detarray(acc,pos + 1) > 1) & (detarray(angle,pos + 1) > 1) &
          (detarray(angle + 1,pos + 1) > 1)
    detarray(angle,pos) = 1;
   end
   if (detarray(acc,pos) < 2) & (detarray(angle + 1,pos) < 2)
    detarray(angle,pos) = 1;
   end
```

```
    if (detarray(acc,pos) < 2) & (detarray(acc,pos + 1) < 2) &
            (detarray(angle + 1,pos + 1) > 1)
     detarray(angle,pos) = 1;
    end
    if (detarray(angle + 1,pos) < 2) & (detarray(angle + 1,pos + 1) <
            2) & (detarray(acc,pos + 1) > 1)
     detarray(angle,pos) = 1;
    end
   end
  end
end
angle = acc;
for pos = 2:(distmax - 1)
 if detarray(angle,pos) == 3
  if (detarray(angle - 1,pos + 1) < 2) & (detarray(angle,pos + 1) < 2)
            & (detarray(1,pos + 1) < 2)
   if (detarray(angle - 1,pos - 1) > 1) & (detarray(angle,pos - 1) > 1)
            & (detarray(1,pos - 1) > 1)
    detarray(angle,pos) = 1;
   end
   if (detarray(angle - 1,pos) < 2) & (detarray(1,pos) < 2)
    detarray(angle,pos) = 1;
   end
   if (detarray(angle - 1,pos) < 2) & (detarray(angle - 1,pos - 1) < 2)
            & (detarray(1,pos - 1) > 1)
    detarray(angle,pos) = 1;
   end
   if (detarray(1,pos) < 2) & (detarray(1,pos - 1) < 2) &
            (detarray(angle - 1,pos - 1) > 1)
    detarray(angle,pos) = 1;
   end
  end
  if (detarray(angle - 1,pos - 1) < 2) & (detarray(angle,pos - 1) < 2) &
            (detarray(1,pos - 1) < 2)
   if (detarray(angle - 1,pos + 1) > 1) & (detarray(angle,pos + 1) > 1)
            & (detarray(1,pos + 1) > 1)
    detarray(angle,pos) = 1;
   end
   if (detarray(angle - 1,pos) < 2) & (detarray(1,pos) < 2)
    detarray(angle,pos) = 1;
   end
   if (detarray(angle - 1,pos) < 2) & (detarray(angle - 1,pos + 1) < 2)
            & (detarray(1,pos + 1) > 1)
    detarray(angle,pos) = 1;
   end
   if (detarray(1,pos) < 2) & (detarray(1,pos + 1) < 2) &
            (detarray(angle - 1,pos + 1) > 1)
    detarray(angle,pos) = 1;
   end
  end
 end
end

% reset all points to nominal values for main algorithm

for angle = 1:acc
 for pos = 1:distmax
  if detarray(angle,pos) == 3
   detarray(angle,pos) = 2;
  end
 end
end

%
% created : 12/09/95
% latest : 13/09/95
%     by : M.J. Steedman
% status : Development version
%

---------------------------------------------------------------------------


function[detarray] = thnval04(detarray,acc,distmax)
%
% function[detarray] = thnval04(detarray,acc,distmax)

% mark edge points

for angle = 1:acc
 for pos = 2:(distmax - 1)
  if detarray(angle,pos) == 0
   if ((detarray(angle,pos - 1) > 0) & (detarray(angle,pos + 1) == 0)) |
((detarray(angle,pos + 1) > 0) & (detarray(angle,pos - 1) == 0))
    detarray(angle,pos) = -1;
   end
  end
 end
end

% delete points on edges which are not required for continuity

for angle = 2:acc - 1
 for pos = 2:(distmax - 1)
  if detarray(angle,pos) == -1
   if (detarray(angle - 1,pos + 1) < 2) & (detarray(angle,pos + 1) < 2)
            & (detarray(angle + 1,pos + 1) < 2)
    if (detarray(angle - 1,pos - 1) > 1) & (detarray(angle,pos - 1) > 1)
            & (detarray(angle + 1,pos - 1) > 1)
     detarray(angle,pos) = 1;
    end
    if (detarray(angle - 1,pos) < 2) & (detarray(angle + 1,pos) < 2)
     detarray(angle,pos) = 1;
    end
    if (detarray(angle - 1,pos) < 2) & (detarray(angle - 1,pos - 1) < 2)
            & (detarray(angle + 1,pos - 1) > 1)
     detarray(angle,pos) = 1;
    end
    if (detarray(angle + 1,pos) < 2) & (detarray(angle + 1,pos - 1) <
            2) & (detarray(angle - 1,pos - 1) > 1)
     detarray(angle,pos) = 1;
    end
   end
   if (detarray(angle - 1,pos - 1) < 2) & (detarray(angle,pos - 1) < 2)
            & (detarray(angle + 1,pos - 1) < 2)
    if (detarray(angle - 1,pos + 1) > 1) & (detarray(angle,pos + 1) >
            1) & (detarray(angle + 1,pos + 1) > 1)
     detarray(angle,pos) = 1;
    end
    if (detarray(angle - 1,pos) < 2) & (detarray(angle + 1,pos) < 2)
     detarray(angle,pos) = 1;
    end
    if (detarray(angle - 1,pos) < 2) & (detarray(angle - 1,pos + 1) <
            2) & (detarray(angle + 1,pos + 1) > 1)
     detarray(angle,pos) = 1;
    end
    if (detarray(angle + 1,pos) < 2) & (detarray(angle + 1,pos + 1) <
            2) & (detarray(angle - 1,pos + 1) > 1)
     detarray(angle,pos) = 1;
    end
   end
   if (detarray(angle - 1,pos) > 1) & (detarray(angle - 1,pos - 1) > 1)
& (detarray(angle,pos + 1) < 2) & (detarray(angle,pos - 1) > 1) &
(detarray(angle + 1,pos + 1) < 2) & (detarray(angle + 1,pos) < 2)
    detarray(angle,pos) = 1;
   end
   if (detarray(angle + 1,pos) > 1) & (detarray(angle + 1,pos - 1) > 1)
& (detarray(angle,pos + 1) < 2) & (detarray(angle,pos - 1) > 1) &
(detarray(angle - 1,pos + 1) < 2) & (detarray(angle - 1,pos) < 2)
    detarray(angle,pos) = 1;
   end
  end
 end
end

% two missing points, 1 and acc

angle = 1;
for pos = 2:(distmax - 1)
 if detarray(angle,pos) == -1
  if (detarray(acc,pos + 1) < 2) & (detarray(angle,pos + 1) < 2) &
            (detarray(angle + 1,pos + 1) < 2)
   if (detarray(acc,pos - 1) > 1) & (detarray(angle,pos - 1) > 1) &
            (detarray(angle + 1,pos - 1) > 1)
    detarray(angle,pos) = 1;
   end
   if (detarray(acc,pos) < 2) & (detarray(angle + 1,pos) < 2)
    detarray(angle,pos) = 1;
   end
   if (detarray(acc,pos) < 2) & (detarray(acc,pos - 1) < 2) &
            (detarray(angle + 1,pos - 1) > 1)
    detarray(angle,pos) = 1;
   end
   if (detarray(angle + 1,pos) < 2) & (detarray(angle + 1,pos - 1) < 2)
            & (detarray(acc,pos - 1) > 1)
```

```
    detarray(angle,pos) = 1;
   end
  end
  if (detarray(acc,pos - 1) < 2) & (detarray(angle,pos - 1) < 2) &
            (detarray(angle + 1,pos - 1) < 2)
   if (detarray(acc,pos + 1) > 1) & (detarray(angle,pos + 1) > 1) &
            (detarray(angle + 1,pos + 1) > 1)
    detarray(angle,pos) = 1;
   end
   if (detarray(acc,pos) < 2) & (detarray(angle + 1,pos) < 2)
    detarray(angle,pos) = 1;
   end
   if (detarray(acc,pos) < 2) & (detarray(acc,pos + 1) < 2) &
            (detarray(angle + 1,pos + 1) > 1)
    detarray(angle,pos) = 1;
   end
   if (detarray(angle + 1,pos) < 2) & (detarray(angle + 1,pos + 1) <
            2) & (detarray(acc,pos + 1) > 1)
    detarray(angle,pos) = 1;
   end
  end
 end
end
angle = acc;
for pos = 2:(distmax - 1)
 if detarray(angle,pos) == -1
  if (detarray(angle - 1,pos + 1) < 2) & (detarray(angle,pos + 1) < 2)
            & (detarray(1,pos + 1) < 2)
   if (detarray(angle - 1,pos - 1) > 1) & (detarray(angle,pos - 1) > 1)
            & (detarray(1,pos - 1) > 1)
    detarray(angle,pos) = 1;
   end
   if (detarray(angle - 1,pos) < 2) & (detarray(1,pos) < 2)
    detarray(angle,pos) = 1;
   end
   if (detarray(angle - 1,pos) < 2) & (detarray(angle - 1,pos - 1) < 2)
            & (detarray(1,pos - 1) > 1)
    detarray(angle,pos) = 1;
   end
   if (detarray(1,pos) < 2) & (detarray(1,pos - 1) < 2) &
            (detarray(angle - 1,pos - 1) > 1)
    detarray(angle,pos) = 1;
   end
  end
  if (detarray(angle - 1,pos - 1) < 2) & (detarray(angle,pos - 1) < 2) &
            (detarray(1,pos - 1) < 2)
   if (detarray(angle - 1,pos + 1) > 1) & (detarray(angle,pos + 1) > 1)
            & (detarray(1,pos + 1) > 1)
    detarray(angle,pos) = 1;
   end
   if (detarray(angle - 1,pos) < 2) & (detarray(1,pos) < 2)
    detarray(angle,pos) = 1;
   end
   if (detarray(angle - 1,pos) < 2) & (detarray(angle - 1,pos + 1) < 2)
            & (detarray(1,pos + 1) > 1)
    detarray(angle,pos) = 1;
   end
   if (detarray(1,pos) < 2) & (detarray(1,pos + 1) < 2) &
            (detarray(angle - 1,pos + 1) > 1)
    detarray(angle,pos) = 1;
   end
  end
 end
end

% reset all points to nominal values for main algorithm

for angle = 1:acc
 for pos = 1:distmax
  if detarray(angle,pos) == -1
   detarray(angle,pos) = 0;
  end
 end
end

%
% created : 12/09/95
%   latest : 12/09/95
%      by : M.J. Steedman
%  status : Development version
%

---
```

```
function [image] = rfvalc01(image)
%
% function [image] = rfvalc01(image)
%

sz1   = size(image);
rmax  = sz1(1,1);
cmax  = sz1(1,2);

for a = 1:rmax
 for b = 1:cmax
  if image(a,b) == 1
   image(a,b) = 0;
  end
  if image(a,b) == 2
   image(a,b) = 1;
  end
 end
end

%
% created : 12/06/95
%  latest : 12/06/95
%      by : M.J.Steedman
%  status : test version
%

---------------------------------------------------------------------------------


function[detarray] = thnroofo(detarray,acc,distmax)
%
% function[detarray] = thnroofo(detarray,acc,distmax)

for angle = 1:acc
 for pos = 1:distmax
  if detarray(angle,pos) > 0
   if (detarray(angle,pos + 1) > 0)
    if detarray(angle,pos + 2) == 0
     detarray(angle,pos + 1) = 0;
    elseif detarray(angle,pos + 3) == 0
     detarray(angle,pos + 0) = 0;
     detarray(angle,pos + 2) = 0;
    elseif (detarray(angle,pos + 4) == 0)
     detarray(angle,pos + 0) = 0;
     detarray(angle,pos + 2) = 0;
     detarray(angle,pos + 3) = 0;
    elseif (detarray(angle,pos + 5) == 0)
     detarray(angle,pos + 0) = 0;
     detarray(angle,pos + 1) = 0;
     detarray(angle,pos + 3) = 0;
     detarray(angle,pos + 4) = 0;
    elseif (detarray(angle,pos + 6) == 0)
     detarray(angle,pos + 0) = 0;
     detarray(angle,pos + 1) = 0;
     detarray(angle,pos + 3) = 0;
     detarray(angle,pos + 4) = 0;
     detarray(angle,pos + 5) = 0;
    elseif (detarray(angle,pos + 7) == 0)
     detarray(angle,pos + 0) = 0;
     detarray(angle,pos + 1) = 0;
     detarray(angle,pos + 2) = 0;
     detarray(angle,pos + 4) = 0;
     detarray(angle,pos + 5) = 0;
     detarray(angle,pos + 6) = 0;
    elseif (detarray(angle,pos + 8) == 0)
     detarray(angle,pos + 0) = 0;
     detarray(angle,pos + 1) = 0;
     detarray(angle,pos + 2) = 0;
     detarray(angle,pos + 4) = 0;
     detarray(angle,pos + 5) = 0;
     detarray(angle,pos + 6) = 0;
     detarray(angle,pos + 7) = 0;
    elseif (detarray(angle,pos + 9) == 0)
     detarray(angle,pos + 0) = 0;
     detarray(angle,pos + 1) = 0;
     detarray(angle,pos + 2) = 0;
     detarray(angle,pos + 3) = 0;
     detarray(angle,pos + 5) = 0;
     detarray(angle,pos + 6) = 0;
     detarray(angle,pos + 7) = 0;
     detarray(angle,pos + 8) = 0;
    elseif (detarray(angle,pos + 10) == 0)
```

```
  detarray(angle,pos + 0) = 0;
  detarray(angle,pos + 1) = 0;
  detarray(angle,pos + 2) = 0;
  detarray(angle,pos + 3) = 0;
  detarray(angle,pos + 5) = 0;
  detarray(angle,pos + 6) = 0;
  detarray(angle,pos + 7) = 0;
  detarray(angle,pos + 8) = 0;
  detarray(angle,pos + 9) = 0;
elseif (detarray(angle,pos + 11) == 0)
  detarray(angle,pos + 0) = 0;
  detarray(angle,pos + 1) = 0;
  detarray(angle,pos + 2) = 0;
  detarray(angle,pos + 3) = 0;
  detarray(angle,pos + 4) = 0;
  detarray(angle,pos + 6) = 0;
  detarray(angle,pos + 7) = 0;
  detarray(angle,pos + 8) = 0;
  detarray(angle,pos + 9) = 0;
  detarray(angle,pos + 10) = 0;
elseif (detarray(angle,pos + 12) == 0)
  detarray(angle,pos + 0) = 0;
  detarray(angle,pos + 1) = 0;
  detarray(angle,pos + 2) = 0;
  detarray(angle,pos + 3) = 0;
  detarray(angle,pos + 4) = 0;
  detarray(angle,pos + 6) = 0;
  detarray(angle,pos + 7) = 0;
  detarray(angle,pos + 8) = 0;
  detarray(angle,pos + 9) = 0;
  detarray(angle,pos + 10) = 0;
  detarray(angle,pos + 11) = 0;
elseif (detarray(angle,pos + 13) == 0)
  detarray(angle,pos + 0) = 0;
  detarray(angle,pos + 1) = 0;
  detarray(angle,pos + 2) = 0;
  detarray(angle,pos + 3) = 0;
  detarray(angle,pos + 4) = 0;
  detarray(angle,pos + 5) = 0;
  detarray(angle,pos + 7) = 0;
  detarray(angle,pos + 8) = 0;
  detarray(angle,pos + 9) = 0;
  detarray(angle,pos + 10) = 0;
  detarray(angle,pos + 11) = 0;
  detarray(angle,pos + 12) = 0;
elseif (detarray(angle,pos + 14) == 0)
  detarray(angle,pos + 0) = 0;
  detarray(angle,pos + 1) = 0;
  detarray(angle,pos + 2) = 0;
  detarray(angle,pos + 3) = 0;
  detarray(angle,pos + 4) = 0;
  detarray(angle,pos + 5) = 0;
  detarray(angle,pos + 7) = 0;
  detarray(angle,pos + 8) = 0;
  detarray(angle,pos + 9) = 0;
  detarray(angle,pos + 10) = 0;
  detarray(angle,pos + 11) = 0;
  detarray(angle,pos + 12) = 0;
  detarray(angle,pos + 13) = 0;
elseif (detarray(angle,pos + 15) == 0)
  detarray(angle,pos + 0) = 0;
  detarray(angle,pos + 1) = 0;
  detarray(angle,pos + 2) = 0;
  detarray(angle,pos + 3) = 0;
  detarray(angle,pos + 4) = 0;
  detarray(angle,pos + 5) = 0;
  detarray(angle,pos + 6) = 0;
  detarray(angle,pos + 8) = 0;
  detarray(angle,pos + 9) = 0;
  detarray(angle,pos + 10) = 0;
  detarray(angle,pos + 11) = 0;
  detarray(angle,pos + 12) = 0;
  detarray(angle,pos + 13) = 0;
  detarray(angle,pos + 14) = 0;
 end
 end
 end
 end
end


%
% created : 28/07/95
% latest : 28/07/95
%     by : M.J. Steedman
% status : Development version
%

-------------------------------------------------------------------------------


function [fr1] = imtidy01(fr1,maxgap,cutang)

% function [fr1] = imtidy01(fr1,maxgap,cutang)
%
% image-tidy
% cut out the heater support wires function.
% the fringe pattern detected by the fringe detector is trimmed just
% before the fringe linking/tracing stage. Maxgap is taken as a
% parameter so that a gap equal to maxgap + 1 can be created
% centred on the angle at which the heater support artifacts lie
% thereby ensuring that no fringe tracing succesfully crosses this line

sz     = size(fr1);
acc    = sz(1,1);
radius  = sz(1,2);

if (cutang - ((maxgap/2) + 0.5)) > 0
 for pos = 1:radius
  fr1(cutang,pos) = 0;
 end

 for stepout = 1:((maxgap/2) + 0.5)
  for pos = 1:radius
   fr1((cutang - stepout),pos) = 0;
   fr1((cutang + stepout),pos) = 0;
  end
 end
end

%
% created : 27/07/95
% latest : 01/05/96
%     by : M.J. Steedman
% status : Development version
%

-------------------------------------------------------------------------------


function [result] = delsim01(fringes,similarity)
%
% function [result] = delsim01(fringes,similarity)
%
% scans a fringe array and deletes any fringes more than similarity
% percent the same as a previous fringe.

fas     = size(fringes);
maxfringe = fas(1,1);
maxang   = fas(1,2);

for fringe = 2:maxfringe
 if max(fringes(fringe:fringe,1:maxang) > 0)
  for scanfrg = 1:(fringe - 1)
   identcount = 0;
   for pos = 1:maxang
    if (fringes(fringe,pos) == fringes(scanfrg,pos)) &
          (fringes(fringe,pos) ~= 0)
     identcount = identcount + 1;
    end
   end
   if identcount > (maxang * similarity)
    fringes(fringe:fringe,1:maxang) = zeros(1,maxang);
   end
  end
 end
end

result = fringes;

%
% created : 22/01/96
% updated : 22/01/96
%     by : M.J. Steedman
% status : development
%
```

```
function [result] = fillfg03(fringes,noiselim)
%
% function [result] = fillfg03(fringes,noiselim)
%
% gaps in fringes filling algorithm
% nominally noiselim is the fraction of a fringe that must be present
% for it to be processed by the algirtihm, this highlights inputs that are
% more than likely going to give silly interpolated results as if less
% than 60% of a fringe is initially present then its overall shape really
% has to be assumed to be basically elliptical regardless and this
% could be plainly not true.

temp   = size(fringes);
nofrg  = temp(1,1);    % number of fringes to be processed
acc    = temp(1,2);    % number of angles at which the fringe
                       % finding algortihm operated
result = zeros(nofrg,acc);


for presfrng = 1:nofrg  % loop for each fringe

%presfrng = 1;

  fringelength                    = 0;
  tempfrng                        = zeros(1,(acc * 3));
  tempfrng(1:1,1:acc)             = fringes(presfrng:presfrng,1:acc);
  tempfrng(1:1,(acc+1):(2*acc))   = fringes(presfrng:presfrng,1:acc);
  tempfrng(1:1,(2*acc+1):(acc*3)) = fringes(presfrng:presfrng,1:acc);

  % first of all find out how many holes there are in the fringe
  % due to the fringe linking algorithm there might be more than 1
  % any holes that are 'small' which shall be defined as 3 points or
  % less just get filled with a line linking the ends each side, easy
  % if that solves the problem then good. Most likely is that there
  % will be one large hole left to be filled in in which case the
  % size of the hole is required and the angle for each end point

  startgap = 0;
  biggaps  = 0;

  for pos = 2:((acc * 3) - 1)
   if (tempfrng(1,pos) == 0) & (tempfrng(1,pos - 1) > 0)
    startgap = pos - 1;
    if (tempfrng(1,pos+1) > 0)
     tempfrng(1,pos) = ((tempfrng(1,pos - 1) +
                        (tempfrng(1,pos+1))) / 2);
    elseif (tempfrng(1,pos+2) > 0)
     tempfrng(1,pos)   = tempfrng(1,pos - 1);
     tempfrng(1,pos+1) = tempfrng(1,pos + 2);
    elseif (tempfrng(1,pos+3) > 0)
     tempfrng(1,pos)   = tempfrng(1,pos - 1);
     tempfrng(1,pos+1) = ((tempfrng(1,pos - 1) +
                        (tempfrng(1,pos+3))) / 2);
     tempfrng(1,pos+2) = tempfrng(1,pos + 3);
    else
     if ((pos > acc) & (pos < ((acc * 2) + 1)))
      biggaps = biggaps + 1;
      % count up the number of problems to be sorted out
     end
    end
   end
  end

  for a = acc + 1:acc * 2
   if tempfrng(1,a) > 0
    fringelength = fringelength + 1;
   end
  end

  % now iterate sorting out one of the big gaps per pass
  % first find the start and end of a gap, as three copies of fringe in
  % array wrap arround the end gaps are not a problem in this case.
  % note that only fringes with a suitably large length are to be gap
  % filled as this algorithm assumes far too much about the shape of
  % missing sections, chose only fringe over 60% complete already

  if fringelength > (acc * noiselim)
   for gaps = 1:biggaps
    gapstart = 0;
    gapend   = 0;

    for pos = acc + 1:3 * acc
     if (tempfrng(1,pos) == 0) & (tempfrng(1,pos - 1) > 0) & (gapstart
           == 0)
      gapstart = pos - 1;  % position of last known fringe point
     end
    end
    for pos = gapstart:3 * acc
     if (tempfrng(1,pos) > 0) & (tempfrng(1,pos - 1) == 0) & (gapend
           == 0)
      gapend = pos;       % position of next actual fringe point
     end
    end

    % now average trend before fringe dangling end
    avtrendstart = 0;
    avstartcount = 0;
    for trend = gapstart:-1:gapstart - (acc / 8)
     if (trend > 0 ) & (trend < (acc * 3) + 1)
                         % make sure we are in range
      if tempfrng(1,trend) > 0
       avtrendstart = avtrendstart + (tempfrng(1,trend) -
                         tempfrng(1,trend - 1));
       avstartcount = avstartcount + 1;
      end
     end
    end
    avtrendstart = avtrendstart / avstartcount;  % create actual average

    % now average trend after fringe dangling end
    avtrendend   = 0;
    avendcount   = 0;
    for trend = gapend:gapend + (acc / 8)
     if (trend > 0 ) & (trend < (acc * 3) + 1)
                         % make sure we are in range
      if tempfrng(1,trend) > 0
       avtrendend = avtrendend + (tempfrng(1,trend) -
                         tempfrng(1,trend + 1));
       avendcount = avendcount + 1;
      end
     end
    end
    avtrendend = avtrendend / avendcount;     % create actual average

    % now gap fill
    gaparray = zeros(1,((gapend - gapstart) - 1));
    for gappos = 1:((gapend - gapstart) / 2)
     gaparray(gappos) = tempfrng(1,gapstart) + (gappos *
                         avtrendstart);
     firstgapcentre   = gaparray(gappos);
    end
    for gappos = ((gapend - gapstart) - 1):-1:((gapend - gapstart) / 2)
     if gaparray(gappos) == 0
      gaparray(gappos) = tempfrng(1,gapend) + (((gapend - gapstart)
                         - gappos) * avtrendend);
      secondgapcentre  = gaparray(gappos);
     end
    end

    % now average out these so that the two fringes meet
    % at the moment the point gappos((gapend - gapstart) / 2) contains
    % the sum of the two interpolation end points, the average is
    % obviously half of this. Having got this value average the
    % interpolated lines so that the two meet.

%   meetpoint = gaparray((gapend - gapstart) / 2) / 2;
    meetpoint = (firstgapcentre + secondgapcentre) / 2;

    % maxmove = size of correction at gap centre, other corrections
    % are linearly proportional to dist into fringe
    maxmove = meetpoint - firstgapcentre;
%   for gappos = 1:(((gapend - gapstart) / 2) - 1)
    for gappos = 1:((gapend - gapstart) / 2)
     gaparray(gappos) = gaparray(gappos) + (maxmove * (gappos /
                         ((gapend - gapstart) / 2)));
    end

    maxmove = meetpoint - secondgapcentre;
%   for gappos = ((gapend - gapstart) - 1):-1:
                         (((gapend - gapstart) / 2) + 1)
    for gappos = ((gapend - gapstart) - 1):-1:
                         (((gapend - gapstart) / 2) - 1)
     gaparray(gappos) = gaparray(gappos) + (maxmove * (((gapend -
           gapstart) - gappos) / ((gapend - gapstart) / 2)));
```

```
    end

    % insert results
    tempfrng(1,(gapstart + 1):(gapend - 1)) = gaparray;

  end
 end

 % copy results from above end of fringe in tempfrng down
 % this gets stuff that took advantage of there being three copies of
 % the fringe in tempfrng back down into the master central copy

 for pos = acc + 1:acc * 2
  if tempfrng(pos) == 0
   tempfrng(pos) = tempfrng(pos + acc);
  end
 end

%  result = tempfrng(1,(acc + 1):(acc * 2));
 result(presfrng:presfrng,1:acc) = tempfrng(1,(acc + 1):(acc * 2));

end

%
% created : 29/03/95
%  latest : 23/01/96
%      by : M.J. Steedman
%  status : Development version - multifringe processing enabled
%

-------------------------------------------------------------------------------


function [plot] = pltarc04(image,fringesin,intr,intc)

% function [plot] = pltarc04(image,fringesin,intr,intc)
%
% plots the arcs from running slistr algorithm onto image
% second version, to continue plots across small gaps in
% the fringe
%
sz       = size(image);
rows     = sz(1,1);
cols     = sz(1,2);
szfrg    = size(fringesin);
plot     = image;
plotint  = 512;            % plot intensity for detect peaks
nooffrg  = szfrg(1,1);
angles   = szfrg(1,2);     % allow for wrap arround fringe array to be
                           % repeated so it occurs twice

fringes = zeros(nooffrg,angles * 2);
fringes(1:nooffrg,1:angles)                 = fringesin;
fringes(1:nooffrg,(angles + 1):(angles * 2)) = fringesin;


% plot linking adajent fringe points with lines

for fringe = 1:nooffrg
 oldrpt       = 0;
 oldcpt       = 0;
 oneoldrpt    = 0;
 oneoldcpt    = 0;
 twooldrpt    = 0;
 twooldcpt    = 0;
 threeoldrpt  = 0;
 threeoldcpt  = 0;
 fouroldrpt   = 0;
 fouroldcpt   = 0;
 fiveoldrpt   = 0;
 fiveoldcpt   = 0;
 for angle = 1:angles * 2
  if fringes(fringe,angle) > 0
   theta = (2*pi*angle)/angles;
   rpoint = intr + (fringes(fringe,angle)*cos(theta));
   cpoint = intc + (fringes(fringe,angle)*sin(theta));
   if oldrpt > 0
    for step = 0:0.1:1     % link two points
     rplot = (rpoint+((oldrpt-rpoint)*step));
     cplot = (cpoint+((oldcpt-cpoint)*step));
     if ((rplot > 0.5) & (cplot > 0.5) & (rplot < rows + 0.5) & (cplot
          < cols + 0.5))
      plot(rplot,cplot) = plotint;
     end
    end
   elseif oneoldrpt > 0
    for step = 0:0.1:1     % link two points
     rplot = (rpoint+((oneoldrpt-rpoint)*step));
     cplot = (cpoint+((oneoldcpt-cpoint)*step));
     if ((rplot > 0.5) & (cplot > 0.5) & (rplot < rows + 0.5) & (cplot
          < cols + 0.5))
      plot(rplot,cplot) = plotint;
     end
    end
   elseif twooldrpt > 0
    for step = 0:0.1:1     % link two points
     rplot = (rpoint+((twooldrpt-rpoint)*step));
     cplot = (cpoint+((twooldcpt-cpoint)*step));
     if ((rplot > 0.5) & (cplot > 0.5) & (rplot < rows + 0.5) & (cplot
          < cols + 0.5))
      plot(rplot,cplot) = plotint;
     end
    end
   elseif threeoldrpt > 0
    for step = 0:0.1:1     % link two points
     rplot = (rpoint+((threeoldrpt-rpoint)*step));
     cplot = (cpoint+((threeoldcpt-cpoint)*step));
     if ((rplot > 0.5) & (cplot > 0.5) & (rplot < rows + 0.5) & (cplot
          < cols + 0.5))
      plot(rplot,cplot) = plotint;
     end
    end
   elseif fouroldrpt > 0
    for step = 0:0.1:1     % link two points
     rplot = (rpoint+((fouroldrpt-rpoint)*step));
     cplot = (cpoint+((fouroldcpt-cpoint)*step));
     if ((rplot > 0.5) & (cplot > 0.5) & (rplot < rows + 0.5) & (cplot
          < cols + 0.5))
      plot(rplot,cplot) = plotint;
     end
    end
   elseif fiveoldrpt > 0
    for step = 0:0.1:1     % link two points
     rplot = (rpoint+((fiveoldrpt-rpoint)*step));
     cplot = (cpoint+((fiveoldcpt-cpoint)*step));
     if ((rplot > 0.5) & (cplot > 0.5) & (rplot < rows + 0.5) & (cplot
          < cols + 0.5))
      plot(rplot,cplot) = plotint;
     end
    end
   end
   oldrpt = rpoint;
   oldcpt = cpoint;
  else
   fiveoldrpt  = fouroldrpt;
   fiveoldcpt  = fouroldcpt;
   fouroldrpt  = threeoldrpt;
   fouroldcpt  = threeoldcpt;
   threeoldrpt = twooldrpt;
   threeoldcpt = twooldcpt;
   twooldrpt   = oneoldrpt;
   twooldcpt   = oneoldcpt;
   oneoldrpt   = oldrpt;
   oneoldcpt   = oldcpt;
   oldrpt      = 0;
   oldcpt      = 0;
  end
 end
 % deal with the wrap arround link and plot if fringe at 1 & angles
 if (fringes(fringe,1) > 0) & (fringes(fringe,angles) > 0)
  rpoint1 = intr + (fringes(fringe,1)*cos(theta));
  cpoint1 = intc + (fringes(fringe,1)*sin(theta));
  rpoint2 = intr + (fringes(fringe,angles)*cos(theta));
  cpoint2 = intc + (fringes(fringe,angles)*sin(theta));
  for step = 0:0.1:1     % link two points
   rplot = (rpoint1+((rpoint2-rpoint1)*step));
   cplot = (cpoint1+((cpoint2-cpoint1)*step));
   if ((rplot > 0.5) & (cplot > 0.5) & (rplot < rows + 0.5)
        & (cplot < cols + 0.5))
    plot(rplot,cplot) = plotint;
   end
  end
 end
end
```

```
%
% created : 16/02/95
% latest : 18/05/95
%     by : M.J. Steedman
% status : Development version
%
```

Main Algorithm as used for processing the flight results. This is only the replacement sections to the files above.

```
function[xvect,yvect] = rfptarc3(photo,xstart,ystart,cutang,noisefract,
xmax,ymax,sang,eang,angmul,tacc)
%
% function[xvect,yvect] = rfptarc3(photo,xstart,ystart,cutang,
noisefract, xmax,ymax,sang,eang,angmul,tacc)
%
% sectional angle capable version
%
% cutang must be specified within range 3 to tacc if correct fringe
% linking is to be achieved as else linking stages will wrap about start
% points on assumption of processing all 360 degrees of fringes about
% centre.
% xmax   = total rows, needed for file read operation
% ymax   = total cols, needed for file read operation
% sang   = starting angle (numerical value) see correction below.
% algorithm uses value 1 less than actual start angle to save lots of 1
% angle maths corrections
% eang   = ending angle (numerical value)
% angmul = accuracy multiplier, process acc angles, out of 360
% degrees = acc*angmul
%
% roof valley vector creating operator.
% this uses the dilate and thin, fringe noise overcoming system. It is
% applied using the starslice pattern and linked into fringes using the
% advanced adaptive fringe linking developed for starslice and its
% derivatives. fringe linking using lnktstxx.m is then carried out to try
% and overcome any remaining breaks, this uses the multiscalar
% version of the fringe end tangent plotting fringe linker. Finally the
% remaining fringes are thresholded for length to minimise rubbish
% due to image edges being detected and the presence of noise or
% small fringe segments.
% the results are all save to disk along with many intermediate results.
%
% to G:\imageres\
%
% rfvect10.dry    diary file
% rfvt_raw.mat  srvdts9d.m    raw sliregch.m fringe roof and valley
%                             detection results
% rfvt_det.mat  srvdts9d.m    result after iterative dilate and thin stage
% rftv_frg.mat  srvdts9d.m    result after fringe linking stage - fringes
% rfvt_res.mat  srvdts9d.m    result after fringe linking stage - plotted
%                             results
% rftv_lnk.mat [not stored]   results of lnkwca05.m fringes after
%               [ tangent ]   tangent limking algorithm. this
%               [ linker  ]   algorithm is allowed to link fringe
%               [ disabled ]  across the heater wires as fringe ends
%                             badly distorted by heater wire will not
%                             be parrallel
% rftv_sim.mat  'int16'       fringes after deletion of overly similar
%                             fringes
% rftv_unf.mat  'int16'       fringes just before gap filling algorithm
%                             but after noise fringe suppresion
% rftv_fcg.mat  'int16'       final centres of gravity
% rftv_cog.mat  'int16'       centres of gravity before noise
%                             suppression
% rftv_mng.mat 'int16'        mean of centres of interpolated
%                             circle centres
% rftv_mdg.mat 'int16'        median of centres of interpolated
%                             circle centres
% rftv_usf.mat  'int16'       fringe sections actually used in
%                             the centre determinations, a part
%                             of the final fringe array.
% rftv_rfg.mat  'int16'       final fringe results afer noise suppresion
% rfvt_frs.mat  'int16'       all final fringes plotted onto photo image
% rfvt_vct.mat  'float'       final gravity vector
% rfvt_cnt.mat  'float'       centres of fringes used for gravity vector
%                             calulation
% rfvt_frv.mat  'int16'       final fringes with centres and vector plotted
%                             onto the image as well
%

diary h:\imageres\rfvectgt.dry

% constants
passnos    = 2;          % number of dilate and thin stages
acc        = tacc; %512% number of angles algoruthm operates at
sang       = sang - 1;   % correct sang, as matlab arrays start at
                         % 1 a one will keep getting added back onto
                         % this value hence giving the correct
                         % starting angle
maxgap     = 1;          % maximum permitted fringe gap in fringe
                         % linking stage, reduced from2 due to errors
                         % with image photo10c.mat, noted good
                         % fringes don't have any gaps at all.
thinsp     = 5;          % operations of thinning stages per pass.
complete   = 0.5;        % fraction of fringe to be present before it
                         % will be completed by gap filler and used
                         % for vector calculations
complete   = complete / angmul;
                         % correction so complete fraction is fraction
                         % of a whole fringe detection over the
                         % angles actually scanned not over 360 when
                         % it could be below complete % of 360
                         % degrees
maxsimilarity = 0.8;     % maximum fringe similarity permitted
                         % before overly similar fringes are deleted
                         % leaving only the first detected fringe

cd h:\matlab\matforh

resfringe = srvdts9e(photo,passnos,acc,maxgap,thinsp,xstart,ystart,
cutang,xmax,ymax,sang,eang,angmul);
% srvdts9c.m algorithm, saves most results to disk
% srvdtis9.m version returns the detected fringes array.

sz1       = size(resfringe);
fringetot  = sz1(1,1);
fringelgth = sz1(1,2);

% but carry out ultra similar fringe suppression, as point reuse
% allowed get rid of fringes over maxsimilarity% similar

resfringe = delsim01(resfringe,maxsimilarity);

cd h:\imageres

fid   = fopen('rftv_sim.mat','w');
count = fwrite(fid,resfringe,'int16');

cd h:\matlab\matforh

% now scan fringe and discard if just noise
% set noise level as 25% of angles

noiselevel = acc * noisefract;

for fringenos = 1:fringetot
 pointsf   = 0;

 for scanpos = 1:acc
  if resfringe(fringenos,scanpos) > 0
   pointsf = pointsf + 1;
  end
 end

 if pointsf < noiselevel   % discard fringe
  for scanpos = 1:acc
   resfringe(fringenos,scanpos) = 0;
  end
 end
end


% having thrown out the noise plot the remaining fringes one at at
```

```
% time and find thier centres also calculate a vector through these
% centres on the assumption though it may be invalid that the first
% fringe is the innermost.

centres   = zeros(fringetot,2);
centrenos = 0;

cd h:\imageres

fid   = fopen('rftv_unf.mat','w');
count = fwrite(fid,resfringe,'int16');

cd h:\matlab\matforh

% start partial arc moment section

% collect fringe sections from the fringes that are present

% resfrgtoscan = zeros(fringetot,1:(minfend - minfstart + 1));
% size(resfrgtoscan)

insfrg = 0;
for fringe = 1:fringetot
 if max(max(resfringe(fringe:fringe,1:fringelgth))) > 0
  insfrg = insfrg + 1;
  resfrgtoscan(insfrg:insfrg,1:fringelgth) =
            resfringe(fringe:fringe,1:fringelgth);
 end
end

scanedfrgs = 0;
szrfts     = size(resfrgtoscan);
scanedfrgs = szrfts(1,1)
lengthsgrf = szrfts(1,2)

% finds centres of gravity for the fringe sections, used to
% correct for their displacment in image from each other

if scanedfrgs > 0
 for fringenos = 1:scanedfrgs
  [xord,yord] = momntd01(resfrgtoscan(fringenos:fringenos,
            1:lengthsgrf),xstart,ystart,sang,eang,angmul);
  centrenos = centrenos + 1;
  centres(centrenos,1) = xord;
  centres(centrenos,2) = yord;
 end
end

% find corrected centres of gravity

% end partial arc moment section

% vector
xvect = 0;
yvect = 0;
vector = [xvect,yvect];

sfcg = size(fincgrav);
rfcg = sfcg(1,1);
cfcg = sfcg(1,2);

if rfcg > 1

 for vectornos = 2:rfcg
  xvect = xvect + (fincgrav(1,1) - fincgrav(vectornos,1));
  yvect = yvect - (fincgrav(1,2) - fincgrav(vectornos,2));
 end

 xvect = xvect / (rfcg - 1);
 yvect = yvect / (rfcg - 1);

 vector = [yvect,xvect];

 rowofcentre = xvect
 colofcentre = yvect

end

% store matalb results to disk for checkups after - final fringes array

cd h:\imageres

fid   = fopen('rftv_rfg.mat','w');
count = fwrite(fid,resfringe,'int16');

cd h:\bitmap

fid = fopen(photo,'r');
pht = fread(fid,[xmax,ymax],'char');

cd h:\matlab\matforh

res = pltparc4(pht,resfringe(1:fringetot,1:tacc),xstart,ystart,sang,eang,
            angmul);

cd h:\imageres

fid     = fopen('rfvt_frs.mat','w');
count = fwrite(fid,res,'int16');
fid     = fopen('rfvt_vct.mat','w');
count = fwrite(fid,vector,'float');
fid     = fopen('rfvt_cnt.mat','w');
count = fwrite(fid,centres,'float');

sz1       = size(res);
maxrow = sz1(1,1);
maxcol  = sz1(1,2);

for icentre = 1:centrenos
 if (centres(icentre,2) > 0.5) & (centres(icentre,2) < (maxrow + 0.5))
& (centres(icentre,1) > 0.5) & (centres(icentre,1) < (maxcol + 0.5))
  res(centres(icentre,2),centres(icentre,1)) = 256;
 end
end

for pvector = 1:20
 if (xstart - (vector(1,2)*pvector/20) > 0.5) & (xstart -
            (vector(1,2)*pvector/20) < (maxrow + 0.5)) & (ystart +
            (vector(1,1)*pvector/20) > 0.5) & (ystart +
            (vector(1,1)*pvector/20) < (maxcol + 0.5))
  res(xstart - (vector(1,2)*pvector/20),ystart +
            (vector(1,1)*pvector/20)) = 256;
 end
end

fid   = fopen('rfvt_frv.mat','w');
count = fwrite(fid,res,'int16');

cd h:\matlab\matforh

diary off      % stop diary for g:\imageres\rfvect10.dry

fclose('all')   % close all files, avoids data loss if computer crashes
later

%
%  created : 05/09/95
%     latest : 22/09/95
% adaption : 22/01/96 fringe reusuing version allows multiple fringes
%                     to use a set of fringe points, no tangent linking
%                     stage.with trap to delete multiple detections of
%                     basically identical fringes
%     latest : 23/01/96
% adaption : 09/02/96 tidied version uses int16 storage to avoid 256 to
%                     0 conversions where undesirable, vector creator
%                     no longer uses start point, produces fringes,
%                     plus centres plus gravity vector final all data
%                     image
%     latest : 12/02/96
% adaption : 29/04/96 gt for ground test version. Process ground test
%                     images generated with the real flight hardware.
%     latest : 01/05/96 : sector processing version
%     latest : 04/06/96
%   no blur : 12/06/96 : rfvcgtnb.m no per operaor overall bluring
%                     stage in srvdts9e.m sliregfh.m
% partarcm : 13/06/96 : moments from partial arcs operation
%  rfptarc1 : 14/06/96 : sorted out moments by partial arcs with some
%                     attempt to control the arc fraction required for
%                     acceptance
%     latest : 18/06/96 : error corrections
%  rfptarc1 : 20/06/96 : circle interpolating with drift correction.
%  rfptarc2 :
%  rfptarc3 : 06/08/96 : less the corrective terative center finder to
%                     give results from the highly broken up flight
%                     results where two overlapping fringe results
%                     are not guaranteed, hence iterative version
```

```
%                          entered continual loop
%     by : M.J. Steedman
%  status : Development version
%

-------------------------------------------------------------------------------



function[fringe] = srvdts9e(photo,passnos,acc,maxgap,thinsp,xstart,
ystart,cutang,xmax,ymax,sang,eang,angmul)
%
% function[fringe] = srvdts9e(photo,passnos,acc,maxgap,thinsp,xstart,
ystart,cutang,xmax,ymax,sang,eang,angmul)
%
% Slice - roof - valley - dilate - thin - iterative - subroutine.
%
% photo   = photograph    = 'xxxxxxxx.xxx'  : typical = 'photo7c.mat'
% passnos = dilate/thin iterations     = x       : typical = 4
% acc     = angles to operate at       = x       : typical = 512
% maxgap = gap allowed by linker  = x       : typical = 2
% thinsp  = thins per dilation          = x       : typical = 5
% xstart  = x coordinate start          = xxx     : typical = 128
% ystart  = y coordinate start          = xxx     : typical = 128
% cutang  = cutangle for heater wire = xxx     : typical = 256
%

%
% This version c allows a fringe to be used multiple times. A fringe
% point is only allowed to start a fringe once but by setting used
% fringe points to another but non zero value they can be crossed
% again by later fringe tracing for use only with rfvect07 without
% tangent linking stage.
%
% using heater wire cut out step as well.
% built on the srvdth11.m algorithm but with iterative dilation and
% thinning. This should minimise interfringe linking while allowing
% more dilation stages, and still accomplish results as the thinning
% used in this algorithm is 'along slices' only so angular spread is
% maintained. Also reworking thinning stage to work with the roof
% and valley image so that the valleys are retained for future dilations.
%
% fringe extraction test using the dilate and thin approach working
% from a detection of both the roofs and valleys in a fringe image
%
% slice, roof, valley, dilate, thin, algorithm
% operates slireg, roof and valley finder on image, followed by
% frvalc01.m which converts this to a roof only image. rvtoad01.m is
% used to generate the image in angle-radius form so the peak
% thinning averaging routine can be applied along image slices. The
% data is then returned to the plain format for the dilation stage. The
% results of this being thinned with passnos + 1 applications of
% thin013.m
%
% version 6 uses group along a slice down to 1 rather than thinning
% algorithm to control the dilation. This 'thins' the dilated fringe
% pattern at right angles to the fringes but does not trim back any
% fringe length like a normal thinning algorithm does, hence no loss
% of detections

cd h:\bitmap

fid  = fopen(photo,'r');
pht7 = fread(fid,[xmax,ymax],'char');

cd h:\matlab\matforh

sz1    = size(pht7);
rmax   = sz1(1,1);
cmax   = sz1(1,2);
fringetot = 50;

rfval = sliregfh(pht7,xstart,ystart,acc,sang,eang,angmul);

% having got detarray a clean up can now be carried out to remove
% multiple detections of the same fringe, it can be safely assumed
% that any sequences of points in the detect array are actually all
% multiple detections of one fringe

cd h:\imageres

% store matalb results to disk for checkups after - raw detections roof
% valley

fid  = fopen('rfvt_raw.mat','w');
count = fwrite(fid,rfval,'char');

cd h:\matlab\matforh

detsize  = size(rfval);
detlength = detsize(1,2);

for passes = 1:passnos

 % dilation first, valleys then roofs

 detarray = rfval;

 for a = 1:acc
  for b = 2:(detlength - 1)
   if (a == 1)
    a1 = acc;
   else
    a1 = a - 1;
   end
   if (a == acc)
    a2 = 1;
   else
    a2 = a + 1;
   end

   if detarray(a,b) == 0;   % case valley, expand
    if rfval(a1,b-1) == 1
     rfval(a1,b-1) = 0;
    end
    if rfval(a,b-1) == 1
     rfval(a,b-1) = 0;
    end
    if rfval(a2,b-1) == 1
     rfval(a2,b-1) = 0;
    end
    if rfval(a1,b) == 1
     rfval(a1,b) = 0;
    end
    if rfval(a2,b) == 1
     rfval(a2,b) = 0;
    end
    if rfval(a1,b+1) == 1
     rfval(a1,b+1) = 0;
    end
    if rfval(a,b+1) == 1
     rfval(a,b+1) = 0;
    end
    if rfval(a2,b+1) == 1
     rfval(a2,b+1) = 0;
    end
   end
   if detarray(a,b) == 2;   % case roof, expand
    if rfval(a1,b-1) == 1
     rfval(a1,b-1) = 2;
    end
    if rfval(a,b-1) == 1
     rfval(a,b-1) = 2;
    end
    if rfval(a2,b-1) == 1
     rfval(a2,b-1) = 2;
    end
    if rfval(a1,b) == 1
     rfval(a1,b) = 2;
    end
    if rfval(a2,b) == 1
     rfval(a2,b) = 2;
    end
    if rfval(a1,b+1) == 1
     rfval(a1,b+1) = 2;
    end
    if rfval(a,b+1) == 1
     rfval(a,b+1) = 2;
    end
    if rfval(a2,b+1) == 1
     rfval(a2,b+1) = 2;
    end
   end
  end
 end
end
```

```
if passes > 1       % ie hugely wide detections are under control
                    % second dilation stage to close larger gaps.

 detarray = rfval;

 for a = 1:acc
  for b = 2:(detlength - 1)
   if (a == 1)
    a1 = acc;
   else
    a1 = a - 1;
   end
   if (a == acc)
    a2 = 1;
   else
    a2 = a + 1;
   end

   if detarray(a,b) == 0;   % case valley, expand
    if rfval(a1,b-1) == 1
     rfval(a1,b-1) = 0;
    end
    if rfval(a,b-1) == 1
     rfval(a,b-1) = 0;
    end
    if rfval(a,b-1) == 1
     rfval(a2,b-1) = 0;
    end
    if rfval(a1,b) == 1
     rfval(a1,b) = 0;
    end
    if rfval(a2,b) == 1
     rfval(a2,b) = 0;
    end
    if rfval(a1,b+1) == 1
     rfval(a1,b+1) = 0;
    end
    if rfval(a,b+1) == 1
     rfval(a,b+1) = 0;
    end
    if rfval(a2,b+1) == 1
     rfval(a2,b+1) = 0;
    end
   end
   if detarray(a,b) == 2;   % case roof, expand
    if rfval(a1,b-1) == 1
     rfval(a1,b-1) = 2;
    end
    if rfval(a,b-1) == 1
     rfval(a,b-1) = 2;
    end
    if rfval(a2,b-1) == 1
     rfval(a2,b-1) = 2;
    end
    if rfval(a1,b) == 1
     rfval(a1,b) = 2;
    end
    if rfval(a2,b) == 1
     rfval(a2,b) = 2;
    end
    if rfval(a1,b+1) == 1
     rfval(a1,b+1) = 2;
    end
    if rfval(a,b+1) == 1
     rfval(a,b+1) = 2;
    end
    if rfval(a2,b+1) == 1
     rfval(a2,b+1) = 2;
    end
   end
  end
 end
end

% detarray = thinroof(detarray,acc,distmax);
% detarray = thinvall(detarray,acc,distmax);

for thin = 1:thinsp
 rfval = thnrof04(rfval,acc,detlength);
end

for thin = 1:thinsp
 rfval = thnval04(rfval,acc,detlength);
 end

end

% first convert results to the right format

detarray = rfvalc01(rfval);

% ensure thinned properly
detarray = thnroofo(detarray,acc,detlength - 20);

cd h:\imageres

% store matalb results to disk for checkups after - rooves only image
fid    = fopen('rfvt_det.mat','w');
count = fwrite(fid,detarray,'char');

cd h:\matlab\matforh

% now operate fringe linking algorithm
% after first removing the heater wires

detarray = imtidy01(detarray,maxgap,cutang);

% now use detarray to extract arcs directly from detected points
% aim is to extract whole features in one algorithm rather than
% marking edge points with starslice and then trying to get something
% from that data at a later stage.

% first arrays to hold fringe point sequences

% fringetot = 5; make this a parameter

fringe = zeros(fringetot,acc);
finingesfound = 0;
lastfringe     = 0;
fringenos      = 0;
distmax   = detlength - 10; % because of old arraysize of distmax + 10
noisefract     = 0.02;

% for fringenos = 1:fringetot

while lastfringe == 0
 fringenos = fringenos + 1;
 if fringenos == fringetot
  lastfringe = 1;
 end

 rad     = 5;    % resets ready for next fringe (was 7)
 angle   = 0;
 firsthit = 0;

 while((rad < (distmax - 2)) & (firsthit == 0))
  rad = rad + 1;
  angle = 0;
  while((angle < acc) & (firsthit == 0))
   angle = angle + 1;
   if (detarray(angle,rad) > 0) & (detarray(angle,rad) < 10)
    fringe(fringenos,angle) = rad;
    detarray(angle,rad) = 10;  % different but non zero value
                               % records used, so not for fringe
                               % starts but may be reused for
                               % tracing through
    firsthit = 1;
    fringesfound = fringesfound + 1;
   end
  end
 end

% trap case of not found enough fringes but have exhausted array
% i.e. asked to find more than there was to find.

 if (firsthit == 0)
  lastfringe = 1;
 end

 firstrad    = rad;
 nodetectsfor = 0;
 stoppedat    = 0;              % position at which forward sweep
                                % was stopped by fringe end, used
                                % as an alternative stop for the
                                % reverse sweep so that it does not
                                % overwrite a fringe that has
```

```
                    % already been found.

for wrapnang = angle + 1:angle + acc
 nang = wrapnang;
 if nang > acc
  nang = nang - acc;
 end
 lastdiff = 2;                  % default radius difference
                                % between last pair of detects
 if (nang > 4)
  if ((fringe(fringenos,nang-1) > 0) & (fringe(fringenos,nang-2) > 0)
& (fringe(fringenos,nang-3) > 0) & (fringe(fringenos,nang-4) > 0))
    lastdiff = fringe(fringenos,nang-1) - ((fringe(fringenos,nang-2) +
          fringe(fringenos,nang-3) + fringe(fringenos,nang-4)) / 3);
  end
 elseif (nang > 3)
  if ((fringe(fringenos,nang-1) > 0) & (fringe(fringenos,nang-2) > 0)
          & (fringe(fringenos,nang-3) > 0))
    lastdiff = fringe(fringenos,nang-1) - ((fringe(fringenos,nang-2) +
          fringe(fringenos,nang-3)) / 2);
  end
 elseif nang > 2
  if ((fringe(fringenos,nang-1) > 0) & (fringe(fringenos,nang-2) >
          0))
    lastdiff = fringe(fringenos,nang-1) - fringe(fringenos,nang-2);
  end
 end
 if lastdiff < 0
  lastdiff = 0 - lastdiff;
 end
 if rad < 9
  rad = 9;
 end
 if nodetectsfor < (maxgap + 1)
  if detarray(nang,rad) > 0
   detarray(nang,rad) = 10;
   fringe(fringenos,nang) = rad;
   nodetectsfor = 0;
 elseif detarray(nang,rad + 1) > 0
  detarray(nang,rad + 1) = 10;
  fringe(fringenos,nang) = rad + 1;
  nodetectsfor = 0;
  if rad < (distmax - 2)
   rad = rad + 1;
  end
 elseif detarray(nang,rad - 1) > 0
  detarray(nang,rad - 1) = 10;
  nodetectsfor = 0;
  fringe(fringenos,nang) = rad - 1;
  if rad > 7
   rad = rad - 1;
  end
 elseif (detarray(nang,rad + 2) > 0) & (lastdiff > -1)
  detarray(nang,rad + 2) = 10;
  nodetectsfor = 0;
  fringe(fringenos,nang) = rad + 2;
  if rad < (distmax - 3)
   rad = rad + 2;
  end
 elseif (detarray(nang,rad - 2) > 0) & (lastdiff > -1)
  detarray(nang,rad - 2) = 10;
  nodetectsfor = 0;
  fringe(fringenos,nang) = rad - 2;
  if rad > 8
   rad = rad - 2;
  end
 elseif (detarray(nang,rad + 3) > 0) & (lastdiff > 0)
  detarray(nang,rad + 3) = 10;
  nodetectsfor = 0;
  fringe(fringenos,nang) = rad + 3;
  if rad < (distmax - 4)
   rad = rad + 3;
  end
 elseif (detarray(nang,rad - 3) > 0) & (lastdiff > 0)
  detarray(nang,rad - 3) = 10;
  nodetectsfor = 0;
  fringe(fringenos,nang) = rad - 3;
  if rad > 9
   rad = rad - 3;
  end
 elseif (detarray(nang,rad + 4) > 0) & (lastdiff > 1)
  detarray(nang,rad + 4) = 10;
  nodetectsfor = 0;
  fringe(fringenos,nang) = rad + 4;
  if rad < (distmax - 5)
   rad = rad + 4;
  end
 elseif (detarray(nang,rad - 4) > 0) & (lastdiff > 1)
  detarray(nang,rad - 4) = 10;
  nodetectsfor = 0;
  fringe(fringenos,nang) = rad - 4;
  if rad > 10
   rad = rad - 4;
  end
 elseif ((detarray(nang,rad + 5) > 0) & (lastdiff > 2))
  detarray(nang,rad + 5) = 10;
  nodetectsfor = 0;
  fringe(fringenos,nang) = rad + 5;
  if rad < (distmax - 6)
   rad = rad + 5;
  end
 elseif (detarray(nang,rad - 5) > 0) & (lastdiff > 2)
  detarray(nang,rad - 5) = 10;
  nodetectsfor = 0;
  fringe(fringenos,nang) = rad - 5;
  if rad > 11
   rad = rad - 5;
  end
 elseif ((detarray(nang,rad + 6) > 0) & (lastdiff > 3))
  detarray(nang,rad + 6) = 10;
  nodetectsfor = 0;
  fringe(fringenos,nang) = rad + 6;
  if rad < (distmax - 7)
   rad = rad + 6;
  end
 elseif (detarray(nang,rad - 6) > 0) & (lastdiff > 3)
  detarray(nang,rad - 6) = 10;
  nodetectsfor = 0;
  fringe(fringenos,nang) = rad - 6;
  if rad > 12
   rad = rad - 6;
  end
 elseif ((detarray(nang,rad + 7) > 0) & (lastdiff > 4))  % 4
  detarray(nang,rad + 7) = 10;
  nodetectsfor = 0;
  fringe(fringenos,nang) = rad + 7;
  if rad < (distmax - 8)
   rad = rad + 7;
  end
 elseif (detarray(nang,rad - 7) > 0) & (lastdiff > 4)
  detarray(nang,rad - 7) = 10;
  nodetectsfor = 0;
  fringe(fringenos,nang) = rad - 7;
  if rad > 13
   rad = rad - 7;
  end
 elseif ((detarray(nang,rad + 8) > 0) & (lastdiff > 5))  % 5
  detarray(nang,rad + 8) = 10;
  nodetectsfor = 0;
  fringe(fringenos,nang) = rad + 8;
  if rad < (distmax - 9)
   rad = rad + 8;
  end
 elseif (detarray(nang,rad - 8) > 0) & (lastdiff > 5)
  detarray(nang,rad - 8) = 10;
  nodetectsfor = 0;
  fringe(fringenos,nang) = rad - 8;
  if rad > 14
   rad = rad - 8;
  end
 else
  nodetectsfor = nodetectsfor + 1;
 end
else
 if stoppedat == 0
  stoppedat = wrapnang - (maxgap + 1);
 end
end

rad        = firstrad;
nodetectsfor = 0;

for wrapnang = angle - 1:-1:angle - acc
 nang = wrapnang;
 if nang < 1
```

```
  nang = nang + acc;
 end
 lastdiff = 2;                  % default radius difference
                                % between last pair of detects
 if (nang < acc - 3)
  if ((fringe(fringenos,nang+1) > 0) & (fringe(fringenos,nang+2) >
0) & (fringe(fringenos,nang+3) > 0) & (fringe(fringenos,nang+4) > 0))
   lastdiff = fringe(fringenos,nang+1) - ((fringe(fringenos,nang+2)
+ fringe(fringenos,nang+3) + fringe(fringenos,nang+4)) / 3);
  end
 elseif (nang < acc - 2)
  if ((fringe(fringenos,nang+1) > 0) & (fringe(fringenos,nang+2) >
          0) & (fringe(fringenos,nang+3) > 0))
   lastdiff = fringe(fringenos,nang+1) - ((fringe(fringenos,nang+2)
          + fringe(fringenos,nang+3)) / 2);
  end
 elseif nang < acc - 1
  if ((fringe(fringenos,nang+1) > 0) & (fringe(fringenos,nang+2) >
          0))
   lastdiff = fringe(fringenos,nang+1) - fringe(fringenos,nang+2);
  end
 end
 if lastdiff < 0
  lastdiff = 0 - lastdiff;
 end
 if rad < 9
  rad = 9;
 end
 if (nodetectsfor < (maxgap + 1)) & (wrapnang > (stoppedat - acc))
  if detarray(nang,rad) > 0
   detarray(nang,rad) = 10;
   nodetectsfor = 0;
   fringe(fringenos,nang) = rad;
  elseif detarray(nang,rad + 1) > 0
   detarray(nang,rad + 1) = 10;
   nodetectsfor = 0;
   fringe(fringenos,nang) = rad + 1;
   if rad < (distmax - 2)
    rad = rad + 1;
   end
 elseif detarray(nang,rad - 1) > 0
  detarray(nang,rad - 1) = 10;
  nodetectsfor = 0;
  fringe(fringenos,nang) = rad - 1;
  if rad > 7
   rad = rad - 1;
  end
 elseif (detarray(nang,rad + 2) > 0) & (lastdiff > -1)
  detarray(nang,rad + 2) = 10;
  nodetectsfor = 0;
  fringe(fringenos,nang) = rad + 2;
  if rad < (distmax - 3)
   rad = rad + 2;
  end
 elseif (detarray(nang,rad - 2) > 0) & (lastdiff > -1)
  detarray(nang,rad - 2) = 10;
  nodetectsfor = 0;
  fringe(fringenos,nang) = rad - 2;
  if rad > 8
   rad = rad - 2;
  end
 elseif (detarray(nang,rad + 3) > 0) & (lastdiff > 0)
  detarray(nang,rad + 3) = 10;
  nodetectsfor = 0;
  fringe(fringenos,nang) = rad + 3;
  if rad < (distmax - 4)
   rad = rad + 3;
  end
 elseif (detarray(nang,rad - 3) > 0) & (lastdiff > 0)
  detarray(nang,rad - 3) = 10;
  nodetectsfor = 0;
  fringe(fringenos,nang) = rad - 3;
  if rad > 9
   rad = rad - 3;
  end
 elseif (detarray(nang,rad + 4) > 0) & (lastdiff > 1)
  detarray(nang,rad + 4) = 10;
  nodetectsfor = 0;
  fringe(fringenos,nang) = rad + 4;
  if rad < (distmax - 5)
   rad = rad + 4;
  end
 elseif (detarray(nang,rad - 4) > 0) & (lastdiff > 1)
  detarray(nang,rad - 4) = 10;
  nodetectsfor = 0;
  fringe(fringenos,nang) = rad - 4;
  if rad > 10
   rad = rad - 4;
  end
 elseif ((detarray(nang,rad + 5) > 0) & (lastdiff > 2))
  detarray(nang,rad + 5) = 10;
  nodetectsfor = 0;
  fringe(fringenos,nang) = rad + 5;
  if rad < (distmax - 6)
   rad = rad + 5;
  end
 elseif (detarray(nang,rad - 5) > 0) & (lastdiff > 2)
  detarray(nang,rad - 5) = 10;
  nodetectsfor = 0;
  fringe(fringenos,nang) = rad - 5;
  if rad > 11
   rad = rad - 5;
  end
 elseif ((detarray(nang,rad + 6) > 0) & (lastdiff > 3))
  detarray(nang,rad + 6) = 10;
  nodetectsfor = 0;
  fringe(fringenos,nang) = rad + 6;
  if rad < (distmax - 7)
   rad = rad + 6;
  end
 elseif (detarray(nang,rad - 6) > 0) & (lastdiff > 3)
  detarray(nang,rad - 6) = 10;
  nodetectsfor = 0;
  fringe(fringenos,nang) = rad - 6;
  if rad > 12
   rad = rad - 6;
  end
 elseif ((detarray(nang,rad + 7) > 0) & (lastdiff > 4))  % 4
  detarray(nang,rad + 7) = 10;
  nodetectsfor = 0;
  fringe(fringenos,nang) = rad + 7;
  if rad < (distmax - 8)
   rad = rad + 7;
  end
 elseif (detarray(nang,rad - 7) > 0) & (lastdiff > 4)
  detarray(nang,rad - 7) = 10;
  nodetectsfor = 0;
  fringe(fringenos,nang) = rad - 7;
  if rad > 13
   rad = rad - 7;
  end
 elseif ((detarray(nang,rad + 8) > 0) & (lastdiff > 5))  % 5
  detarray(nang,rad + 8) = 10;
  nodetectsfor = 0;
  fringe(fringenos,nang) = rad + 8;
  if rad < (distmax - 9)
   rad = rad + 8;
  end
 elseif (detarray(nang,rad - 8) > 0) & (lastdiff > 5)
  detarray(nang,rad - 8) = 10;
  nodetectsfor = 0;
  fringe(fringenos,nang) = rad - 8;
  if rad > 14
   rad = rad - 8;
  end
 else
  nodetectsfor = nodetectsfor + 1;
 end
end

% now scan fringe and discard if just noise
% set niose level as 15% of angles

noiselevel = acc * noisefract;
pointsf    = 0;

for scanpos = 1:acc
 if fringe(fringenos,scanpos) > 0
  pointsf = pointsf + 1;
 end
end

if pointsf < noiselevel    % discard fringe
 for scanpos = 1:acc
  fringe(fringenos,scanpos) = 0;
```

```
  end
  fringenos = fringenos - 1;
 end

end                     % now increment fringetot

clear detarray;         % used so clear to save memory

% array of fringes now aquired so plot and save the results

res = pltparc4(pht7,fringe,xstart,ystart,sang,eang,angmul);

cd h:\imageres

% store matalb results to disk for checkups after - fringes array
fid     = fopen('rftv_frg.mat','w');
count = fwrite(fid,fringe,'char');
fid     = fopen('rfvt_res.mat','w');
count = fwrite(fid,res,'char');

cd h:\matlab\matforh

fclose('all')

%
%       9e : 12/06/96 : no initial bluring, call sliregfh.m
%    latest : 21/05/96
%       9d : 01/05/96 : sector processing version, called by
%                       rfvectgt.m master algorithm
% adapted : 22/01/96   multi fringe using
% created : 05/09/95
%    latest : 15/09/95
%       by : M.J. Steedman
%   status : Development version
%
```

```
function[result,fringe,detections] = sliregfh(image,initcr,initcc,acc,
sang,eang,angmul)
%
% function[result,fringe,detections] = sliregfh(image,initcr,initcc,acc,
sang,eang,angmul)
%
% slireg07 and on, development
% set out to improve the performance of the algorithm for fringes near
% the limits of the image by offsetting the slices by a distance into the
% image this gives some zeros at the centre so the operators can go
% much closer to the image centre than before and also the outside
% overrun is extended
%
% slireg array of ones valleys are set to zeros and roofs to 2, format
% is a matrix rows by cols rather than the dist and angle arrangement
%
% slice-store, save detections to allow direct line rather than peak
% detection
% slistl13 - tighten up lastdiff constraint, this proves that angle
% limiter is too constraining and also very time consuming to use.
% checking detailed results from slistl12 shows that in good fringes
% the change in radius from one radius to the next is only +/- 2 from
% the last and the largest hole in a fringe is one point missed and that
% is very rare.
% slistl12.m - as slistl09 (notl1) adds maxgap parameter, equal to the
% maximum permitted run of zeros between points to be detected as
% in the same fringe - but no angle limit
% slistl09.m - as slistr08 but with extra blurring of image slices in
% order to reduce the effect of high frequency noise on the operator
% slistl08.m - as slistr07.m but multiscalar detectors turned on
% widths are 13, 19, 25, 31, 37 : basic slistr07 only runs the 13 wide
% detector
%
% new starslice algorithm based on the idea of the star algorithm but
% using a completely new feature finding method involving gradients,
% this version detects zero crossings of the image gradient that look
% like non local maxima ( i.e. it filters out noise spurs ) in this case
% points where the gradient goes form +ve to -ve without bouncing
% back across the zero point right alongside. the algorithm takes the
% average of the slope between four sequential pairs of points as the
% gradient, it then scans this vector of locally averaged gradients. in
% this scanning pass it sums the five gradients before the present
% point and if the result is greater than +3 assumes that the image has
% been increasing intensity up to this point along this vector. The five
% points on from the present scan point are likewise summed and
% assumed to represent a drop if their sum is below -3. A fringe ( a
% `bright` ridge in the image ) is declared to have been found if both
% the above tests are true for a centre point which has a gradient value
% of > -2 and < 2
%
% this tooled up version ignores points with plus or minus slopes of
% over 150 as these generally only occur if the image edge has been
% overlapped giving a false edge as the scan arrays are pre-set at zero
% beyond these points. It also checks that rad-1 > 0 and rad+1 < 0 to
% ensure that the point in question is actually a place where the
% gradient crosses zero and not a dip below gradient +2 just before a
% crossing, something that has been giving several false detects up to
% now.
%
% success of the new algorithm as described above this has been
% tested on photo5.mat and over half the vectors detect the main
% fringe in the image with the only spurious detects being in the noise
% at the edges of the picture well outside the main image area.
%
% this version reduces the number of multiple detects and is perfect
% on the vertically downwards slice which has been used as the one
% for which graphs of the intermediate stages are extracted to see
% what is actually going on the problem is though that it does loose
% some of the real limit detects on the very poor fringes that are
% heavily disrupted by the high frequency noise that this image
% contains from being a screen photograph
%
% latest starsl11 version without the gradient at radius rad >-2 and <2
% detector, this reduces the safety net but much decreases the chance
% that a sharp swing of the gradient through zero will cause the
% crossing point to be missed by this safeguard. now contains a
% sensible progress indicator output, no `junk` output no dummy
% parameter requirements.
%
% star algorithm for finding the centre of gravity of shapes in a grey
% level image
% see Paper No 36 for basics of the star algorithm
% The general idea is that you scan out along a number of equally
% radially spaced points from an initial start point until you detect an
% image feature by a suitable thresholded function becoming true.
% This is for use in grey level images with the intention that the `stop`
% points for each angle will highlight the desired image feature. In
% order to improve the boundary detection the algorithm having
% found all the stop points for the initial centre now works out their
% centre of gravity and then repeats using this point as the new start
% point, and so it goes on until the centre of gravity of the stop points
% `matches` the start point.
%
% starsl12.m : higher speed version has been hacked to streamline
% execution pre computation used where possible to speed operations.
%
% starsl13.m distmax now corrected, more 'junk' removal
% starsl14.m multiscalar version
%

sz1     = size(image);
rmax    = sz1(1,1);
cmax    = sz1(1,2);
xy      = zeros(64,2);
cr      = initcr;
cc      = initcc;
oldcr   = 0;
oldcc   = 0;

%result = ones(rmax,cmax);

% presmoothing - cut.

% image = blurim01(image);

% calculates the maximum distance from the start point specified to
% an image corner and sets this as the longest required slice length.

if cr > rmax / 2
 rmaxsquare = cr*cr;
else
 rmaxsquare = (rmax - cr) * (rmax - cr);
end
if cc > cmax / 2
 cmaxsquare = cc*cc;
else
 cmaxsquare = (cmax - cc) * (cmax - cc);
```

```
end
distmax = sqrt(rmaxsquare + cmaxsquare) + 40;
                                % +20 * 2 due to need extra
                                % zeros required if processing
                                % is to reach the image edge
                                % and centre due to the
                                % width of the operators.
result = ones(acc,distmax);

% now slice length is known create array to hold detected points.

detarray = zeros(acc,distmax+10);
                            % + 10 to avoid out of range errors from
                            % the point linking algorithm

gdsph   = 2;                % constant used for results visibility

totangs = acc*angmul;

for angle = 1:acc

  theta = 2*pi*((angle + sang) / (acc * angmul));
  slsp  = zeros(distmax,1);   % slicespace holds the slice of the image
                              % saves repeated complex references to
                              % the main image.
  % first copy image information into slicespace array the 20 offset on
  % slsp gives the 20 zeros in the image centre to allow the operators
  % to start scanning before the image points are reached therby
  % allowing them to detect fringes much closer to the start point

  for rad = 1:distmax - 30
   if ((cr+(rad*cos(theta)) < rmax) & (cr+(rad*cos(theta)) > 1))
    if ((cc+(rad*sin(theta)) < cmax) & (cc+(rad*sin(theta)) > 1))
     slsp(rad + 20,1) =
              image(cr+(rad*cos(theta)),cc+(rad*sin(theta)));
    else
     slsp(rad + 20,1) = 0;
    end
   else
    slsp(rad + 20,1) = 0;
   end
  end

  % pre - smooth the present image slice in order to ease the effects
  % of high frequency noise
  % uses the vertical over 5 point smoother blur_line_vertical
  % two applications brings in some effect from a nine pixel region

  slsp  = burlnv01(slsp);
  slsp2  = burlnv01(slsp);    % incremental bluring
  slsp3  = burlnv01(slsp2);


  % now calculate the gradient at each point along the present radius in
  % preparation for finding the zero crossing points.

  gradsave = zeros(distmax,1);

  for rad = 3:(distmax - 2)
   gradtemp = [(slsp(rad+2)-slsp(rad+1)),(slsp(rad+1)-
             slsp(rad)),(slsp(rad)-slsp(rad-1)),(slsp(rad-1)-slsp(rad-2))];
   for aa = 1:3
    for bb = 1:3
     if gradtemp(bb) > gradtemp(bb + 1)
      temp = gradtemp(bb);
      gradtemp(bb) = gradtemp(bb + 1);
      gradtemp(bb + 1) = temp;
     end
    end
   end
   gradsave(rad) = (gradtemp(3) + gradtemp(2)) / 2;
  end

  gradsave2 = zeros(distmax,1);

  for rad = 3:(distmax - 2)
   gradtemp = [(slsp(rad+2)-slsp(rad+1)),(slsp(rad+1)-
             slsp(rad)),(slsp(rad)-slsp(rad-1)),(slsp(rad-1)-slsp(rad-2))];
   for aa = 1:3
    for bb = 1:3
     if gradtemp(bb) > gradtemp(bb + 1)
      temp = gradtemp(bb);
      gradtemp(bb) = gradtemp(bb + 1);
      gradtemp(bb + 1) = temp;
     end
    end
   end
   gradsave2(rad) = (gradtemp(3) + gradtemp(2)) / 2;
  end

  gradsave3 = zeros(distmax,1);

  for rad = 3:(distmax - 2)
   gradtemp = [(slsp(rad+2)-slsp(rad+1)),(slsp(rad+1)-
             slsp(rad)),(slsp(rad)-slsp(rad-1)),(slsp(rad-1)-slsp(rad-2))];
   for aa = 1:3
    for bb = 1:3
     if gradtemp(bb) > gradtemp(bb + 1)
      temp = gradtemp(bb);
      gradtemp(bb) = gradtemp(bb + 1);
      gradtemp(bb + 1) = temp;
     end
    end
   end
   gradsave3(rad) = (gradtemp(3) + gradtemp(2)) / 2;
  end

  % now have found somethign if the gradient = zero and the five
  % gradient values before it total over +5 end the five after total
  % below -5; version 12 use 3 scales +3 to -3, +6 to -6 and +9 to -9

  a     = gradsave;
  b     = gradsave2;
  c     = gradsave3;

  result = slirsub1(result,a,distmax,angle,gdsph);
  result = slirsub1(result,b,distmax,angle,gdsph);
  result = slirsub1(result,c,distmax,angle,gdsph);

  % all done now go to next angle

end

result = result(1:acc,21:distmax);  % 20 offset to get rid fo the 20
                                    % values 'inside' the centre of the
                                    % image that were there simply for
                                    % algorithm operating requirements
%
%       created : 09/06/95
%     fh  latest : 12/06/96 : no initail bluring stage, call by srvdts9e.m
%     eh  latest : 03/05/96 : sector processing version, called by
%                             srvdts9d.m which is run by rfvectgt.m
%                             master algorithm
% slireg  latest : 15/09/95 : this version improvements on slireg06.m
%                           - all angle rad version
% starsl created : 9/8/94
%          latest : 3/2/95
% slistl created : 16/02/95
%          latest : 23/03/95
%             by : M.J. Steedman
%         status : Development version
%
```

Appendix 3

Mechanical Design Drawings

These diagrams show the mechanical design of the flight payload. They were used for the construction of the various subsections along with further diagrams giving precise details of the component parts. Only the overall diagrams and elevations of major components are given here. A more detailed description of the payload is presented in the G-490 SDP [154].

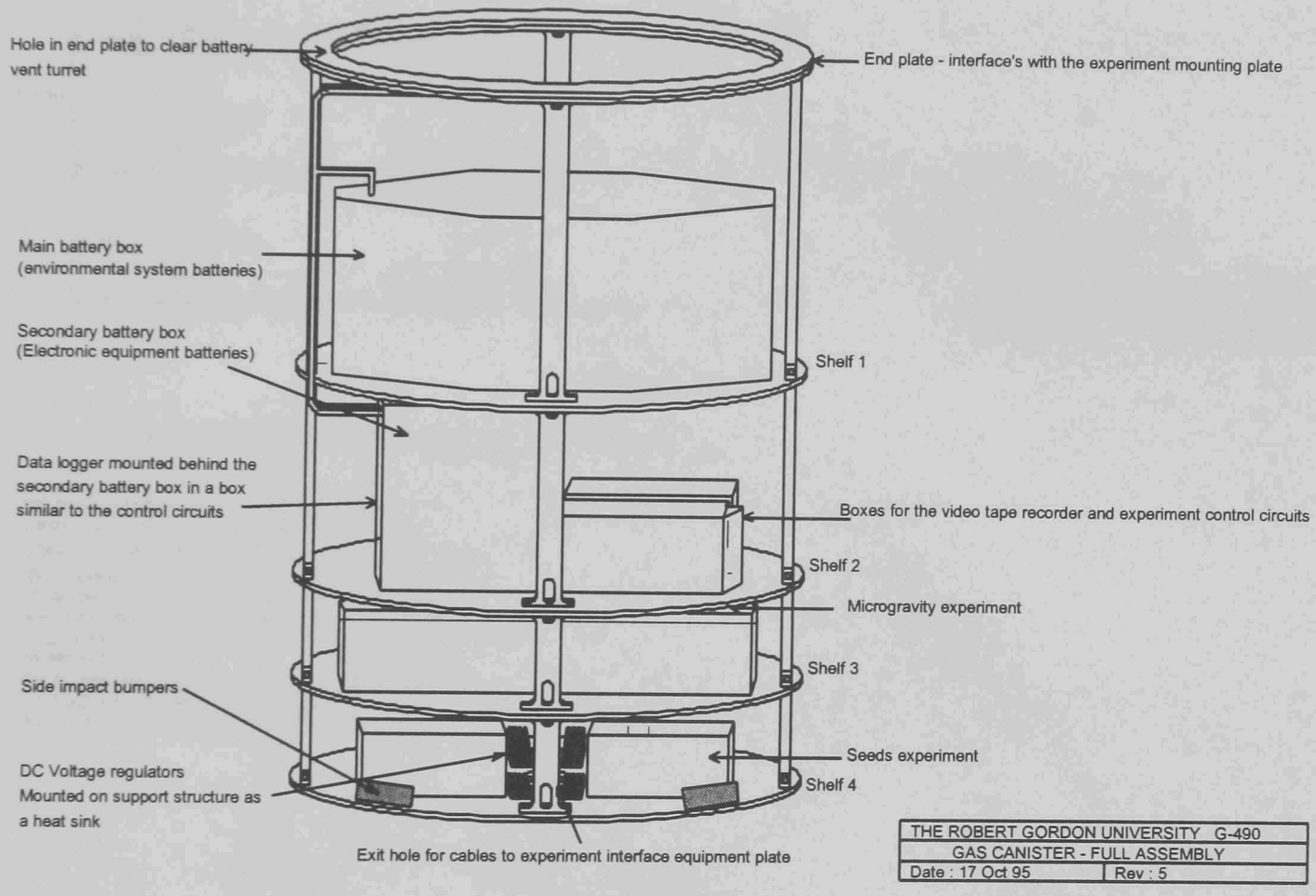


Figure A3.1

The whole experiment in isometric view. This shows the principal components on the payload in the nominal upright position. The payload hangs from the NASA experiment mounting plate which is bolted two the rings at the top of the experiment and which in turn bolted to the upper end of the NASA supplied GAS canister. Details of the NASA carrier hardware can be found in the GAS Experimenters Handbook [151]. The lower end of the payload is supported only in the lateral axis. This support is provided by four side impact bumpers which rest against the inner walls of the GAS canister.

In this view, Figure A3.1, the NASA interface equipment plate which provides the payloads electrical interface with the GAS carrier hardware would fit just below the payload similar to a fifth shelf.

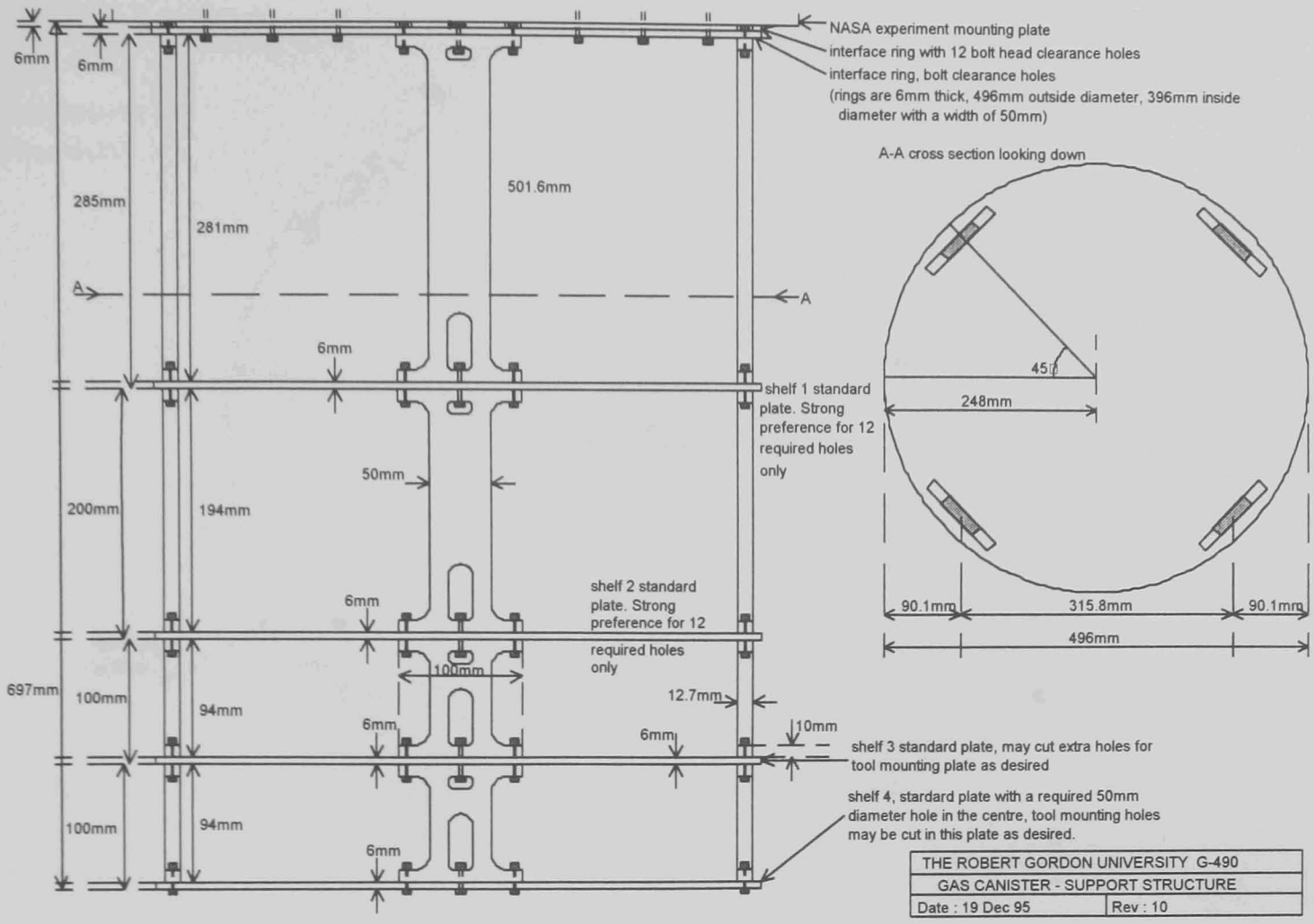


Figure A3.2

Elevation of the main support structure only.

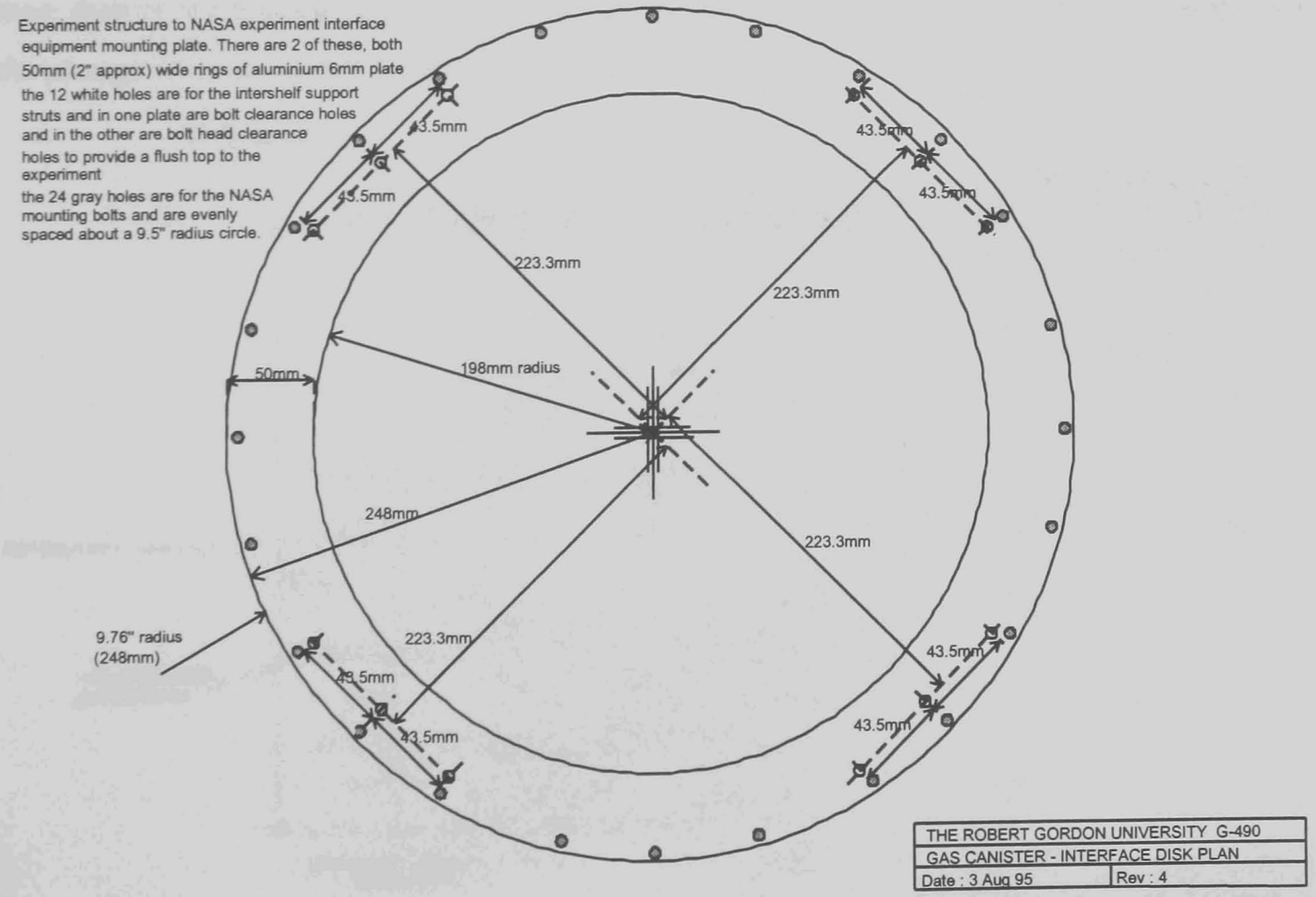


Figure A3.3

Dimensions and hole centres of the rings which interface between the 'I' style experiment support struts and the NASA experiment mounting plate. The latter attaches to the 24 out holes.

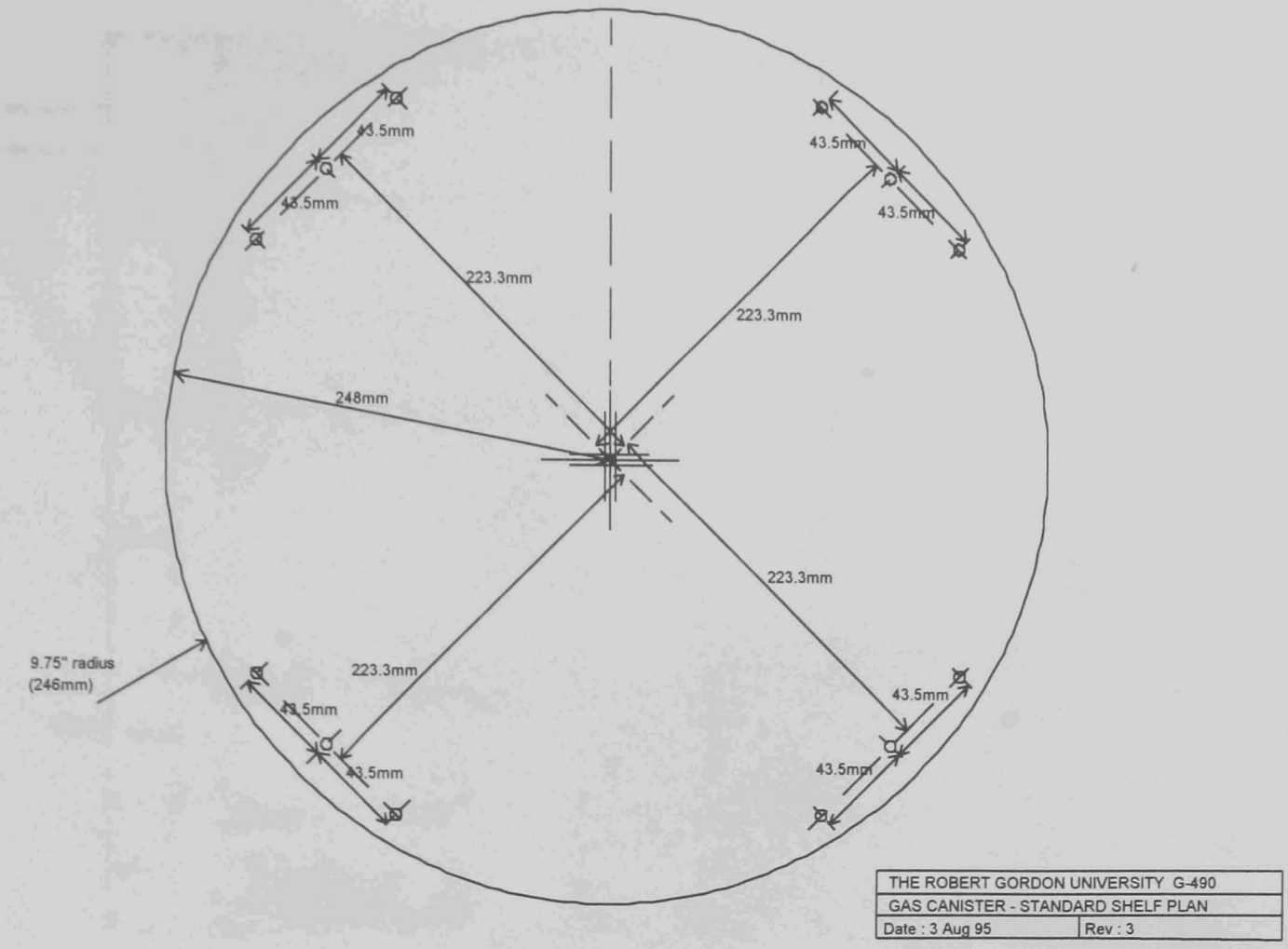


Figure A3.4

Basic shelf dimensions and holes for the support struts. This is the basic pattern of the four main shelves. Each in addition has attachment points for the shelf specific equipment and cutouts for the passage of wiring looms as required.

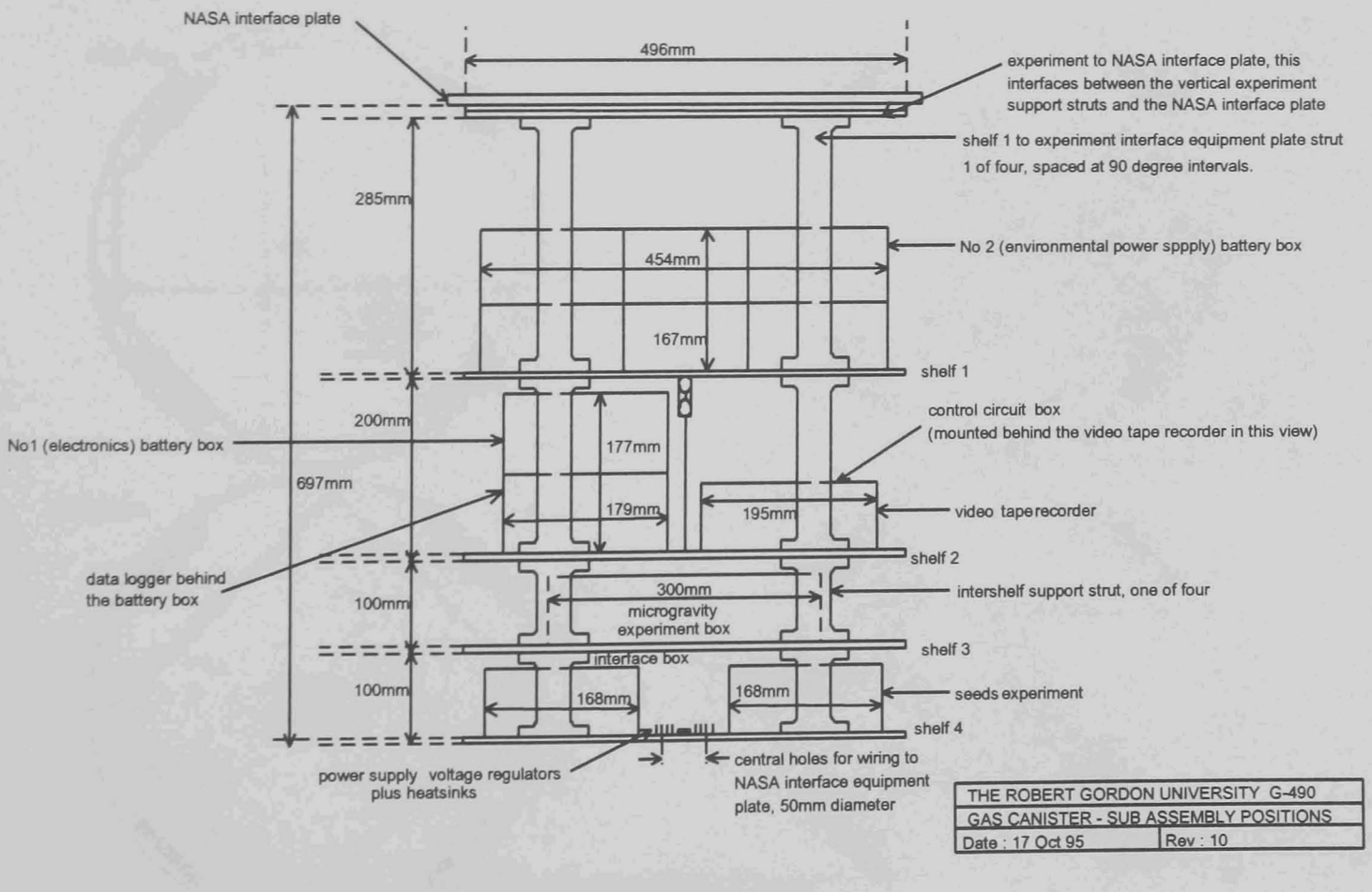


Figure A3.5

Side elevation showing only the major components and principal dimensions.

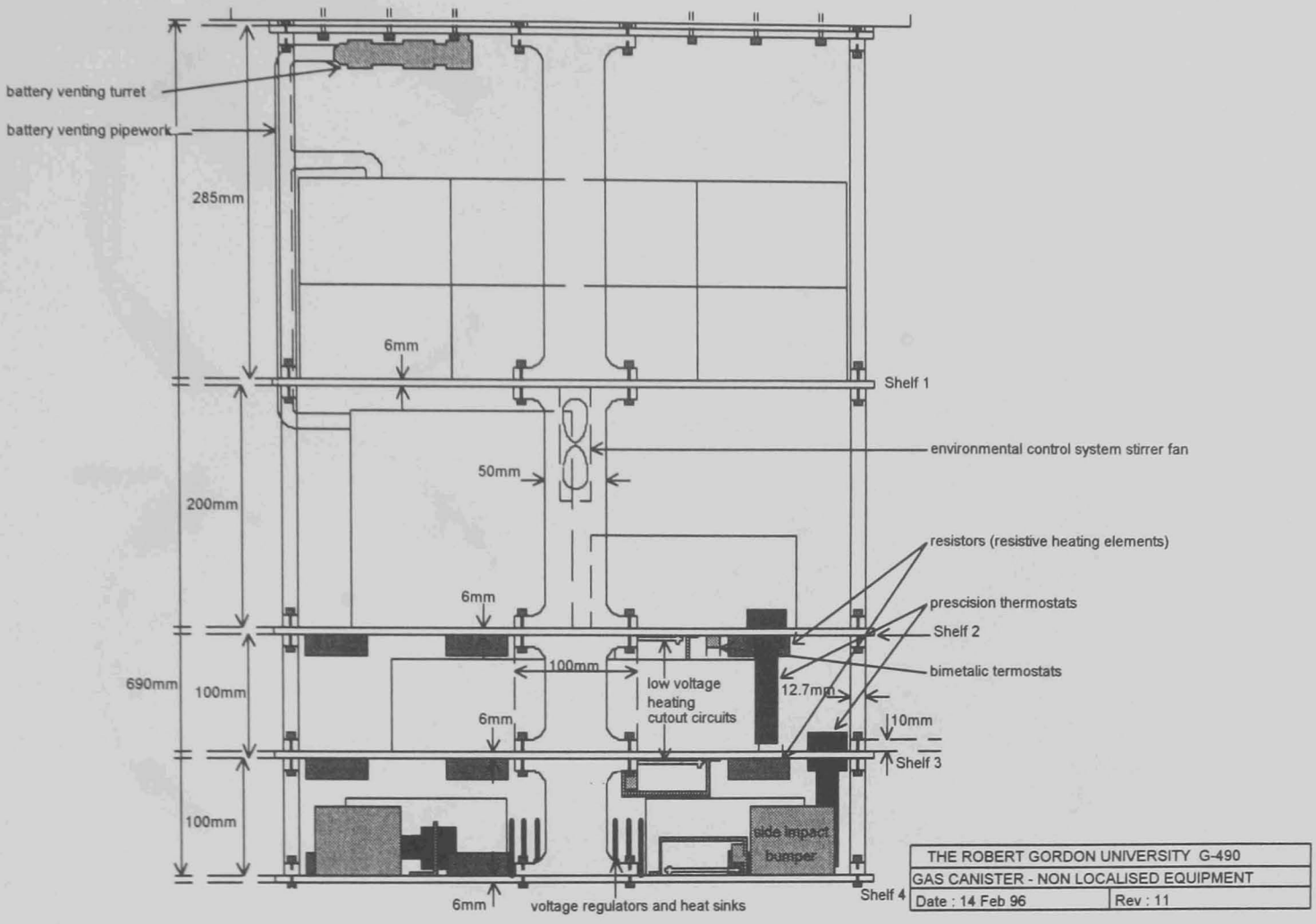


Figure A3.6

Locations of the distributed equipment such as the heating system. This is a general design view, the precise locations of these components were established in situe on the payload.

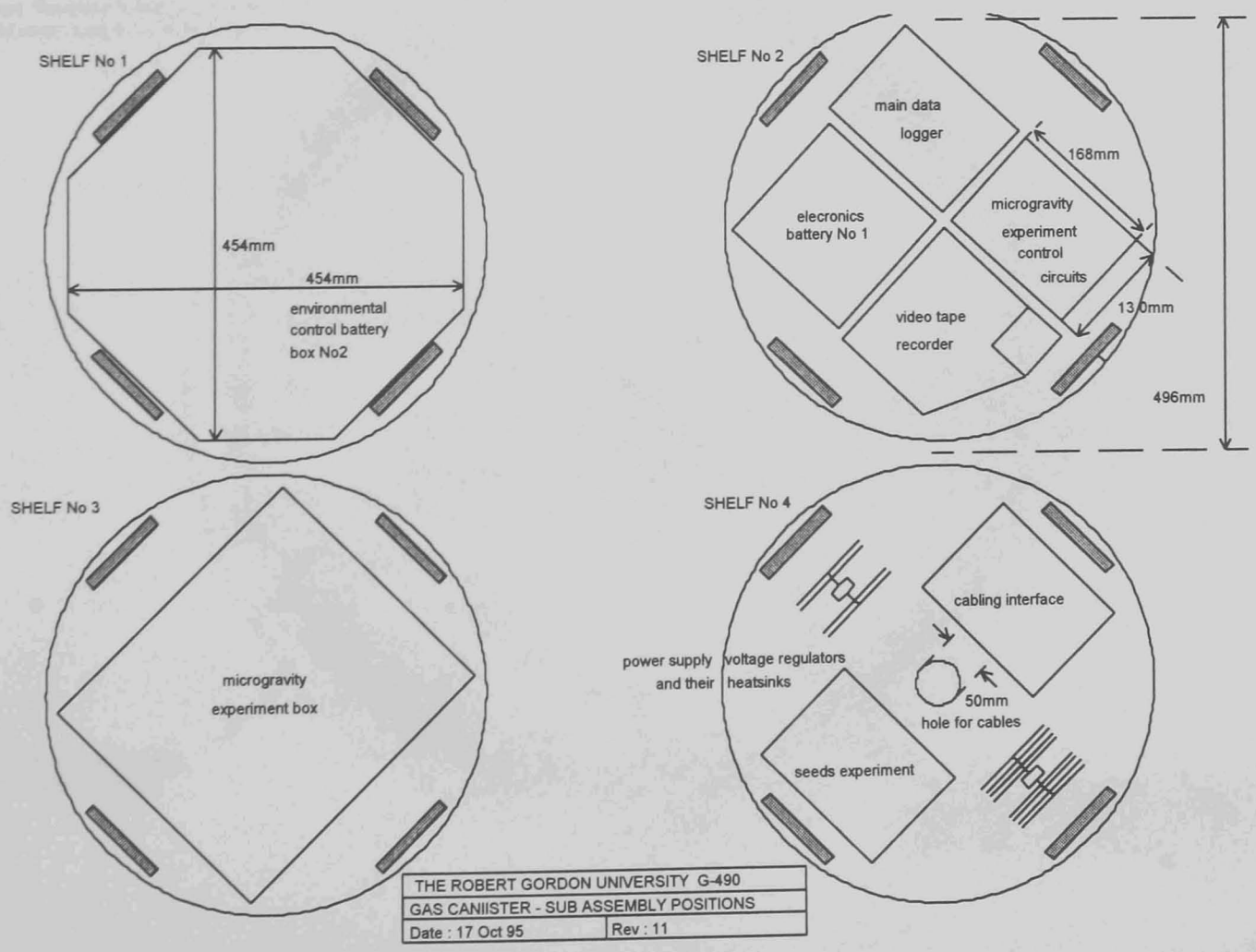


Figure A3.7

Shelf by shelf plan view of principal equipment locations.

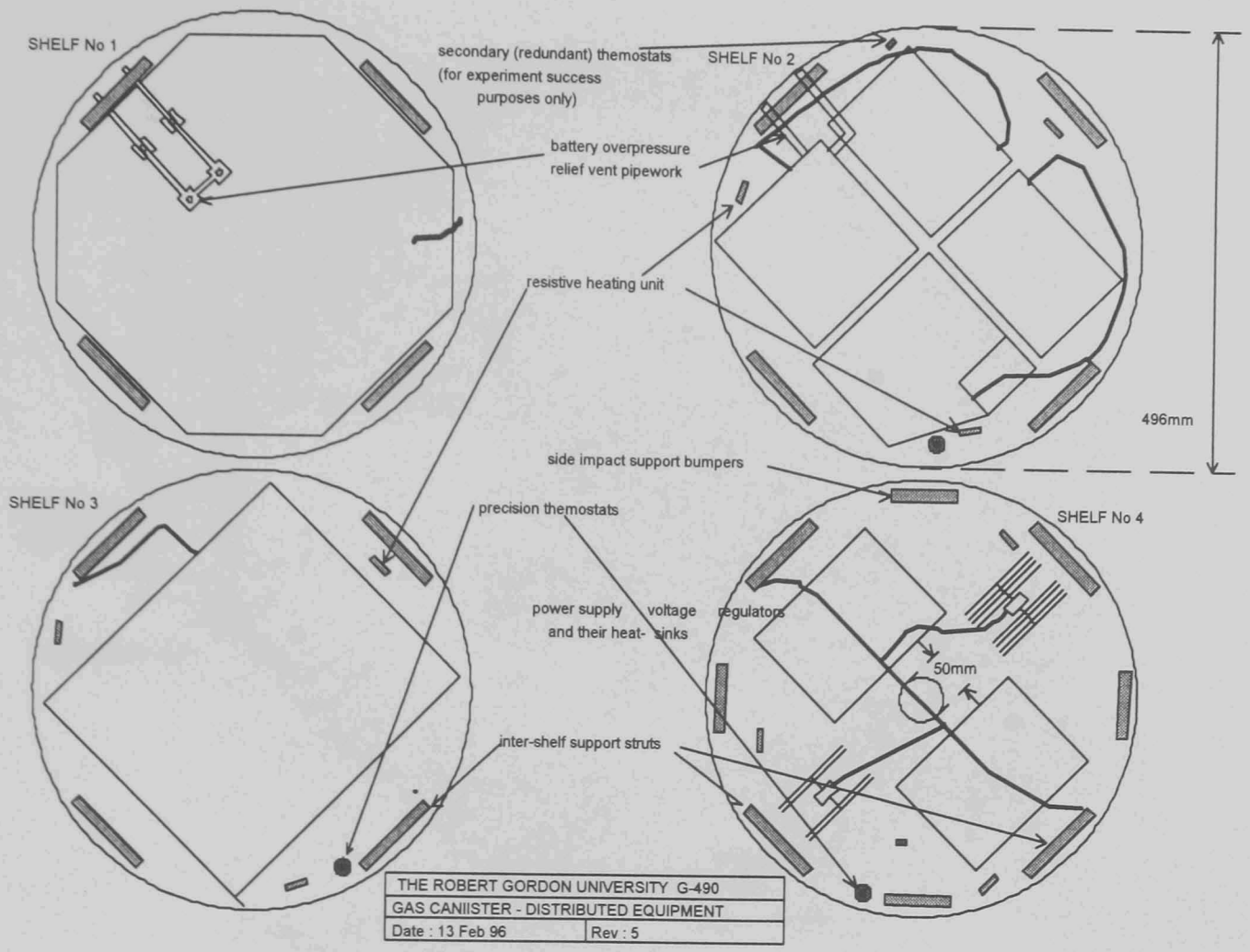


Figure A3.8

Plan view of distributed equipment, principally thermal control system components.

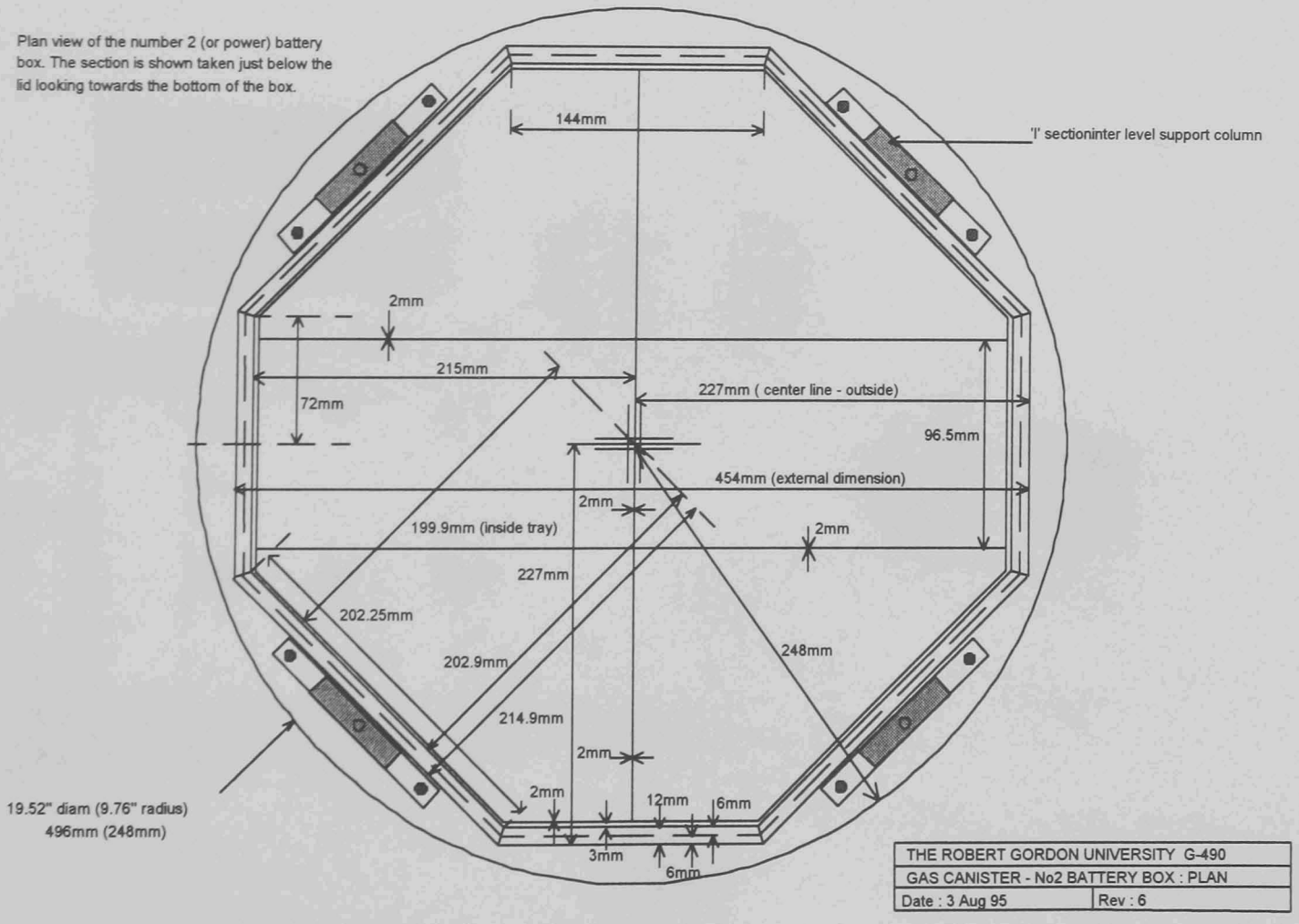


Figure A3.9

Detailed plan of Shelf 1 showing a section through the environmental battery box just below the lid. One tray is illustrated with its six battery compartments.

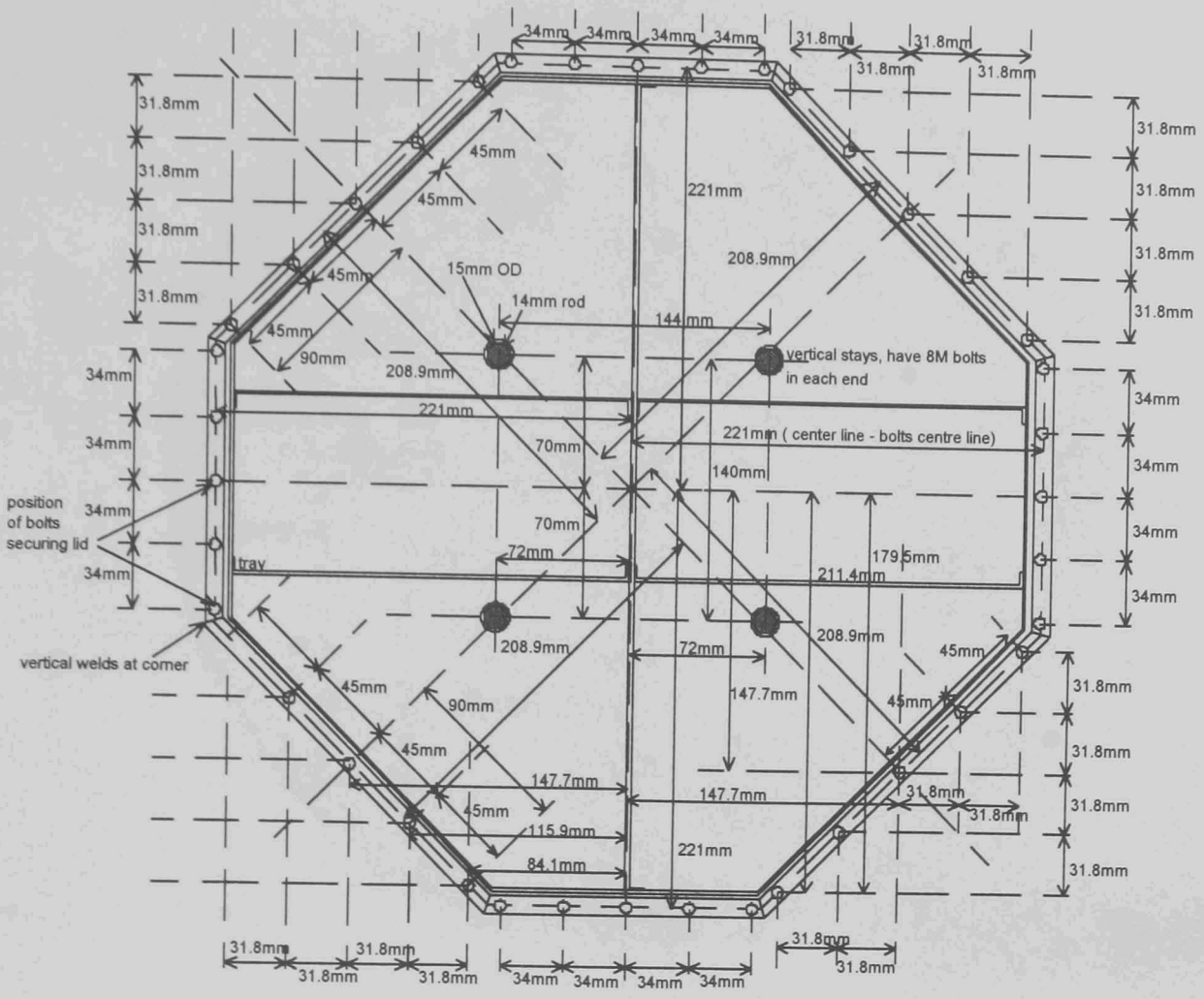


Figure A3.10

Battery box details, lid bolts, lid stays and plan view of one battery tray in situ.

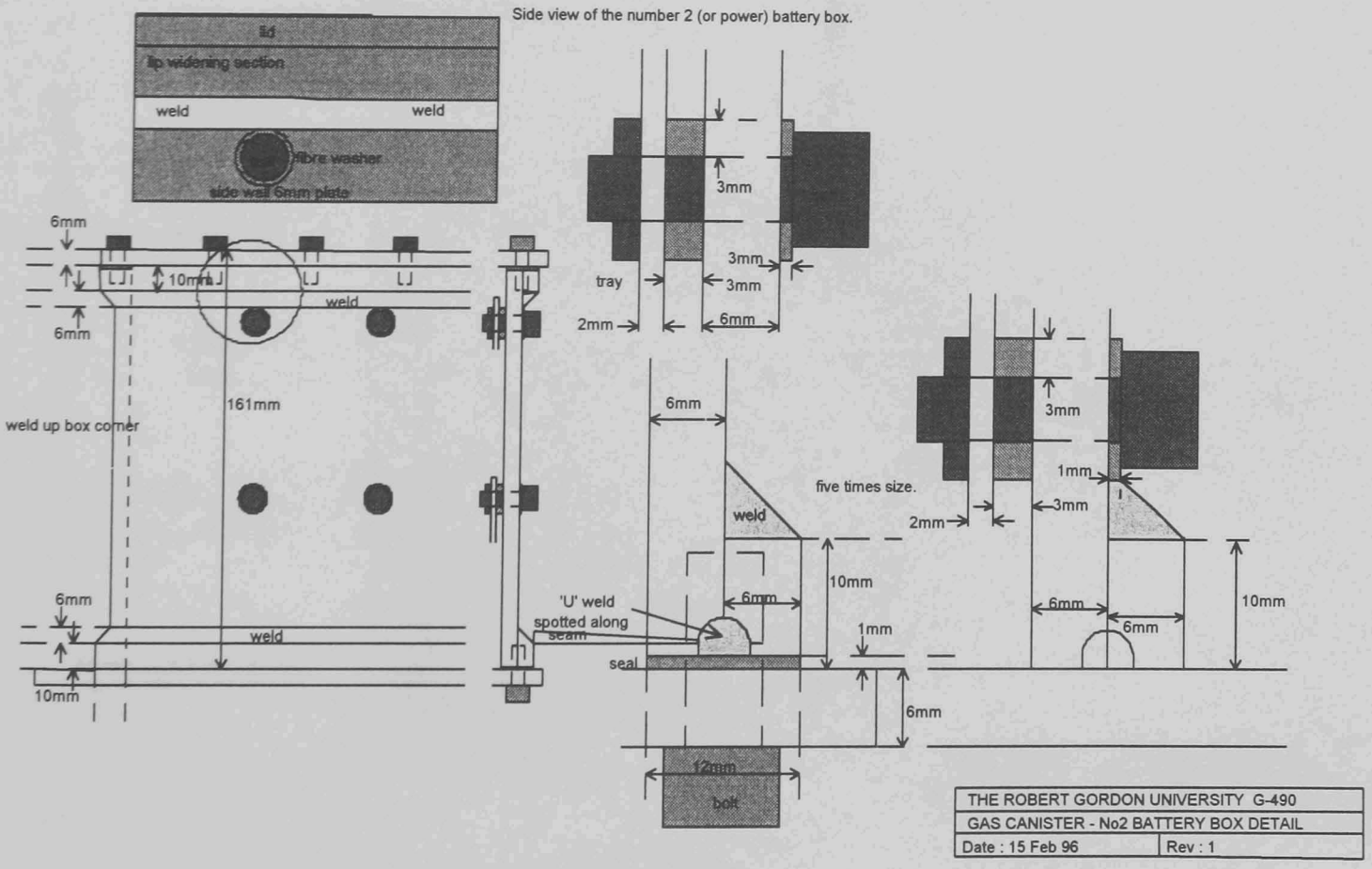


Figure A3.11

Battery box side elevation details, wall cross section and details of critical sections.

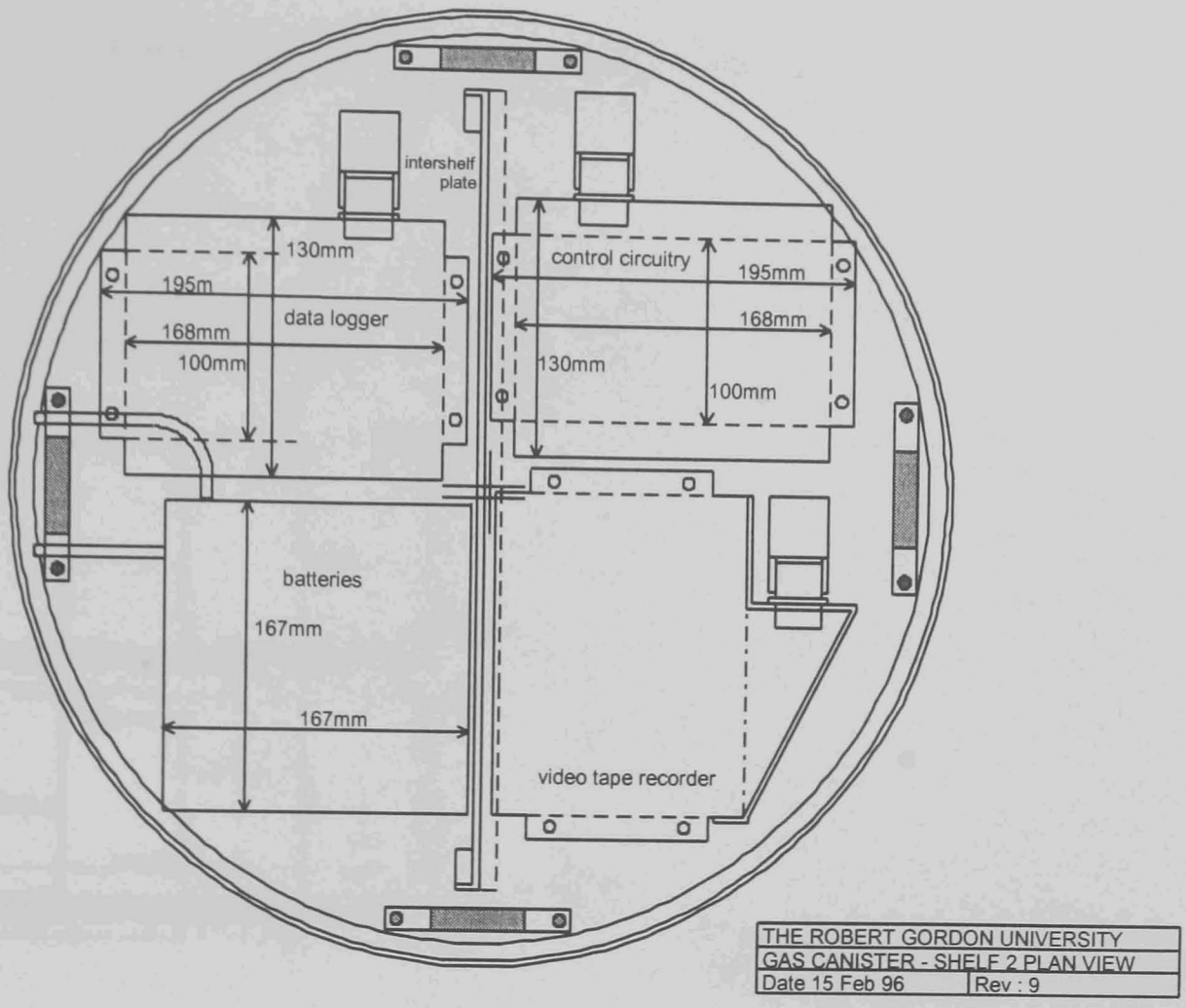


Figure A3.12

Detailed plan view of shelf 2, electronics batteries, control and data logging.

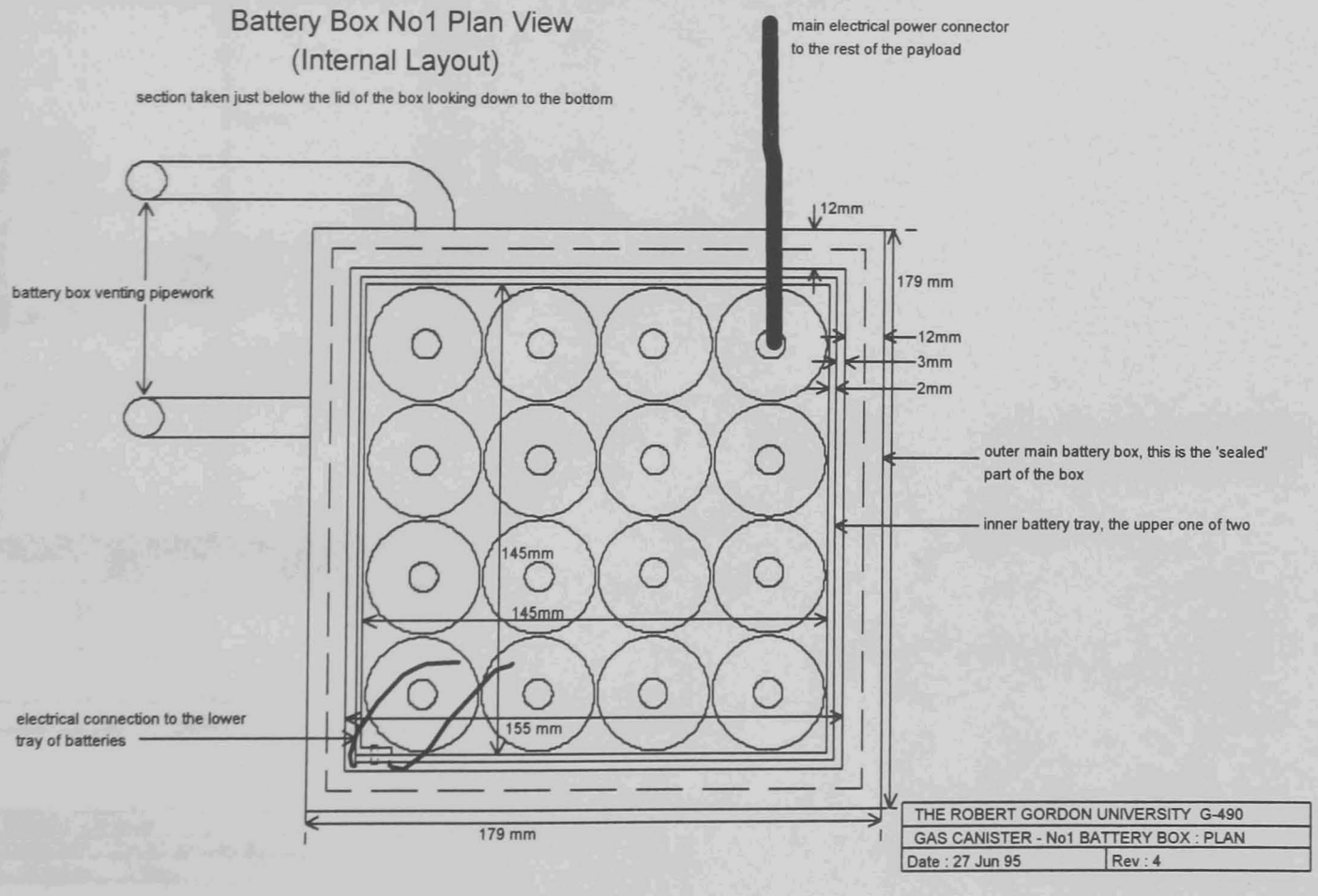


Figure A3.13

Electronics battery, mounted on shelf 2, box plan view.

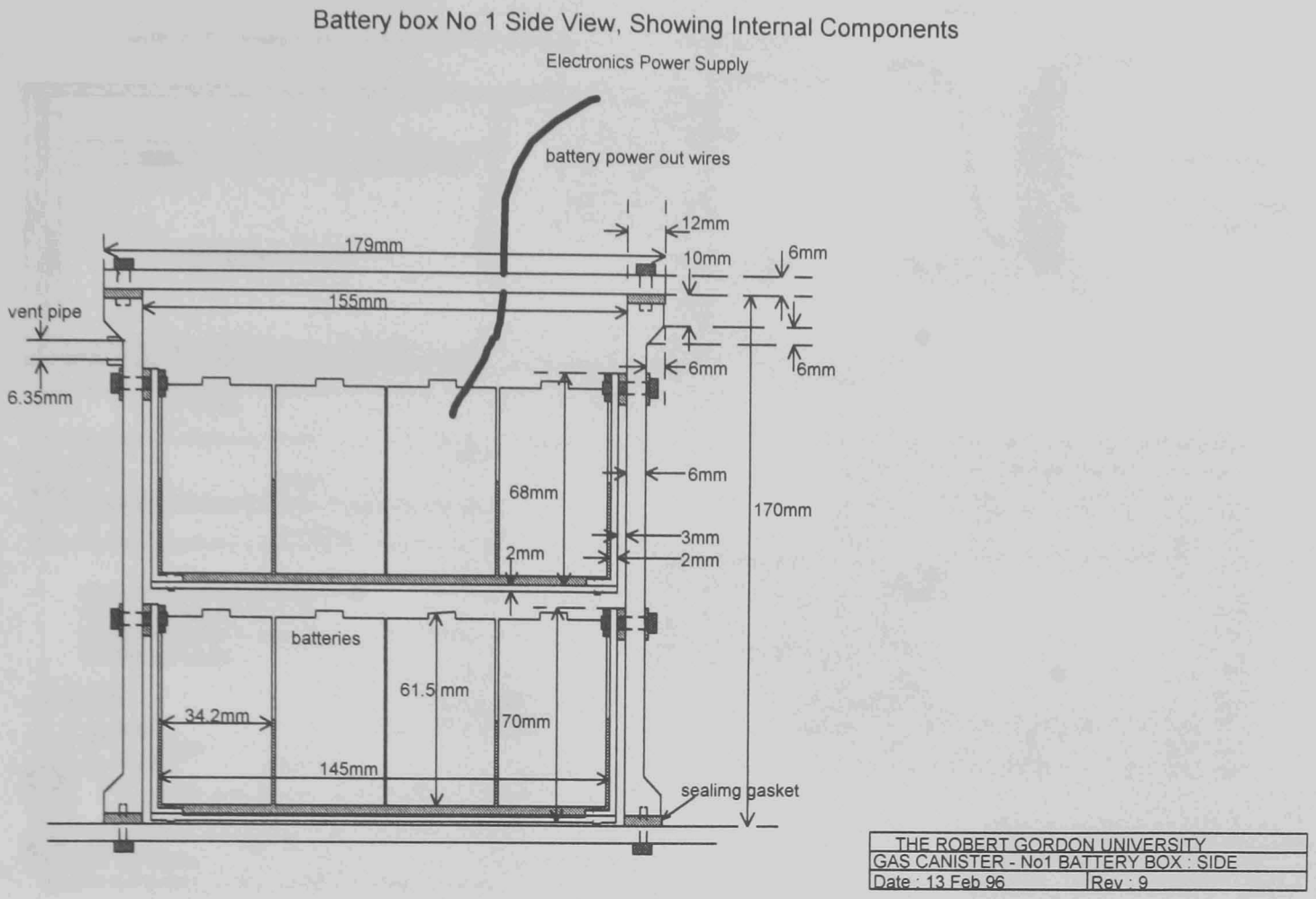


Figure A3.14

Battery box mounted on shelf, cross section.

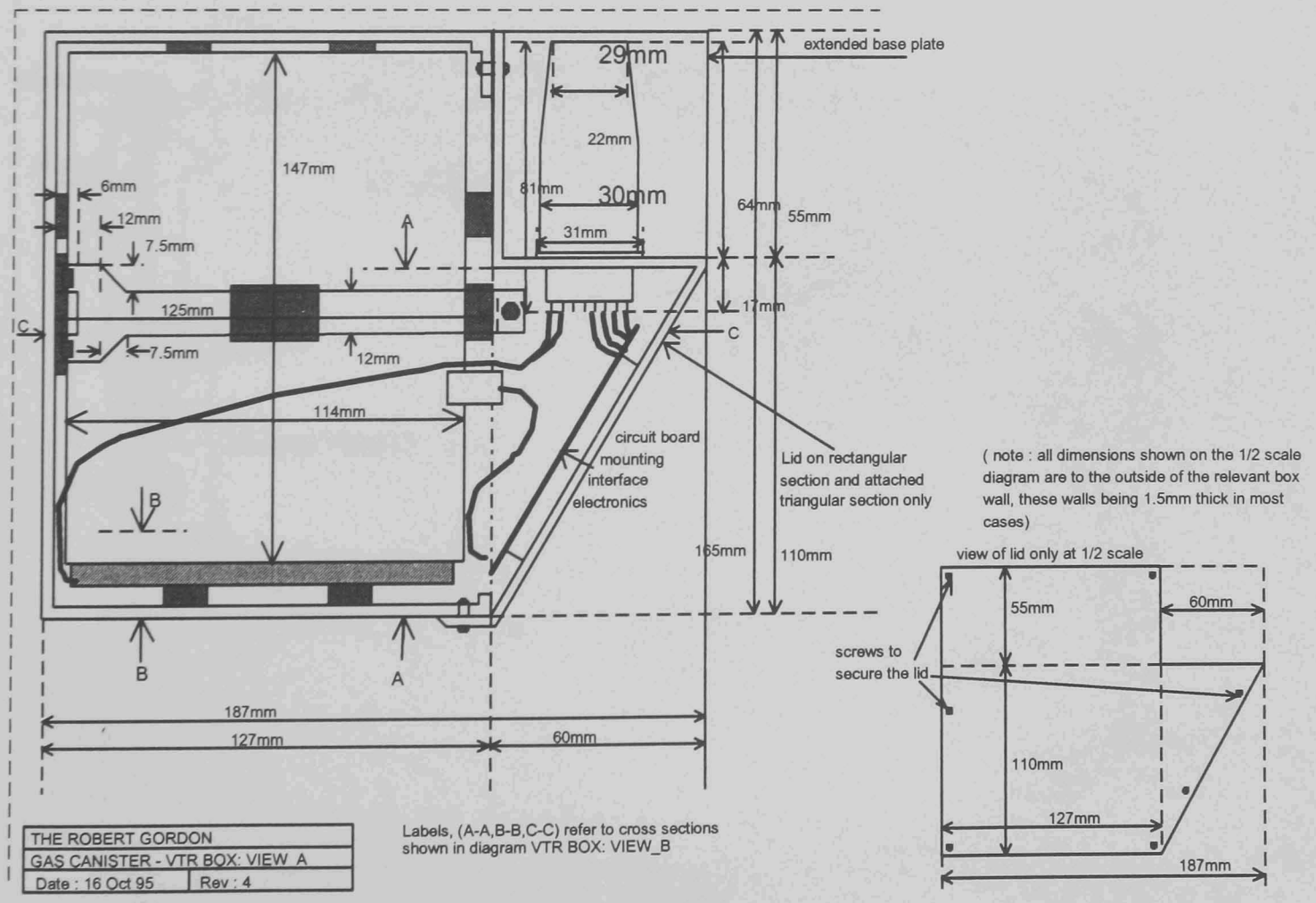


Figure A3.15

Mounting of the video tape recorder and its control circuitry.

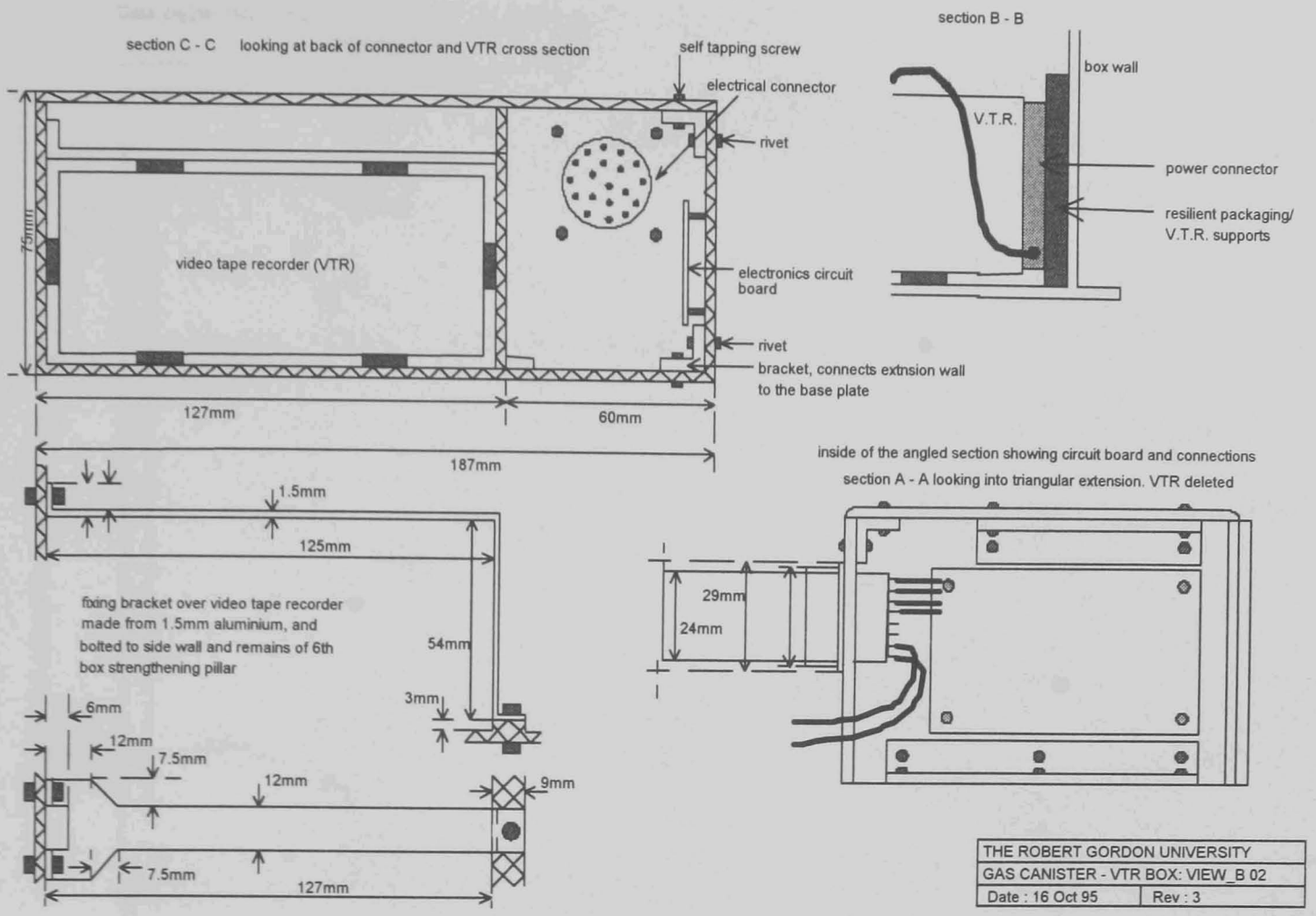


Figure A3.16

Cross sections and retaining bracket details.

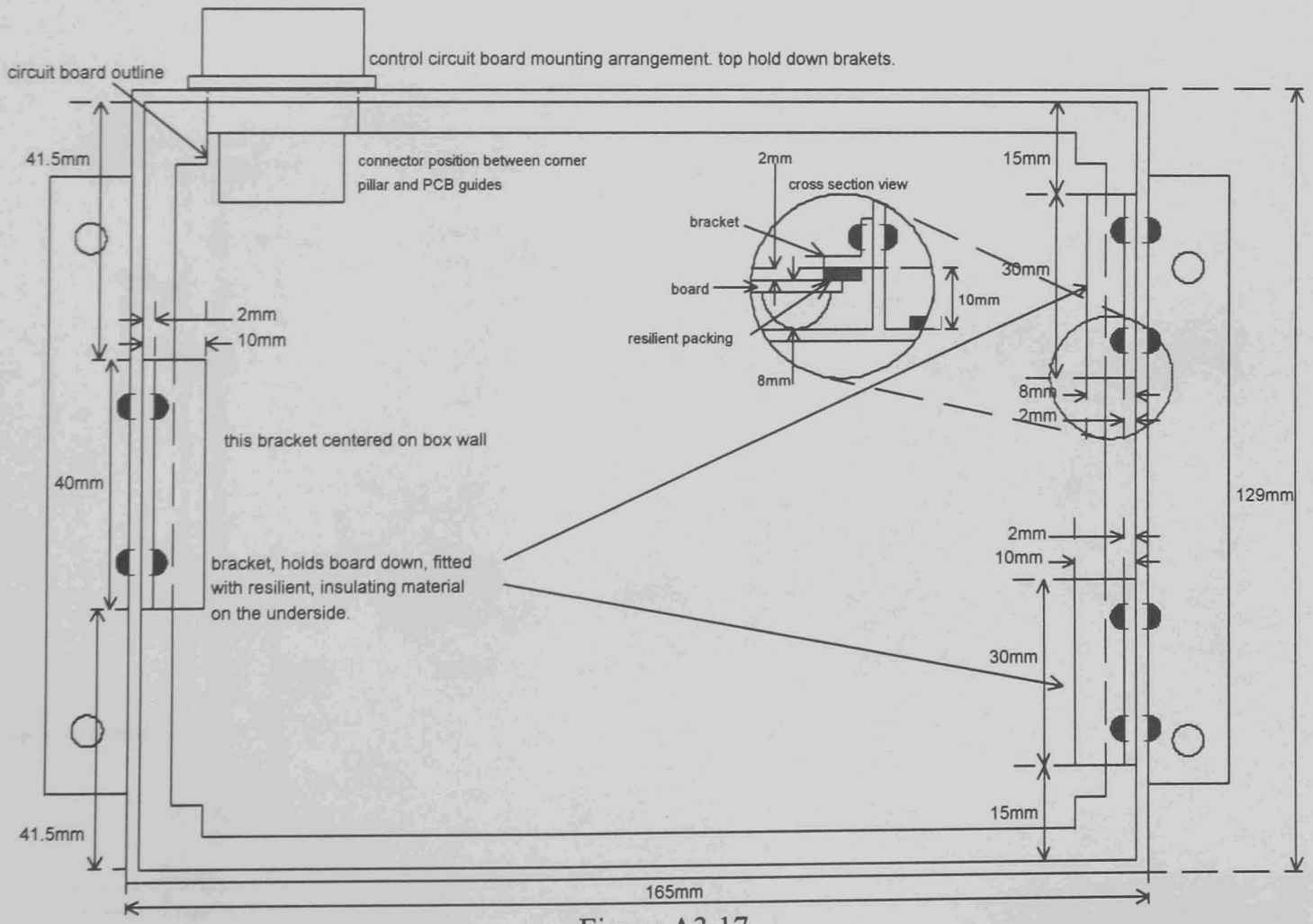


Figure A3.17

Control circuit housing and retaining brackets. These brackets plus a central support column provided edge and centre support thereby limiting board resonances which affected earlier designs.

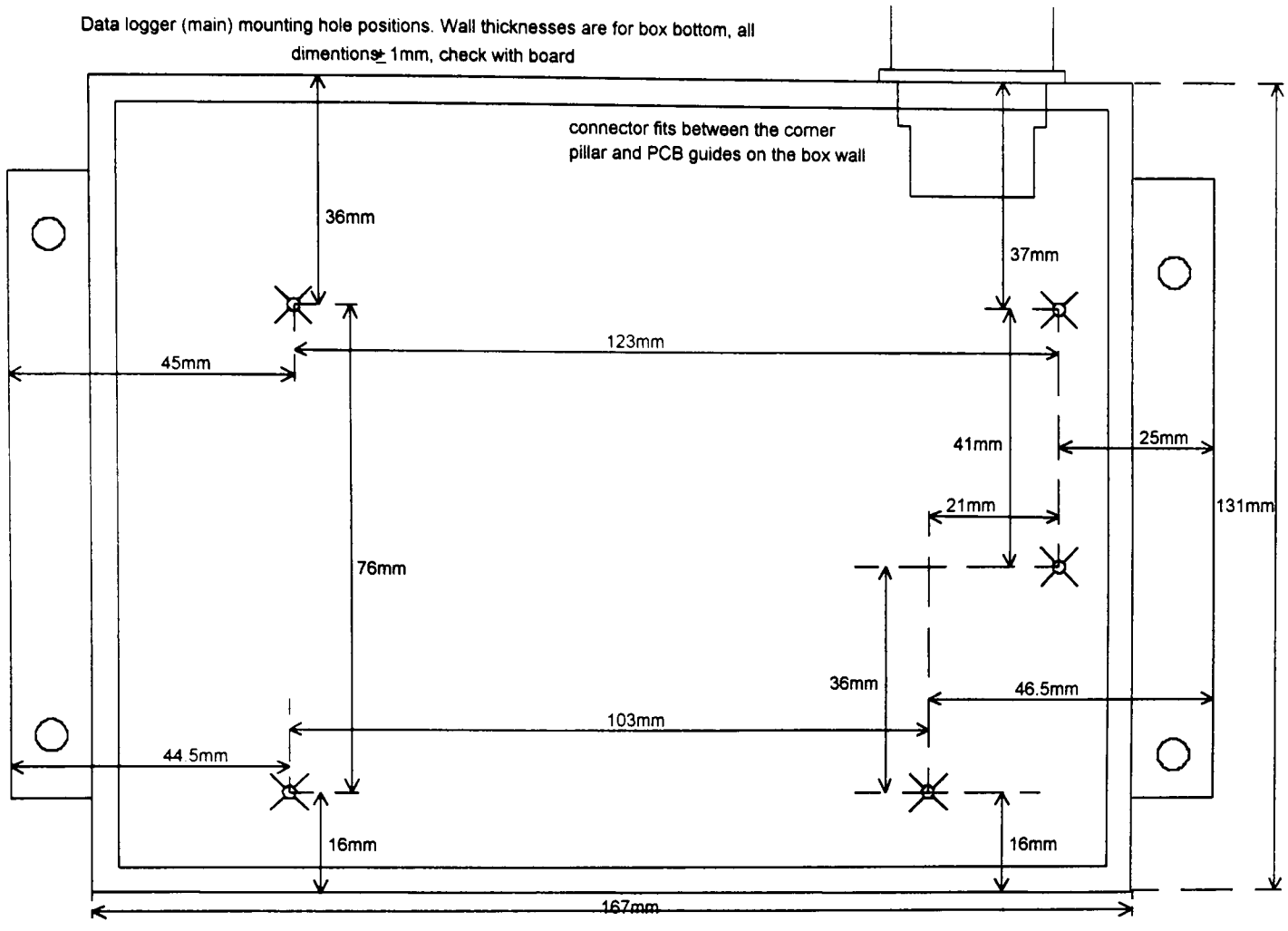


Figure A3.18

Data logger mounting, two circuit boards were stacked on pillars, sufficient as a large inter-board connector provided enough stiffening to avoid resonances.

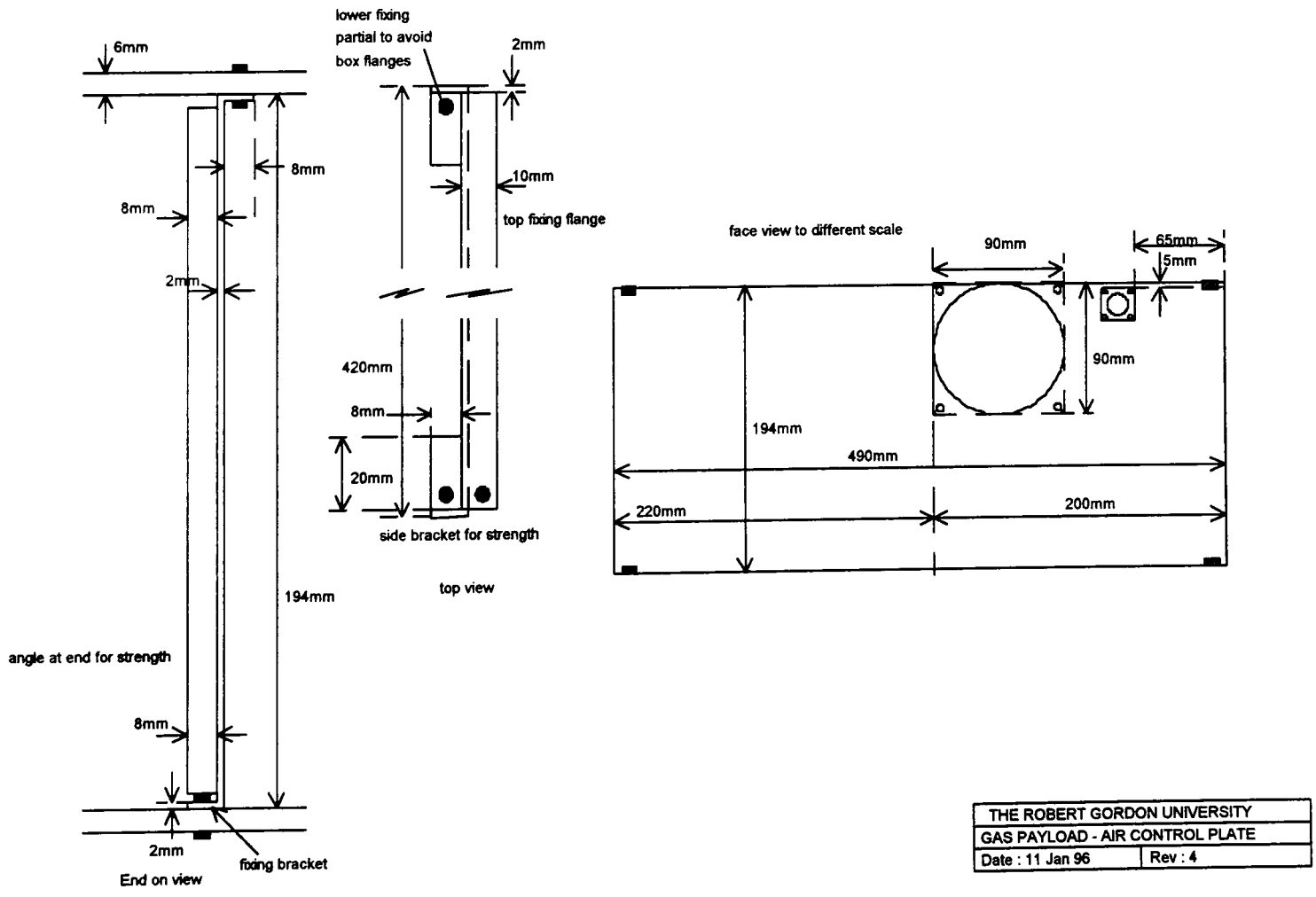


Figure A3.19

Stirrer fan on divider plate. This ensured air circulation in the experiment to avoid hot spots.

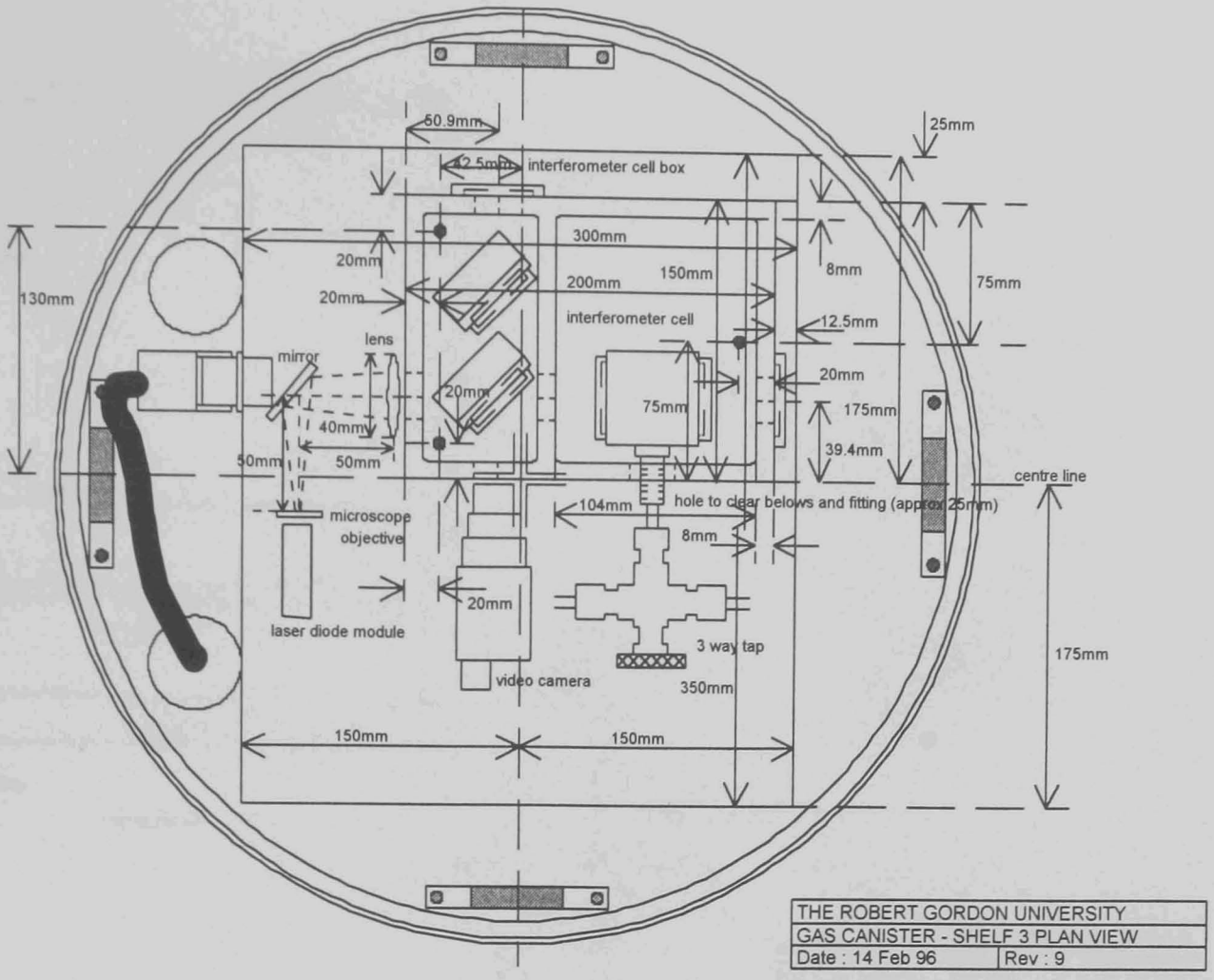


Figure A3.20

Shelf 3 showing the interferometer details with the light blocking protective cover removed.

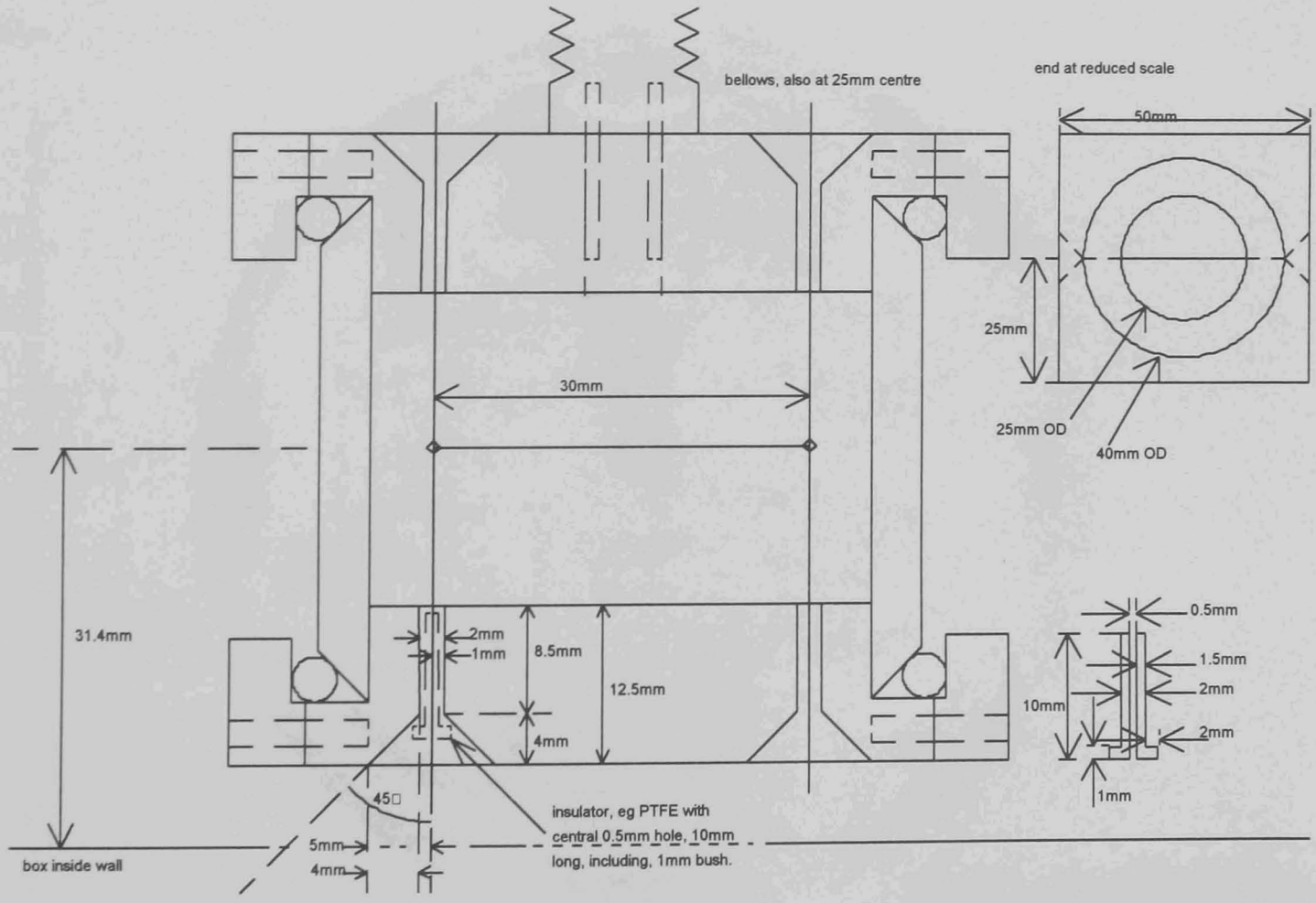


Figure A3.21

Detail of the experiment cell showing the heating element in its cylindrical cavity. The ends are plain glass windows sealed with 'O' rings, the insulators around the heater wires being epoxyed.

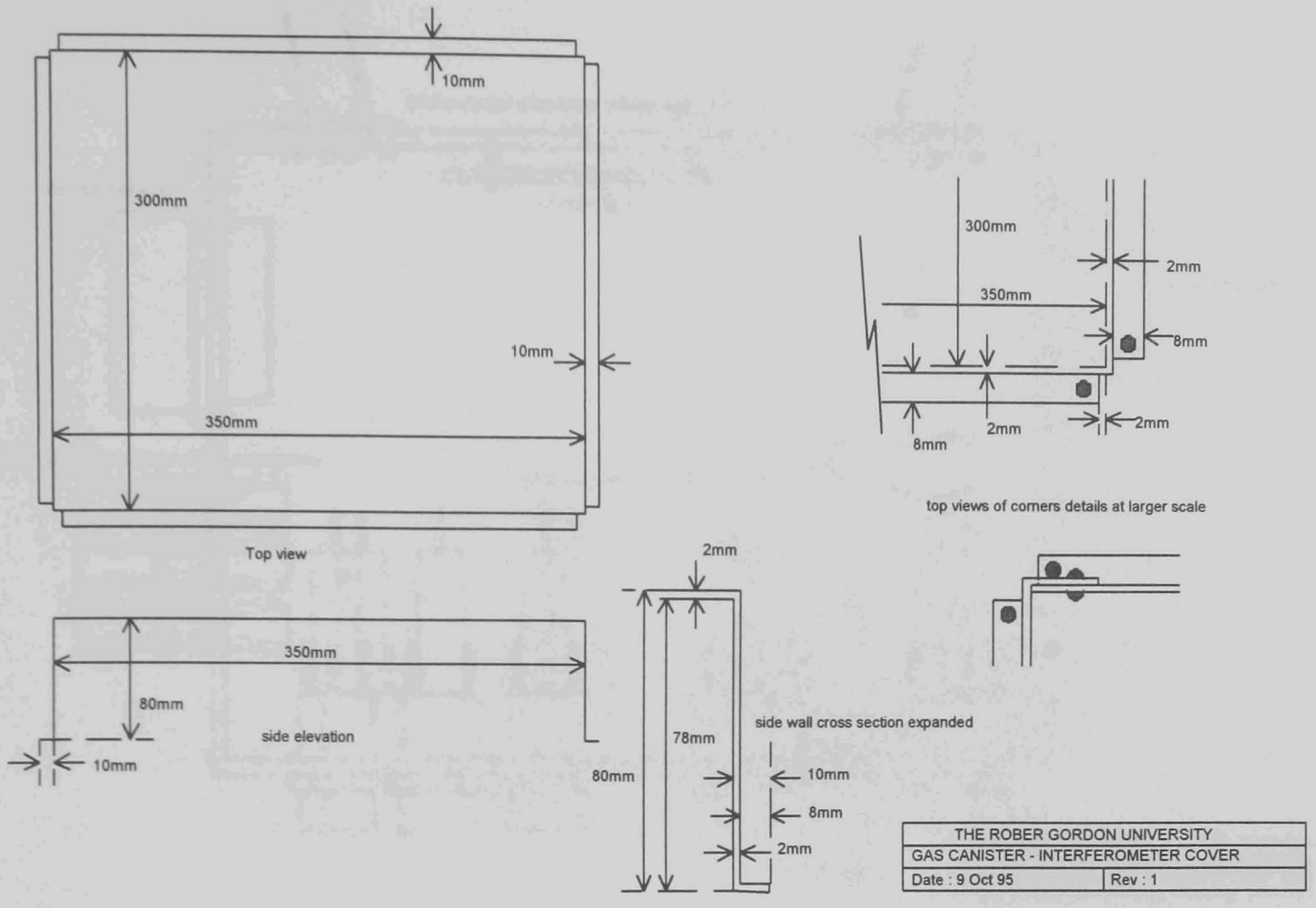


Figure A3.22

Interferometer cover, protects the interferometer and separates personnel from the laser.

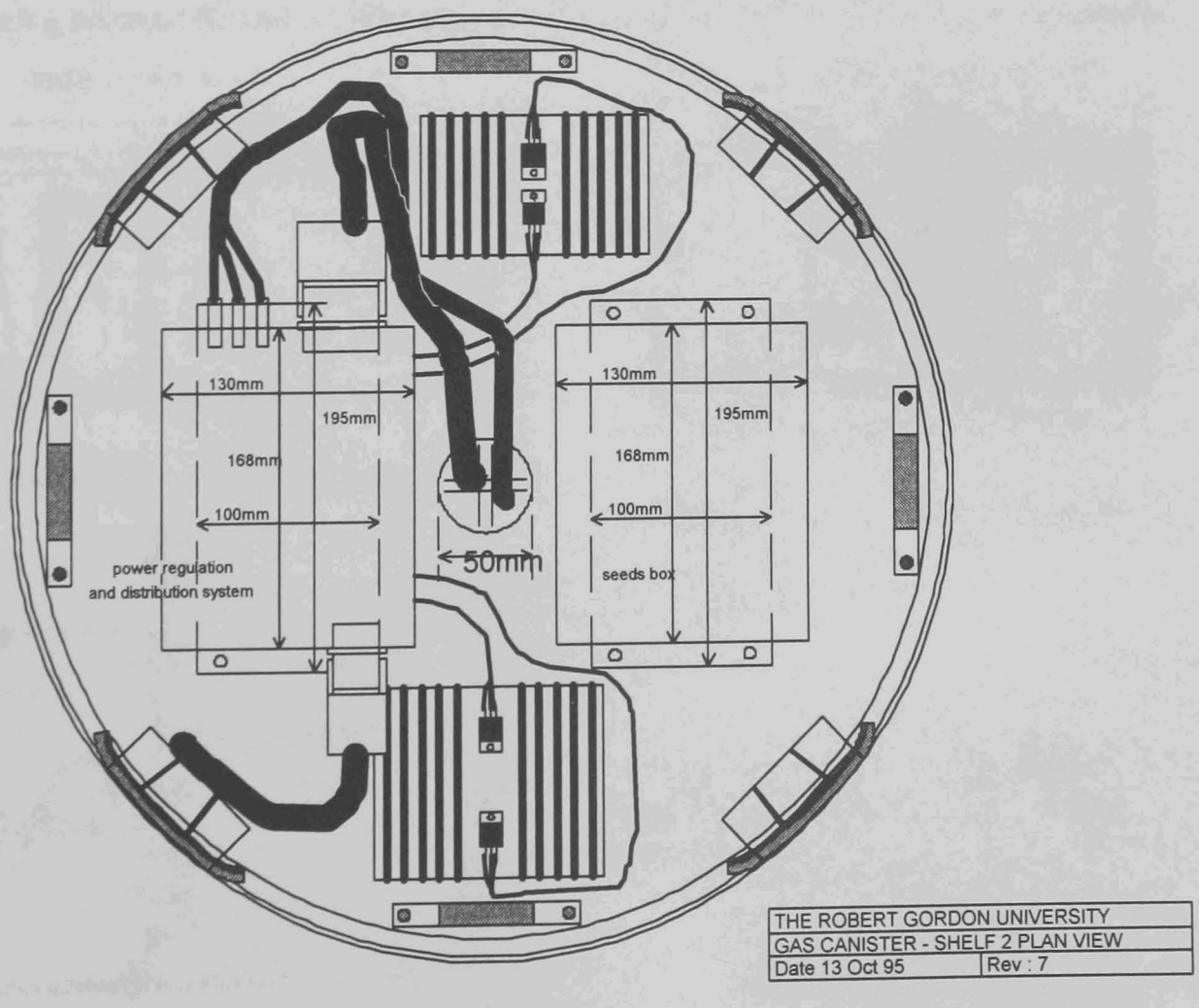


Figure A3.23

Shelf 4 contains the voltage regulators, lateral supports, main interconnect box and the subsidiary seeds experiment.

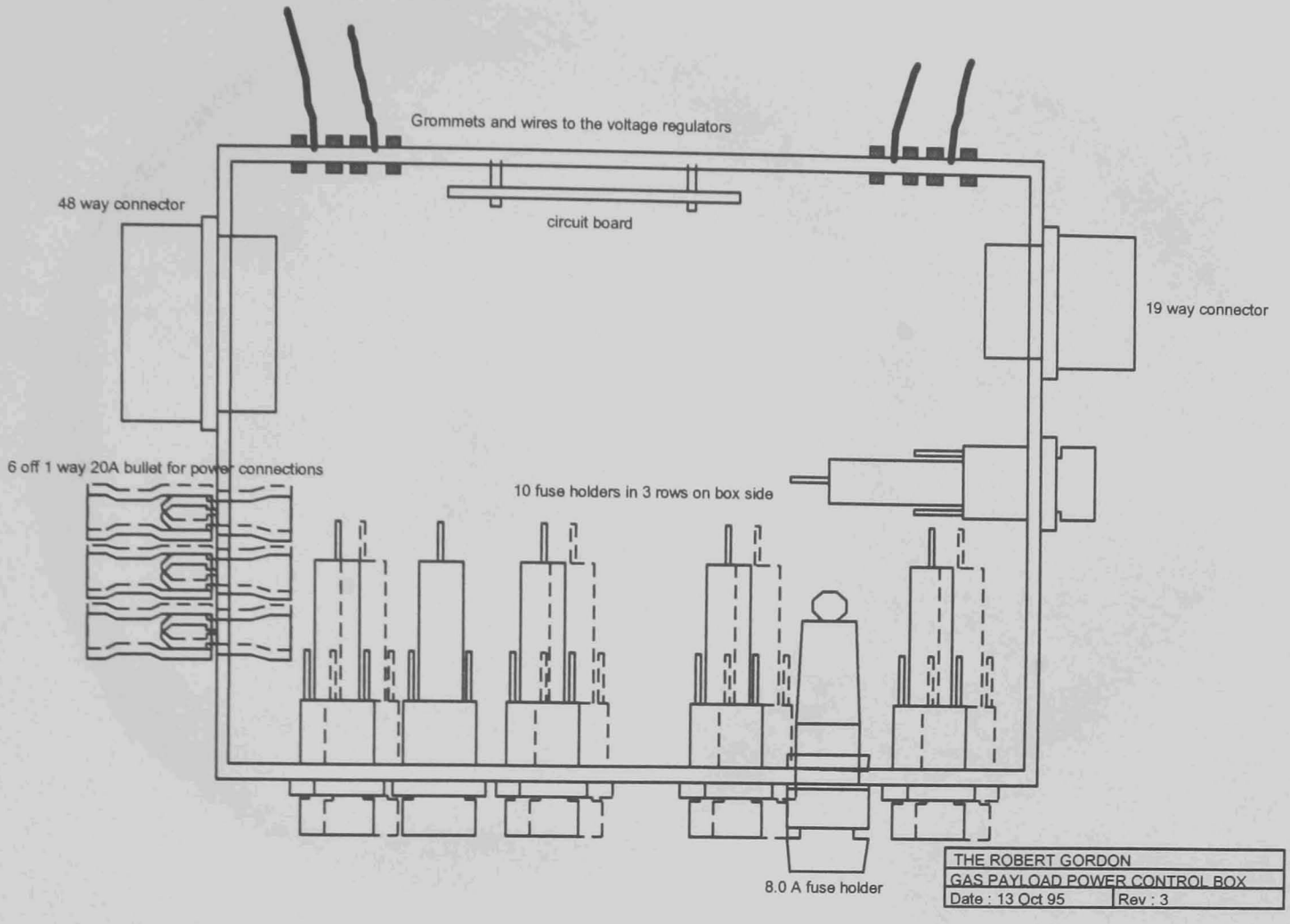


Figure A3.24

Interconnection box, contains the main fuses and forms the primary wiring interconnect point for interfacing between the main wiring looms, voltage regulators, fuses and NASA equipment.

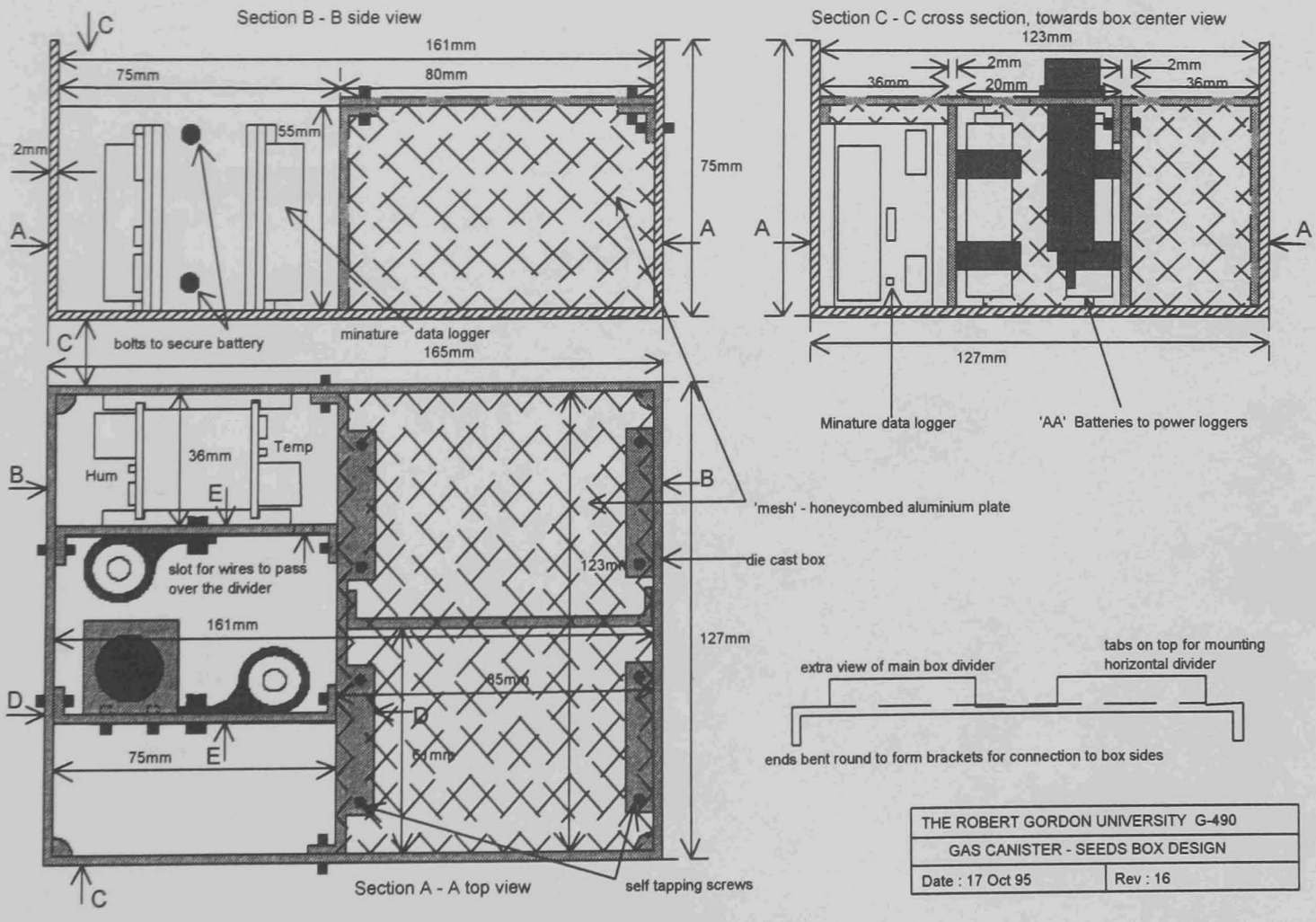


Figure A3.25

Seeds box, designed and built for the seeds experiment to ease its integration into the mechanical structure and safety paperwork.

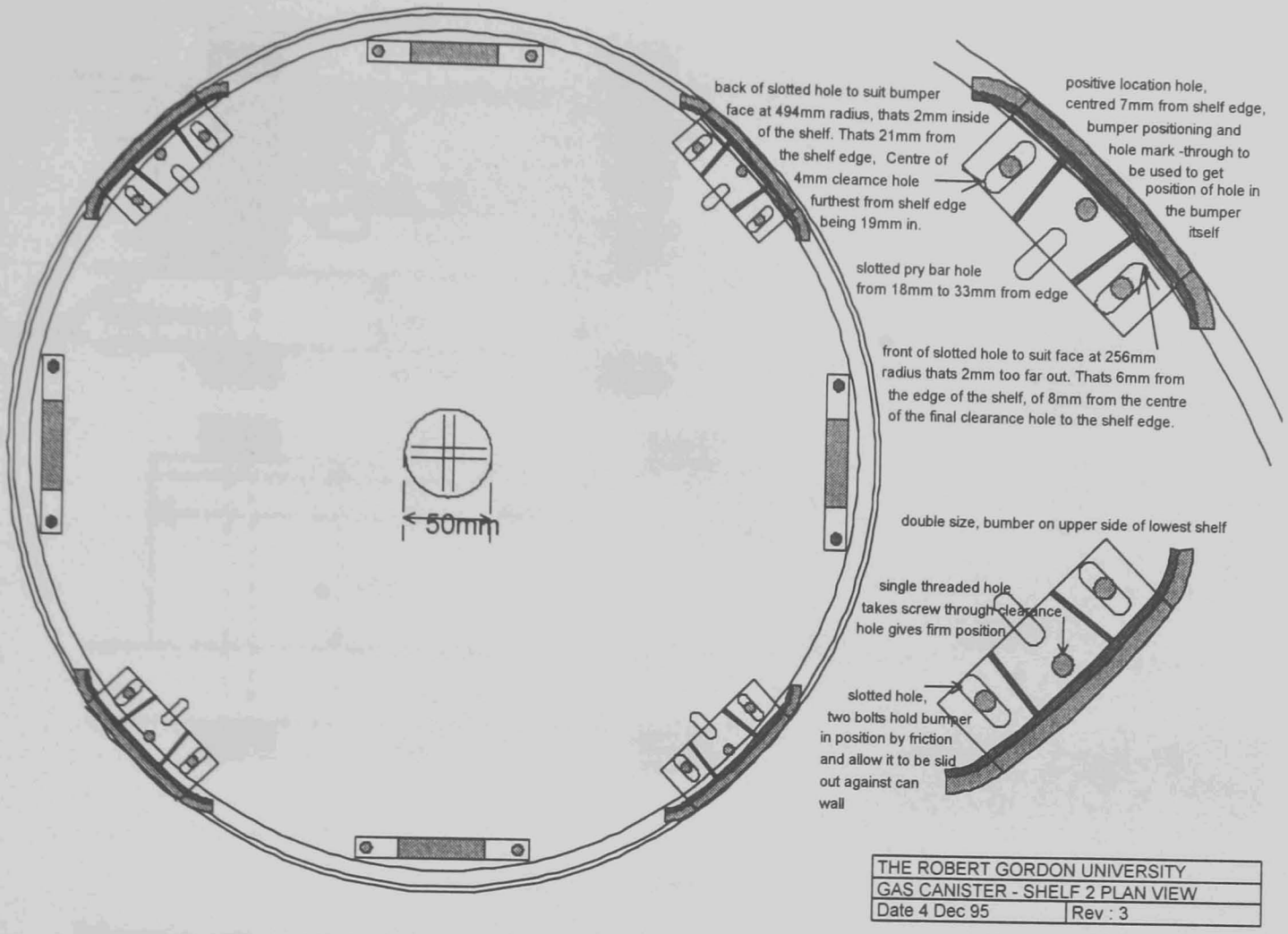


Figure A3.26

Lateral support bumpers and slotted holes which facilitate their adjustment against the sides of the canister. Access is available from below as the bottom of the canister may be removed.

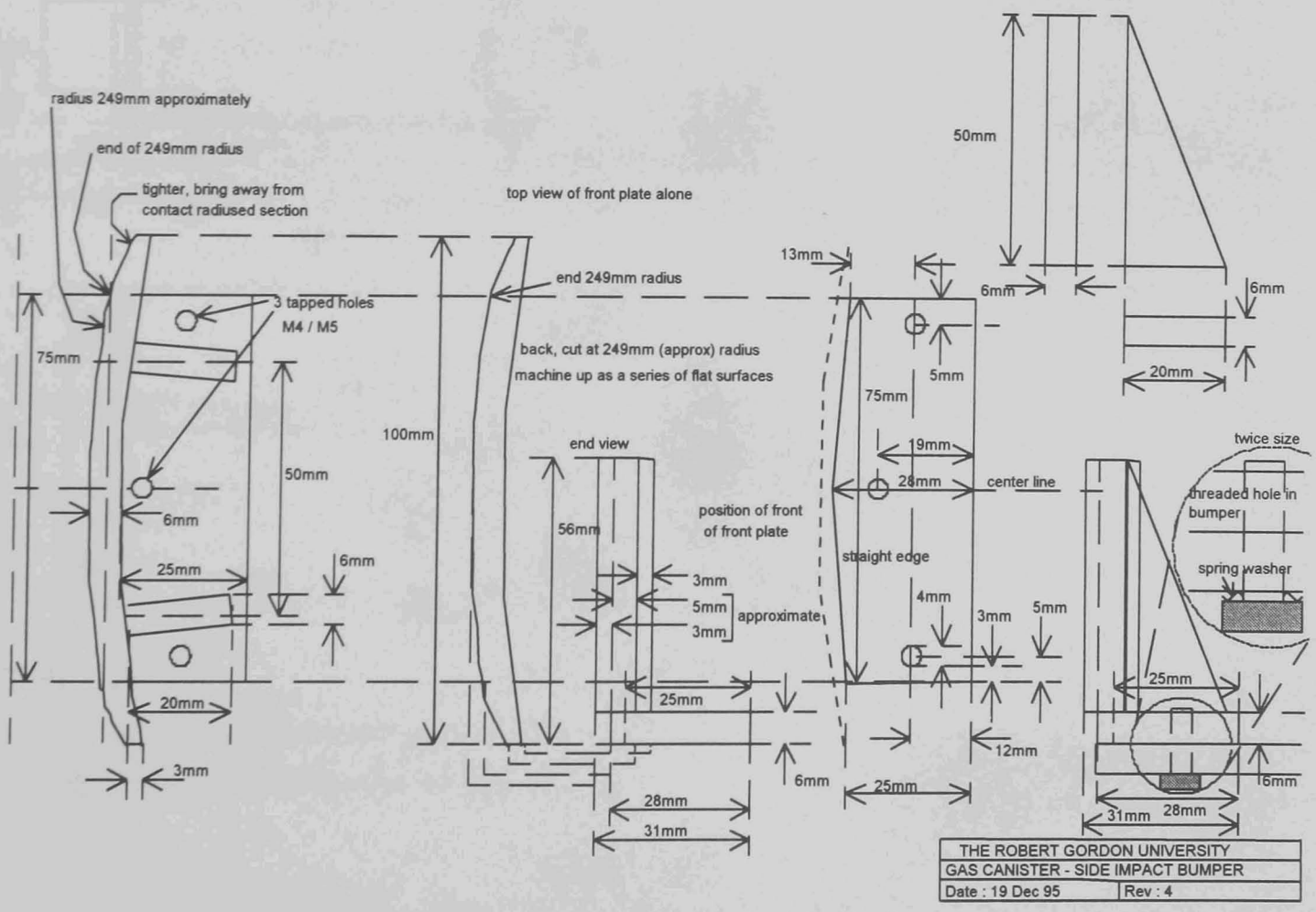


Figure C.27

Bumper construction details. The front face is covered with 5mm of resilient material to cushion it against the canister wall, protecting the canister and absorbing any slight mis-alignments.

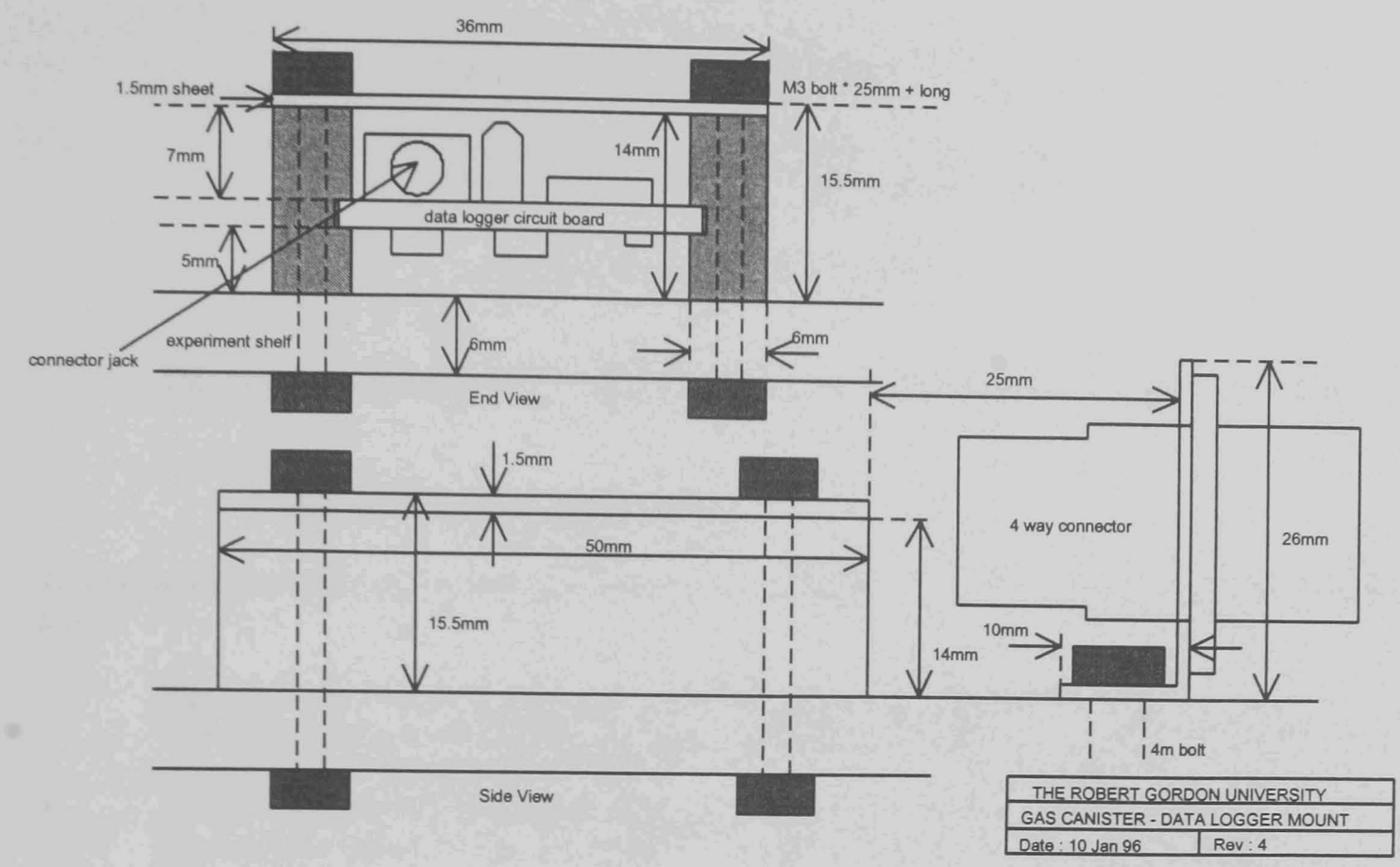


Figure A3.28

Miniature temperature data logger mounting. There are four of these, one on each shelf.

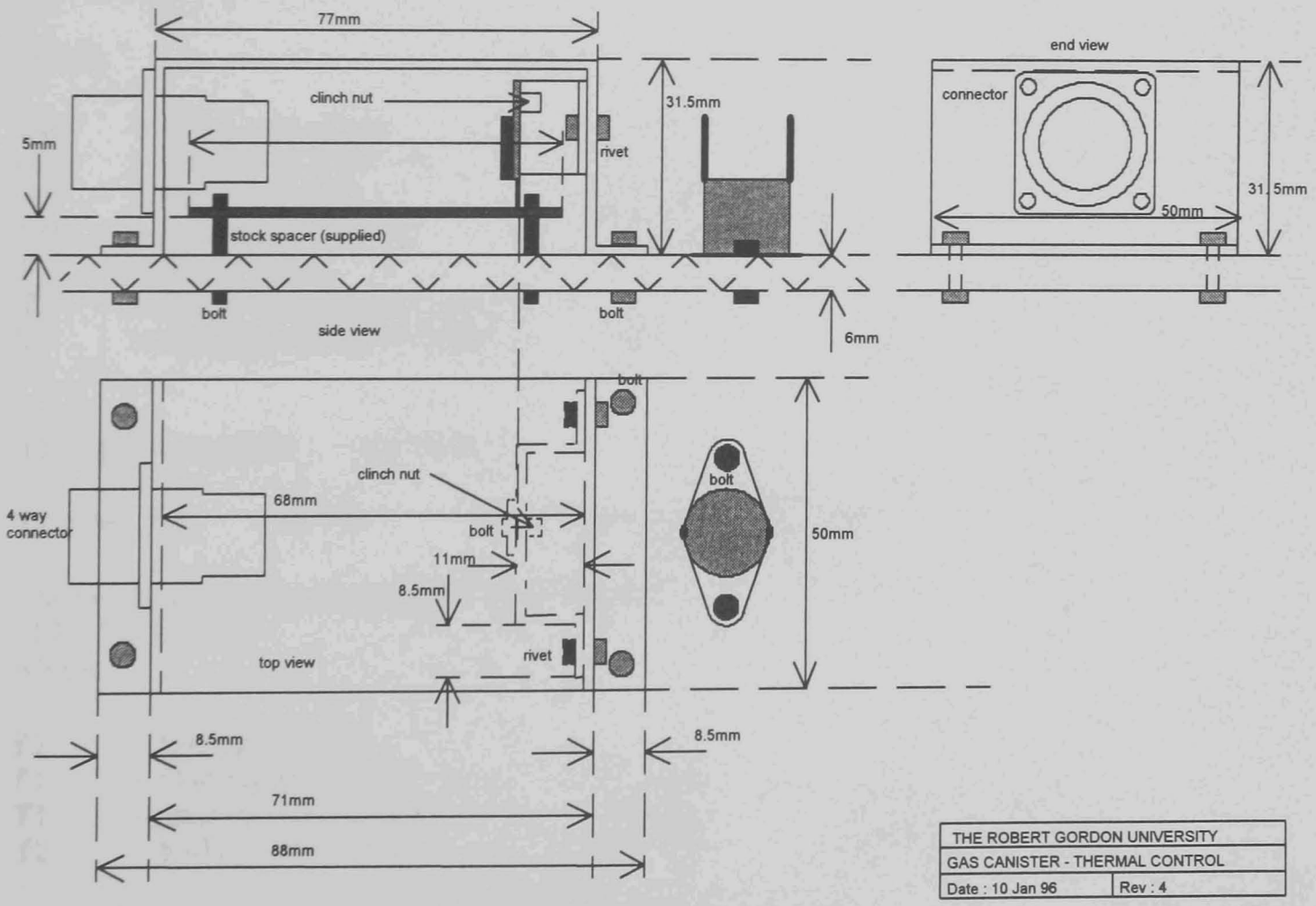


Figure A3.29

Thermal control system low voltage cut-out circuit mounting and backup bi-metallic thermostat. This also serves as the connector mounting, in several cases shared with the miniature dataloggers.

Appendix 4

ENERGY CONTAINMENT ANALYSIS

G - 490 ENERGY CONTAINMENT ANALYSIS
REPORT #ECA-490-31494 -- FEBUARY 13, 1996

G - 490 (5.0 ft GAS CANISTER)

This analysis is based on the NASA thermal analysis guidelines for GAS payloads, [326].

STEADY STATE ANALYSIS

To determine the worst case steady state canister temperature, the following three assumptions were made:

1. The cargo bay solar orientation corresponds to a canister temperature of 40°C.
2. The worst case power dissipation assumes all heat dissipating devices fail on. This also assumes that all inhibits fail to operate.
3. All generated heat is transferred to the surrounding dry nitrogen in an adiabatic process (no conductive or convective heat transfer to experiment structure or GAS canister occurs).

The following equation is used to determine the worst case temperature based on the above assumptions:

$$QR = o^{-}A2EffF_{2-1}(T2^{2}*T2^{2} - T1^{2}*T1^{2}) \qquad (1)$$

where:

QR	= Worst case power	= 59.55W	
o^{-}	= 5.669E-08 $W/m^{2}°K4$		
A2	= Radiating area	= $2.15m^{2}$	
Eff	= Effective Emittance	= 0.065	
F_{2-1}	= Shape Factor	= 1	
T2	= Worst case canister temperature		
T1	= Starting canister temperature	= 40°C	= 313°K

$$T2 = (QR / o^{-}A2EffF_{2-1} + 313^{2}*313^{2})^{1/4} \qquad (2)$$

T2 = 361.7°K = 88.7°C (maximum canister temperature)

The resultant increase in pressure is determined by:
P2 = P1(T2/T1)
where:

P2	= Ending pressure		
P1	= Starting pressure	= 15 psi	
T1	= Starting temperature	= 25°C	= 298°K
T2	= Ending temperature	= 88.7°C	= 362°K

P2 = 18.2 psi (maximum canister pressure)

CONCLUSION

The GAS canister will be able to contain the energy in the worst case power dissipation. The GAS canister will remain sealed throughout the flight since the GAS canister pressure relief valve set at 30 psi will not be breached. (This excludes any venting from the battery venting system due to battery failures due to causes other than excessive temperature as this is not into the main part of the canister)

TRANSIENT THERMAL ANALYSIS

To determine the worst case transient canister temperature, the following assumptions were made:

1. The cargo bay solar orientation corresponds to a canister temperature of 40°C.
2. The worst case power dissipation assumes all heat dissipating devices fail on. It is assumed that one of the inhibits for the environmental heating system will fail but that the backup will shut it down.
3. The temperature calculations are taken every 10 minutes.
4. Energy is dissipated through the nitrogen atmosphere, the experiment support structure and GAS canister.

The following equation is used to determine the transient temperature based on the above assumptions:

$$T2(t+ t) = T2(t) + [(Qin/CpM) - (o^{-}A2Eff/Cpm)(T2(t)^{2}*T2(t)^{2} - T1^{2}*T1^{2})]\, t \qquad (3)$$

T2(0) = 313°K
Qin = 57.25W
o^{-} = 5.669E-08 W/(m^{2}°K4)
A2 = dissipating area (2.15m^{2} for GAS canister)
Eff = effective emittance (0.065 for aluminium)
T1 = 313°K
t = 600s
CpM = (CpM)N2 + (CpM)payload + (CpM)can

The thermal model used for the transient analysis is shown in Figure A4.1.

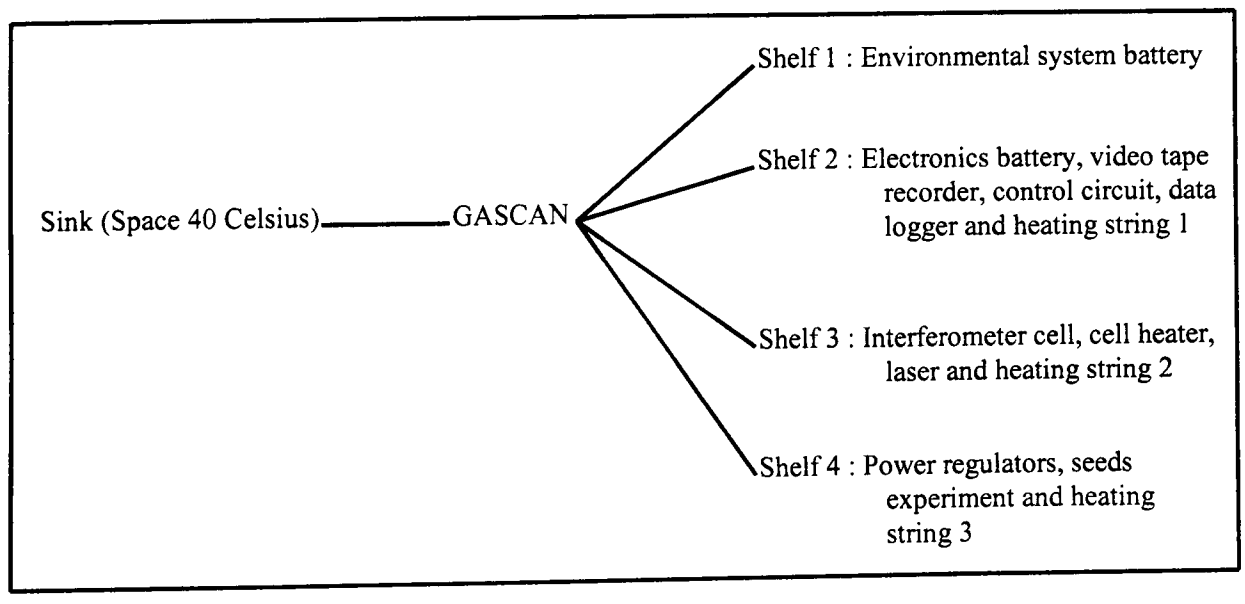


Figure A4.1 : Thermal model of the GAS canister for G-490

Due to the complex and highly iterative nature of these calculations, the actual calculations were carried out using the 'Matlab' mathematical software package (The 'Matlab 4' 'm' file is included as Appendix 4.1).

At worst case power dissipation, the environmental control battery will deplete after operating the environmental heating system for 97 hours. In practice, the battery will be charged for longer than this, as backup cutouts will disable the system before the unacceptably high temperatures shown by the steady state analysis are approached. (The depletion time is not Wh/W as the current drawn is voltage dependant, the values used were taken from manufacturers data for a comparable discharge time).

At worst case power dissipation, the electronics battery will deplete in 20 hours.

For both batteries it is assumed that any voltage difference between their actual output voltage and rated e.m.f. is lost internally as heat is dissipated into the batteries (the worst thing it could heat up). The values also assume fully charged cells, as the self-discharge of the cells used is negligible over the anticipated integration-to-launch time.

The resultant peak temperatures for a 133 hour simulation are listed below, (zero canister power being reached in under 100 hours due to flat batteries).

Shelf 1	: 79°C	: 352°K
Shelf 2	: 86°C	: 359°K
Shelf 3	: 87°C	: 360°K
Shelf 4	: 84°C	: 357°K
GASCAN	: 73°C	: 346°K

These temperatures are not considered liable to cause hazards in themselves though component failures may occur in some of the electronics due to thermal overstress.

The resultant increase in pressure is determined by :

$P2 = P1(T2/T1)$

P2 = Ending pressure
P1 = Starting pressure = 14.7 psi
T1 = Starting temperature = 25°C = 298°K
T2 = Ending temperature = 87°C = 360°K assuming the worst to the nitrogen

P2 = 17.7 psi. (maximum canister pressure)

CONCLUSION

The Gas canister will be able to contain the energy in the worst case power dissipation. The GAS canister will remain sealed throughout the flight since the GAS canister pressure relief valve set at 30 psi will not be breached. (note this excludes any venting from the battery venting system due to battery failures from causes other than excessive temperature as this is not into the main part of the canister).

PAYLOAD SUBSECTION CONSIDERATIONS

In addition to the main shelves, there is one further heating unit in the canister. This is the '1W' heating element in the interferometer cell. This is the only 'seperately' heated section of the payload as all the other 'heating' systems are thermally coupled to whichever of the payload shelves they are mounted on. This element is mounted in a small experimental cell containing silicone oil and is therefore of interest were it to heat this water to boiling point.

-- this section will have to be filled out once detailed interferomketer cell design has been done so that the thermal conductivity between the cell and the structure is known, this is an important parameter as this is the only way significant control can be placed on the rate of temperature rise in the cell during the experimental runs.

-- The cell is not hazardous from a safety veiwpoint even if it exploded as it is low mass and is physically contained in the interferometer cell, assuming 1W of power stood any chance of generating enough power to burst the container (rather unlikely).

APPENDIX 4.1

```
function [result] = ther20d1()
%
% function [result] = ther20d1()
%
% 08/02/96 20 degrees celcius version
%
% 09/01/96 final hot case verion
%
% 21/12/95 move shelf 1 heating string to shelf 4
%
% GAS CANISTER thermal analysis calculations package -
% for Matlab 4.
% this is being developed to carry out the thermal analysis
% of the transient thermal case. The calculations will be
% based on a six node model with the nodes being each of
% the four experimental package shelves, the GAS CAN's
% inner wall and the general environmental sink. the case of
% interest for this calculation is the 'hot' case for sun facing
% in which the background sink temperature is 40 degrees
% (313 kelvin), as given in the notes on carring out a
% thermal containment analysis.
%
% UPDATE 22/11/94 TELECONFERENCE
%
% all back up, be they thermostats, thermal fuses or
% whatever are assumed to fail. As this file is a general
% calculation system it inculdes code to allow you to
% simulate thermal fuses and thermostats. These are
% included for EXPERIMENTAL SUCCESS PURPOSES
% ONLY and are therefore not used for this analysis. For
% simplicity they are disabled by simply setting their
% operating temperatures to over 500 degrees celcuis for
% worst case energy containment calculations. Note that
% ALL temperatures in this file are in Kelvin with zero
% celsuis taken as +273 degrees kelvin.
%

% general constants and variables

tor       = 0.00000005669;
          % Stefan-Boltvmann constant
          W/m(sqrd)kelvin(tothe4)
powtot    = 202.5;
          % in watt`s, value used is actually calculated on
          % a step by step basis this value is for
          % information only.
mcpcan    = 0.21 * 60.641;
          % specific heat of GAS can (Al) times its mass
          % (Kg) mass includes mounting plate which is in
          % direct contact with the GAS canister
mcpshf1   = 0.21 * 14.110 + 0.317 * 27.072 + 0.40 * 4.115;
          % aluminium + batteries + battery holding filler
mcpshf2   = 0.21 * 6.495  + 0.20 * 0.70;
          % aluminium + electronics
mcpshf2   = mcpshf2 + 0.21 * 2.895 + 0.317 * 4.512 + 0.40
* 0.577;
          % aluminium + batteries + filler
mcpshf3   = 0.21 * 6.671 + 0.20 * 0.20 + 1.0 * 0.025;
          % aluminium + electronics + silicone oil (in cell)
mcpshf4   = 0.21 * 4.805 + 0.20 * 0.10;
          % aluminium + electronics
timestep  = 1 / 6;    % calculation timestep in hours
Eeffal1   = 0.1;      % effective emittance of structural
                      % aluminium shelf1
Eeffal2   = 0.1;      % effective emittance of structural
                      % aluminium shelf2
Eeffal3   = 0.1;      % effective emittance of structural
                      % aluminium shelf3
Eeffal4   = 0.4;      % effective emittance of structural
                      % aluminium shelf4
Eeffal    = 0.1;      % effective emittance of structural
                      % aluminium used
sink      = 313;      % effective sink temperature in kelvin
hotcase   = 1;        % 0 = cold , 1 = hot, controls
                      % elecronics output
regtemp   = 1293;     % setting on the thermostats, i.e. the
                      % temperature which the
                      % environmental control system will
                      % try to maintain
fusetemp  = 1343;     % thermal fuse temperature, note
                      % above regtemp set huge to equate to
                      % all regulation failed.
timevolt  = 1;        % effective time elapsed for the
                      % environmental control battery. As
          % the current load on this battery is not constant
          % it`s elapsed time that is of interest is the
          %  ammount of  amps*hours
          % that have been drawn from it and therefore it`s
          % present state of charge as this determines the
          % voltage supplied to any active heaters and
          % therefore their energy dissipation
timevele  = 1;        % elapsed power usage for electronics
                      % battery
thrfuse1  = 1;        % thermal fuse 70 degrees celsius,
                      % this is placed
thrfuse2  = 1;        % in series with the environmental
                      % control battery
thrfuse3  = 1;        % once blown that`s it  power is zero.

% results matrices
% contain the temperature plots for the entire flight for each
% node with a variable temperature

result          = zeros(9,500);
shelf1temp      = zeros(500,1);
shelf2temp      = zeros(500,1);
battyetemp      = zeros(500,1);
shelf3temp      = zeros(500,1);
shelf4temp      = zeros(500,1);
insidecan  = zeros(500,1);
shelf1temp(1) = 298;
% these put the starting temperature of 25 degrees
shelf2temp(1) = 298;
% in each of the matrices as the `temperature at
shelf3temp(1) = 298;
% last time step` values are taken from these
shelf4temp(1) = 298;
% matrices and they are initialised as all zeros
battyetemp(1)   = 298;
insidecan(1)    = 298;
result(7,1)     = 24;
% initial battery voltages, have to be put in
result(8,1)  = 24;
% manually as time loops from 2 due to how the
```

```
% matrix indexing works.

% look up charts for the battery voltage levels. As the battery voltage does
% not fall linearly with time as the cells discharge the voltage histories
% have to be encoded manually. The battery discharge curves that result from
% these arrays have been worked out by using voltage against time curves for
% the batteries for similar cell loadings given in the cell manufacturers data.

envbt =
[24,20.5,19.2,18.9,18.4,17.8,17,16,14.4,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0];
%elebt =
[24,20.8,20,19.2,18.9,18.7,18.4,17.8,17.4,17,16.2,14.4,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0];
elebt =
[24,20.6,19.5,18.9,18.6,18.0,17.4,16.8,15.0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0];
envbat = zeros(1500,1);   % environment battery
elebat = zeros(500,1);    % electronics battery

% loops to extrapolate intermediate values, saves work typing in lots of
% values that vary linerarly form one to the next

for a = 1:30
 for b = 1:45
  envbat((a - 1)* 45 + b) = envbt(a + 1) + ((envbt(a) - envbt(a+1)) * ((45 - b) / 45));
 end
end

for a = 1:30
 for b = 1:15
  elebat((a - 1)* 15 + b) = elebt(a + 1) + ((elebt(a) - elebt(a+1)) * ((15 - b) / 15));
 end
end

% current constants (amps)
diode     = 0.033;
camera    = 0.138;
cellheat  = 0.083;
logger    = 0.030;
fan       = 0.106;
controlcir = 0.005;
recorder  = 0.530;
% resistive heating unit resistance (ohms)
heatres1  = 30;
heatres2  = 90;
heatres3  = 60;
heatres4  = 30;
heatres   = 1 / ((1 / heatres2) + (1 / heatres3) + (1 / heatres4));

%
% *** main loop ***
%
for time = 2:(480 * 30/heatres);  % where prestime is in increments of ten minutes

%
% current calculations
% the environmental control battery current is that drawn by the fan plus
% the current draw of the heating elements that are presently active.
% envbat = heaters on those shelves that are below 313 kelvin + fan
% elebat = camera + logger + laser + cell heat + tape recorder + controlcir
%

timevoltinc = 0;
result(6,time) = envbat(timevolt);
result(7,time) = elebat(timevele);


bat1cur = (fan / 24) * envbat(timevolt);
if shelf2temp(time - 1) < regtemp % 313 kelvin = 40 celcius desired temperature
 bat1cur = bat1cur + (envbat(timevolt) / heatres2);
end

if shelf3temp(time - 1) < regtemp % 313 kelvin = 40 celcius desired temperature
 bat1cur = bat1cur + (envbat(timevolt) / heatres3);
end

if shelf4temp(time - 1) < regtemp % 313 kelvin = 40 celcius desired temperature
 bat1cur = bat1cur + (envbat(timevolt) / heatres4);
end

%
% battery useage curve. environmental battery has a power capacity of
% 216,000 mAH and depletes by timevolt = 360 , hence 1 time step is
% worth 600 mAH. power used is current times time, here current is Amps
% and one time step is timestep hours.
% therefore bat1cur * timestep = AH used
%         AH used / 0.6     = timesteps woth of battery used
% finally account for the fact that not all the nominal battery life is
% delivered at grater than nominal output.
% for 30ohms * 3 at 20V = 2A + 0.1 (fan) 12 ways = 0.175A
% at 0.175A battery discharge in about 74 hours delivering some 12.9 AH
% as nominally 18AH correction factor is
% nominal useage * ( 18 / 12.9 )  (factor 18 / 12.9 = 1.395)
%

timevolt = timevolt + ((((bat1cur * timestep) / 0.6) / 0.53)* heatres / 30);  %* 1.395);

% now supply No : 2

bat2cur = 0;
if elebat(timevele) > 8
```

```
 bat2cur = bat2cur + logger + controlcir;
 if hotcase == 1
  bat2cur = bat2cur + recorder;
 end
 if (elebat(timevele) > 14) & (hotcase == 1)
  bat2cur = bat2cur + diode + camera + cellheat;
 end
end                  % battery No2 current falls as the voltage
gets
                     % very low as the linear regulators start to
shut
                     % down due to insufficient voltage to allow
them
                     % to work correctly

%
% battery useage curve. electronics battery has a power
capacity of
% 36,000 mAH and depletes by timevolt = 360 , hence 1
time step is
% worth 100 mAH. power used is current times time, here
current is Amps
% and one time step is timestep hours.
% therefore bat2cur * timestep = AH used
%         AH used / 0.1      = timesteps woth of battery used
%

timevele = timevele + ((bat2cur * timestep) / 0.1);

%
% shelf 1
% environmental control system battery, power dissipation
is battery losses
% in the worst case thes are its nominal voltage (24V) minus
the actual
% output voltage at the batteries present state of charge
multiplied by the
% current being drawn.
% environmental control system heating string no1. power is
volts times amps
% where the current is determined by ohm's law as the
element is resistive
% amps = volts/heatres
%

shf1pow = (24 - envbat(timevolt)) * bat1cur;
% heating string moved
% if shelf1temp(time - 1) < regtemp % 313 kelvin = 40
celcius desired temperature
%   shf1pow = shf1pow + (envbat(timevolt) *
envbat(timevolt) / heatres1);
% end

%
% shelf 2 and electronics battery
% main electronics shelf, dissipation due to electronics
battery losses,
% heating string 2, the data logger, control circuit fan and
vigeo tape
% recorder
%

bat2pow = (24 - elebat(timevele)) * bat2cur;

% battery on shelf

shf2pow = bat2pow;
bat2pow = 0;

if ((shelf2temp(time - 1) < regtemp) & (thrfuse2 == 1))
                     % 313 kelvin = 40 celcius desired
temperature
  shf2pow = shf2pow + (envbat(timevolt) * envbat(timevolt)
/ heatres2);
end
if (bat2cur > 0)
  shf2pow = shf2pow + logger * elebat(timevele);
end
shf2pow = shf2pow + fan * envbat(timevolt);
if (elebat(timevele) > 8) & (bat2cur > 0)
  shf2pow = shf2pow + controlcir * 5;
  if hotcase == 1
    shf2pow = shf2pow + recorder * 6;
  end
end

%
% shelf 3
% microgravity experiment has the laser diode, fluid cell
heater, video
% camera and heating sting no 3
%

shf3pow = 0;
if ((shelf3temp(time - 1) < regtemp) & (thrfuse3 == 1))
                     % 313 kelvin = 40 celcius desired
temperature
  shf3pow = shf3pow + (envbat(timevolt) * envbat(timevolt)
/ heatres3);
end
if (elebat(timevele) > 14) & (bat2cur > 0) & (hotcase == 1)
  shf3pow = shf3pow + diode * 12 + cellheat * 12 + camera
* 12;
else
  if (elebat(timevele) > 2) & (bat2cur > 0) & (hotcase == 1)
    shf3pow = shf3pow + diode * (elebat(timevele) - 2);
    shf3pow = shf3pow + cellheat * (elebat(timevele) - 2);
    shf3pow = shf3pow + camera * (elebat(timevele) - 2);
  end
end

%
% shelf 4
% seeds and NASA interface shelf, vlotage regulators are all
that dissipates
% significant power on this shelf.
%

shf4pow = 0;
if (elebat(timevele) > 2) & (bat2cur > 0)
  if elebat(timevele) > 8
    if (elebat(timevele) > 14) & (hotcase == 1)
      shf4pow = (elebat(timevele) - 12) * (diode + cellheat +
camera);
    elseif (hotcase == 1)
      shf4pow = 2 * (diode + cellheat + camera);
    end
    shf4pow = shf4pow + (elebat(timevele) - 5) * controlcir;
```

```
  if hotcase == 1
   shf4pow = shf4pow + (elebat(timevele) - 6) * recorder;
  end
 else
  if hotcase == 1
   shf4pow = 2 * (diode + cellheat + camera);
   shf4pow = shf4pow + (2 * recorder);
  end
  shf4pow = shf4pow + (2 * controlcir);
 end
else
 shf4pow = 0;
end

if shelf4temp(time - 1) < regtemp % 313 kelvin = 40 celcius
desired temperature
 shf4pow = shf4pow + (envbat(timevolt) * envbat(timevolt)
/ heatres4);
end



%
% total power output, used for canister cooling
%

powtot = shf1pow + shf2pow + shf3pow + shf4pow;
result(8,time) = powtot;

%
% calculate new node temperatures
% for simplicity we will assume that the gascan only
dissipates heat once
% it has built up there so it will always be one time step
behind in the
% maths. This is not that inaccurate as long as the time steps
are small,
% and it should overestimate canister temperatures anyway
which is safer.
%
% equation
% Tnew(node) =
Told(node)+(Qin/MCp-(torA2Eeff/MCp)(T2p4(t)-T1p4(t))d
elta(t)
% new_temperature = old_temperature + change
% Qin  = payload power          delta(t) = change in time
(hours)
% M    = items mass             T2p4(t)  = this node temp at
last step
% Cp   = specific heat                    to the power 4
% tor  = stefan-boltzman constant  T1p4(t)  = sink node
temp at last step
% A2   = item radiating area              to the power 4
% Eeff = effective emittance
%

tcano = insidecan(time - 1);
brac1 = tor * 2.150 * 0.065 / mcpcan;
brac2 = (tcano*tcano*tcano*tcano) - (sink*sink*sink*sink);
insidecan(time) = tcano + (powtot/mcpcan - brac1 * brac2) *
timestep;
tshfo = shelf1temp(time - 1);
brac1 = tor * 0.604 * Eeffal1 / mcpshf1;
brac2 = (tshfo*tshfo*tshfo*tshfo) -
(tcano*tcano*tcano*tcano);
shelf1temp(time) = tshfo + (shf1pow/mcpshf1 - brac1 *
brac2) * timestep;

tshfo = shelf2temp(time - 1);
brac1 = tor * 0.625 * Eeffal2 / mcpshf2;
brac2 = (tshfo*tshfo*tshfo*tshfo) -
(tcano*tcano*tcano*tcano);
shelf2temp(time) = tshfo + (shf2pow/mcpshf2 - brac1 *
brac2) * timestep;

%tshfo = battyetemp(time - 1);
%brac1 = tor * 0.625 * Eeffal2b / mcpbtel;
%brac2 = (tshfo*tshfo*tshfo*tshfo) -
(tcano*tcano*tcano*tcano);
%battyetemp(time) = tshfo + (bat2pow/mcpbtel - brac1 *
brac2) * timestep;

tshfo = shelf3temp(time - 1);
brac1 = tor * 0.587 * Eeffal3 / mcpshf3;
brac2 = (tshfo*tshfo*tshfo*tshfo) -
(tcano*tcano*tcano*tcano);
shelf3temp(time) = tshfo + (shf3pow/mcpshf3 - brac1 *
brac2) * timestep;

tshfo = shelf4temp(time - 1);
brac1 = tor * 0.393 * Eeffal4 / mcpshf4;
brac2 = (tshfo*tshfo*tshfo*tshfo) -
(tcano*tcano*tcano*tcano);
shelf4temp(time) = tshfo + (shf4pow/mcpshf4 - brac1 *
brac2) * timestep;

end      % of the main loop

maxima = zeros(5,1);
minima = ones(5,1);
minima = minima*1000;

% for a = 1:500

for a = 1:(480 * 30/heatres)
 result(1,a) = shelf1temp(a);
 result(2,a) = shelf2temp(a);
%  result(3,a) = battyetemp(a);
 result(3,a) = shelf3temp(a);
 result(4,a) = shelf4temp(a);
 result(5,a) = insidecan(a);
 if shelf1temp(a) > maxima(1)
  maxima(1) = shelf1temp(a);
 end
 if shelf1temp(a) < minima(1)
  minima(1) = shelf1temp(a);
 end
 if shelf2temp(a) > maxima(2)
  maxima(2) = shelf2temp(a);
 end
 if shelf2temp(a) < minima(2)
  minima(2) = shelf2temp(a);
 end
%  if battyetemp(a) > maxima(3)
%   maxima(3) = battyetemp(a);
%  end
 if shelf3temp(a) > maxima(3)
```

```
  maxima(3) = shelf3temp(a);
 end
 if shelf3temp(a) < minima(3)
  minima(3) = shelf3temp(a);
 end
 if shelf4temp(a) > maxima(4)
  maxima(4) = shelf4temp(a);
 end
 if shelf4temp(a) < minima(4)
  minima(4) = shelf4temp(a);
 end
 if insidecan(a) > maxima(5)
  maxima(5) = insidecan(a);
 end
 if insidecan(a) < minima(5)
  minima(5) = insidecan(a);
 end
end

prnt = maxima          % print maximum temperature
reached by each node during the
                % simulation in degrees kelvin
prnt = maxima - 273     % repeat in celcius
prnt = minima - 273


%
% this version created  : 09/01/96
%             update : 09/01/96
%                by: M.J.Steedman
%             status: 5 degree clecius regtemp calculations
with
%                  possibly different heater powers on each
shelf
%                  version with heaters shelves 2,3,4 battery
2 part
%                  of shelf 2 once more
%
%                  corrected version; changed to reflect the
actual
%                  energy containment analysis version
requirements
%                  as outlined at NASA teleconference
22/11/94
%
%             For: Thermal energy containment analysis
(worst case)
%                  for Get Away Special Payload G-490 of
The Robert
%                  Gordon University Aberdeen
%
% full modelling version : 20/02/94
%             status : writing in progress
%
```

List Of References

[1] K. Hehl, J. Fritsch, G. Xu, 'Carborne Gravimetry-merely A Trial Or A Method For Determining Gravity On A Profile?', IEEE 1994 Position Location And Navigation Symposium. p.376-80., 11-15 April 1994, Loc.: Las Vegas, NV, USA.

[2] Joseph J. Howell, 'Microgravity Monitoring Instrument Development And application To Vernir Guidance, Navigation, and Vehicle Control', SPIE, Vol. 1694 (1992).

[3] D.I. Jones, 'Vibration Control In Space', Compute. & Control Eng. J., Vol. 5, No 2, 1994.

[4] Y. Jafry, 'STEP, A Laboratory For Experimental Gravitation In Space.', Experimental Gravitation. Proceedings of the International Symposium, conf. 26 Jun - 2 Jul 1993, Nathiagali, Pakistan. Pub: IOP Publishing Bristol UK.

[5] Abs only, 'Microgravity Accelerometer Data Systems For Space Experiments', J. Of Metals, Vol. 39, Issue 10, 1987.

[6] W.M. Neupert, G.L. Epstein, J. Houston, A. Zarechnak, ' Measurements Of The STS Orbiters Angular Stability During In-Orbit Operations', 1995 Shuttle Small Payloads Symposium, Camden Yards, Baltimore, Maryland, Sept. 25-28, 1995, NASA Conf. Pub. 3310.

[7] J.M. Mardones, C.P. Garcia, 'Linear instabilities In Viscoelastic Fluid Convection', J. Physics And Condensed Matter, Vol. 2, Issue 5, 1990.

[8] T.C. Jue, B. Ramaswamy, 'Natural Convection With Thermocapillary And Gravity Modulation Effects In Low Gravity Environments', J. Spacecraft & Rockets, Vol. 29, No 6, Nov-Dec 1992.

[9] R. Clever, G. Schubert, 'Three-Dimensional Oscillatory Convection In A Gravitationally Modulated Fluid Layer', Physics. Fluids, Vol. 5, No 10, Oct 1993.

[10] A.M. Vetoshkin A.V. Korol'kov V.S. Kuptsova V.V. Savichev, 'Features Of The Development Of Convective Processes Under Nearly Weightless Conditions', Translation From Russian, Inz.-fiz. Zhu. Vol. 62, No 2,Feb 1992.

[11] S.A. Kassemi, S. Ostrach, 'Nature Of Buoyancy-Driven Flows In A Reduced-Gravity Environment', AIAA Journal, Vol. 30, No 7, July 1992.

[12] M. Kassemi, M.H.N. Naraghi, 'Analysis Of Radiation-Natural Convection Interactions In 1-g And Low-g Environments Using The Discrete Exchange Factor Method', Int. J. Heat Mass Transfer, Vol. 36, No 17, 1993.

[13] A. Ito, S.K. Choudhury, T. Fukano, 'A Technique For Measuring An Unsteady Temperature Distribution In A Liquid Using Holographic Interferometry', JSME Int. J.-S2 Fluids Eng. Heat Transfer.

[14] Y.M. Chen, F. Mayinger, 'Holographic Interferometry Studies Of The Temperature Field Near A Condensing Bubble', Opt. Meth. Dynamics flu. Solid, IUTAM Symp. Liblice Czech. 1984.

[15] A. Verga, 'Spaceborne Autoranging Infrared Viewing System For Thermal Mapping Of Fluids In A Microgravity Condition', Opt. Eng., Vol. 33, No 1, Jan 1994.

[16] S. Ushijima, H. Takeda, N. Tanaka, 'Image Processing System For Velocity Measurements In Natural Convection Flows', Nuclear Eng. And Design, Vol. 132, No 2, 1991.

[17] J.M. Coupland, N.A. Halliwell, 'Particle Image Velociometry: Three-Dimensional Fluid Velocity Measurements Using Holographic Recording And Optical Correction', Applied Optics, Vol. 31, No 8, 1992.

[18] X.H. Yan, L.C. Breaker, 'Surface Circulation Estimation Using Image-Processing And Computer Vision Methods Applied To Sequential Satellite Imagery', Photogrammetric Eng. & Remote Sensing, Vol. 59, No 3, 1993.

[19] L.G. Napolitano, G. Golia, A. Viviani, 'Effects Of Variable Transport Properties On Thermal Marangoni Flows', Acta Astronautica, Vol. 13, No 11/12 1986.

[20] P.L.G. Ybarra, M.G. Verlarde, 'Oscillatory Marangoni-Bernard Interfacial Instability And Capillary-Gravity Waves In Single- And Two- Component Liquid Layers With Or Without Soret Thermal Diffusion', Physics Fluids, Vol. 30, No 6, June 1987.

[21] H.F. Bauer, 'Heat Transfer In A Liquid Column In Weightless Conditions Due To Thermal Marangoni', Numerical Heat Transfer, Vol. 8, 1985.

[22] N. Ramachandran, C.A. Winter, 'Effects Of G-Jitter And Marangoni Convection On Float Zones', J. Spacecraft & Rockets, Vol. 29, No 4, July-Aug 1992.

[23] A. Bergeon, D. Henry, H. Benhadid, 'Marangoni-Bernard Instability In Microgravity Conditions With Soret Effect', International Journal Of Heat And Mass Transfer, Vol. 37, No 11, 1994, pp 1545-1562.

[24] R.F. Dressler, N.S. Sivakumaran, 'Non-Contaminating Method To Reduce Marangoni Convection In Microgravity Float Zones', J. Crystal Growth, No 88, 1988.

[25] H.C.J. Hoefsloot, H.W. Hoogstraten, A. Hoven, L.P.B.M. Janssen, 'Marangoni Instability In A Liquid Layer Bounded By Two Coaxial Cylinder Surfaces', Applied Scientific Research, No 47, 1990.

[26] H.C.J Hoefsloot, H.W. Hoogstraten, L.P.B.M. Janssen, 'Marangoni Instability In A Liquid Layer Confined Between Two Concentric Spherical Surfaces Under Zero Gravity Conditions', Applied Scientific Research, No 47, 1990.

[27] J.H. Lichtenbelt, A.A.H. Drinkenburg, 'Marangoni Convection And Mass Transfer From The Liquid Phase To The Gas Phase Under Microgravity Conditions', Naturwissenschaften, 73 (1986) .

[28] Y. Kamotani, S. Ostrach, A. Pline, 'Analysis Of Velocity Data Taken In Surface-Tension Driven Convection Experiment In Microgravity', Physics Of Fluids, Vol. 6, No 11, 1994, pp 3601-3609.

[29] B. Hines, 'Optical Design Issues For The Micro- Precision Interferometer Test-bed For Space-Based Interferometry.', Proc. SPIE. The International Soc. For Optical Engineering, Vol. 1947, 1993, p 114-25.

[30] M.C. Muckerheide, 'Post Flight Operation Of A High Peak Power Neodymium Yag Laser Aboard The G-449 Payload Flown On Space Shuttle Columbia Mission 61-C', 1992 Shuttle Small Payloads Symposium, Sheraton-Greenbelt, Lanham, Maryland, Oct. 20-23, 1992, NASA Conf. Pub. 3171.

[31] Gresho, P.M., Sani, R.L., 'The Effects Of Gravity Modulation On The Stability Of A Heat Fluid Layer', Journal of Fluid Mechanics, Vol. 40, No , 1970, pp 783-806.

[32] Biringen, S., Danabasoglu, G., 'Computation Of Convection Flow With Gravity Modulation In Rectangular Cavities', J. Thermophysics & Heat Transfer, Vol. 4, No 3, 1990, pp 357-365.

[33] Carpenter, B.M., Hornsy, G.M., 'Combined Buoyant Thermocapillary Flow In A Cavity', Journal of Fluid Mechanics, Vol. 207, No , 1989, pp 121-132.

[34] K.T. Yang, 'Numerical Modelling Of Natural Convection -radiation Interactions In Enclosures.', Heat Transfer, 1986, pp 131-140 Hemisphere Washington D.C.

[35] Kamotani, Y., Prasad, A., Ostrach, S., 'Thermal convection In An Enclosure Due To Vibrations Aboard A Spacecraft', AIAA Journal, Vol. 19, No 4, 1981, pp 511-516.

[36] C. Saltiel, M.H.N. Naraghi, 'Analysis Of Radiative Heat Transfer In Participating Media Using Arbitrary Nodal Distribution.', Numerical Heat Transfer, Vol. 17, No B, 1990, pp 227-243.

[37] M.H.N. Naraghi, M. Kassemi, 'Analysis Of Radiative Transfer In Rectangular Enclosures Using A Discrete Exchange Factor Method.', ASME J. Heat Transfer, Vol. 111, No , 1989, pp 1117-1120.

[38] W.A. Fiveland, 'Discrete-Ordinate Solution Of The Radiative Transport Equation For Rectangular Enclosures', J. Heat Transfer, Vol. 106, No , 1984, pp 699-706.

[39] Ramaswamy, B., Jue, T.C., Akin, J.E., 'Finite Element Analysis Of Oscillatory Flow With Heat Transfer Inside A Square Cavity', AIAA Journal, Vol. 30, No 2, 1992, pp 412-422.

[40] G. de Vahl Davis, 'Natural Convection Of Air In A Square Cavity : A Bench Mark Numerical Solution', Int. J. Numerical Methods Fluids, Vol. 3, 1983, pp249-264.

[41] S. Ostrach, 'Natural Convection In Enclosures', J. Heat Transfer, Vol. 110, No ?, 1988. pp 1175-1195.

[42] G. Lauriat, 'Combined Radiation-Convection In Grey Fluids Enclosed In Vertical Cavities', ASME J. Heat Transfer, Vol. 104, No ?, 1982, pp 609-615.

[43] Robert Gordon's Institute of Technology (Robert Gordon University), 'IEE/British Sugar Space Shuttle Competition, RGIT Grampian Get Away Special', Private correspondence/proposal submission to IEE competition, Jan. 1990.

[44] J. Illingworth J. Kittler ,'A Survey Of The Hough Transform', Comp. Vis. Graph. & Image Proc.

[45] V.F. Leavers, 'Which Hough Transform (Survey)', CVGIP Image Understanding, Vol. 58, No 2, 1993.

[46] L. Davis, 'A Survey Of Edge Detection Techniques', Compute. Graphics Image Process, Vol. 4, 1975.

[47] T. Pavlidis, 'A Critical Survey Of Image Analysis Methods', Expand. Paper: 8th Int. Conf. Pattern Recognition, Paris, Oct 1986.

[48] B. Hajek, 'A Tutorial Survey Of Theory And Applications Of Simulated Annealing', Proc. 24th IEEE Conf. Decision and Control, 1985, pp 755-760.

[49] W.A. Carpenter, R.G. Stiles, S.K. Sheppard ,'Colour Map Of Contrast Enhancement On MR Images : Use Of Desktop Computers', American J. Roentgenology, Vol. 162, No 1, 1994.

[50] S. Aghagolzadeh, O.K. Ersoy, 'Transform Image Enhancement', Optical Engineering, Vol. 31, No 3, 1992.

[51] D.C.D. Hung, 'Enhancement And Feature Purification Of Fingerprint Images', Pattern Recognition, Vol. 26, No 11, 1993.

[52]Jia-Guu Leu, 'Image Contrast Enhancement Based On The Intensities Of Edge Pixels', GVGIP: Graph. Mod. & Image Proc., Vol. 54, No 6, Nov 1992.

[53]G. Deng, L.W. Cahill, 'Multiscale Image Enhancement Using The Logarithmic Image Processing Model', Electronics Letters, Vol. 29, No 9, 29 April 1993.

[54] D.D. Sworder, P.F. Singer, D. Doria, R.G. Hutchins, 'Image-Enhanced Estimation Methods', Proceedings Of The IEEE, Vol. 81, No 6, June 1993.

[55] Sung-Jea Ko, T.M. Forest, 'Image Sequence Enhancement Based On Adaptive Symmetric Order Statistics', IEEE Trans. Cir. & Sys. II, Analogue % Dig. Sig. Proc., Vol. 40, No 8, Aug 1993.

[56] Y. Yamashita, H. Ogawa, 'Optimum Image Restoration Filters And Generalised Inverses Of Operators', Systems And Computers In Japan, Vol. 24, No 6, 1993.

[57] T.M. Cannon, H.J. Trussel, B.R. Hunt, 'Comparison of Image Restoration Methods', Applied Opt., Vol. 17, pp 3385-3390, 1978.

[58] T.R. Crimmins, 'Geometric Filter For Speckle Reduction', Applied Optics, Vol. 24, No 10, 1985, p 1438-1443.

[59] D.T. Kuan, A.A. Sawchuk, T.C. Strand, P. Chavel, 'Adaptive Restoration Of Images With Speckle', IEEE Trans. ASSP, Vol. 135, No, 1987, p 373-383.

[60] H.H. Aresnault, M. Denis, 'Image Processing In Signal-Dependent Noise', Can. J. Physics, Vol. 61, 1983, p 309-317.

[61] B. Gildas, 'A Renormalisation Group Approach to Image Processing Problems', IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 11, pp 164-180, Feb 1989.

[62] S. Geman, D, Geman, 'Stochastic Relaxation, Gibbs Distributions, And The Bayesian Restoration Of Images', IEEE Trans. Pattern Anal. Machine Intelligence, Vol. PAMI-6, pp 721-741, 1984.

[63] K. Komai, K. Minoshima, S. Ishii, 'Recognition Of Different Fracture Surface Morphologies Using Computer Image Processing Technique', JSME Int. J. Series A, Vol. 36, No 2, 1993.

[64] Editorial, 'Detection And Comparison Of Patterns In Images', The Journal Of Nuclear Medicine, Vol. 35, No 1, 1994.

[65] C.-H. Chuang, C.-C., 'A Wavelet Deformable Model For Shape Description And Mulitscale Elastic Matching', Proc Of The SPIE - The Int. Soc. For Opt. Eng., Vol. 2094, No 2, 1993.

[66] A. Samal, P.A. Iyengar, 'Natural Shape Detection Based On Principal Component Analysis', Journal of Electronic Imaging, Vol. 2, No 3, 1993.

[67] R.L. Brown, 'The Fringe Distance Measure: An Easily Calculated Image Distance Measure With Recognition Results Comparable To Gaussian Blurring', IEEE Trans. Sys. Man & Cybernetics, Vol. 24, No 1, Jan 1994.

[68] W. Philips, 'A New Fast Algorithm For Moment Computation', Pattern Recognition, Vol. 26, No 11, 1993.

[69] H. Bunke, U. Buhler, 'Applications Of Approximate String Matching To 2D Shape Recognition', Pattern Recognition, Vol. 26, No 12, 1993.

[70] S. Hambrusch, 'Parallel Algorithms For Grey-Scale Digitised Picture Component Labelling On A Mesh Connected Computer', J. Of Parallel. & Dist. Compute., Vol. 20, No 1, 1994.

[71] J.R. Ullmann, 'Edge Replacement In The Recognition Of Occluded Objects', Pattern Recognition, Vol. 26, No 12, 1993.

[72] T. Nagao, T. Agui, H. Nagahashi, 'Extraction Of Two-Dimensional Arbitrary Shapes Using A Genetic Algorithm', Proc. Of The SPIE - The Int. Soc. For Opt. Eng., Vol. 2094, No 1, 1993.

[73] D. Marr, E. Hildreth, 'Theory Of Edge Detection', Proc. R. Soc. London Series B', Vol. 207, 1980.

[74] F. Kammoun J.P. Astruc, 'Optimum Edge Detection For Object- Background Picture', CVGIP Graph. Mod. & Im. Proc., Vol. 56, No 1, Jan 1994.

[75] I. Crooks, B.G. Fallone, 'A Novel Algorithm For The Edge Detection And Edge Enhancement Of Medical Images', Med. Physics, Vol. 20, No 4, 1993.

[76] T.J. Herbert, D. Malagre, 'Optimisation Approach To Edge Detection', J. Opt. Soc. America, Vol. 11, No 1, 1994.

[77] A.M. Taratorin, S. Sideman, 'Constrained Regularised Differentiation', IEEE Trans. On Pattern Analysis & Machine Intelligence, Vol. 16, No 1, 1994.

[78] Y. Zhou, R. Chellappa, 'Edge Detection Using Zero Crossings Of Second Order Directional Derivatives Of A Random Field Model', Proc. of ICASSP (conference), New York 1986. (IEEE).

[79] A. Gruen, D. Stallmann, 'High Accuracy Matching Of Object Edges', Proc. Of The SPIE - The Int. Soc. For Opt. Eng., Vol. 1820, 1993.

[80] H. Wang, B.G. Fallone, 'A Robust Morphological Algorithm For Automatic Field Extraction And Correlation Of Portal Images', Med. Physics, Vol. 21, No 2, 1994.

[81] K.N.V.L.N. Koundinya, B. Chanda, 'Detecting Lines In Grey Level Images Using Search Techniques', Signal Processing, Vol. 37, No 2, 1994.

[82] X.W. Xu, K, Doi, 'Accurate Automated Delineation Of Rib Cage Boundary On Digital Chest Images Foe Computer-Aided Diagnosis Schemes', Radiology, Vol. 185, No S, 1992.

[83] M. Meribout, K.M. Hou., 'A Real Time Re-programmable Low Level Image Processing: Edge Detection And Edge Tracking Accelerator', Proc. Of The SPIE - The Int. Soc. For Opt. Eng., Vol. 2094, No 3, 1993.

[84] C.-W. Tao, W.E. Thompson, J.S. Taur, 'A Fuzzy If Then Approach To Edge Detection', 2nd IEEE Int. Conf. On Fuzzy Sys., San Francisco, CA, USA, 1993.

[85] L.-H. Chen, W.-H. Tsai, 'Location Of Curved Edges To Sub-pixel Values', J. Of The Chinese Inst. Of Eng., Vol. 16, No 5, 1993.

[86] A.D. Calway, R. Wilson, 'Curve Extraction In Images Using A Multi-resolutional Framework', CVGIP. Image Understanding, Vol. 59, No 3, 1994.

[87] M. Becker, D. Ley, 'Extracting A Symbolic Object Description From Grey-Level Images Using A Kalman-Filter-Based Contour Tracer Combined With A Complex Edge-Operator', SPIE Intelligent Robots And Computer Vision, Vol. 1825, 1992.

[88] D. Geman, 'Stochastic Model For Boundary Detection', Image and Vision Computing, Vol. 5, No 2, 1987, pp61-65.

[89] M. Petrou, A. Kolovas, 'The Recursive Implementation Of The Optimal Filter For The Detection Of Roof Edges And Thin Lines', Sig. Proc VI - Theories & Apps. Proc. Of EUSIPCO-92, 6th European Sig. Proc. Conf., Amsterdam, Vol. 3, 1992

[90] R.O. Duda, P.E. Hart, 'Use of the Hough Transform to Detect Lines and Curves in Images', Communications ACM., Vol. 15, 1972, pp 11-15.

[91] R.F. Vaz, D. Cyganski, 'Generation Of Affine Invariant Local Contour Feature Data', Pattern Recognition Letters, Vol. 11, No 7, 1990, pp 479-483.

[92] B.D. Carlson, E.D. Evans, S.L. Wilson, 'Search Radar Detection And Track With The Hough Transform : Part 1 System Concept', IEEE Trans. Aerospace & Elect. Sys.

[93]B.D. Carlson, E.D. Evans, S.L. Wilson, 'Search Radar Detection And Track With The Hough Transform. : Part 2 Detection Statistics', IEEE Trans. Aerospace & Elect. Sys.

[94]B.D. Carlson, E.D. Evans, S.L. Wilson, 'Search Radar Detection And Track With The Hough Transform. : Part 3 Detection Performance With Binary Integration', IEEE Trans. Aerospace & Elect. Sys.

[95] M. Bober, J. Kittler, 'Estimation Of Complex Multi-modal Motion - An Approach Based On Robust Statistics And Hough Transform', Image And Vision Computing, Vol. 12, No 10, 1994, pp661-668.

[96] C. Pieralli, G. Tribillion., 'Contouring Using Interferometric Fringe Digital Processing With The Hough Transform: Application To The Detection Of Cavities On Polished Surfaces.', Signal Processing, Vol. 40, No 2-3, 1994, pp 217-225.

[97] D. Casasent, R. Krishnapuram, 'Curved Object Location By Hough Transformations And Inversions', Pattern Recognition, Vol. 20, No 2, 1987.

[98] B.C. Arrue, R.M. Inigo, 'High Order Curve Recognition With The Hough Transform Using A Connectionist Approach', Proc. Of The SPIE - The Int. Soc. For Opt. Eng., Vol. 1966, 1993.

[99] A. Kavianpour, S. Shoari, N. Bagherzadeh, 'A New Approach For Circle Detection On Multiprocessors', J. Of Parallel & Distributed Computing, Vol. 20, 1994.

[100] X. Cao, M.G. Rodd, F. Deravi, Q.M. Wu, 'Detection of Multiple Circles Based on Adaptive Hough Transform', Eng. Applied Artificial Intelligence, Vol. 1, No 2, 1988, pp97-101.

[101] S. Kumar, N. Ranganathan, D. Goldgof, 'Parallel Algorithms For Circle Detection In Images', Pattern Recognition, Vol. 27, No 8, 1994, pp 1019-1028.

[102] C. Kimme, D. Ballard, J. Sklansky, 'Finding Circles By An Array Of Accumulators', Communications of the ACM, Vol. 18, No 2, Feb 1975.

[103] R.K.K. Yip, P.K.S. Tam, D.H.K. Leung, 'Modification Of Hough Transform For Circle And Ellipse Detection Using A Two Dimensional Array', Pattern Recognition, Vol. 25, No 9, 1992, pp1007-1022.

[104] S. Tsuji, F. Matsumoto, 'Detection Of Ellipses By A Modified Hough Transformation', IEEE Transactions on Computers, Vol. c-27, No 8, Aug 1978, pp777-781.

[105] H.K. Yuen, J. Illingworth, J. Kittler, 'Ellipse Detection Using The Hough Transform', In Avley Vision Conference, Manchester, 1988.

[106] A.M. Cross, 'Detection of Circular Geological Features Using The Hough Transform', Int. J. Remote Sensing, Vol. 9, No 9, 1988, pp1519-1528.

[107] H. Taud, J.F. Parrot, 'Detection of Circular Structures on Satellite Images', Int. J. Remote Sensing, Vol. 13, No 2, 1992, pp 319-335.

[108] D.H. Ballard, 'Generalising The Hough Transform To Detect Arbitrary Shapes', Pattern Recognition, Vol. 13, No 2, 1981.

[109] C.J Sheng, H.T. Wen, 'Fast Generalised Hough Transform', Pattern Recognition Letters, Vol. 11, No 11, 1990, pp 725-733.

[110] V.F. Leavers, 'The Dynamic Generalised Hough Transform: Its Relationship To The Probabilistic Hough Transforms And An Application To The Concurrent Detection Of Circles And Ellipses', CVGIP Image Understanding, Vol. 56, No 3, Nov 1992, pp 381-398.

[111] J. Illingworth, J. Kittler, 'The Adaptive Hough Transform', IEEE Trans. on Pattern Anal. & Machine Intelligence.', Vol. 9, No 5, 1987, pp690-698.

[112] V.F. Leavers, J.F. Boyce, 'The Radon Transform And Its Application To Shape Parameterisation In Machine Vision', Image And Vision Computing, Vol. 5, No 2, May 1987.

[113] V.F. Leavers, G.F. Miller, 'Radon Transformation Of Delta-Function Curves. A Geometric Approach', Alvey Vision Club Conference, pp 335-340, Cambridge, Sept. 1987b.

[114] N. Friedland, D. Adam, 'Automatic Ventricular Cavity Boundary Detection From Sequential Ultrasound Images Using Simulated Annealing', IEEE Trans. Biomedical Eng., BME-34 May 1987.

[115] D.M. Tsai, M.F. Chen, 'Curve-Fitting Approach For Tangent Angle And Curvature Measurements', Pattern Recognition, Vol. 27, No 5, 1994, pp 699-711.

[116] P.L. Rosin. G.A.W. West, 'Detection Of Circular Arcs In Images', Proceedings of the fourth Avley Vision Conference, University Of Manchester, 31 Aug - 2 Sept 1988.

[117] H.L. Tan, S.B. Gelfand, E.J. Delp, 'A Comparative Cost Function Approach To Edge Detection', IEEE Trans. On Sys. Man & Cyber., Vol. 19, No 6, 1989.

[118] D. Chang, S. Hashimoto, 'Finding Edges In Images: A Hierarchical Relaxation Method', Trans. Of The Soc. Of Instrument & Control Eng., Vol. 29, No 12, 1993.

[119] D. Geman, S. Geman, C. Graffigne, P. Dong, 'Boundary Detection By Constrained Optimisation', IEEE Trans. Pattern Mach. Intelligence, Vol. 12, No ?, 1990.

[120] K. Nagao, M. Sohma, K. Kawakami, S. Ando, 'Detecting Contours In Image Sequences', IEICE: Transactions On Information And Systems, Vol. E76-D, No 10, 1993.

[121] S. Ghosal, R. Mehrotra, 'Detection Of Composite Edges', IEEE Trans. On Image Proc., Vol. 3, No 1, 1994.

[122] M. Kisworo, S. Venkatesh, G. West' ,Modelling Edges At Sub-pixel Accuracy Using The Local Energy Approach', IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 16, No 4, 1994.

[123] K.B. Eom, R.L. Kashyap, 'Composite Edge Detection With Random Field Models', IEEE Trans. On Sys. Man & Cyber., Vol. 20, No 1, 1990.

[124] D. Mintz, 'Robust Consensus Based Edge Detection, 'CVGIP: Image Understanding', Vol. 59, No 2, 1994.

[125] R. Sai Prasad, T.M. Srinivasan, 'An Image Processing Method For Cardiac Motion Analysis', IEEE Trans. Biomedical Eng., BME-34 March 1987.

[126] Guy E. Mailloux, Andre Bleau, Michel Bertrand, Robert Petitclerc, 'Computer Analysis of Heart Motion From Two-Dimensional Echocardiograms', IEEE Trans. Biomedical Eng., BME-34 May 1987

[127] Liang-Fu Zhang, Edward A. Geiser, 'An Effective Algorithm For Extracting Serial Endocardial Borders From 2-Dimensional Echocardiograms', IEEE trans. Biomedical Eng., BME-31 June 1984.

[128] J.H. Duncan, T.C. Chou, 'On The Detection Of Motion And The Computation Of Optical Flow', IEEE Trans. Pattern Analysis & Machine Intelligence, Vol. 14, No 3, 1992.

[129] Q. Lin, J.F. Vesecky, H.A. Zebker, 'Phase Unwrapping Through Fringe-Line Detection In Synthetic-Aperture Radar Interferometry', Applied Optics, Vol. 33, No 2, 1994, pp 201-208.

[130] M. Servin, R. Rodriguez-vera, A.J. Moore, 'A Robust Cellular Processor For Phase Unwrapping', J. Modern Optics, Vol. 41, No 1, 1994.

[131] L. Crescentini, G. Fiocco, 'Automatic Fringe Recognition And Detection Of Sub-wavelength Phase Perturbations With A Michelson Interferometer', Appl. Opt., Vol. 27, 1988, pp 118-123.

[132] H.E. Cline, A.S. Holik, W.E. Lorensen, 'Computer Aided Surface Reconstruction Of Interference Contours', Applied Optics, Vol. 21, No 24, 1982, p 4481-4488.

[133] H.E. Cline, W.E. Lorensen, A.S. Holik, 'Automatic Moire Contouring', Applied Optics, Vol. 23, No 10, 1984, p 1454-1459.

[134] D.W. Robinson, 'Automatic Fringe Analysis With A Computer Image Processing System', Appl. Opt., Vol. 22, No 14, 1983, pp 2169-2176.

[135] W. Osten, W. Juptner, U. Mieth, 'Knowledge Assisted Evaluation Of Fringe Patterns For Automatic Fault Detection.', Proc SPIE. The International Soc. for Opt. Eng., Vol. 2004, 1994, p 256-68.

[136] H. Glunder, R. Lenz, 'Fault Detection In Non-destructive Testing (NDT) By An Opto-electronic Hybrid Processor', Proceedings SPIE, Vol. 370, No, 1982, p 157-162.

[137] D.A. Tichenor, V.P. Madsen, 'Computer Analysis Of Holographic Interferograms For Non-destructive Testing.', Optical Engineering, Vol. 18, No 5, 1979, pp 469-472.

[138] Ming Chang, P.P. Lin, Wen Chin Tai, 'A Fringe Centre Detection Technique Based On A Sub-Pixel Resolution, And It's Applications Using Sinusoidal Gratings.', Optics and Lasers In Engineering, Vol. 20, No 3, 1994, p 162-76.

[139] E.W.S. Nakadate, N. Magome, T. Honda J. Tsujiuchi, 'Hybrid Holographic Interferometer For Measuring Three-Dimensional Deformations', Opt. Eng., Vol. 20, No 2, 1981, p 246-252.

[140] G.A. Mastin, D.C. Ghiglia, 'Digital Extraction Of Interference Fringe Contours', Applied Optics, Vol. 24, No 12, 1985, p 1727-1728.

[141] D.J. Bone, H.A. Bachor, R.J. Sanderman, 'Fringe-Pattern Analysis Using A 2-D Fourier Transform', Applied Optics, Vol. 25, No 10, 15 May 1986.

[142] K.J. Gasvik, K.G. Robbersmyr, 'Error Analysis Of A Zero-Crossing Algorithm For Fringe Location' Optical Engineering, Vol. 33, No 1, 1994, pp 246-250.

[143] R. Hofling, W. Osten, 'Displacement Measurement By Image- Processed Speckle Patterns', J. Mod. Optics, Vol. 34, No, 1987, p 607.

[144] W. Osten, R. Hofling, J. Saedler, 'Two Computer-Aided Methods For Data Reduction From Interferograms', Proc SPIE, Vol. 863, No, 1987, p 105-113.

[145] N. Eichhorn, W. Osten, 'An Algorithm For The Fast Derivation Of Line Structures From Interferograms', J. Mod. Opt., Vol. 35, No 10, 1988, pp 1717-1725.

[146] K. Paler, K.M. Crenell, J. Kittler, B.N. Dobbins, B.L. Button, C. Wykes., 'Identification Of Fringe Minima In Electronic Speckle Pattern Images.', Pattern Recognition Letters, Vol. 6, No 1, 1987, p 33-44.

[147] I.W. Bowler, K.A. Paler, 'A Gabor Filter Approach To Fringe Analysis', Proceeding of the eight international conference on pattern recognition., Paris, (IEEE), 1986, p 558-560.

[148] W.R.J. Funnell, 'Image Processing Applied To The Interactive Analysis Of Interferometric Fringes', Applied Optics, Vol. 20, No 18, 1981, pp 3245-3250.

[149] F. Becker, H. Yung, 'Digital Fringe Reduction Techniques Applied To The Measurement Of Three- Dimensional Transonic Flow Fields', Opt. Eng., Vol. 24, No 3, 1985, p429-434.

[150] T. Yatagai, S. Nakadate, M. Idesawa, H. Saito, 'Automatic Fringe Analysis Using Digital Image Processing Techniques', Opt. Eng., Vol. 21, No 3, 1982, p 432-435.

[151] NASA, 'Get Away Special (GAS) Small Self Contained Payloads, Experimenter Handbook', Special Payloads Division, Goddard Space Flight Centre, 1995.

[152] R.M. Taylor, 'Critical Analysis of Support Structure', G-490 Structural Analysis, Private Correspondence, Dec. 1995.

[153] M.J. Steedman, 'G-490 Payload Accommodations Requirements', Safety Document for NASA Safety Review, Private Correspondence, Dec. 1994.

[154] M.J. Steedman, 'G-490 Safety Data Package', Safety Document for NASA Safety Review, Private Correspondence, Feb. 1996.

[155] M.J. Steedman, 'G-490 Material List', Private Correspondence to NASA, Feb. 1996.

[156] M.J. Steedman, 'G-490 Ground Test Technical Operating Procedures', Private Correspondence to NASA, Feb. 1996.

[157] D. Butler, 'Get Away Special (GAS) Thermal Design Summary', X-732-83-8, Goddard Space Flight Centre, July, 1983.

[158] D. Durham, 'G-490 Fluid Advise', School of Pharmacy, The Robert Gordon University, private correspondence, 1995.

[159] M.J. Steedman, 'G-490 Battery Energy Capacity Calculations', NASA projects work, NASA Project logs, Private Correspondence, 1994.

Bibliography

1) R.C. Gonzalez, R.E. Woods, 'Digital Image Processing', 2nded, Addison-Wesley, 1987.

2) E.R. Davies, 'Machine Vision : Theory, Algorithms, Practicalities', Acad. Press, Jan 1990.

NASA, 'Get Away Special (GAS) Small Self Contained Payloads, Experimenter Handbook', Special Payloads Division, Goddard Space Flight Centre, 1995.

NASA, '1992 Shuttle Small Payloads Symposium', Sheraton-Greenbelt, Lanham, Maryland, Oct. 20-23, 1992, NASA Conf. Pub. 3171.

NASA, '1993 Shuttle Small Payloads Symposium', Holiday Inn, Annaplois, Maryland, Oct. 4-7, 1993, NASA Conf. Pub. 3233.

NASA, '1995 Shuttle Small Payloads Symposium', Camden Yards, Baltimore, Maryland, Sept. 25-28, 1995, NASA Conf. Pub. 3310.

R.D.Taylor, 'Structural Analysis', G-490, The Robert Gordon University, February 96, private correspondence, report for NASA.