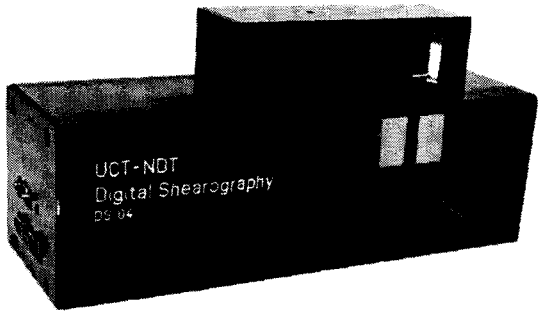
DEPARTMENT OF MECHANICAL ENGINEERING

RONDEBOSCH, CAPE TOWN, SOUTH AFRICA

DEVELOPMENT OF A PORTABLE LOW-COST DIGITAL SHEAROGRAPHY SYSTEM



Barry Pitman

BSc Electro-Mechanical Engineering

November 2008

Submitted in partial fulfilment of the academic requirements for the degree of Master of Science in Engineering

DECLARATION

1. I know the meaning of plagiarism and declare that all the work in the document, save for that which is properly acknowledged, is my own.

2. Each significant contribution to, and quotation in, this project from the work, or works, of other people has been cited and referenced.

3. This report is my work.

4. I have not allowed, and will not allow, anyone to copy my work with the intention of passing it off as his or her own.

Signed by candidate

Barry Matthew Pitman

ACKNOWLEDGEMENTS

The completion of this dissertation would not have been possible without the assistance of several people. Firstly, thanks go to my project supervisors, Dirk Findeis and Jasson Gryzagoridis, who made themselves available at all times to answer questions, give advice and provide valuable assistance.

Secondly, thanks must also go to other UCT staff who offered guidance: Glen Newins for his patience and light-hearted attitude in the Mechanical Engineering Workshop and Samuel Ginsberg, whose knowledgeable advice in designing the high-resolution digital-to-analogue converter and creating a PCB board for the controller was much appreciated.

Thirdly, thanks must go to the individuals who maintain the freeware and open-source tools that were used in the course of this dissertation. Dennis Bareis' freeware *MakeMsi* Windows Installer scripting language was used to write an installer package for the custom-developed software. David Tschumperlé's open-source *CImg* toolkit was used to smooth phase-stepped images as well as open and save images.

Finally, I would like to thank my parents, family and friends for the support which they have offered me during the course of this dissertation and throughout my university career. This dissertation would certainly not have been possible without their love, patience and support.

SUMMARY

Introduction

Non-Destructive Testing (NDT) forms an important part of the production and maintenance cycles of safety-critical components in many industries. While aerospace, automotive and defence-related industries are making increased use of composite materials in their most demanding applications, there is a lack of suitable NDT techniques for these components.

Shearography is an optical non-destructive testing technique that is well-suited, but not limited, to testing composite materials. The method takes advantage of the special properties of lasers, namely their high coherence length and monochromatic light, to detect flaws such as cracks, debonds and delaminations in materials.

Figure 1 below shows a typical hardware setup for shearography. The specimen is illuminated using a laser beam which is passed through a beam expander to increase its divergence. The light reflected from the specimen is then passed through a set of shearing optics, similar to that used in a Michelson interferometer. These two beams, which are laterally sheared, are recombined using the beamsplitter, and are projected onto the camera's sensor.

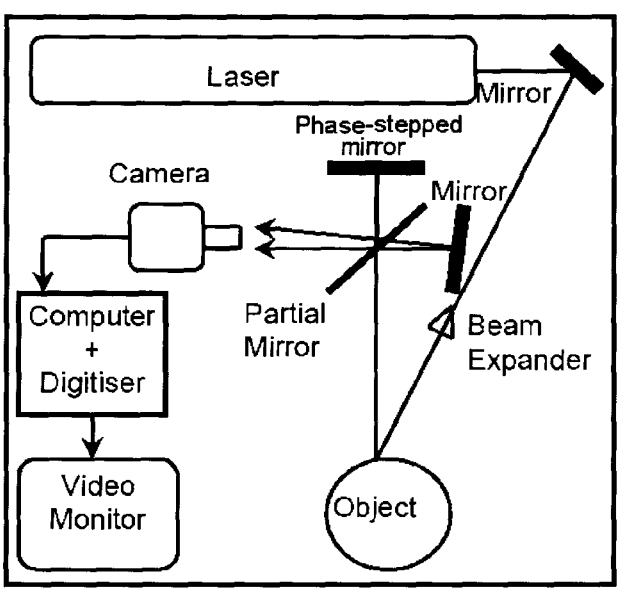


Figure 1: Typical setup for shearography

The light reflected from the specimen is known as a 'speckle pattern': a field of spatially random intensity light that is formed when coherent, monochromatic light is reflected off a rough surface. The speckle pattern, while spatially random, is also unique, and will only change if the relative positions of the laser and specimen change. Small deflections in the surface of the specimen will cause the speckle pattern to change and it is this principle is used to detect flaws in specimens, as flaws will influence the local deformation of the surface.

The resulting speckle pattern is recorded and saved for later use. Next, the specimen is stressed (usually thermally), and the new speckle pattern is compared to the old one by subtracting the two images. This process reveals strain concentrations in the way which the specimen moved or expanded during the stressing process: flaws and discontinuities are visible as concentric fringes of constructive and destructive interference, as in Figure 2 below.

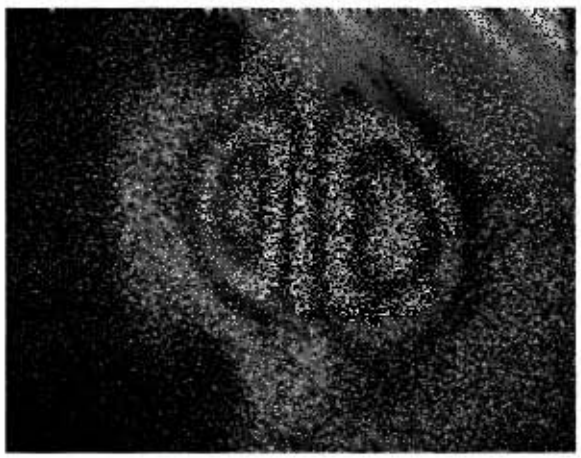


Figure 2: An example interferogram

The method described briefly above is known as conventional shearography. A relatively recent improvement to this method, known as phase-stepped shearography [1], uses an actuator to shift one of the mirrors in the shearing device by fractions of the laser's wavelength, producing a phase-modulated interferogram which is typically less noisy. These images can then be filtered to further decrease the noise level and clarify the fringes [2]. These filtered images are usually of sufficient quality to be used for phase-unwrapping, whereby the displacement of the object is recovered from the fringes, and a 3D contour plot is rendered. Figure 3 shows the progression from the acquired image, via filtering and unwrapping to a 3D plot.

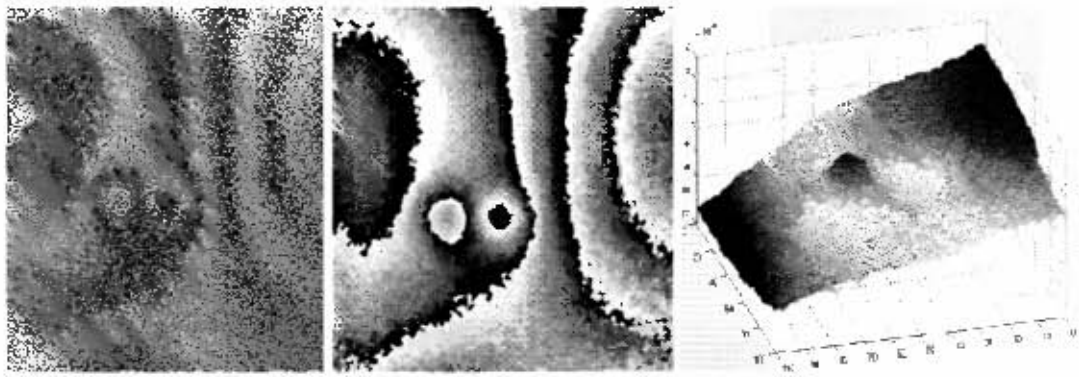


Figure 3: Phase-stepped fringes, filtered phases and 3D unwrapped phases

Shearography is an emerging technology which has found usage in several industrial settings [3], but is yet to be widely adopted, possibly as a result of its cost and relative complexity. It does however offer significant advantages over other NDT methods:

- It can be performed in real time without extensive specimen preparation.
- It is a full-field technique; comparatively large areas can be inspected at one time.
- It is optical and non-contacting, making it applicable to a wide range of materials.
- It is less sensitive to vibration than other optical techniques, such as ESPI.

UCT's NDT Laboratory has developed a portable shearography testing unit and PC application which implements phase-stepping, filtering and phase-unwrapping to detect flaws in components [4]. However, the existing testing unit uses high-end components and imaging libraries, and is difficult to install on a target machine. It is envisaged that a cost-effective and user-friendly shearography system may find increased usage in a wider variety of applications.

The main aim of this dissertation is to design, build, and test a low-cost, portable digital shearography testing unit which may be used to investigate the mainstream NDT potential of shearography. In trying to reduce the overall cost, there are several important sub-objectives:

- Use a more cost-effective digital camera, with a more generic PC interface.
- Eliminate software dependence on costly imaging libraries.
- Enable operation from a laptop (eliminate dependence on PCI cards).
- Allow "Plug-and-Play" operation of the testing unit.

Procedure

The new shearography system was completely redesigned, including mechanical construction and software. The shearography head was designed using *ProEngineer* CAD package and built by the Mechanical Engineering Workshop (see figure 4). The box is constructed from bent sheet-metal instead of machined from a solid aluminum block in order to reduce the cost.

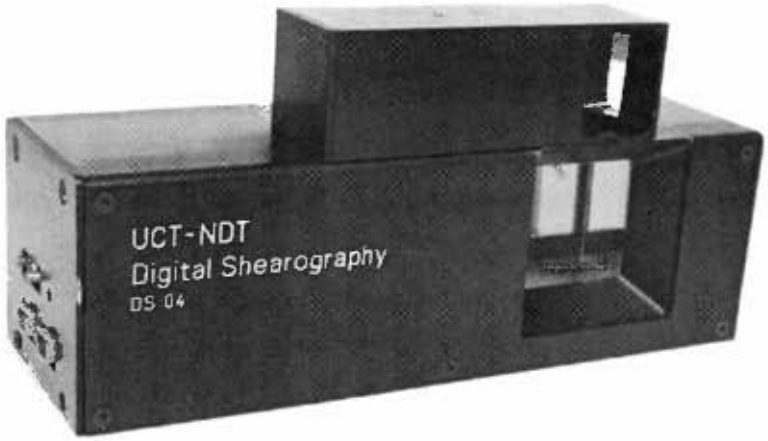


Figure 4: Front view of shearography head

The new CMOS camera uses a FireWire port, which is more common and more cost-effective than the existing system's Camera Link camera. The use of a FireWire camera also eliminates the need for a framegrabber PCI card. The FireWire camera used offers flexibility in the form of external communication ports, allowing control of peripheral devices through the same camera cable. This feature was used to communicate with the piezoelectric actuator through a custom designed peripheral controller board. This controller board also provides regulated power to the laser, eliminating the need for additional power cables.

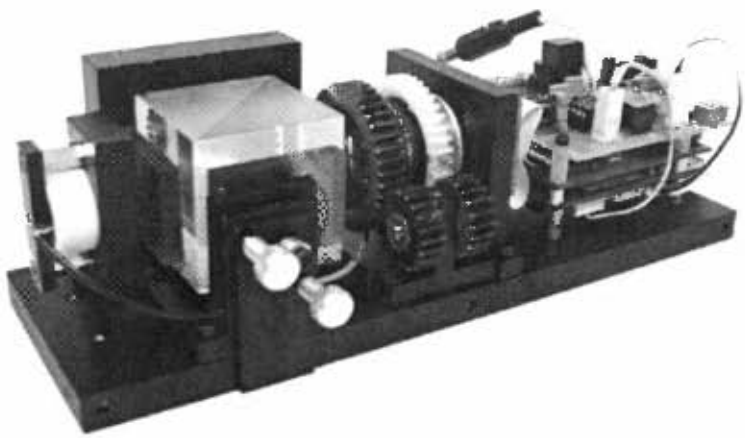


Figure 5: Internal assembly of shearography head

Solution

The main objective, namely reducing the total cost, was achieved: the new unit is 52% more cost-effective than the existing one. All of the sub-objectives were also met, while in addition offering improved performance in many areas;

- The system is capable of operating from a desktop PC via one FireWire cable (no target PC modifications are necessary). Laptop usage requires that auxiliary 12V power is supplied.

- A custom Windows Installer package deploys the new *Inspector 2.0* application, installing drivers, shortcuts and help files.

- The *Inspector 2.0* application allows in-situ calibration of the piezoelectric actuator, eliminating the need for external calibration.

- The new phase-stepping algorithms implemented in *Inspector 2.0* take advantage of two-dimensional *arctan* lookup tables to achieve display rates of 25fps in phase mode, a large improvement over the old methods. This makes viewing of highly transient behavior possible, and may be used to detect flaws in more heat conductive materials in future.

Validation tests were performed on a section of Oryx helicopter rotor blade which had artificial flaws created by drilling specified depths into the rear surface. The specimen was stressed thermally, using an infra-red lamp, as illustrated in figure 6 below.

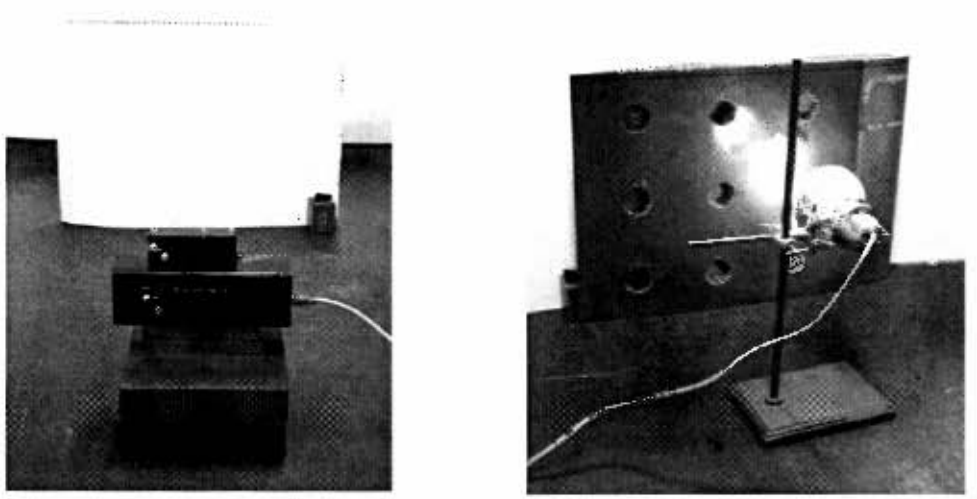


Figure 6: Testing setup – arrangement (left) and specimen stressing (right)

All of the flaws were successfully detected by the new system using both simple intensity-based shearography as well as phase-stepped shearography. The images produced were of good quality, allowing them to be phase-unwrapped using the *MATLAB*-compiled phase unwrapping application written as part of another project [5]. These unwrapped images offer the user the opportunity to visualize the surface displacement in 3D.

Figures 7 to 9 were obtained by performing phase-stepped shearography on the entire specimen (in figure 6), from a distance of 1.5m.

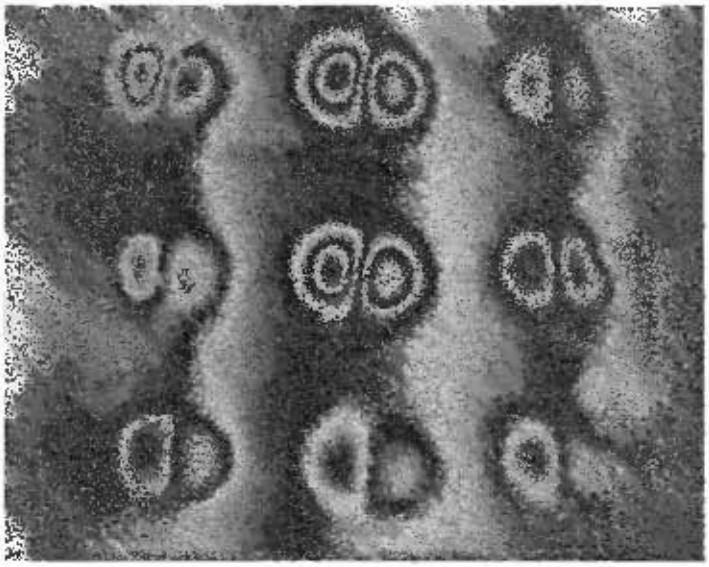


Figure 7: Phase-stepped shearography image, acquired using the new system

One can easily see the nine flaws as double lobed anomalies. This phase-stepped image was then filtered and unwrapped to obtain the gradient surface.

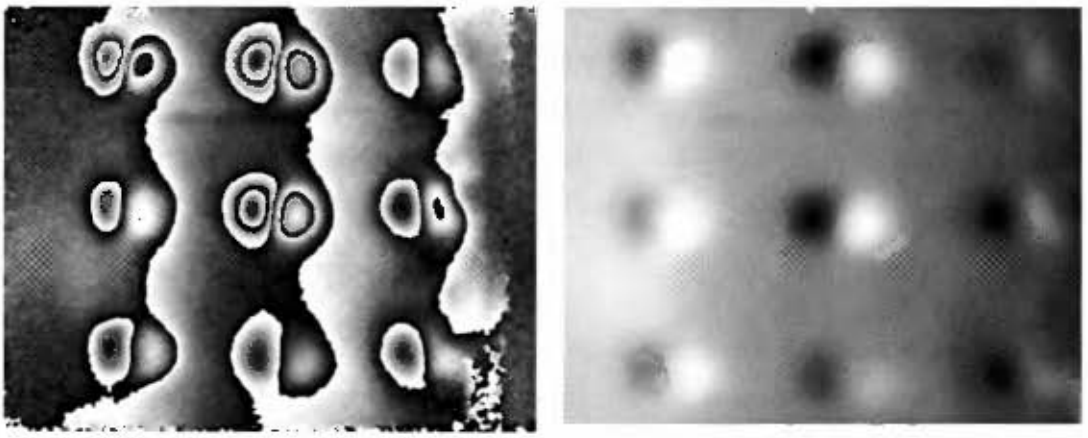


Figure 8: Filtered phases (left) and unwrapped phases (right)

Once the phases have been 'unwrapped', the resulting image does not represent the surface of the specimen but rather the surface's first derivative, its gradient (as in figure 9). This surface may then be integrated to find the true surface displacement that occurred during the stressing process.

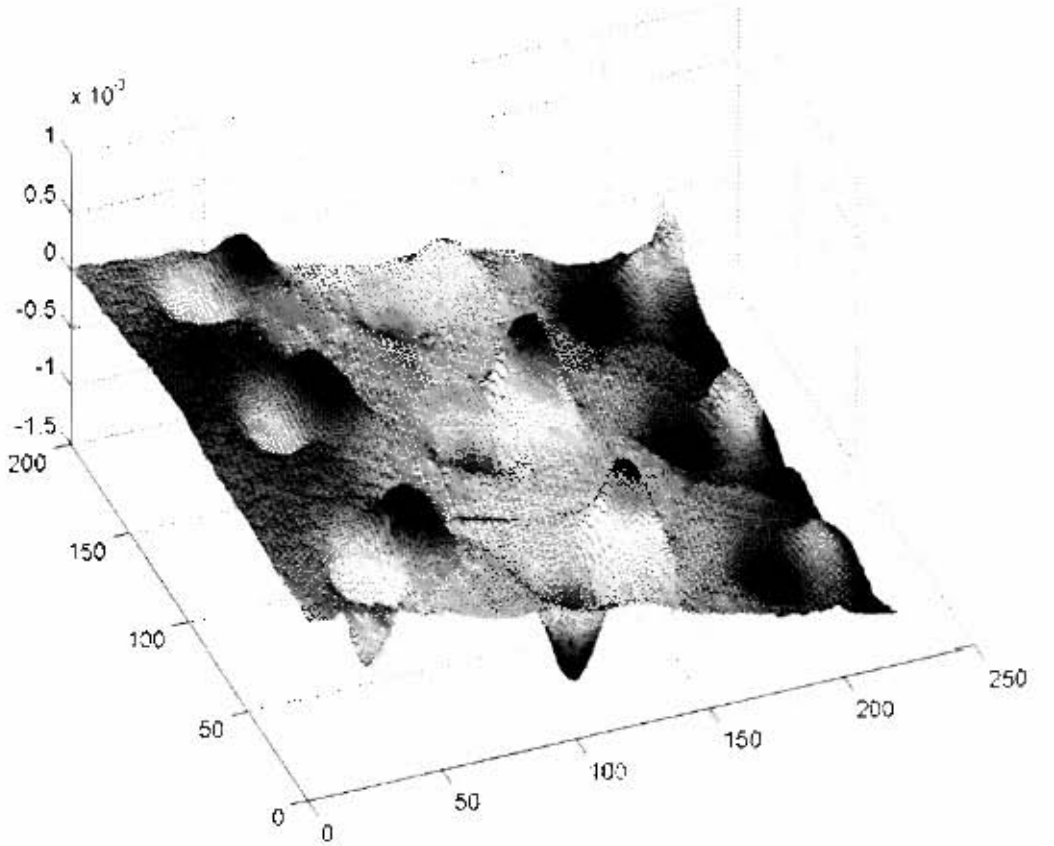


Figure 9: Gradient surface

Some additional features were added that made the acquisition of these particular example images possible; for example, the new *Inspector 2.0* application gives the user direct access to some on-board camera settings, which allows for the camera's exposure time to be manually extended. This is particularly useful in low-light conditions, where even with an open aperture, the acquired images are too dark to analyse.

Finally, the system was successfully tested on a laptop with the use of an external 12V power supply for the shearography head. As the shearography head does not draw its power from the laptop, the laptop was able to operate using only its battery for the duration of the tests.

Conclusions & Recommendations

The project is regarded as a success. A new shearography head was designed, manufactured and successfully tested. The new design achieved a cost reduction of over 50% by using more cost-effective parts, generic software, and a higher level of OEM integration. It is envisaged that the new design will be simpler to install and use as a result of the custom installer package, and the simple 'plug-and-play' operation of the unit. In addition, the capability to operate from a laptop is an important advantage as it increases the mobility of shearography.

While the project is regarded as a success, there are several areas in which the current design may be improved:

1. It is recommended that the mechanical design return to the original concept of a solid box instead of using sheet-metal, owing to its ease of assembly, dust-proofing and strength. These solid parts may also be injection-moulded.

2. One of the main limitations of the controller board is that is gives a limited piezo displacement range. Using longer travel PZTs or modifying the controller board's design to include a wider dynamic range would solve this problem.

3. Investigate the influence of using a CCD camera instead of the CMOS camera. It is expected that this is the cause for the slightly lower quality of interferograms produced, when compared with the existing system.

In addition, the current design lends itself to further expansion using the same FireWire cable relaying communication with peripheral devices over the camera's COM port. There are many opportunities, for example in automated inspection: using remote adjustment of lens and aperture with servo motors and remote head panning and tilting with stepper motors.

TABLE OF CONTENTS

LIST OF APPENDICES

LIST OF FIGURES

LIST OF TABLES

GLOSSARY OF TERMS & ACRONYMS

API	Application Programming Interface. This is the 'front-end' of a library or application that has been designed for other applications to interface with.
BNC	A common type of connector used in RF applications.
CCD	Charge-Coupled Device
CMOS	Complimentary Metal-Oxide Semiconductor
COM	Generic term used to describe a serial interface
DAC	Digital-to-Analogue Converter
Daisy-Chain	Serial connection of multiple devices on the same cable.
DCAM	1394-based Digital Camera Specification – a standard communication protocol for FireWire cameras. The standard can be purchased from the 1394 Trade Association [6].
DOF	Degrees-Of-Freedom refers to the number of axes or planes about which an object can move.
FireWire	Also known as IEEE1394, FireWire is a serial communications protocol similar to USB.
Framegrabber	PC expansion card that is used to digitize video data into a matrix of pixels. Framegrabbers are required for Camera Link cameras
I^2C	Pronounced "Eye-Squared-C", is a Philips-developed serial communication protocol that uses two lines, known as *data* and *clock*. It is a multi-device and –master protocol that is used to communicate small amounts of data between IC chips [7].
IC	Integrated Circuit
IIDC	Instrumentation and Industrial Control Working Group – this group is responsible for managing the DCAM specification (see DCAM).
GFRP	Glass-Fibre Reinforced Plastic
GPIO	General Purpose Input/Output. These are pins offered by some

	cameras for controlling auxiliary equipment.
LabVIEW	A popular graphical programming language for measurement and instrumentation, by National Instruments [8].
LUT	Look-Up Table
MFC	Microsoft Foundation Classes. These classes come standard with *Visual C++* 6.0, and the MFC template gives *Visual C++* applications the standard Windows 'look-and-feel'.
MIL	Matrox Imaging Library [9]. This is the imaging library used by the existing *Inspector* application.
NDT	Non-Destructive Testing
OCR	Optical Character Recognition
OEM	Original Equipment Manufacturer. This refers to a company that uses other company's products in a product of its own. Similar to the concept of a Value-Added Reseller (VAR).
PCB	Printed Circuit Board
PCI	Peripheral Component Interface – extension card that plugs into a PC motherboard, usually to control external components or give the PC extended functionality.
PZT	Commonly refers to the material used in piezoelectric actuators, but may also be used to refer to the actuator itself.
SDK	Software Development Kit
SMPS	Switched-Mode Power Supply (used interchangeably with DC-DC converter)
TECRL	Thermoelectrically-Cooled Red Laser
USB	Universal Serial Bus, a common serial communication protocol

LIST OF SYMBOLS

$\Phi_a(x, y)$	Phase distribution before stressing
$\Phi_b(x, y)$	Phase distribution after stressing
λ	Wavelength of the laser
S	Magnitude of image shear
d	Out-of-plane displacement
x	In-plane distance, such that d(d)/dx is the gradient of the displaced surface.
α	Angle between the direction of the object displacement and camera-viewing angle.
β	Angle between the direction of the object displacement and object beam

1. INTRODUCTION

1.1 Background

Non-destructive testing (NDT) is regarded as a critical part of the manufacturing, validation and maintenance testing cycles in many different industries [3].

As advancing technology offers designers new, strong and lightweight composite materials, so the technology for their validation must also adapt; traditional ferrous-based non-destructive testing methods such as magnetic induction, eddy currents as well as acoustics are not suitable for modern composites.

There is a need for NDT techniques that are reliable and cost-effective. Currently, there is no single NDT technique that can be used in all situations; most techniques are deemed to be complimentary [10], as different techniques are suited to certain materials and conditions. Ideally, one should use a variety of techniques on components to ensure reliable and accurate results, however this approach has both time and cost implications; in order for this approach to be feasible, the methods used should be both quick and cost-effective.

Shearography is an optical NDT technique that is capable of improving the NDT capabilities in many industries, owing to its numerous advantages over other methods, such as real-time and full-field inspection. Shearography is receiving increased attention for detecting flaws in safety-critical aerospace components; e.g. being incorporated into the production lines of aerospace components [11] as well as being included in the maintenance testing routines for various components.

The University of Cape Town's NDT Laboratory has been involved in the research and development of optical interference techniques for many years, culminating in the development of portable ESPI (Electronic Speckle Pattern Interferometry) and shearography testing units [4] (figure 10). The shearography prototype, currently under development, incorporates a laser diode illumination source and capabilities for phase-stepped shearography using a piezoelectric actuator. The picture on the following page shows the existing shearography head.

Figure 10: Existing portable shearography head on a tripod stand

The utility of shearography has been well documented [12],[13],[14], and with recent advancements made by the NDT Lab [1], the new portable prototypes produced by the NDT lab are receiving interest from industry. In addition, the number of commercially available shearography systems is increasing rapidly [15],[16],[17]. However, the cost of shearography in the past has been high, as it requires high quality digital cameras, piezoelectric actuators and controllers. The development of successful portable shearography units has offered practical insight into the method and hardware-related requirements of shearography. This practical insight, coupled with advancements in digital camera and piezoelectric technology have made the possibility of a low cost shearography system possible.

It is envisaged that a low-cost shearography system may find increased usage in a wider range of applications than what was previously the case, opening new avenues of development where, in the past, the cost associated with shearography was considered prohibitive. By adopting and learning from the existing portable shearography unit, the proposed new low-cost unit should make shearography more cost-effective and user-friendly while still providing useful results.

1.2 Problem clarification

The main aim of this dissertation is to design, build and test a low-cost, portable shearography testing unit. Besides attempting to reduce the cost of the unit, the new design should also attempt to overcome some of the limitations of the previous design solution, thus it should be:

- Easier to use
- Simpler to install
- Capable of stand-alone operation (i.e. controlled from a laptop instead of a PC)

Thus a clean-slate approach is to be taken; the system should be fully redesigned, including:

- Physical layout and construction
- Component selection and custom electronic design, if necessary
- All software and imaging libraries, if necessary

The new low-cost shearography system should be tested and validated in a similar manner to the methods used to validate the existing shearography system; by performing tests on a high-value aerospace component with artificial flaws introduced.

1.3 Design requirements

The design requirements for the new system were established using the existing shearography system as a reference point; the new system should match or exceed its performance and features. Thus the requirements are:

1. Design a new shearography head using cost-effective off-the-shelf components where possible.
   1.1. The shearography head should have capabilities for phase-stepped shearography.
   1.2. Develop new shearing optics or adopt the optics layout of the existing shearography system.
   1.3. Design the shearography head such that no host PC modifications are necessary. This will include:
      1.3.1. Use a camera-PC interface that does not require a framegrabber card.
      1.3.2. Find a method for controlling the PZT actuator that does not require a digital-to-analogue converter to be fitted to the PC.
   1.4. Enable panning and tilting of the shear-mirror.
   1.5. Enable lens focus and aperture adjustment.
   1.6. Enable panning and tilting of the laser beam to match the field-of-view of the camera.
2. Adapt or rewrite the existing software application (called *Inspector*) that has been written to control the shearography head.
   2.1. The new application should be able to control both the new and old shearography heads with minimal changes required to swap between the two.
   2.2. It should implement image processing algorithms for simple intensity-based and phase-stepped shearography.
   2.3. It should be able to perform brightness and contrast adjustment of images.
   2.4. It should be able to filter phase-stepped images, and unwrap them using the phase-unwrapping plug-in used as part of another project [5].
   2.5. It should be able to open and save images.
   2.6. It should eliminate, where possible, dependencies on 3rd party software packages which require licensing.

1.4 Plan of development

This report starts with a literature review of several different fields related to shearography. The theory of shearography is discussed in detail, followed by a review of the hardware and software implementations commonly used in shearography systems. A survey is conducted into the commercially available shearography systems, and a review of UCT's existing shearography system is also performed. This information can be used as a point of reference for the performance and specifications of the new low-cost shearography unit.

Section 3 discusses all of the different concepts and potential components for use in the new design. This section includes discussions on the layout of the shearography head, the type of camera to be used, the selection of piezoelectric actuator, the piezo controller design as well as all software-related concepts.

Section 4 details the resolution of the conceptual design and the implementation of the final solution, including mechanical construction, the piezoelectric actuator's controller and the custom software that was written for controlling the shearography head. The breakdown of the overall cost and list of specifications are also included in this section.

Section 5 describes the validation procedure that was carried out; a series of tests were performed on an aerospace component. The results of these tests and the experimental procedure followed are outlined here.

Section 6 contains a discussion on some of the practical limitations of the design solution, as experienced during the testing procedure.

Finally, conclusions are drawn and recommendations are made.

Appendix A offers an operation manual for software and hardware, including troubleshooting information, while Appendix B contains the manufacturing drawings for the new shearography head.

2. LITERATURE REVIEW

2.1 Theory of shearography

Shearography is an optical non-destructive testing technique that uses lasers to perform full-field inspections of surface deformation. It is used to detect flaws such as cracks and delaminations in materials by inspecting small out-of-plane surface displacements. Due to a lack of other viable testing methods and their suitability, this technique is being used increasingly on composite materials.

2.1.1 Speckle patterns

Shearography makes use of a single mode, coherent light sources to illuminate the specimen. When a light source with these properties illuminates a specimen with a rough surface, a speckle pattern is observed. A speckle pattern is an area where the spatial intensity of reflected light appears to vary randomly, owing to the fact that the reflected light travels different fractions of its own wavelength off the rough surface. Figure 11 shows an example of a speckle pattern

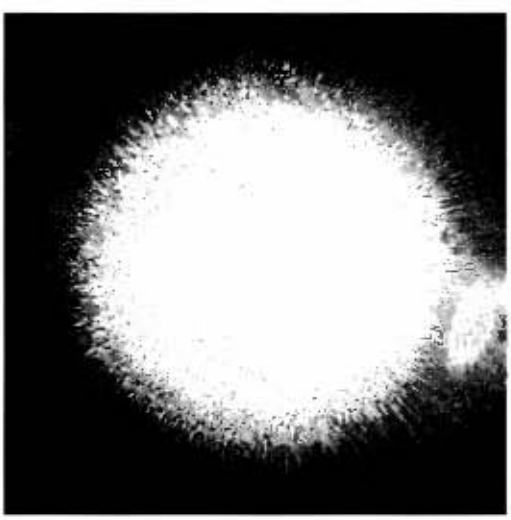


Figure 11: Example of a speckle pattern from coherent laser light

The speckle pattern contains useful information about the surface of the specimen; if the optical length that the light travels changes, the pattern will change. This is the basis for many optical interference techniques.

2.1.2 Hardware setup

Most often a laser is used to illuminate the specimen, usually with the use of a beam expander. The light reflected from the specimen is passed through a set of Michelson-type shearing optics. A beamsplitter reflects 50% of incident light, and transmits the other 50%. The beamsplitter is used to both split and recombine the sheared image.

The shearing optics superimpose two laterally sheared images onto the sensor of the camera. This superposition of speckle patterns means that the light intensity of any particular pixel is the result of the complex addition (due to constructive and destructive interference) of the two speckle patterns.

Figure 12 illustrates the how the reflected beam is split, sheared, recombined and then projected onto a camera sensor.

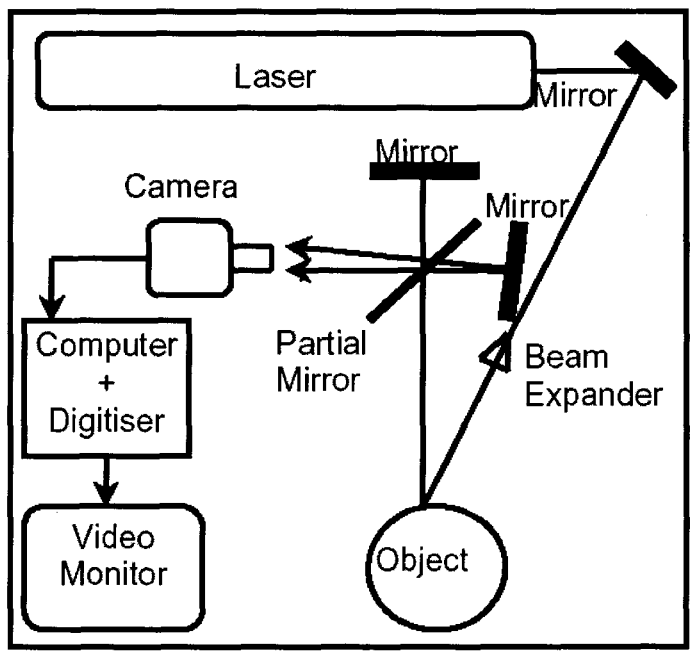


Figure 12: Typical hardware setup for shearography

The fact that both interfering beams are derived from the reflected beam makes this technique less sensitive to vibration than other similar techniques, such as ESPI. This is one of the main reasons that shearography is currently the preferred line of development in portable interferometry systems.

2.1.3 Deriving intensity fringes

Defects in a material influence the local deformation when stressed, and thus internal flaws can be detected optically by surface inspection as displacement anomalies.

In order to measure the displacement during stressing, an image of the specimen in an unstressed state is acquired, and compared to an image of the same specimen after it has been stressed (using mechanical, pressure or thermal methods). The subtraction of these two images will reveal how the surface displaced during the stressing process.

The equations given here were taken from "*Holographic and Speckle Interferometry*" by R. Jones and C. Wykes [18]. The intensity of a given pixel on the image sensor can be described mathematically by the following equation:

$$I_a = 2A^2(1+\cos\theta x+\delta x-\theta x))$$

Where A is the amplitude of the complex addition of the two overlapping wave-fronts, dx is the shear magnitude and θx is a term that accounts for the object beam and surface orientation.

Stressing the specimen introduces a phase shift in the reflected light, owing to the surface deformation. This phase shift is described by the term $\Delta\Phi$ in the following equation:

$$I_b = 2A^2(1+\cos\theta x+\delta x-\theta x)+\Delta\Phi)$$

When these two images are 'subtracted' from each other, the remaining interference fringes represent the displacement due to the stressing of the specimen, and are used to identify flaws.

$$I_r = I_a - I_b = 4A^2\sin\left((\theta x+\delta d-\theta x)+\frac{\Delta\Phi}{2}\right)\sin\frac{\Delta\Phi}{2}$$

Flaws will appear as anomalies in the fringe pattern produced in shear images, as in figure 13. Fringes, or whiter areas, are formed where there is decorrelation between

the phases of light, i.e. $\Delta\Phi = \Pi(2n+1)$ and darker correspond to correlations, i.e. $\Delta\Phi = 2n\Pi$ where $n = 1,2,3....$

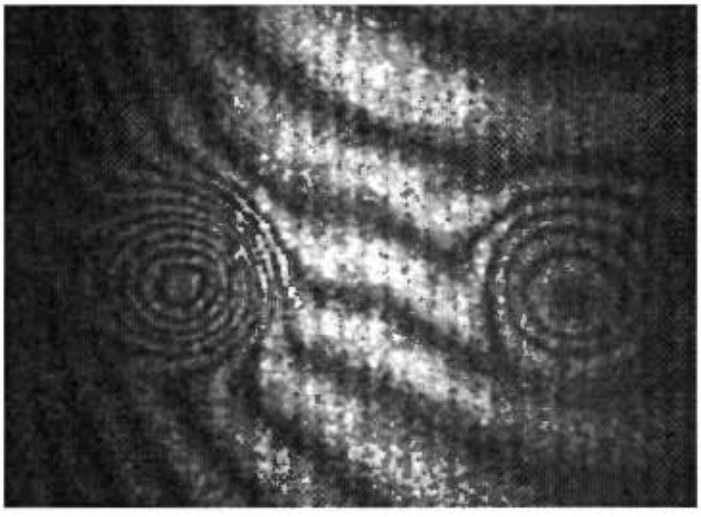


Figure 13: Example of interference fringes showing two flaws

The fringes can be represented using the following equation:

$$\Delta\Phi = \frac{4\Pi}{\lambda}\left(\frac{\delta l}{\delta x}\right)S$$

Where $\Delta\Phi$ is the correlation phase, $\frac{\delta l}{\delta x}$ is the rate of surface displacement, S is the magnitude of image shear, and λ is the wavelength of the laser used. Solving for the fringes ($\Delta\Phi = 2n\Pi$ where $n = 1,2,3...$), we get the equation:

$$\frac{\delta l}{\delta x} = \frac{n\lambda}{2S}$$

Where n is the number of fringes. This equation shows that the surface gradient is proportional to the number of fringes, i.e. the greater the number of fringes surrounding a particular flaw, the greater the severity of the flaw. Also, increasing the shear magnitude for a given displacement gradient will increase the number of fringes, i.e. increase the sensitivity of the device.

2.1.4 Phase-stepping and filtering

Intensity fringes reveal the magnitude of changes in the surface's displacement and where they are taking place. These fringes do not however, describe whether the displacements are in or out of the viewing plane. In order to determine this, phase-stepping is implemented.

By progressively altering the distance travelled by one of the interference beams by fractions of the laser's wavelength, one can determine whether the changes in the interference patterns are caused by in- or out-of-plane displacements. Physically, this requires that the position of one of the reflecting mirrors is shifted by fractions of the laser's wavelength. The fraction of the wavelength to be shifted, $^{2\pi}/_{N}$ where N is an integer, depends on the data-reduction algorithm used [19]. Two commonly used methods are commonly referred to as the three-bucket (N = 3) and four bucket (N = 4) methods. Using these algorithms, multiple images are recorded for each state (unstressed and stressed), each with a different phase step. The following equations are taken for the four bucket system, where N = 4.

$$I_i = I_B(x,y) + I_{MP}(x,y)\cos\left(\Phi(x,y) + i.\frac{\Pi}{2}\right) \text{ where } i = 1,2,3... \qquad [20]$$

Where $I_B(x,y)$ is the background intensity, $\Phi(x,y)$ is the phase difference between the two overlapping beams, and $I_{MP}(x,y)$ is the modulation intensity. These four images are then combined mathematically, as shown below, to derive a single interferogram. This interferogram eliminates the background noise ($I_B(x,y)$) and thereby produces better quality images.

$$\Phi(x,y) = \arctan\left(\frac{I_3(x,y) - I_1(x,y)}{I_4(x,y) - I_2(x,y)}\right)$$

Finally, this 'composite' interferogram is subtracted from the first interferogram to get the phase image:

$\beta(x,y) = \Phi_b(x,y) - \Phi_a(x,y)$

$\Phi_b(x,y)$ = phase distribution after stressing

$\Phi_a(x,y)$ = phase distribution before stressing

The difference between in- and out-of-plane displacements can now clearly be seen owing to the different intensity gradients around the flaw, figure 14.

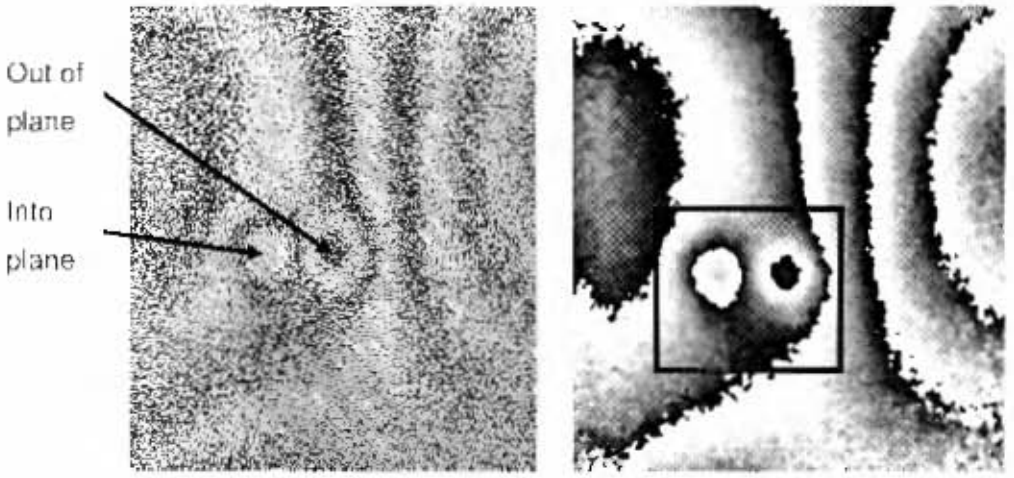


Figure 14: Fringes types shown in phase stepped and filtered images (left)

A sinusoidal filter can now be applied to clean the image and enhance the clarity of the phase steps. Another advantage of using phase-stepping and filtering is that background noise is reduced, yielding better quality images, in figure 15.

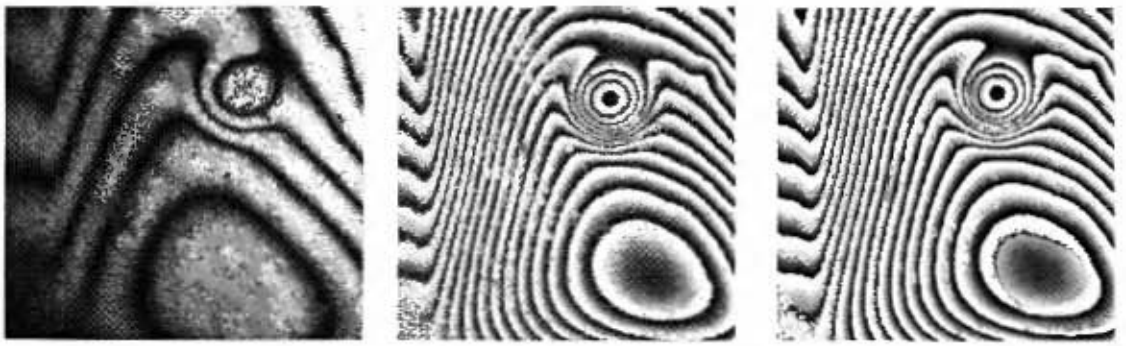


Figure 15: ESPI results: i. Intensity fringes, ii. Phase fringes, iii. Filtered fringes

The filtered images represent the surface profile in the case of ESPI images and the surface's gradient in the case of shearography images. The discontinuities around the fringes are where the phase stepping and filtering have introduced 2π discontinuities.

The fringes are analogous to a contour map: fringes connect points of equal gradient, with each new fringe being a wavelength's displacement above or below the previous fringe.

2.1.5 Phase-unwrapping, or demodulation

Phase-unwrapping is the process of reconstruction of the surface (or gradient) displacement by resolving the 2π discontinuities in the filtered phase images. A previous UCT project [5] achieved this by identifying the fringes using Canny edge detection [21], classifying the fringes based on the pixels surrounding the edge, and finally filling in the image regions using a flood-fill operation.

The method is successful when clear, continuous fringes can be obtained. This is not always the case however, and other methods should also be investigated; novel methods have been proposed which seem to offer superior results [22],[23]. Figures 16 to 20, show how a surface plot is derived from a filtered phase diagram:

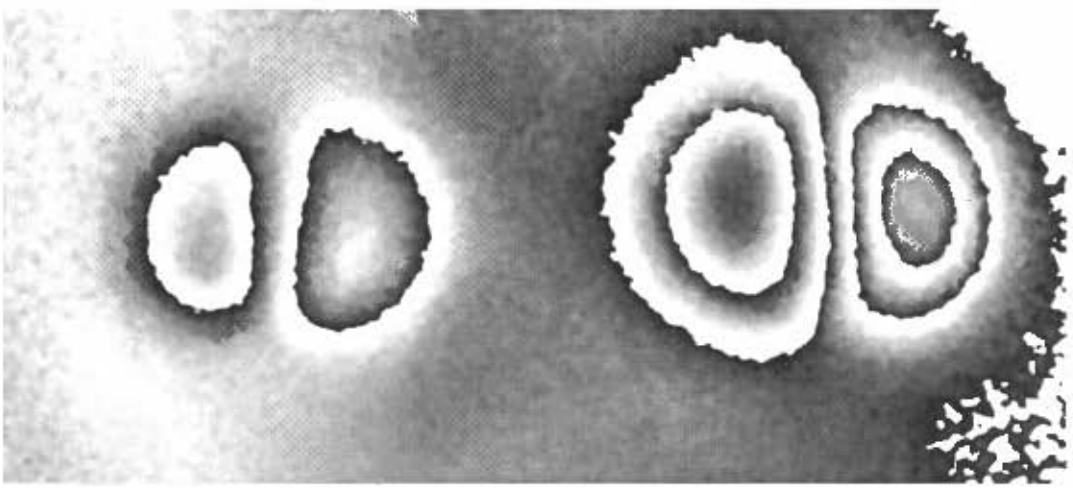


Figure 16: Filtered phase image

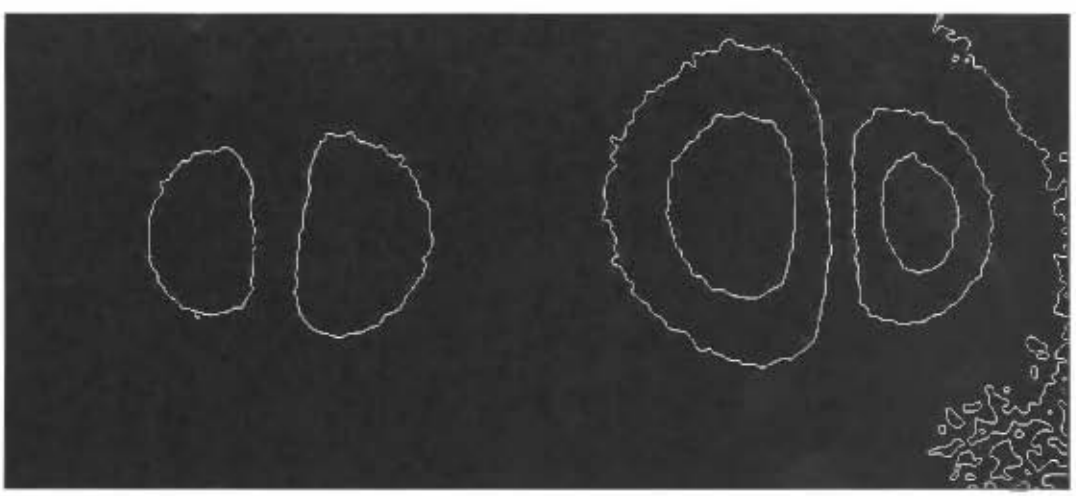


Figure 17: Edge detection using a Canny edge detector

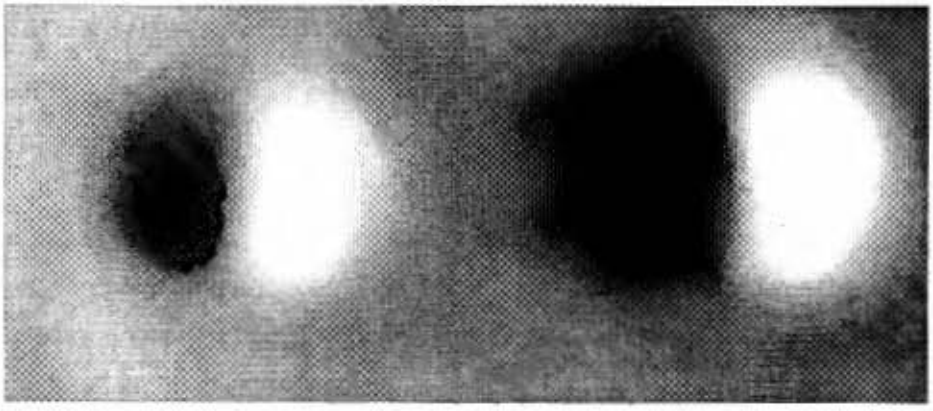


Figure 18: Unwrapped image

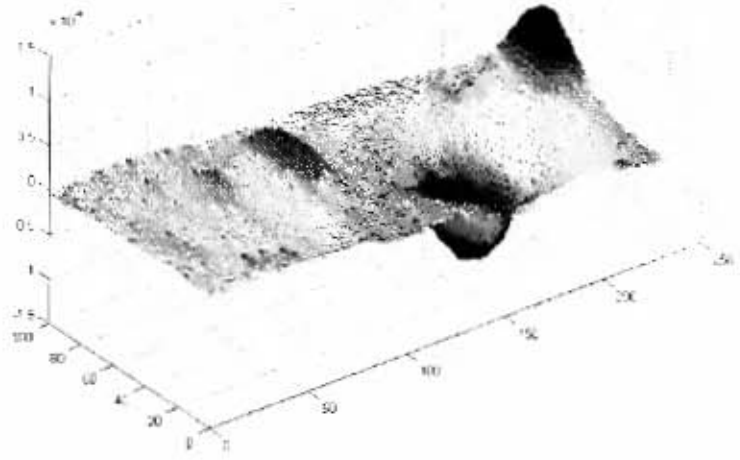


Figure 19: Gradient plot

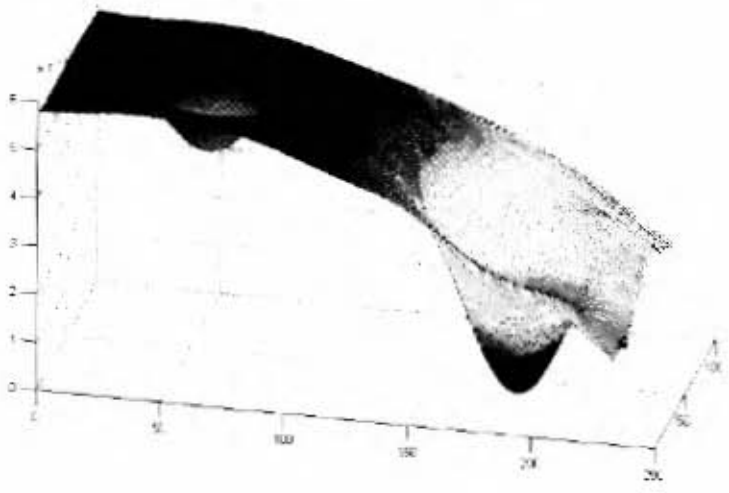


Figure 20: Surface plot

2.2 Hardware requirements of shearography

Physical implementations of this technique must take into account various different practical considerations which make available different hardware configuration options. There are many ways to implement shearography, which is evidenced by the large number of commercially available shearography systems, each using different types of hardware components. This section discusses the hardware requirements for implementing shearography, with a particular emphasis on configurations that could be used in the low-cost system.

The basic hardware requirements for a phase-stepping digital shearography system are as follows:

- An illumination source that is both monochromatic and coherent.
- A recording and storage medium for capturing and storing speckle patterns.
- A set of shearing optics used in the superposition of speckle patterns.
- A means of phase-stepping one of the shear mirrors.

An extensive survey of the types of suitable hardware was conducted. The cost and performance of each of the surveyed components was analyzed to determine the component best suited to the application. A large amount of information, gleaned from specification documents, user manuals and email correspondence is summarized here.

2.2.1 Illumination source

Lasers are used for illumination in most shearography applications due to their high coherence length (the distance over which laser light can travel before becoming incoherent) and high level of monochromicity. There are many different types of lasers that can be used in interferometry applications, but in portable applications, thermoelectrically-cooled laser diodes are favoured owing to their lower power consumption, small size and relatively low cost.

The quality, chromaticity and intensity of the laser illumination determine many of the shearography system's specifications – the maximum inspection area, maximum working distance and affect the quality and clarity of fringes developed. The higher the power of the laser, the more versatile the system can become in terms of

performing inspections. However, the power delivered by laser diodes is limited as they require means by which to disperse the heat they generate.

Much of the current work at the University's NDT laboratory is focused on combining laser diodes in arrays and ensuring their stability [24].

2.2.2 Recording medium

Before the advent of digital cameras, photographic slides were used to capture and store speckle patterns and then superimpose sheared speckle patterns to obtain shearography images. This was a time-consuming process, and has since been abandoned in favour of using digital cameras. Digital cameras, coupled with a PC or laptop can be used to capture, store and process images in a similar process to using photographic slides, with the advantage that it is faster, requiring less setup time and results are stored digitally for analysis and further processing.

Digital cameras for machine vision can be broadly classified either by their PC interface or sensor type (CCD or CMOS). USB and FireWire cameras can plug directly into generic PC ports. Their transfer rates are slower than those achieved by Camera Link and Gig-E connections, which require framegrabber PCI boards. Most cameras offer extended features such as onboard LUTs (lookup tables), GPIO (general purpose input/output) pins and advanced image capture triggers.

2.2.2.1 FireWire

FireWire, also known as IEEE1394 and i.Link, is a serial bus interface standard for high speed communications, especially between digital video devices [25]. FireWire 400, or IEEE1394a, is the widely used version of this interface and supports transfer speeds of 400 Mbit/s. FireWire 800, or IEEE1394b, achieves 800Mbit/s but is not yet widely supported. Unlike USB2.0, the quoted transfer rates are realistically achievable as data transfer does not consume host PC's resources.

FireWire ports are common in laptops and many PCs. FireWire PCI boards are widely available and inexpensive. All Sony and Apple laptops come with standard with FireWire ports. The ports come in two different versions, 4-pin and 6-pin. Most laptops have 4-pin ports while desktops generally have 6-pin ports. This is because the extra 2 pins are power pins and exceed the power capabilities of laptops. The

standard allows for a voltage of up to 30V, with a maximum power of 45W. However, unlike USB, the voltage level is not specified in the standard and one may encounter anything from 8V to 30V between the power pins, and typically be able to draw 7 to 8W.

The length of FireWire cables is limited to 4.5 metres, but via other means can be extended to a maximum of 72 metres. FireWire is not as popular as its competing standard, USB2.0; one of the main reasons for this is the greater cost of FireWire ports and cables [26].

Most FireWire cameras support the DCAM (1394-based Digital Camera Specification) standard. The term 'DCAM' is used interchangeably with IIDC (Instrumentation and Industrial Control Working Group), the group responsible for managing the standard. This standard is supported by most imaging software packages. However, while the DCAM specification controls video feed, many of these cameras have their own advanced features which lie outside the specification. As such, camera manufacturers supply their own custom drivers and SDK's which they claim are DCAM compliant, but this compliance only refers to the video stream. If one wishes to use the extended camera features, the manufacturer's software must be used.

There is a wide range of high-quality FireWire cameras available, as FireWire was originally designed for digital video; AVT, Basler, PixeLINK and Sony all have ranges of FireWire cameras. Most FireWire cameras use the FireWire 1394a standard, but some manufacturers have recently launched ranges of 1394b cameras, which support the new FireWire protocol, offering higher transfer rates and using 9-pin connectors.

All of the 1394a cameras have 6-pin ports, as camera power is supplied over the same cable. Where these devices must be connected to laptops (or other 4-pin ports) mains-powered FireWire hubs or 'laptop adaptor kits' are usually used [26]. The FireWire specification allows daisy-chaining of devices, and most FireWire cameras support this feature by having two onboard FireWire ports. A maximum of 63 devices may be connected to one host port. These devices must share the same processing time and as such lower frame-rates should be expected.

2.2.2.2 USB2.0

USB (Universal Serial Bus) is the de facto standard for many peripheral devices. The standard claims to support transfer speeds of up to 480Mbit/s, but in practice this is rarely achieved. The USB3.0 specification, due for release in 2008, will offer transfer rates of 4800Mbit/s using a fibre optic link. The maximum length of a USB cable is 5m.

The power supply over USB is somewhat limiting – the specification allows a maximum of 500mA at 5V (2.5W). Some USB devices draw more than this limit, but are then classified as non-standard devices. PoweredUSB [27] was developed to overcome this limitation. It comprises two plugs, a standard USB connection and another mains power plug, capable of supplying additional power at 5V, 12V or 24V.

USB cameras are ubiquitous, but generally only in low-quality imaging applications such as webcams. There are, however, some manufacturers of higher quality USB cameras, such as PixeLINK, Lumenera and Edmund Optics.

These manufacturers claim that USB cameras are better suited for laptop applications than FireWire cameras are, owing to the fact that laptop FireWire ports are 4-pin whereas the cameras are 6-pin. There is no distinction amongst USB connectors, so a USB camera can operate directly from a laptop, albeit at the limit of its power supply. However, USB will generally achieve slower frame rates than corresponding FireWire cameras.

The advantage to using USB cameras over FireWire is that they are generally slightly more cost-effective, and USB ports are more common.

Software imaging applications have been slow to provide support for USB cameras; MIL (Matrox Imaging Library) does not have any support for USB cameras. Similarly to FireWire cameras, these cameras often offer extended features that are only accessible through the manufacturer's software.

2.2.2.3 Camera Link

Camera Link has been the industry standard for high-end imaging applications, but is likely to be supplanted by Gig-E Vision cameras. The Camera Link specification allows transfer rates of up to 2.04 Gbit/s. Neither Camera Link nor Gig-E Vision provides power over their cables and both require framegrabber cards.

2.2.2.4 Gig-E Vision

This interface is another high-end imaging interface which is being developed by a host of large machine vision companies and is based on the Gigabit Ethernet protocol. It aims to overcome some of the limitations of the Camera Link specification by using cost-effective cables which can be up to 100m long. It supports transfer rates of 1000Mbit/s over standard Ethernet cables (CAT5e or CAT6).

2.2.2.5 CMOS vs. CCD

CMOS and CCD are two different types of image sensor used in digital cameras. They have different methods of operation, and are manufactured using different processes. Historically, CCD cameras have been preferred in high-end applications because of their higher image quality and clarity. Recently however, manufacturing advances have been made in CMOS sensor technology that has seen them finding increased usage. The pixel wells in both sensors convert light to electrons, but the way in which they digitize the signal differs; CCD sensors have one analogue-to-digital converter for all the pixels, and the electron charge from each pixel is carried across the CCD to one corner of the sensor to be digitized by the ADC. This means that the full sensor area is available for pixel light capture. CMOS sensors on the other hand have transistors situated next to every pixel for converting the pixel electrons to a digital signal. This enables faster processing, but limits the amount of light reaching every pixel, as space is also taken up in the sensor area by all the conversion transistors. Digital noise from these transistors affects the pixel light capture, which results in images from CMOS sensors being more noisy than CCD. The advantage to using CMOS sensors instead of CCDs is that CMOS sensors are more cost-effective and consume less power, which makes them ideal for low-cost camera interfaces such as USB and FireWire, where power is supplied over the interface bus instead of over a separate cable, and as such must be kept to a minimum.

2.2.3 Shearing optics

The process used by shearography systems to split the light reflected from the specimen into two separate beams and recombine them onto the camera's sensor differs significantly between different implementations. Often the configuration of the shearing optics is proprietary. One of the most common configurations uses a modified Michelson interferometer [28], in which one mirror of the interferometer is tilted slightly to introduce a small shear into the recombined image. This is then known as a shearing Michelson interferometer. Typically the amount of tilt, or shear, is controllable as this also has an effect on the sensitivity of shearography. It does not matter which arm of the interferometer is used for shearing, but usually the mirror not used for shearing is actuated for use in phase-stepping. Theoretically, there is no reason why the same arm of the interferometer cannot be used as both a phase-stepped and shear mirror, but practical limitations mean that it is simpler for one mirror to be phase-shifted while the other is used for shearing. The layout of the shearing optics is illustrated in figure 21:

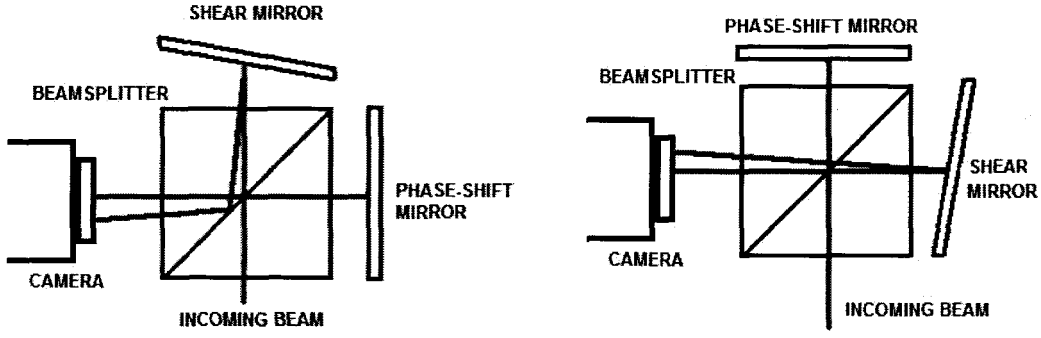


Figure 21: Equivalent optics layouts

A beamsplitter is used to separate the incoming reflected laser light into two beams, and is an essential part of the interferometer. Beamsplitters are partial mirrors, making use of various different coatings to achieve different transmission to reflection ratios, though a ratio of 50% is typically used in shearography applications.

There are essentially three different types of beamsplitter, namely plate, cube and pellicle beamsplitters, each of which are found in different shearography systems. Plate beamsplitters can suffer from 'ghost' images, due to internally reflected waves

through their thickness. These beamsplitters are however more cost-effective than the other alternatives.

Pellicle beamsplitters also suffer from the problem of internally reflected light, but this effect is minimized as they are extremely thin (2μm). This however also diminishes their usefulness in portable applications as they are generally not as robust as the plate or cube beamsplitters.

Cube beam splitters comprise two precision right-angled prisms which are cemented together using a transparent adhesive, and reflection/transmission takes place via a reflective coating applied to the hypotenuse of one of the prisms. They suffer less from 'ghost' images as the interface between the two prisms is thin, and reflected light usually travels back out through the perpendicular prism surfaces. Cube beamsplitters are more durable than the other alternatives as the reflective coating is contained within the prism and cannot be scratched or degraded.

Table 1 shows the relative prices of beamsplitters:

Table 1: Standard beam splitters

Type	Size	Price	
	mm	USD	R
Plate	35	$31	R248
Cube	40	$229	R1832
Pellicle	50	$200	R1600

Notes:

1. Prices are taken from Edmund Optics [29]
2. Currency conversion: US$1 = R8.00

2.2.4 Phase-stepping hardware

There are several different ways in which to achieve phase-stepping in interferometry applications. The basic requirement is that the actuator used should introduce known phase-shifts into the reflected light from one of the interferometer's arms, in synchronization with the image capture routine. The exact phase-steps that are required depend on the wavelength of the light source and the data reduction algorithm used; typically three- or four-bucket.

Several of these methods are discussed:

- Directly shifting the mirror using piezoelectric actuators
- Using piezoelectric actuators to introduce a phase-shift in fibre optic cables.
- Using laser diode sinusoidal phase modulation interferometry.

2.2.4.1 Piezoelectric mirror-shifting

The most widely used method for phase-shifting makes use of a piezoelectric actuator to shift the mirror. Piezoelectric actuators are generally used in applications where precise sub-micrometer positioning is required at high frequencies. For instance, piezoelectric devices are finding increased use in digital cameras for active lens stabilization [30]. Because of their wide range of applications, piezoelectric actuators come in many different forms, but mirror-shifting only requires single-axis, low frequency control, using cylindrical stacked actuators, which are the simplest form. These come in two varieties – steel packaged (enclosed) and epoxy-sealed (bare stacks); see figure 22. The bare stacks are prone to damage by shear forces, and must be specially mounted using half ball bearings to avoid damage [31]. The packaged units are easier to mount, although less cost-effective.

Figure 22: Epoxy-sealed stack (left) and stainless steel-cased (right)

These devices take advantage of the converse piezoelectric effect, where certain ceramic materials produce a small strain in response to a voltage applied across their surfaces. They are available in two broad categories: open- and closed-loop actuators. Closed-loop control is achieved through the use of strain gauges mounted on the actuator, from which signals are fed into a control circuit. Closed-loop actuators are used in applications where there are large dynamic loads on the actuator or where ultra-precise positioning is required.

In order to achieve travel that is of practical use, piezoelectric materials are stacked on top of each other in series, as shown in figure 23. Flexure hinges are also used to magnify travel. The amount of internal stacking determines the travel-to-voltage ratio, and also influences the length of the actuator.

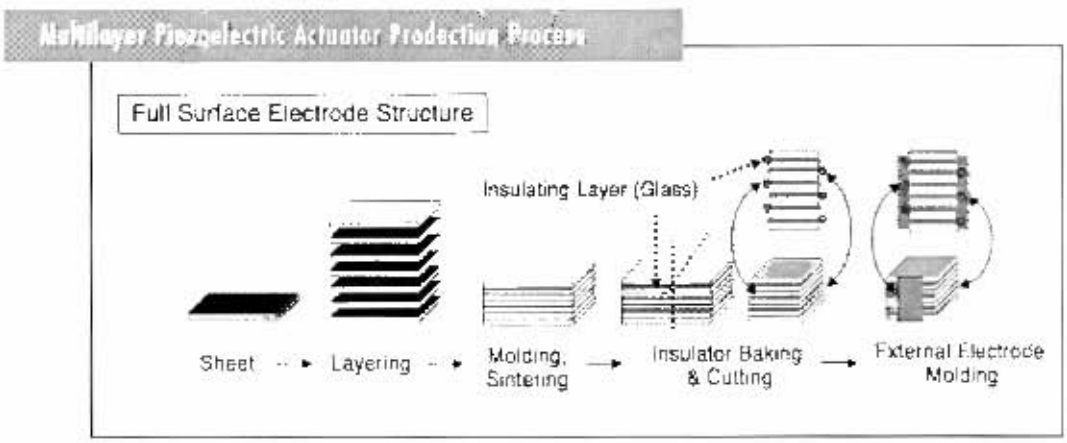


Figure 23: Multilayer piezoelectric actuator production process

When using a piezoelectric actuator, some type of voltage controller is necessary, as the piezo responds to voltage actuation. There are many different ways of achieving this control. One of the methods commonly employed is to use a digital-to-analogue converter (DAC) PCI card, which is an extension card for PCs. This enables the piezo voltage to be controlled from a software application. Other methods use larger external DACs which resemble power supplies. Due to the high voltages that are usually used to actuate piezo's, these controllers must be able to supply these voltages, sometimes up to 300V. While the actual power requirements are not great (the typical supply current does not exceed 100mA in most applications), obtaining these voltages can be difficult in portable applications.

2.2.4.2 Stretching fibre-optic cable to introduce phase shifts

While this technique is not usually applied to shearography, it is included to illustrate some of the different ways of achieving phase-stepped interferometry. In some interferometry applications such as ESPI, fibre-optic cables are used to transmit light, e.g. the reference beam, instead of reflecting mirrors. Where fibre-optic cables are used, phase-shifts may be introduced by stretching the fibre-optic cable using a piezoelectric actuator.

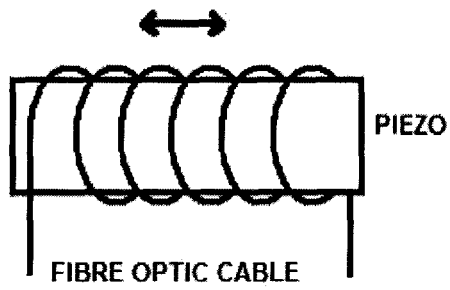


Figure 24: Piezoelectric actuator phase-shifting using fibre-optic cable

The fibre-optic cable is wrapped around a cylindrical piezoelectric actuator (figure 24), such that an expansion of the actuator will stretch the fibre-optic cable. This stretching of the cable introduces phase-shifts into the light travelling in the cable.

The use of multiple turns of fibre-optic cable and larger diameter actuators can result in lower voltages being required for equivalent phase-shifts, lowering the maximum voltage that is required for phase-shifting to approximately 5V [32].

2.2.4.3 Laser diode sinusoidal phase modulation interferometry

This phase-shifting technique (LD-SPM) was proposed by Kim *et al* [33], to overcome some of the limitations of using piezoelectric actuators and to make interferometry systems more compact. It achieves phase modulation without the use of a piezo by directly altering the laser diodes injection current. Changing the laser's illumination characteristics will affect both arms of the shearing interferometer, thus new phase mapping algorithms are also required.

This technique is still in the developmental phase, and the research papers describing it were only published in 2007.

2.3 Software options for shearography

The various hardware components used in shearography must be controlled in a synchronized manner, requiring some form of external control. Modern digital shearography requires the use of a host PC for processing and displaying the results of the image processing that is performed. In virtually all commercially available shearography systems, as well as in UCT's existing shearography system, custom software has been developed for controlling hardware and displaying results.

There are many different software packages and languages that can be used to achieve the objectives set out in this project. This section serves to review some of the available software that may be used in the new low-cost shearography system.

2.3.1 Matrox Imaging Library (MIL)

This is the library currently in use in the NDT laboratory. The Matrox Imaging Library is an industry-leading image processing toolbox that incorporates many advanced features such as optical character recognition (OCR), edge detection, blob analysis and pattern recognition. Using MIL is costly, as development licenses cost in the region of R40 000. In addition, every runtime license costs approximately R4 000.

The MI Library is designed to work in conjunction with Matrox framegrabbers. However, FireWire and USB cameras don't use framegrabbers. MIL claims to support FireWire cameras, but it only supports FireWire cameras connected through the Matrox Meteor II/1394 framegrabber, which is less cost-effective than a generic FireWire PCI card, and not available as a laptop card. It will only support FireWire cameras connected to a generic FireWire port if another "s-licence" is purchased, at a cost of R6000. Matrox provides no support for USB cameras.

Note that MIL is only an imaging library and not a programming language or development environment. MIL would typically be used in a C++ application and built in a development environment such as *Visual C++*.

2.3.2 *MATLAB* Image Acquisition and Processing toolboxes

MATLAB is both a development environment and a programming language. It also has a wide range of powerful toolboxes available, making it a stand-alone tool for developing a wide variety of applications.

The Image Processing toolbox was used for a previous project, and its high-level processing ability makes it ideal for rapid application development.

The *MATLAB* Image Acquisition toolbox supports FireWire and USB interfaces. It supports USB through the WDM (Windows Driver Model) and FireWire through DCAM [6]. The DCAM driver used by *MATLAB* is an open-source driver developed by Carnegie-Mellon University [34]. This driver complies strictly with the DCAM 1.3 specification.

Both USB and FireWire options would be able to capture video streams from most cameras; however these drivers would not support control of extended camera features such as GPIO and COM interfaces.

Another drawback to using *MATLAB* is that is because it is an interpreted language, it runs considerably slower than a compiled language such as C++.

The cost of using *MATLAB* is also large – *MATLAB* costs around R60 000, while the two image toolboxes and compiler cost approximately R10 000 each. However, distribution as compiled applications is free once the development license has been purchased.

2.3.3 *PixeLINK* Software Development Kit

Each camera manufacturer sells an image acquisition library for controlling their camera; the PixeLINK SDK is discussed here as an example of an imaging library offered by a camera manufacturer. It should be noted that software written with this SDK, although written for a camera that claims DCAM compliance, will not be DCAM compliant itself. It is implemented in C++, .NET and LabVIEW. It offers basic functionality: simple functions for grabbing images, previewing video streams and saving images. It also offers the user advanced control over the camera features such as custom frame descriptors (for specifying image options on a frame-by-frame basis), onboard lookup tables, and video callbacks (for modifying the captured frame before display). It also offers control that is not available via any other means – one can control extended camera features such as a COM port, GPIO pins and advanced triggering. It doesn't offer, however, any support for additional image processing beyond image acquisition.

PixeLINK manufactures FireWire and USB cameras, and the SDK supports both through the same interface. This means that software written with this SDK will work with both their FireWire and USB cameras. This is highly desirable, as it offers the user the opportunity to swap cameras without the need for recompilation.

The advantage of the PixeLINK licensing scheme is that only one SDK needs to be purchased, thereafter there is no restriction on redistribution of development or runtime licenses; in fact once the camera driver is installed, SDK-produced software will be ready to run.

2.3.4 *CImg* Imaging Toolkit [35]

CImg is an open-source C++ toolkit for image processing written by David Tschumperlé. This toolkit was considered for usage because it offers more powerful image processing than what can be easily achieved from the C++ basic types, and offers an alternative to 3rd party libraries which require licensing. It uses simple classes and methods for manipulating images. The entire toolkit is contained in a single header file, *CImg.h*, which makes using the library simple; one only needs to include the header file and namespace statement:

```
#include "CImg.h"
using namespace cimg_library;
```

The toolkit offers various functions for opening, saving, and processing images, as well as transformations, matrix operations, drawing and 3D rendering. The toolkit is governed by two different licenses:

1. The CeCILL-C license governs only the *CImg.h* header file. It allows the user to include the unmodified *CImg.h* in their project without disclosing the rest of their source code. This license is similar to the *GNU LGPL*. Any changes to *CImg.h* must be released as open source.
2. The CeCILL license applies to all other toolkit files including documentation and examples. The CeCILL license is more restrictive, and prevents *CImg.h* from being used in closed-source projects. This license is compatible with the *GNU GPL*.

The header file is large, and typically exceeds the page memory of the *Visual C++* compiler. The solution is to increase the memory by adding the switch, /Zm200 to the compiler options. This increases the internal heap size and allows successful compilation.

2.3.5 ImageMagick [36]

ImageMagick is a powerful open-source image processing library designed for command-line or API-based (it has been translated and adapted for many languages) usage. The reason it was considered for this application was because it offers support for many different image formats. The *CImg* library has built-in support for *ImageMagick*; this would allow the supported image formats to be extended from just bitmap (*.bmp) images to almost every image format available.

2.4 Survey of existing shearography systems

Shearography is rapidly gaining acceptance in industry as a useful NDT tool. As such, there are several commercially available shearography systems that are used in a wide range of industries. This section reviews some of the portable shearography systems available in addition to UCT's existing portable testing unit.

2.4.1 Review of UCT's existing shearography unit

UCT's NDT Laboratory's shearography testing unit consists of a shearography head, in which a camera, piezoelectric actuator, shearing optics and laser are housed, as well as a ruggedized PC loaded with the laboratory's custom-designed software. The shearography head is a 250mm × 100mm × 100mm portable box, and implements the technique of phase-stepping using a piezoelectric actuator. Figure 25 shows the layout of the unit.

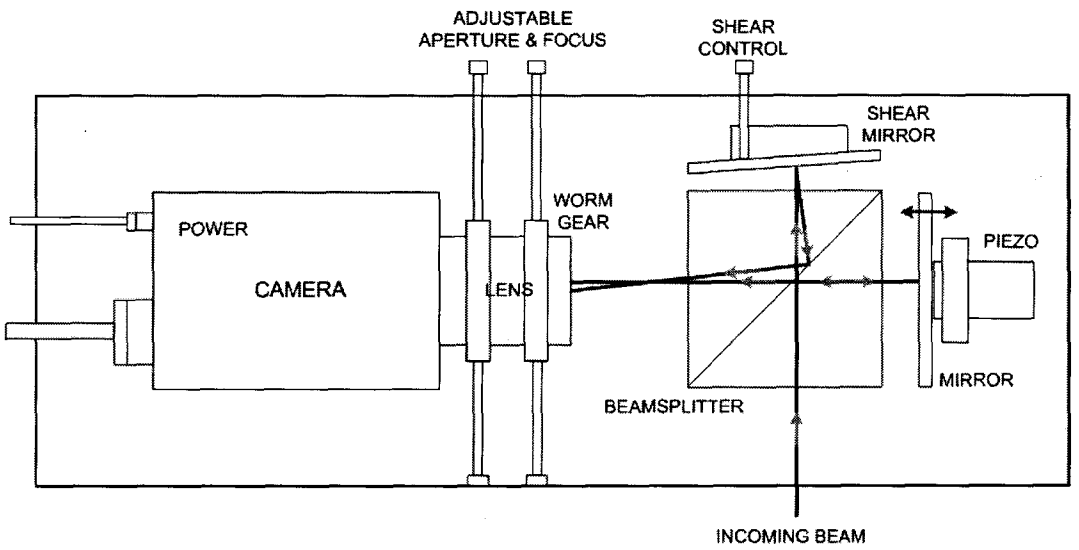


Figure 25: Layout of UCT's shearography head

2.4.1.1 Hardware

2.4.1.1.1 Housing

The housing (base and walls) is machined from a solid block of aluminum which is anodised black to absorb errant light. Components are mounted on the bottom of this surface, and the laser is mounted on the lid of the box.

2.4.1.1.2 Camera and peripheral equipment

The testing unit uses a Pulnix digital camera with a Camera Link interface. The Camera Link interface necessitates a framegrabber PCI card for the host PC. The Matrox Meteor II (or Solios) framegrabber is used. The resolution of the camera is 1392×1040 pixels and uses a Fujinon, C-mount, fixed focal length (16mm) lens. Power to the camera cannot be supplied over the Camera Link cable, and is therefore supplied over a separate cable from the host PC. Figure 26 diagram illustrates the setup of the complete shearography system.

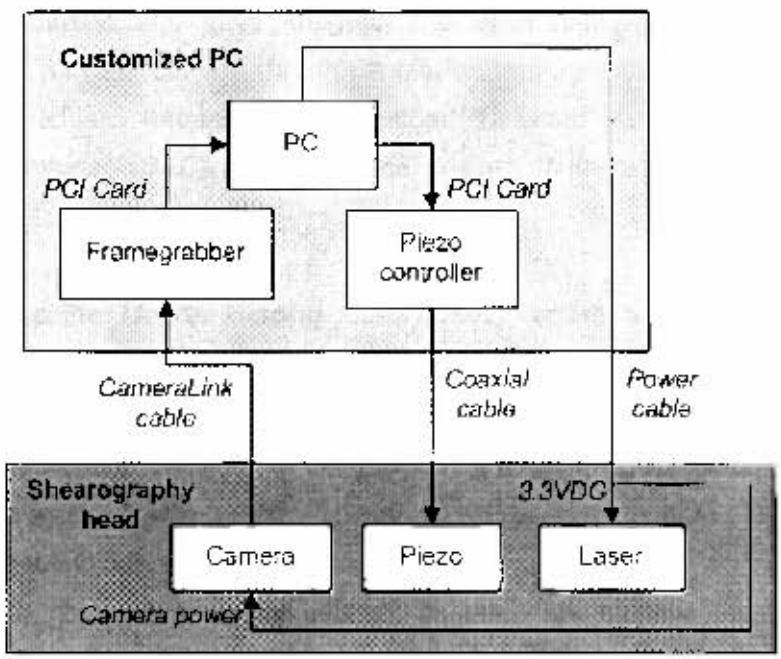


Figure 26: Schematic design of existing system

2.4.1.1.3 Phase-stepping piezoelectric actuator and controller

The piezoelectric actuator is specifically designed for mirror shifting in interferometry applications and is imported from Germany. Its displacement curve is nearly linear, offering 0-6μm travel over a range of 0-150V. Another custom PCI board from the same company is used as a DAC controller, and allows software control of the mirror displacement. The DAC controller has 5mV noise with 14-bit resolution. This gives a displacement resolution of 0.37nm, and error of 0.20nm. The PCI controller and actuator are connected via a coaxial BNC-terminated cable.

2.4.1.1.4 Illumination

Illumination is provided by a thermoelectrically-cooled laser diode with 100mW laser power and 660nm wavelength. The laser is mounted on the lid of the box. Power is provided over the same cable that supplies the power to the camera.

2.4.1.2 Software

The existing testing unit is controlled by a *Visual C++* application, called *Inspector*, which performs all image processing and display, and allows the user to open phase-unwrapping utilities and save images. The application interfaces directly and automatically with the camera through a Matrox framegrabber, as well as with the piezoelectric actuator through another custom PCI board which acts as a digital-to-analogue converter (DAC). However, the design of the application has several drawbacks:

- It uses the *Matrox Imaging Library (MIL)*, where a simpler library would suffice.
- The camera model, resolution and path to camera files were hard-coded in the application, requiring re-compilation on target machines when deploying.
- Code was added and removed in an ad hoc basis by multiple authors, making it difficult to debug and understand.
- It has no version control system, so there are multiple versions all with different bugs and problems, which can make troubleshooting difficult.

2.4.1.3 Deploying testing stations

The high degree of PC hardware customization means that the testing PC and shearography unit are intricately linked. This requires that the PC and testing unit must be distributed in conjunction, raising the cost of the combined system significantly. Currently, the PC supplied is a ruggedized industrial unit. A laptop cannot be used as this would not allow space for the PCI cards.

2.4.2 Dantec Dynamics' Q-810 Portable Shearography System

Dantec Dynamics [37] is a Danish company that specializes in the development and sales of integrated measurement systems for, amongst others, strain measurement. They offer ESPI and portable shearography systems.

Their portable shearography system, the Q-810, uses a remote head, or 'vacuum hood', which has integrated vacuum excitation capabilities, laser diode illumination and a digital display. This hood is attached to the specimen to eliminate any global vibrations, and to allow vacuum stressing of up to 150mbar. A specimen area of 225 × 175mm can be inspected at one time.

The hood also contains controls for operating the unit, and claims to be able to deliver interference images at 1 image every 30 seconds. This acquisition time presumably includes the time that is taken for repositioning the hood.

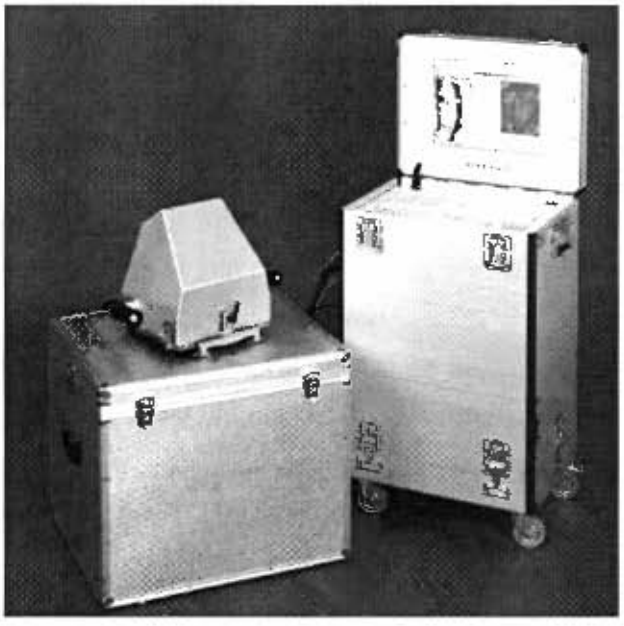


Figure 27: Dantec Dynamics' Q-810 portable shearography system

The hood (which weighs 7kg) connects to a host PC which is contained in a rolling housing. This housing is large and heavy (35kg), but rollers serve to make it more portable (as illustrated in figure 27). The system implements phase-stepped shearography and phase unwrapping, but offers little other detail regarding the technology that is used. The overall cost of this system could not be obtained.

3. CONCEPTUAL DESIGN

Once the design and layout of the existing system were well understood, and research had been conducted into the relevant and available hardware and software, several different concepts were developed and analysed. The aim of these concepts was to come up with a solution that would satisfy all of the design specifications, and in particular be low-cost. Different layout, hardware design and software concepts were generated, which are presented below. The outcomes of these discussions and concepts are discussed in the following section, which details the final solution.

3.1 Physical construction

3.1.1 Type of housing: specimen- vs. tripod-mounted

The type of housing employed to contain the components of the shearography head plays a significant role in determining how the shearography system will be used. For example, the Dantec Dynamics' Q-810 shearography system comes packaged in a vacuum hood which is manually attached to the specimen surface using suction pads. Other systems, such as UCT's existing shearography head are positioned using tripod stands and are not directly connected to the specimen. The following sections analyse the advantages and disadvantages of each solution.

The two major advantages to using a specimen mounted housing over tripod mounting are firstly that the stressing mechanism (thermal, pressure and mechanical) can be incorporated in the hood, simplifying the operation. Secondly, the fact that the shearography head is physically attached to the specimen eliminates any vibration between the two, which can lead to better quality results due to decreased environmental disturbance.

Another advantage of using a specimen-attachable hood is that because of the enclosed design, a less powerful laser might be used. Less powerful lasers are more cost-effective and often provide better quality results, as the decreased power allows the laser illumination to be more stable. Also, owing to the reduced distance between the specimen and shearography head, a lower resolution camera can be used, which is also more cost-effective.

However, one of the limitations of the hood design is that a limited area can be inspected at one time and every time a new area is to be inspected, the head must be decoupled, moved and then reattached to the specimen surface. This can make the inspection or large areas time consuming, whereas with a trip-mounted head, repositioning is simple and can be achieved more quickly.

Also, it is likely that on occasion the vacuum pads which support the shearography head will fail, causing the vacuum hood to fall which may result in the shearography head being damaged. This is unacceptable for a high-value testing unit that can easily be damaged by such a fall.

Finally, when using a specimen-mounted system, the fact that the shearography head is integrated in the vacuum hood:

- Precludes testing components which are irregularly shaped.
- Limits the types of testing that can be carried out.
- Limits the excitation or stressing methods that can be used to those performed by the hood itself.

3.1.2 Housing construction: box design vs. sheet metal

A box design (where the housing was machined from a solid piece of aluminium – see figure 28) was used in the housing of the existing system, and was considered for use in the low-cost system. The advantage to using a solid box design is that the housing is extremely strong, and presents an aesthetically pleasing facade as there are no clear edges.

However, the use of a large block of aluminium for machining out a box is wasteful as it requires many machining hours and wastes a large amount of aluminium in addition to being cost ineffective. Also, the fact that components are mounted on the bottom surface of the inside of the box can make accessing components difficult.

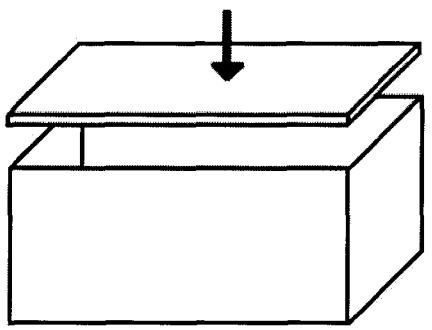


Figure 28: Solid aluminium box housing

As an alternative to this approach, another concept was considered, which made use of a base plate for mounting components and a sheet metal fold-over to cover the box, with end plates to cover the ends, illustrated in figure 29. This is less wasteful as less material is required, but also requires more components to make up the box; 6 instead of 2 in the case of the solid machined block. Additionally, the sheet metal approach will not be as strong as the machined solution.

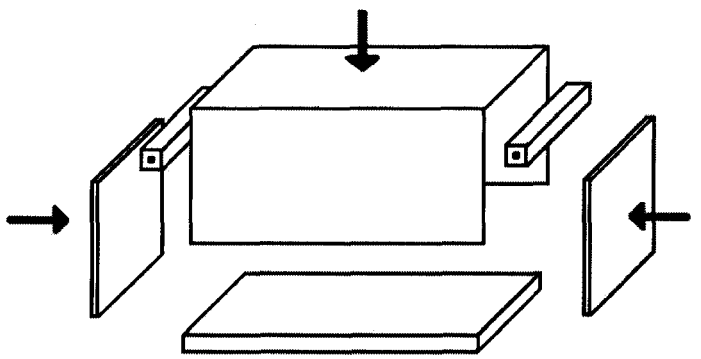


Figure 29: Housing constructed from folded sheet metal

3.1.3 Internal layout concepts

3.1.3.1 Layout of the shearing optics

While there are different ways of shearing and recombining the speckle images on the image plane, the most common method uses a shearing Michelson interferometer. Even when this configuration is used, the relative placement of the shearing mirror with respect to the phase-stepped mirror can be set up differently. Figure 30 shows how this placement can differ.

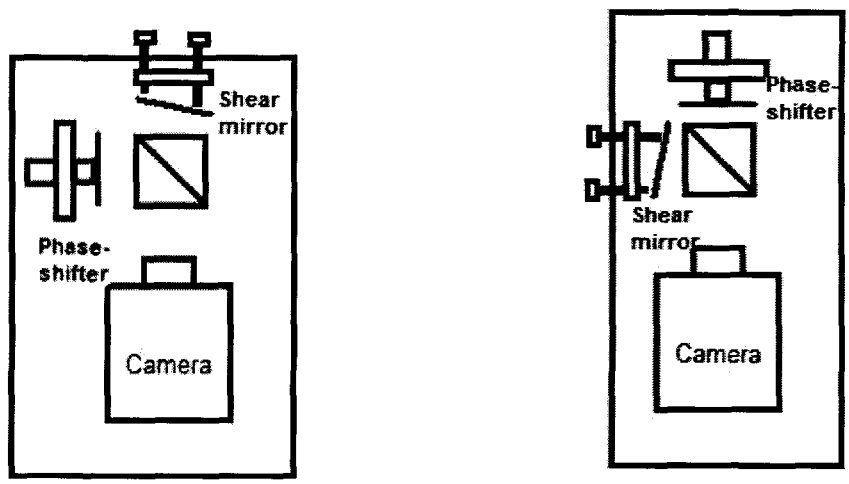


Figure 30: Different layouts for a Michelson shearing interferometer

Both of these configurations are functionally equivalent, but have different practical considerations associated. The first consideration is the placement of the shear mirror controls. The amount is shear should be user controllable, as this has the effect of adjusting the sensitivity of the system. These controls should be easily accessible, and collocated with other relevant controls.

Secondly, the relative placement of the different arms of the interferometer has an effect on the overall size of the unit. The phase-stepping device (typically a piezoelectric actuator) is usually larger than the thin shear mirror plate, and its placement will have a direct impact on the overall dimensions of the unit, for instance making the unit wider or longer.

3.1.3.2 Methods for performing focus and aperture adjustment

Most of the digital cameras used in shearography use manual iris and focus lenses (this will certainly be the case in a low-cost implementation), and thus a means of adjusting the focus and aperture must be provided for in the design. Often the lens is housed inside the shearography head and cannot be accessed directly. This discussion concerns the means of addressing this problem.

One common configuration that was encountered during a survey of current methods and products was one in which both of the mirrors and the beamsplitter are housed after the image has passed through the lens. This allows for the lens to be mounted on the outside of the housing, giving the user direct manual access to the focus and aperture, as in figure 31. Also, since the beam is narrowed by the lens before passing through the beamsplitter and lens, smaller versions of these components may be used, thereby decreasing the overall cost.

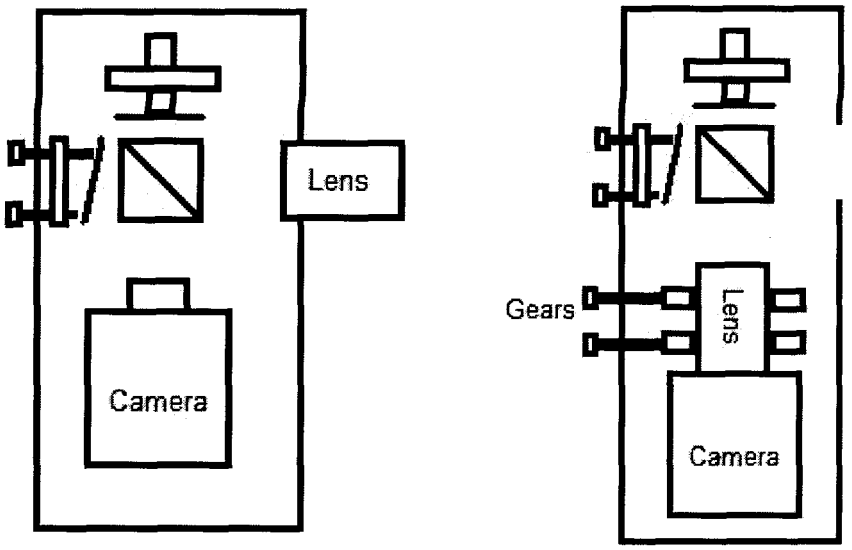


Figure 31: Difference between external lens (left) and the existing design's internal lens (right)

One example of this configuration is that used in the commercially available HG7000 shearography unit made by Pcholo [15].

This approach also has several disadvantages, namely that it leaves the lens exposed to the dirt and dust encountered in industrial settings. A previous MSc project aimed at achieving this goal had already been conducted [38], which met with success, but in the end was not adopted due to decreased image quality and increased complexity.

One alternative to externally-mounted lenses is the use of gears for adjusting the internally-mounted lens. An advantage to using gears instead of manual adjustment is that gears can be used to provide a gear reduction, allowing finer control of the properties. Two different gear-adjustment approaches were considered:

- Using worm gears, as is the case in UCT's existing shearography system.
- Using spur gears as adjustment wheels, protruding through the outer case of the shearography head

Figure 32 illustrates how these solutions might be implemented.

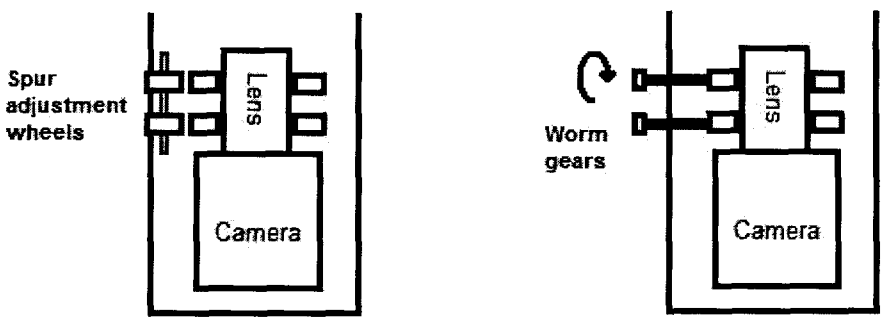


Figure 32: Adjusting the lens: spur gears (left) and worm gears (right)

The worm gear solution has a significant advantage over the use of spur gears, as it allows for the box to be completely dust-sealed. When using spur gears, some gaps in the housing would have to be left in order for the gears to turn smoothly.

Spur gears are however, more cost-effective and are more easily obtainable, making them suited to use in a low-cost shearography implementation.

3.1.3.3 Top-mounted electronics

The final layout concept considered was aimed at miniaturizing the shearography head by making maximum use of the available envelope space. It was aimed at taking advantage of the fact that the board level camera head and board are connected by a cable and need not be located together as was the case in the past when an enclosed camera was used. This design also aimed to take advantage of the fact that there is a lot of unused space in the overall envelope of the shearography head, located behind the laser mounting. By placing the laser inside the main box and moving the electronics behind the laser housing (and on top on the beamsplitter), the overall dimensions of the envelope could be decreased. Figure 33 shows a design in its advanced stages that uses this approach.

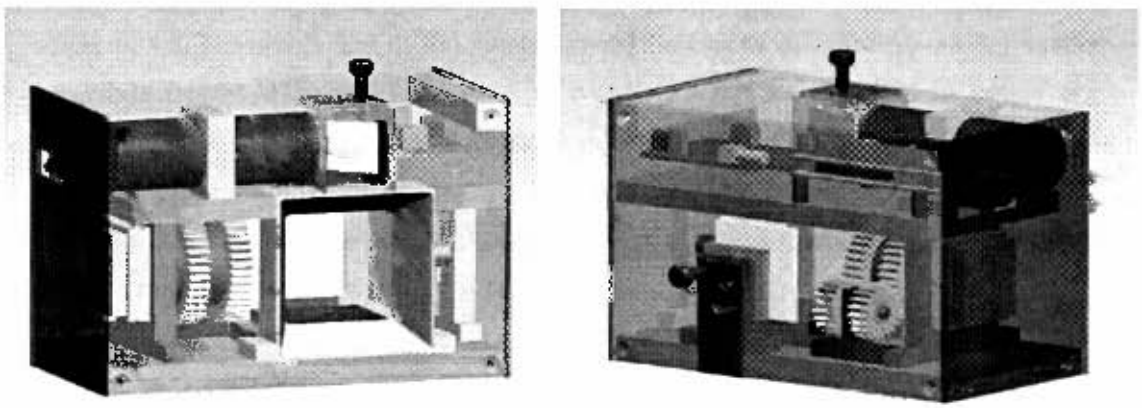


Figure 33: *ProEngineer* model of alternative layout

This layout has some disadvantages:

- Assembly would be difficult as one can't easily access the lower plate's components without first removing the top plate.
- It requires the cable connecting the camera board with the camera head to be fully extended and limits the position of their relative placement severely. Extending the cable would also be difficult but not impossible, as its Molex connectors are not readily available.
- The FireWire connectors would stick out in the top left corner of the cut-out plate, which is not as ergonomic as a lower centre mount.
- Mounting adjusting screws for the laser beam would be difficult, as electronics boards obscure access to the laser mirror from the rear.

3.2 Hardware options

3.2.1 Digital cameras and framegrabbers

The design requirement that the shearography head should require no host PC modifications (and be able to operate from a laptop) eliminates the use of Camera Link and Gig-E interfaces, as they both require framegrabbers. Both USB and FireWire cameras generally use CMOS sensors, so there was no option to use a CCD camera with the remaining camera interfaces. As such, the only two available interface options for choosing a suitable digital camera were USB and FireWire.

USB cameras are generally slightly more cost-effective than equivalent FireWire cameras. This fact, coupled with the fact that USB ports are more readily available than FireWire ports, makes USB cameras ideally suited for use in low-cost portable applications. However, one of the limitations of the use of a USB camera is that the USB bus power is limited to 250mA at 5V. While this is sufficient to power the camera, there will not be any additional power available for auxiliary devices. In contrast, the amount of power available on the FireWire bus is significantly greater than the amount required by the camera, thus enabling the possibility that other devices might be powered from the same cable.

In the case of a portable shearography system that must be both user-friendly and compact, minimizing the number of external cables required for operation can greatly simplify and enhance usability. The potential for a single cable to power and control the camera as well as some of the peripheral devices required (e.g. the laser and piezoelectric actuator) is significant, and means that the extra cost associated with using a FireWire camera instead of a USB camera might be mitigated by the additional ease-of-use which it may offer.

Table 2 on the following page offers a comparison between the different camera interfaces, their performance and specifications. Gig-E and Camera Link cameras are significantly more expensive than the equivalent FireWire and USB cameras. Most of these cameras offer similar performance characteristics, with similar frame rates and bit depth. While the bit depth of the cameras varies between 8 and 10 bits, it should be noted that most imaging applications will automatically convert 10-bit images (which have to be represented internally as at least 16-bit) to 8-bit images.

Table 2: Comparison between cameras

	Part number	Sensor[1]	Interface	Resolution pixels	Frames per sec	Bit depth	Cost[2,3] USD	Cost[2,3] R[4]
Edmund	*EO-1312*	½"	*USB*	*1280 x 1024*	*25*	*8*	*$835*	*R6680*
PixeLINK	PL-B771U	½"	USB	1280 x 1024	30	10	$895	R7160
PixeLINK	PL-B771F	½"	FireWire	1280 x 1024	30	10	$1 095	R8760
PixeLINK	PL-B741	⅔"	FireWire	1280 x 1024	25	10	$1 195	R9560
AVT	F-146	½"	FireWire	1392 x 1024	18	8	$1 390	R11 120
Basler	A622F	⅓"	FireWire	1280 x 1024	24	10	$1 475	R11 800
JAI	TM-1402L	½"	Camera Link	1392 x 1040	30	10	$1 902	R15 216
Sony	XCL-U1000	1/1.8"	Camera Link	1600 x 1400	15	10	$3 607	R28 856
DALSA	Genie 1400	½"	Gig-E	1392 x 1040	15	10	$2 188	R17 504
JAI	TM-1405GE	½"	Gig-E	1392 x 1040	30	10	$2 295	R18 360

Notes:

1. All cameras are monochrome for the sake of comparison. All of the USB and FireWire camera have CMOS sensors, while the other have CCD's.
2. Where available, prices are for board-level cameras.
3. Prices are quoted from Edmund Optics website [29] and do not include transport.
4. Currency conversion: US$1 = R8

In the range of FireWire cameras, the most cost-effective manufacturer is *PixeLINK*, while the Edmund Optics' USB camera is the most cost-effective camera overall. Most of these cameras offer additional features which can be used by OEM manufacturers to further integrate their products for compactness or usability. These features should be evaluated on a per-camera or per-manufacturer basis.

3.2.2 Phase-shifting actuators

In digital shearography, there are several different ways of achieving phase-stepping, some of which were discussed in the literature review. However, the most common and simple method uses a piezoelectric actuator (simply referred to as a piezo) to physically shift the mirror.

The literature review also discusses some of the different types of piezo actuators available. In interferometry applications, only single axis motion is required, and in general simpler open-loop actuators may be used as there is no dynamic loading and the small level of hysteresis present in open-loop is not a critical problem.

3.2.2.1 Piezo type: packaged vs. bare stacks

The ceramic materials used in piezo actuators are usually pre-packaged to make usage simpler. The packaging provides preloading and protects the actuator from damage. However, unpackaged piezo's are also available for some suppliers, and these are significantly more cost-effective than the equivalent packaged unit, and are also generally smaller.

However, the bare-stack (or unpackaged) piezo's require special mounting considerations, which may negate some of the cost savings that are made by using an unpackaged actuator. Nevertheless, unpackaged piezo's are ideally suited for integration in low-cost shearography systems where their cost-effectiveness and small size are important.

3.2.2.2 Selection parameters

Piezoelectric actuators offer a nearly linear displacement response to an applied voltage. The maximum required travel (and therefore piezo voltage) is a function of the laser's wavelength as the mirror must be shifted by approximately half of the laser's wavelength. The lower the voltage required for a given amount of travel makes design of the system simpler, because obtaining a high supply voltage can be difficult in portable applications.

The process of selecting a piezo involves calculating the required displacement-to-voltage ratio as well as ensuring that the piezo can shift the mirror by the minimum

amount, as piezo's are often specified by their maximum travel (in μm) and maximum voltage. However, the maximum supply voltage from the piezo controller is still unknown, as the available voltage will be a function which piezo controller concept is adopted. This makes selecting a piezo based on its displacement-to-voltage ratio difficult.

A pragmatic approach was taken, where the piezo controller voltage was initially estimated, to find if there were actuators capable of delivering the required performance using that maximum voltage. As previously mentioned, the lower the maximum voltage required, the simpler the design of the controller will become. This consideration, combined with the fact that the standard voltage reference for many DACs is 5V, led to 5V being initially chosen as the reference voltage.

In order to determine what magnitude phase shift is required, we also need to know what the wavelength of the laser will be, as the piezo should be able to shift the mirror by half of the lasers wavelength. The piezo mirror need only shift by half the laser's wavelength in order to change to optical path travelled by the light by a full wavelength, as the light travels to and from the mirror surface, effectively doubling the change in path length that was introduced by the piezo mirror.

The typical laser used in shearography applications is a red laser, which has a wavelength of approximately 660ηm. Other lasers may also be used, but by assuming that a red laser is used, we ensure that the piezo will be able to shift sufficiently for other lasers as well, since red light has the longest wavelength of the colours in the visible spectrum.

Thus, the piezo should be able to shift at least 330ηm with a 5V supply, i.e. a displacement-to-voltage ratio greater than 66ηm/V is required.

While the requisite displacement-to-voltage ratio is an important selection specification, there are other specifications which are important when selecting a piezo for phase-shifting shearography. For instance, the length of the actuator is a design factor because the layout of the testing unit dictates that the piezoelectric actuator length has a direct effect on the minimum testing unit size.

Several piezoelectric manufacturers' offerings were investigated, namely Physik Instrumente [39], PiezoMechanik [40], PiezoSystemJena [41], and Mad City Labs [42]. Table 3 shows the prices and design specifications of the investigated components.

Table 3: Piezoelectric actuators

Type	Code[1]	Part number	Travel	Voltage	Length[3]	Price	
			µm	V	mm	USD	R[4]
Stack, open-loop	PSJ	P-216-40	16	150	18	$268	R2 144
Stack, open-loop	MCL	PZT3	18	150	20	$130	R1 040
Enclosed, open-loop	PSJ	P-112-00	16	150	33	$410	R3 280
Enclosed, open-loop	PI	P820.10	15	100	26	$380	R3 040
Enclosed, open-loop	PM	STr25/150/6	6	150	26	$710	R5 680
Enclosed, open-loop	PM	STr25/150/20	16	150	33	$1050	R8 400
Enclosed, closed-loop[2]	PI	P841.10	15	100	32	$1400	R11 200
Enclosed, closed-loop[2]	MCL	Nano-P15	15	150	34	$6250	R50 000

Notes:

1. The manufacturer's code refers to the following manufacturers:
    a. Physik Instrumente (PI)
    b. PiezoMechanik (PM)
    c. PiezoSystemJena (PSJ)
    d. Mad City Labs (MCL)
2. Price for closed-loop actuators includes the controller and actuator.
3. The diameters of actuators are not considered here.
4. Currency conversion: US$1 = R8

3.2.3 Piezoelectric controller: custom vs. OEM

The voltage to be applied to a piezo actuator corresponds to fractions of the laser's wavelength, and can be calibrated before processing starts. The exact voltages that need to be applied depend on the voltage-to-displacement ratio of the piezo actuator used, the wavelength of the laser, and the data reduction algorithm used (typically three- or four-bucket).

Many existing shearography applications use a PCI expansion card that acts as a digital-to-analogue converter to control the piezo. The requirement that the host PC should not require any modifications eliminates this method as a possibility for the new low-cost system.

There are many ways in which to step the phase voltages. Three different methods for achieving this voltage control were considered.

3.2.3.1 Purchasing a USB digital-to-analogue converter

The complexity of designing a DAC controller that is capable of shifting the mirror in synchronization with the image capture, with high resolution, repeatability and low noise may mean that purchasing an OEM DAC module may be a viable solution.

High resolution DAC's are available as USB devices, which are commonly used as high quality replacements for generic PC soundcards. This type of device might be modified to act as a piezo controller.

However, this would require that an extra cable is required; a USB/FireWire cable for the camera and a USB cable for the controller; using two separate cables is not as ergonomic as just one. Also, these controllers are likely to be less cost-effective than designing a custom controller.

3.2.3.2 Multiplexer switching using the camera's I/O pins

It is highly desirable that the phase-step voltages are software configurable, but this not completely necessary; the different voltages required could be manually set on the board first, and cycled through at the correct times with software triggers, using the camera's General Purpose Input Output (GPIO) pins. Note that the number of preset voltages required depends on the data reduction scheme used.

These on-board, preset analogue voltages can then be selected using a multiplexer and the GPIO pins. Figure 34 illustrates this concept.

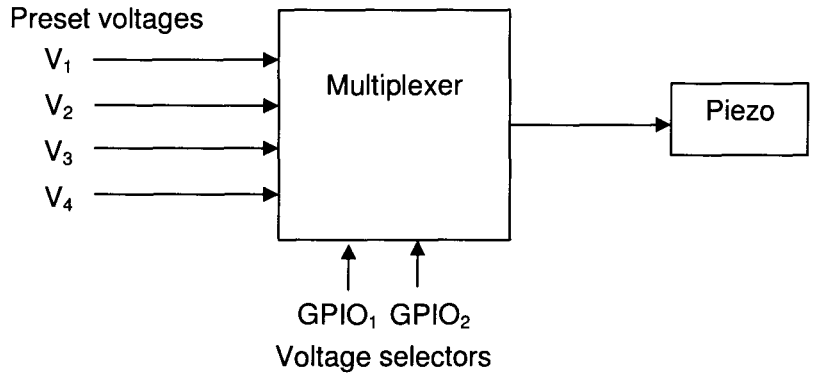


Figure 34: Multiplexer-based piezo controller

This method is simpler than using the serial bus as there is a lower level of electronic design involved, but also has several drawbacks:

- The on-board voltages cannot be easily manipulated; the casing must be opened for them to be manually set; there is no possibility of soft calibration.
- The requisite degree of accuracy may be difficult to achieve with potentiometers and human error.
- Resistors typically have 5% tolerance and suffer from temperature drift, so analogue voltages may change/drift with time and temperature.

3.2.3.3 Communication over the camera's serial bus

Many machine vision cameras, particularly board-level cameras, have been designed with OEM integrators in mind, and have onboard serial buses for communicating with peripheral devices, relaying information using the same FireWire cable as the camera. This capability was investigated to determine whether or not it could be used to control the piezo. While the nature of the serial communication is digital, it may be converted to an analogue signal using a custom-designed DAC board, as in figure 35:

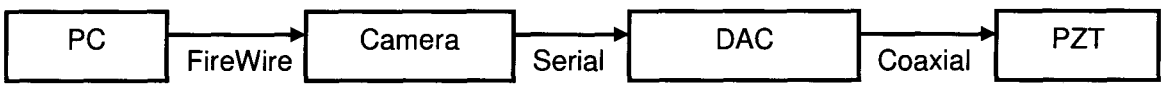


Figure 35: Serial bus communication for piezo controller

If the custom DAC could be designed to be compact enough to fit inside the shearography head housing, no external controller or cable will be required, as the serial communication is relayed over the camera's FireWire cable.

Using a 12-bit DAC, the phase-step voltages are configurable from within the shearography application, enabling these to be changed without manual adjustment. This might also enable in-situ calibration of the piezo.

3.3 Software-related concepts

3.3.1 Choice of programming language and imaging software

3.3.1.1 Selection criteria

The choice of programming language is an important design decision that should be made at the start of the project. While there are many different alternatives; the final decision should be made based on a combination of factors:

- The suitability of the language for the task at hand, for instance the platforms supported, speed and type of application (e.g. desktop, distributed, web).
- The availability of libraries in that language which are suitable for the task.

A software application for controlling the shearography hardware would most likely be a self-contained desktop application. Cross-platform compatibility would not be a major concern for such an application, as its small distribution requirement could be used to ensure that the application ran only on Windows or other PCs.

The processing speed of the application is an important concern however, as the application would need to process live video from a high resolution digital camera, and perform image processing on the recorded images, while displaying the results in real-time. The fact that results can be obtained in real-time when using shearography is an important benefit over other NDT techniques, and thus the application's processing speed is important.

Some of the advanced image acquisition and processing requirements of shearography dictate that 3rd party image acquisition and processing tools will be required. While image processing software is available in almost every programming language, the relevant image acquisition libraries and drivers are often only available in a few languages and as such the image acquisition tool or library used may limit the number of programming languages available. The literature review revealed that in order for a program to interface with a camera, those specific camera drivers are required, except where the camera uses a well-defined protocol, such as the DCAM specification for FireWire. Even then, many cameras offer additional features that are not supported by standard drivers. Therefore, in order to take advantage of a camera's features, the camera, drivers and software should be purchased as a suite.

3.3.1.2 Discussion of software alternatives

Two programming languages were considered, namely *MATLAB* and C++. Both of these languages and their support for camera drivers are discussed in the literature review.

C++ is well-suited to desktop applications, and as such there are many 3rd party imaging libraries which are available. In addition, C++ runs faster than most other compiled and interpreted languages, making it ideal for imaging applications. Virtually all camera vendors supply the imaging APIs in amongst other languages, C++. Finally, development environments which support C++, such as Microsoft's Visual Studio, make available desktop application frameworks (such as MFC) which remove much of the hassle of developing the user-interface by providing templates which conform to the standard Windows application look-and-feel.

There are two distinct disadvantages to using C++ for this application:

- C++ is a relatively low-level language, so developing a complete application is more time-consuming than languages which allow developers to be more productive, such as *MATLAB* or Python.
- The standard C++ library is very limited, and additional software libraries will be required. This may not necessarily be problematic, but many 3rd party imaging libraries are proprietary and may not suit the low-cost paradigm of this dissertation. Examples of 3rd party packages that were considered for additional functionality include MIL, the *CImg* toolkit and ImageMagick, all of which are described in the literature review.

The *MATLAB* environment is known to the author, and was considered as an alternative to MIL and C++. *MATLAB* is ideal for rapid application development and prototyping owing to its ease of use and high-level functions. The standard *MATLAB* Image Acquisition Toolbox uses a generic driver for image acquisition, and as such could not take advantage of advanced camera features. However, *MATLAB* also has support for accessing C++ code through the use of MEX files. This functionality might be used to access the cameras proprietary API from *MATLAB*.

However, *MATLAB* runs significantly more slowly than C++ and is also less cost effective.

3.3.2 Application architecture

The means by which the shearography application splits processing time between user-interface tasks, such as responding to button clicks, and image acquisition and processing is an important design decision. This decision is largely independent of the programming language and software libraries used. Three different methods were investigated. The three methods investigated are:

- Timers
- Callbacks
- Custom multithreading

Both callbacks and user-defined threads create a separate thread of execution in the application. Threads introduce a new level of complexity, but also offer greater control. Timers on the other hand, offer a simple way to process and grab images without starting a new thread, by triggering a timer event within the main thread at predetermined intervals.

3.3.2.1 Callbacks

Callbacks are offered by most acquisition libraries. In short, a callback is a function that is called just before an image is displayed, which may modify the captured frame for display. Callbacks are usually used to overlay graphics, such as a logo, on an incoming image. Typically a callback would not need to access to any previously acquired images, as is the case in shearography. Callbacks are usually called within a separate worker thread, owned by the acquisition library, and so access to previously acquired images or data structures must achieving using a persistence level, and must be synchronized. A callback will affect the processing speed, as it is executed in series with the image capture. This is similar to synchronized capture in that the returned image was captured after the piezo was shifted in the previous callback iteration.

One of the main problems associated with using callbacks in the shearography application is that callbacks use the imaging library's proprietary code to display images, not the custom 'draw' function and as such, every callback's function declaration will be different and will have different input parameters. Thus different

image capture and processing API's will require different callback functions signatures and will be unable to share the same code.

Also, one cannot achieve fine-grained control over the image display; for instance some APIs will always fit images to the screen size, whereas others will always display the image at its true size, overlapping the window or leaving blank areas. Unfortunately, overriding this behavior is often impossible, as control resides with the 3rd party library's callback thread.

When capture is halted, the callback thread is terminated and control returns to the original application thread. In the case of the most APIs, this causes the on-screen image to be lost with the screen returning to the background colour. This is undesirable, as typically the user would want to review the result of an inspection after stopping. As a solution, a reference to the display image must be maintained outside the callback thread so the image can be retrieved and displayed after recording has stopped, using the custom 'draw' function.

Also, callbacks must always return the output frame in the same buffer as they were input. In shearography, the display image is a totally different image to the captured frame, so the captured frame must first be copied out of the source buffer, and calculations must be performed before the display image is copied back to the source buffer. This copying and replacing of large buffers is an unnecessary waste of resources.

3.3.2.2 Timers

This is the method for processing employed by UCT's existing shearography application. The advantage of this approach is that it is simple; the user-interface and image processing both operate in one thread, with the processing time split between the two using a system-dependent message system.

As a specific example, the Windows message system in Microsoft Foundation Classes (MFC) is a means for different parts or threads of the same application to communicate. Messages (typically referring to UI tasks), each with a specific priority, wait in a message queue and are processed when the application thread is idle. Timer events (the WM_TIMER) are such messages, and are assigned a low priority. Whenever a WM_TIMER message is processed, a function is called (the function is specified as a parameter of the timer message).

One common problem encountered when processing timer messages is when the called function performs heavy processing, which prevents other messages from being processed, leaving the user interface unresponsive. The solution is to either:

i. Ensure that the timer function returns before being called again, leaving time for processing other messages during idle-processing.
ii. Place message-handling loops within the intensive timer function, to allow other messages to be handled in between processing.

3.3.2.3 Threads

Using custom multithreading is a similar approach to using callbacks, but it allows more fine-grained control of drawing operations, image processing, and initialization by ensuring that the relevant previously acquired frame buffers remain in scope for each iteration of the frame processing.

A new worker thread is initialized and started within the main application thread when the 'Play' button first pressed. The application thread passes the new worker thread a handle to the underlying document, granting the worker thread access to the data structures that determine which mode it should operate in. The new thread initializes buffers and captures the reference frames that will be used later in intensity-based shearography and phase-stepped shearography. The thread then enters a 'while' loop which performs the capture and processing of images. In each iteration of this loop, the worker thread posts a message to the user interface to update its display (the custom 'draw' function). This worker thread will terminate by exiting the 'while' loop only when the 'Stop' button is pressed.

This approach enables the last frame to be saved when capture stops, preventing the screen from blacking out as is often the case when using callbacks. Also, the entire user-interface will remain responsive as the user-interface thread is not busy with image-processing tasks.

Using custom-multithreading has an important advantage over callbacks; the function that draws images to screen when playing (during image processing) is the same function that draws images when not playing (this is the custom 'draw' function). This centralization of the drawing operation ensures that images will always be drawn in the same way, with the same brightness and contrast and the same screen-fit characteristics.

Finally, using custom multithreading allows different image buffers to be filled every time a grab call is made, removing the need for copying buffers as is the case when using a callback. This is important in phase-stepped shearography, as captured frames are treated differently depending on which order they were grabbed. It is primarily this characteristic that makes custom multithreading achieve higher display rates compared to the equivalent callback-implementation.

3.3.4 Installer packages

It was decided that an installer package should be written for the developed software, so that installation could be automated and standardized across different installations. Most enterprise-level software use automated installers, which remove the responsibility and complexity of the installation procedure from the user. There were different options available when choosing how to package the developed software, namely:

- *MakeMsi*
- Visual Studio Installer

3.3.4.1 *MakeMsi* Installer scripting language

MakeMsi is a free installer scripting language which offers the user fine-grained control of every aspect of the installer package. It creates Windows Installer Packages (*.msi) by reading a build script and compiling the installer file.

Every aspect of the installer can be configured, from creating shortcuts and performing 'custom actions' to performing nested installs. The disadvantage to using this build system is that it requires knowledge of the scripting language, and is not available using a graphical environment.

3.3.4.2 Visual Studio Installer

InterDev, a component of the Visual Studio installation, contains the Visual Studio Installer program, which is capable of making Window's Installer packages. The graphical environment is significantly simpler than using *MakeMsi* scripts, but also lacks the fine-grained control of *MakeMsi*. Visual Studio Installer lacks a simple mechanism for performing custom actions, which are required for installation and uninstallation of camera drivers.

4. FINAL SOLUTION

The final solution to the design problem was formulated based on the concepts that were generated, where each concept was investigated and the most suitable chosen. This section starts by detailing the way in which each of the conceptual designs was considered and presents the resolution of the conceptual design process. Thereafter, the final solution is described in detail, with specific information regarding the implementation of each of the subsystems.

4.1 Resolution of conceptual design

4.1.1 Type of housing adopted

The types of housing that were considered for use in the project were a specimen-mounted housing (a 'vacuum hood') and a tripod-mounted housing. A tripod-mounted housing was adopted for two main reasons:

- Designing the shearography head to be specimen-mounted limits the ways in which the housing can be used. The low-cost shearography system is aimed at opening up new fields of application to the use of shearography as an NDT tool, thus the design solution adopted should be applicable to the widest range of applications possible; the tripod-mounted head can be used a greater number of different applications.

- Making use of a tripod-mounted head design does not preclude the use of the same shearography head in a special specimen-mounted application. In fact, the development of a specimen-mounted housing for UCTs existing tripod-mount shearography head is the subject of a concurrently-running MSc dissertation.

Thus the housing adopted is to design a shearography head that can be mounted on a tripod-stand and be repositioned manually by the user.

4.1.2 Housing construction used

Two different housing construction methods were considered:

- Machining the housing from a solid block of material (aluminium).
- Creating a new housing by using folded sheet metal and cover plates, making use of a solid base plate for mounting components.

The solution adopted was to manufacture the housing from folded sheet metal, as it was anticipated that this would offer cost savings, and be less wasteful.

4.1.3 Internal layout adopted

Several different concepts were considered for the internal layout of the testing unit. Firstly, an evaluation of the relative merits of different configurations of a Michelson shearing interferometer was performed to determine whether or not one configuration had any particular benefits over the other. As the two different layouts were functionally equivalent, the discussion was limited to the ergonomics of the placement of the shearing optics.

It was decided that, since the camera lens adjustment controls (lens and focus) as well as the laser direction controls were on the back side of the shearography head, the controls for adjusting the shearing optics (pan and tilt) should also be located there. Figure 36 shows the adopted configuration of the Michelson shearing interferometer, where the top (in this picture) component is a phase-shifting actuator.

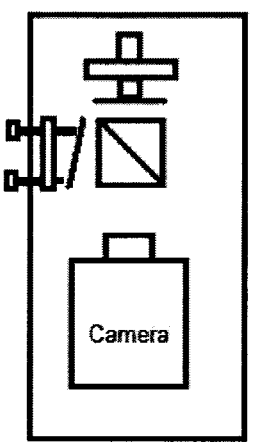


Figure 36: Michelson shearing interferometer layout adopted

A cube beamsplitter was employed instead of a more cost-effective plate beamsplitter. Among the reasons for this was that cube beamsplitters do not suffer from 'ghost' images, and are less prone to damage by normal use. In addition, a cube beamsplitter fits well into the physical layout of the shearography head by blocking and sealing the viewing window. High quality first-surface mirrors were employed in the shearing optics, as normal rear-surface mirrors also propagate 'ghost' images.

As the low-cost shearography system would be making use of a manual adjustment lens, three different means of performing focus and aperture adjustment were proposed. These methods were:

- Manual adjustment by positioning the lens externally, before passing through the beamsplitter
- Using worm gears to adjust an internally mounted lens
- Using spur gears to adjust an internally mounted lens

The last of these three solutions was adopted, as it was simple and cost effective; it avoided the use of more expensive worm gears which could not be sourced locally, and it avoided potential problems that have been experienced in the past by the NDT laboratory with externally mounted lenses. Figure 37 illustrates the use of spur gears.

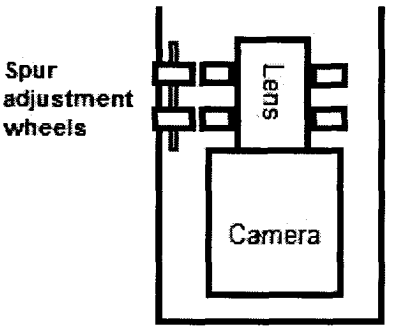


Figure 37: Spur adjustment gears

The position of the laser module on the shearography head is not of critical importance, except for the fact that it should illuminate the same area that is in the field-of-view of the camera. As the laser used was longer than the width of the shearography head, it was mounted longitudinally on the top of the head, with an adjustable 45° mirror plate to direct the beam onto the specimen.

There was also one different layout alternative that was considered, where the electronics boards were top-mounted behind the laser module, so as to minimize the overall envelope of the shearography head. This concept was eventually not adopted, although it was pursued until the manufacturing drawing stage. The reason that it was not adopted was because of the fact that the further miniaturization of the shearography head required that assembly would be significantly more difficult and cluttered. While the compactness of the head is an important concern, the new low-cost shearography system is still in the developmental phase where the focus on functionality and technology are more important that compact size and miniaturization. This was the primary reason that this layout was not adopted, though it may be further pursued in further design iterations.

4.1.4 Camera selection

The selection of the type of camera to be used was narrowed down to two options based on the design requirements, USB or FireWire. A FireWire camera was chosen, owing to the greater power bus available on FireWire cameras. The extra bus power might be used in the final solution to power other peripheral devices in the shearography head, namely the laser and piezo controller.

Care was also taken to choose a camera that offered additional features that might allow serial communication with the piezo controller board. It was found that many of the cameras offered such a feature.

Finally, the most cost-effective camera was selected. The camera is a PixeLINK PL-B771F-BL camera was purchased. This board-level FireWire camera has several desirable additional features:

1. It enables communication with auxiliary devices on the same cable using the I^2C serial communication protocol.
2. It enables other devices to be powered from the FireWire cable by offering a direct connection via an onboard connector to the FireWire power bus.
3. It has an auxiliary power connector for use in the case where there is no FireWire bus power.
4. It offers a simple and easy-to-use SDK (Software Development Kit) for controlling camera functions.

4.1.5 Piezo controller board concept adopted

The type of piezo controller board employed was decided on before the piezo as chosen, so that the available reference voltage for controlling the piezo would be known.

Three different alternatives for the piezo controller were considered:

- Purchasing a USB controller
- Designing a preconfigured multiplexer switching method using GPIO pins
- Designing a custom serial DAC, communicating with the host application via the camera cable, the camera and a serial link.

The choice between these alternatives was dependent on the type of camera which was used and which features it supported. Ideally, the latter solution would be adopted as it allows for the phase voltages to be software-configurable. Fortunately, a cost-effective camera (see the previous section) was found which supported the necessary serial communication link, thus the concept of a custom DAC with serial communication capabilities was adopted.

This solution was also ideal in that the same cable that controlled the camera might be used to control the piezo actuator. The elimination of extra cables was considered an important advantage to the usability of the new shearography system.

4.1.6 Piezo selection

When selecting a suitable piezo actuator, there was an option to use a packaged or unpackaged actuator. Unpackaged actuators are significantly more cost-effective, but require more specialized mounting considerations. It was decided that one of each type would be purchased for the sake of comparison.

As discussed in the conceptual design, the selection of the piezo is based primarily on the travel required and maximum controller voltage available. Much of the following discussion concerns the maximum piezo controller voltage that will be available using the configuration adopted so far.

By the time that the selection of the piezo actuator was being made, it was known that a FireWire camera, the PL-B771F-BL would be used in addition to a custom

designed controller, which would derive its power from the FireWire bus. Using this configuration means that the available voltage for piezo actuation is limited by the FireWire bus voltage (unless DC-DC converters are used to raise this voltage, but these have several other undesirable characteristics which are discussed later under the DAC design section).

However, the FireWire bus voltage is typically very noisy and needs to be regulated for the following reason:

- Optical phase-shifting applications require that the mirror is shifted accurately and precisely. The displacement of the piezo is sensitive to electrical noise on the controller's signal, hence care should be taken that the piezo controller's voltage is free of noise, otherwise the quality of the phase-shifting results may be adversely affected. The 12V that is typically available on the FireWire bus has a significant level of digital noise superimposed (approximately 400mV of high frequency noise), and this needs to be filtered before it can be used to actuate the piezo.

The process of filtering usually requires that voltage 'headroom' is left between the input and output voltages. This means that the maximum available voltage for actuating the piezo is less than the 12V available from the FireWire bus. In fact, as line regulation (due to the variation in different FireWire bus voltages) and aggressive noise filtering will be required; several volts of 'headroom' will likely be required.

Thus, a conservative estimate of the maximum piezo controller voltage of 5V was made, leaving headroom for filtering and line regulation. A supply of 5V was estimated used in the conceptual design, so using the specification derived there; a displacement-to-voltage ratio of at least 66ηm/V is required.

Hence the two most cost effective alternatives for each of the unpackaged and packaged options were purchased:

- Mad City Lab's unpackaged PZT3 (120ηm/V)
- Physik Instrumente packaged P820.10 (150ηm/V) were purchased.

4.1.7 Illumination source used

The ongoing development of laser diode arrays and temperature controllers at UCT's NDT laboratory is offering new alternatives of laser illumination. Unfortunately, at the time of development, none of the alternatives were ready for deployment in a low-cost, compact application.

In addition, the development of a custom laser illumination solution was deemed outside the scope of this project. As such, a 3rd party laser diode module (with integrated TEC controller) was used. The laser module was a compact red laser, with a wavelength of 660nm and 100mW laser power. The same laser module is used by UCT's existing shearography system.

4.1.8 Programming language and libraries used

C++ was chosen over *MATLAB* as a programming language primarily on the bases of cost and speed. *MATLAB* is less cost effective than a C++ development environment such as *Visual C++*. Also, the processing speed of *MATLAB* is slower as it is an interpreted language, whereas C++ is compiled.

Visual C++ was chosen as a development environment, as it allows simple use of MFC, and offers complete templates for Windows desktop applications.

In order to take advantage of the advanced features of the *PixeLINK* camera that was used (such as serial communication which was used in the design of the controller board), the *PixeLINK* software development kit was also purchased and used.

Due to the limited image-processing ability of the *PixeLINK* SDK (it is primarily an image acquisition library), most of the image processing was written using the C++ basic types. Where additional image processing utility was required, the open-source *CImg* toolkit was used instead of the MIL system, as *CImg* is free and contains the imaging functionality that was required.

4.1.9 Application architecture adopted

Three different concepts were considered for the way in which the shearography application splits time between user-interface tasks and image-processing tasks. These options were timers, callbacks and custom multithreading.

In order to determine which of these options was best suited to the shearography application, all three were implemented and tested for speed and reliability.

While all three concepts were found to work, by displaying the video feed and allowing the user to interact with the user interface, the custom multi-threading approach worked the best by being more configurable, running faster, and maintaining a more responsive user interface.

4.1.10 Data reduction algorithm, or 'bucket system' adopted

There are several different data reduction algorithms that may be used in the process of phase-stepped shearography. Most common among these are the three- and four-bucket systems, where either three or four speckle images are combined to form a composite shear image. The literature review contains further discussion on the implementation of these algorithms.

It was decided that the four-bucket system should be used, as this system is widely recognized and has been used successfully in the past by UCTs NDT laboratory. Modifying the application to use a three-bucket system is still possible and would require some minor modifications.

4.2 Conceptual description of final solution

Figure 38 illustrates the conceptual design of the final solution. The use of a FireWire camera allows excess bus power to be used to power the laser and piezo. Furthermore, the camera used has an external serial port, which allows the shearography software application to send phase-step information to the controller, which converts it to an analogue voltage and applies it to the piezoelectric actuator, which causes the mirror to shift in time with the image processing routine in phase stepped shearography.

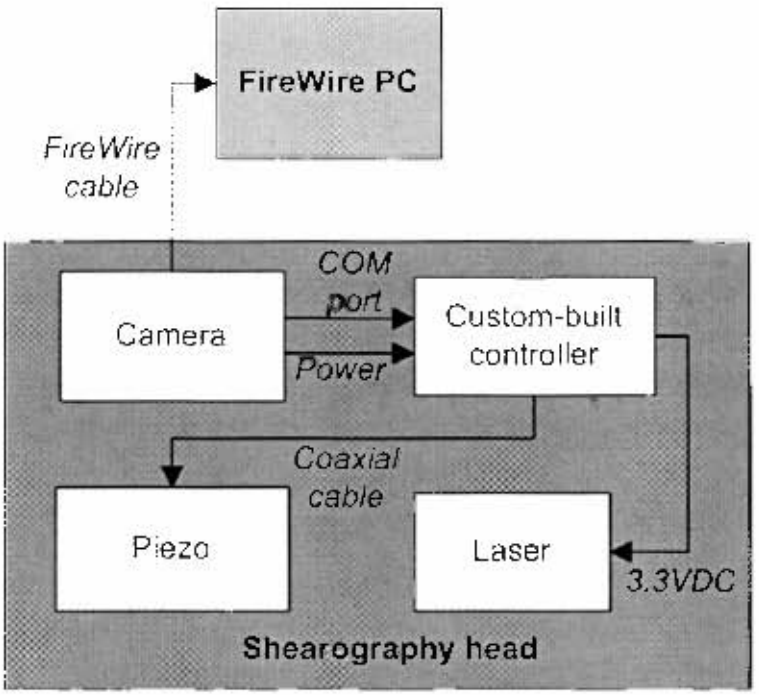


Figure 38: System design for the low-cost testing unit

The software application that was written to operate the shearography head is called *Inspector 2.0*. The application was written using *Visual C++* and custom multithreading and in contained in a *MakeMsi* installer. The next section discusses the implementation of the final solution in detail, including the mechanical construction, piezo controller and software design.

4.3 Mechanical construction

The shearography head measures 72 × 74 × 224 mm, and including the laser housing measures 113 × 74 × 224 mm. It has been designed for operation from a tripod stand, and includes a laser diode module, shearing optics, digital camera, piezoelectric actuator and controller. The housing is constructed from folded sheet metal and cover plates enclosing a base plate, upon which components are mounted. Figure 39 shows an exploded view of the shearography head.

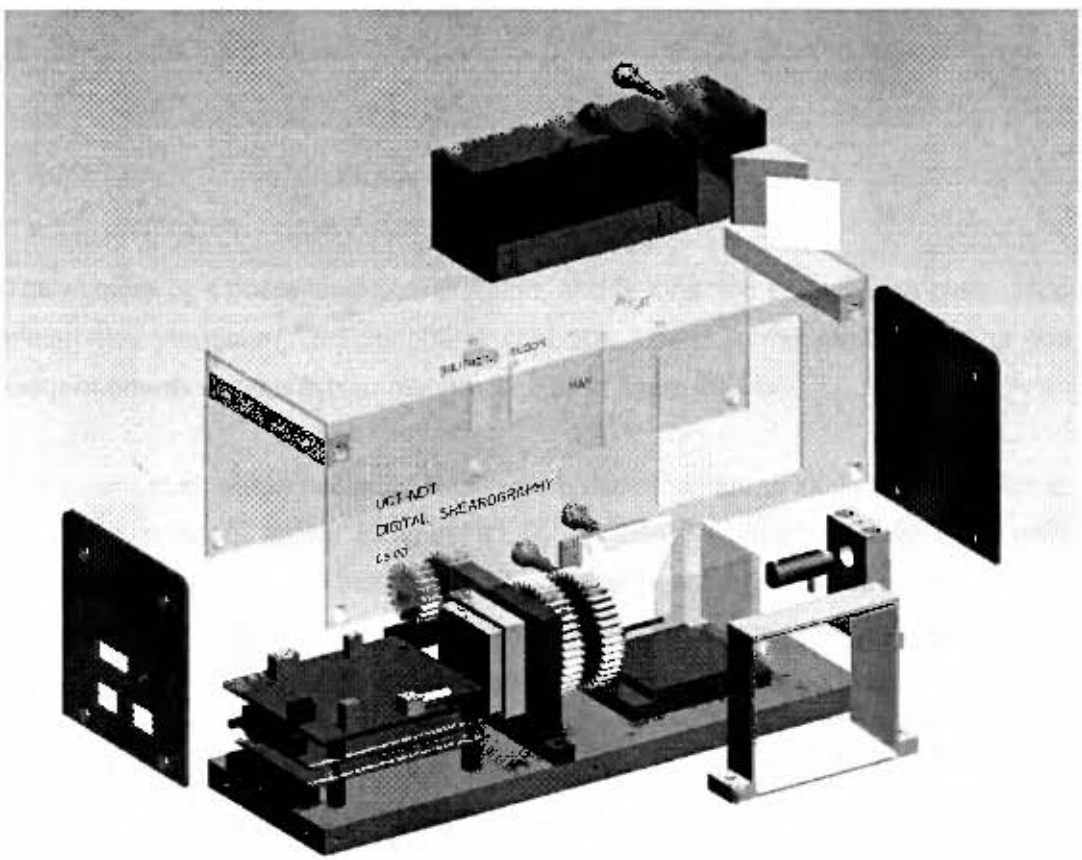


Figure 39: Exploded view of the *ProEngineer* model

The 2mm sheet metal was laser cut and then bent to size. One of the cover plates has slots cut out for the camera's FireWire connectors, which protrude out from the housing. In order to fasten the cover pates to the folded sheet metal, rectangular plate-locks are required.

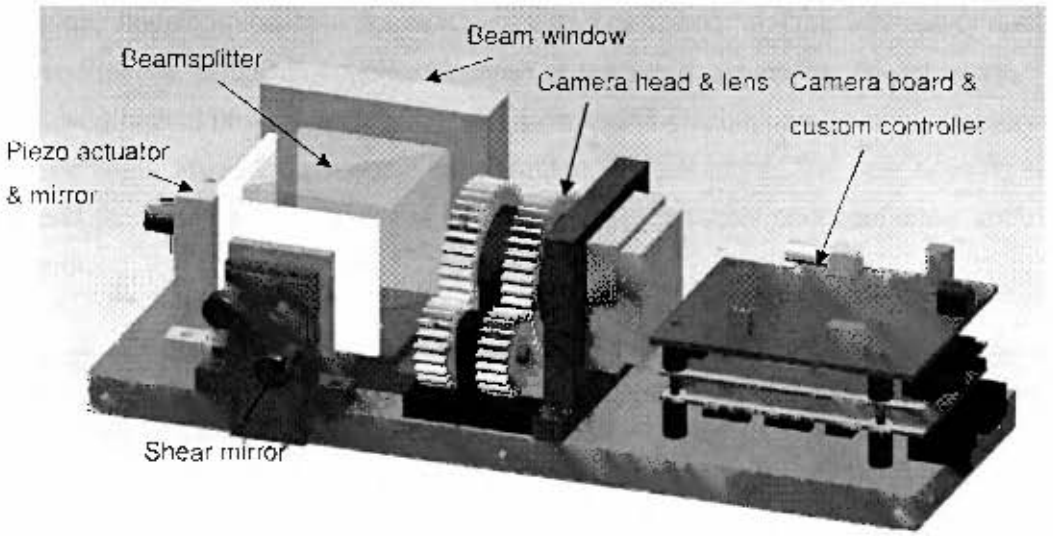


Figure 40: Rear view of layout

The camera is a board-level configuration, and is mounted on the base plate using electronics standoffs. The peripheral controller board is mounted on top of the camera boards using a similar mechanism, as in figure 40.

The laser housing was designed to allow the user simple and intuitive adjustment of the beam position, shown in figure 41. A 45° mirror plate was designed to offer Cartesian response of the beam. Using traditional 0° flat mirror plates inclined at 45°, the laser beam will not move up and down but rather in arcs, which is less intuitive. This housing is aligned with the camera to illuminate the camera specimen in the camera's field-of-view.

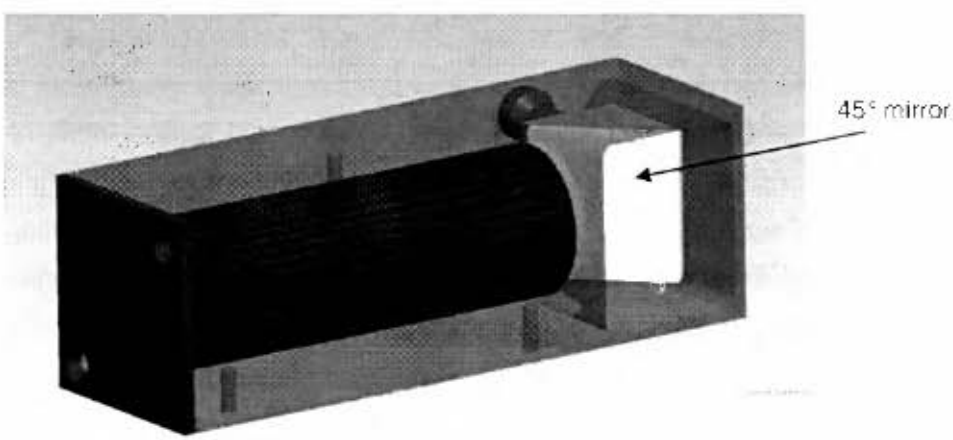


Figure 41: Laser housing

A piezo housing (shown in figure 42) was also designed, which made use of a more cost-effective, unpackaged piezo instead of the usual pre packaged actuators. The housing had to provide preloading, as unpackaged piezo's require special mounting to eliminate shear forces which can damage the actuators. The new housing was found to work well, as the piezo was able to successfully and accurately shift the mirror.

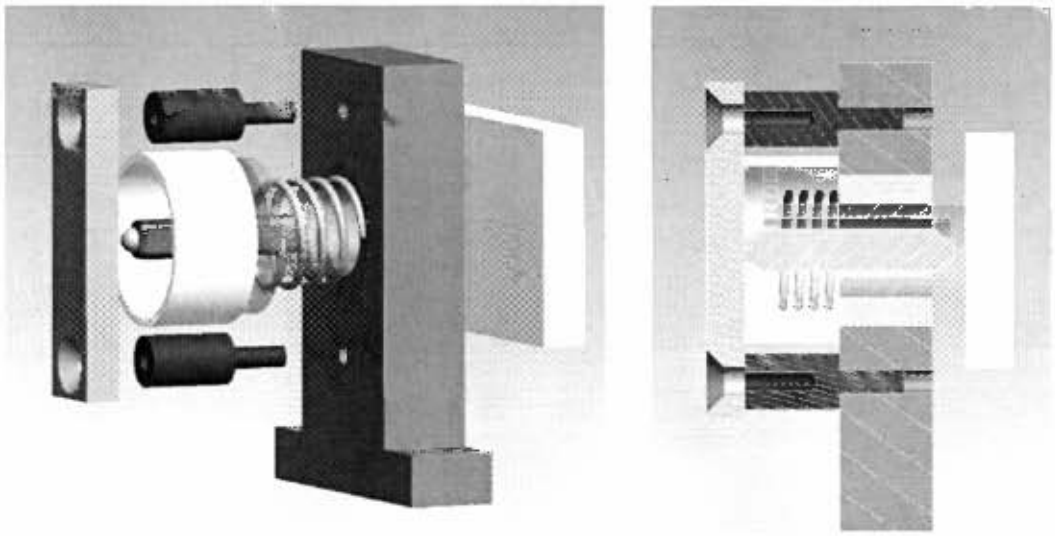


Figure 42: Exploded and cross-sectional view of piezo housing

This housing provides proof-of-concept for using unpackaged piezo's in simple phase-shifting applications. The advantage to using this approach is that unpackaged piezo's are between 3 and 6 times more cost-effective than the equivalent packaged actuators.

The method for relieving shear stress in the actuator is simple; two half-ball bearings are glued to either end of the actuator. These hemispheres sit in special locating grooves on either end of the containing surfaces. The two containing surfaces are pushed together using a preloaded spring. This assembly allows the actuator to rotate, allowing shear stresses to be relieved. It also has the benefit of offering a larger area on which to mount the mirror, compared to mounting the mirror on the piezo's end surface.

The lens is adjusted from outside the shearography housing through the use of spur gears as shown in figure 43. The spur gears protrude through the housing by approximately 2mm, allowing the user to push the gear down or pull it up to perform adjustment. The two gears are located on the same shaft but are able to turn independently on brass bushes. The spur gears on the lens are fastened to the aperture adjustment and focus rings using setscrews.

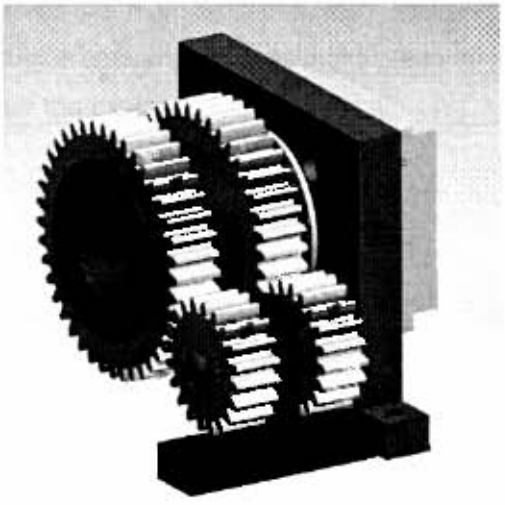


Figure 43: Use of spur gear for lens adjustment

All of the necessary controls are situated on the back panel of the unit (figure 44):

- Shear magnitude adjustment knobs
- Laser beam position adjustment knobs
- Focus and aperture adjustment wheels

Figure 44: Rear view of shearography head controls

4.4 Piezoelectric actuator controller

In an effort to reduce the cost of the shearography system, a simple DAC controller which is controlled via the camera's serial port was designed, tested and built on PCB. In addition, the same board contains a power regulator for the laser, supplying it with the requisite 3.3V, up to 1A.

The controller board serves two main functions, also illustrated in figure 45:

1. A digital-to-analogue converter to control the piezo voltage
2. Provide power for the laser module.

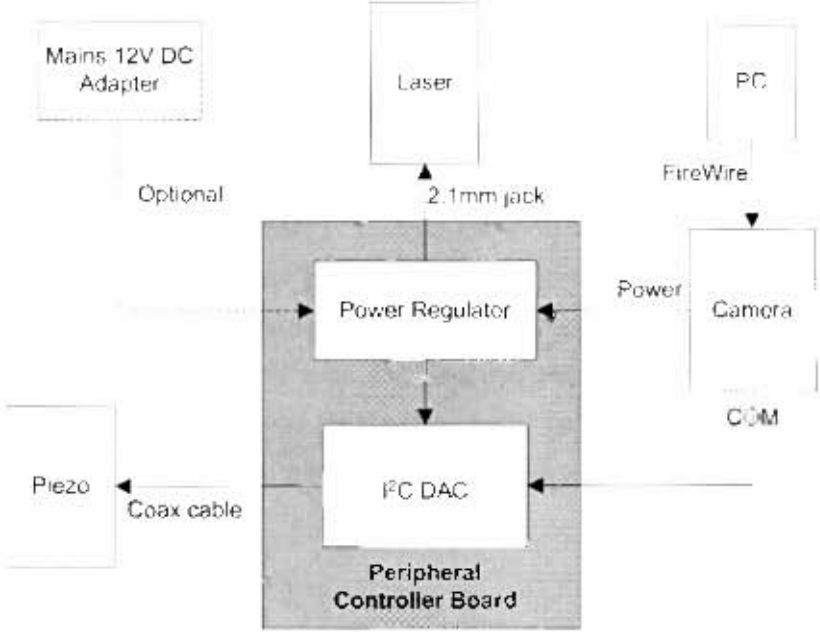


Figure 45: Conceptual design of peripheral controller board

The shaded portion of figure 45 represents the physical controller board. The controller board first regulates the power that it takes from the camera's external power connector (which is originally derived from the FireWire power bus). It regulates this voltage to 3.3V for use by the laser and a stable, regulated 5V for use as a reference voltage for the DAC.

The DAC receives serial communication (which represents phase voltage information) from the camera's COM port (which was originally sent from the *Inspector 2.0* application), and converts this digital information to an analogue voltage and applies it to the piezo actuator.

The detailed design of this board is influenced by many external factors, such as:

- The wavelength, voltage and current draw of the laser.
- The power available from the camera, and therefore the power available from the FireWire port used.
- The permissible level of noise and frequencies thereof supplied to the piezoelectric actuator.

This section describes the design of the peripheral controller board. It divided into two sections: the first describes the design of the power regulation unit, while the second involves the design of the DAC controller itself. The final solution is described last, the output of which is a PCB diagram.

4.4.1 Power regulator

The peripheral controller board derives its power from the FireWire bus. Unfortunately, the voltage and power available from the bus are not explicitly specified, and in practice varies from 8-30V (though typically 250mA at 12V, or 4W). Hence a power regulation unit was required on the controller board, to supply regulated power to the laser and DAC. The power requirements for each of these subsystems were different. The following two sections describe the detailed design of each.

4.4.1.1 Laser power regulator

In order to design the power regulator for the laser, some of its specifications must first be known. The specifications of the laser module that was selected are tabulated in table 4.

Table 4: Laser specifications

Specification	Value
Laser supply voltage	3.3V
Maximum laser power	2W
Laser operating current	0.5A
Laser connector	2.1mm jack

There are essentially two types of voltage regulation available in this application, which operate using different principles:

1. Linear or shunt regulators (e.g. LM317)
2. Switched-Mode Power Supply (SMPS) or DC-to-DC converter

4.4.1.1.1 Linear regulators

Linear regulators are cost-effective and use solid-state technology to drop the voltage and reduce noise, but are inefficient when dropping a large voltage. The power lost by typical linear regulators is equal to the drop in voltage multiplied by the current supplied, hence the power that must be supplied to the laser power regulator is:

$P_{supply} = P_{laser} + P_{lost} = P_{laser} + V_{drop} \times I_{supplied}$
Therefore, $P_{supply} = 2W + (12V - 3.3V) \times 0.5A = 6.4W$

This power required exceeds the power that may be supplied from the FireWire bus (approx. 4W); hence external power would be required. This is possible yet undesirable, as the original intention was that all necessary power would be supplied over the FireWire cable.

4.4.1.1.2 DC-DC converters

DC-DC converters use a more complicated switching design to both raise and drop supply voltages. They achieve far greater efficiency (around 80-95%), but are less cost-effective and more noise-susceptible than comparable linear regulators. They are available for many different voltage levels, and can be large and bulky, particularly where large currents are required. The advantage of using a DC-DC converter is primarily the greater efficiency which is achieved.

A 3.3V output DC-DC converter was sourced, which claimed an efficiency of 85%. Calculating the power that would need to be supplied to the laser,

$P_{supply} = P_{laser} / \eta$
Therefore, $P_{supply} = 2W / 0.85 = 2.36W$

This power can be supplied via the camera; as such a 3.3V DC-DC converter was used in the power regulator for the laser, instead of a linear regulator.

4.4.1.2 DAC power regulator and reference voltage

The digital-to-analogue converter requires a reference voltage from which it performs conversions. In order to ensure that a changing reference voltage does not interfere with reliable phase-stepping by influencing the piezo voltage, the reference voltage must not:

- Vary over time, or vary significantly between different FireWire connections.
- Have a large noise component.

Deriving a stable voltage from the power available from the FireWire bus was a challenging task, as the power bus is extremely noisy due to the large component of digital noise, and the fact that the DC-DC converter used in the design of the laser's power regulator also superimposed high frequency switching noise on the power lines. It was found that the noise component on the 12V power bus was approximately 400mV.

Fortunately, the maximum required piezo voltage was calculated to be 5V for a full phase step, allowing significant 'headroom' for filtering the supply. Additional line regulation was required, as the FireWire bus voltage is not explicitly specified and can range from 8V to 40V. A two stage regulator was implemented, with low-pass filters between each stage for filtering.

The first regulation stage used a variable linear regulator, an LM317, to regulate the voltage to approximately 7V. The benefit of using a linear regulator such as the one used is that they also excel at noise rejection. This 7V was then regulated in the second stage using a precision 5V reference IC, a REF02 chip. This 5V signal was then used as a reference voltage for the DAC.

However, it was found during testing that the reference voltage still had a 20mV noise component, which was deemed unacceptable, as this translates to a significant displacement error for the piezo.

In order to solve this, more aggressive noise suppression was used; a 1mH inductor and 1mF capacitor were used in a LC low-pass filter to isolate the DAC reference and power supply from the rest of the circuit. This modification resulted in a significant improvement; the reference voltage was stable to within 2mV.

4.4.1.3 Auxiliary power input

The peripheral controller board has been designed with an auxiliary power input, for use when connecting to a laptop. The connector is visible in the board schematic as a dotted, optional connector.

A Ø2.1mm 12V connector on the controller board routes power directly to the camera's auxiliary power pin. How exactly the 12V is derived does not matter; one may use a 12V DC wall adapter or even a 12V battery. The entire unit only draws 6W, making battery option viable. In general however, it is preferable to use a laptop adapter kit, as it would likely contain better over-current protection.

4.4.2 DAC piezo controller design

The DAC, or piezo controller, functions as an actuator for the piezo in phase-stepped shearography. Hence, the performance requirements of the DAC depend on the overall displacement requirements for the piezo and also the specifications of the piezo used. Two different piezo's were used in the shearography application, for the purposes of comparison between packaged and unpackaged actuators. As the specifications for both piezo's were similar, the controller board was specifically designed for only one in particular. The specifications of this piezo (the unpackaged piezo) are presented in table 5.

Table 5: Specifications for unpackaged piezo

Specification	Value
Voltage-displacement response	120ηm/V
Maximum displacement/voltage	18μm/150V
Capacitance	0.35μF
Push/pull force capacity	200/20 N
Open-loop resolution	0.15μm

When designing the DAC controller, it was considered important that, since this is a custom design, it should be benchmarked against other piezo controllers to ensure that its resolution and noise specifications were suitable for actuating the piezo. The piezo controller that the new design was benchmarked against was the PCI extension card that UCTs existing shearography system uses, as this card is specifically designed for use in phase-shifting applications. The specifications for the benchmarked controller are shown in table 6.

Table 6: Specification for benchmarked piezo controller and piezo system

Specification	Value
Piezo voltage-displacement response (max)	40ηm/V (6μm/150V)
PCI Controller voltage resolution	150V/14bits (9.1mV)
PCI Controller noise level	5mV
Overall displacement resolution	0.4ηm
Overall displacement noise	0.2ηm

Note that while the voltage output characteristics of the controller board are important, these should also be combined with the piezo specifications used to determine the piezo displacement resolution and noise, as in table 6..

In order to achieve a similar displacement resolution to the benchmarked system, the required resolution for the DAC should first be established. To achieve a resolution of 0.4ηm, using a 5V reference DAC actuating a 120ηm/V piezo, a DAC resolution of '10.6' bits is required. A 12-bit DAC was used, though a 10-bit one would probably also have sufficed. The 12-bit device coupled with the 120ηm/V piezo gives a displacement resolution of 0.15ηm; better than the benchmarked system.

The camera that is used in this application is the *PixeLINK* PL-B771F-BL camera, whose serial port uses the I^2C protocol. It would be convenient if the digital-to-analogue IC used could communicate directly with the camera's I^2C port. A range of DACs were found which listed I^2C as their communication interface. An I^2C DAC from Texas Instruments was found to be a good solution: it operates directly from the I^2C bus, has 12-bit resolution, uses a 2.7 to 5.5V supply and is capable of a settling time of 10μs. Thus the DAC selected was the TI DAC7571.

4.4.3 Timing considerations

Another consideration in the controller board design is that the DAC controller has to respond fast enough for the piezo mirror to be adjusted during video capture, without causing excessive delays in capture. Video capture happens at a maximum of 25fps, giving 40ms between frames. The DAC should be faster than this in order for it not to significantly impede the video feed. The general guideline followed was that the complete mirror shift should take place within 5ms.

There are three places in the DAC-piezo actuation process where timing delays which may affect image-processing may occur:

- The communication between the camera and the DAC controller.
- The settling time of the DAC IC.
- The actuation time of the piezo, once the voltage is set by the controller.

Note that the time that is taken to communicate the phase-voltage information from the host application to the camera over FireWire will be significantly faster than the time taken for the same information to be relayed from the camera to the controller board over I^2C, as the FireWire baud rate is more at 1000 times faster that the I^2C protocol. It is for this reason that any delay in FireWire communication is not considered here. The three delays above are discussed further in the following sections.

4.4.3.1 Controller communication delay

The camera's COM port operates at approximately 100kb/s (the DAC IC is capable of higher speeds, up to 3.4Mb/s). An I^2C baud consists of 9 bits, and 3 bauds are required to set the DAC voltage for DAC IC used. Thus, the communication between camera and piezo takes approximately 270µs.

4.4.3.2 DAC settling time

The settling time of most digital-to-analogue converters is in the order of microseconds, but the exact figure depends on the IC used. The settling time of the IC selected was 10µs.

4.4.3.3 Piezo actuation time

The time taken for a piezo to shift depends on both the magnitude of the shift and the current supplied by the controller. Piezoelectric actuators can be modelled roughly as capacitors (the selected piezo has a capacitance of 0.35μF). Hence, the piezo driver is an RC circuit, where the resistance is in the cable or present as an additional current-limiting resistor. In figure 46, V_{in} is the output voltage of the piezo controller.

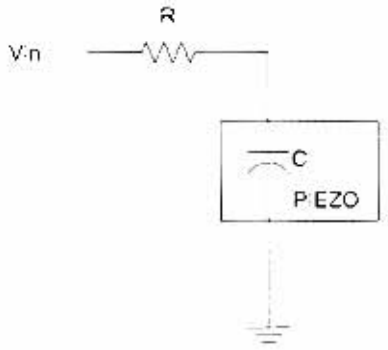


Figure 46: RC circuit representing piezo circuit

The greater the current supplied, the faster the shift will be. A current-limiting resistor is required as there is a limit on the current available (the power available from the camera's FireWire bus is limited). A series resistor is also recommended for driving large capacitive loads (such as a piezo-electric actuator) from an operational amplifier, in order to maintain output stability.

Modelling the piezo as a capacitor, one can model the output voltage and current, as in figure 47:

$$V(t) = V_{step}\left(1 - e^{\frac{-t}{RC}}\right)$$

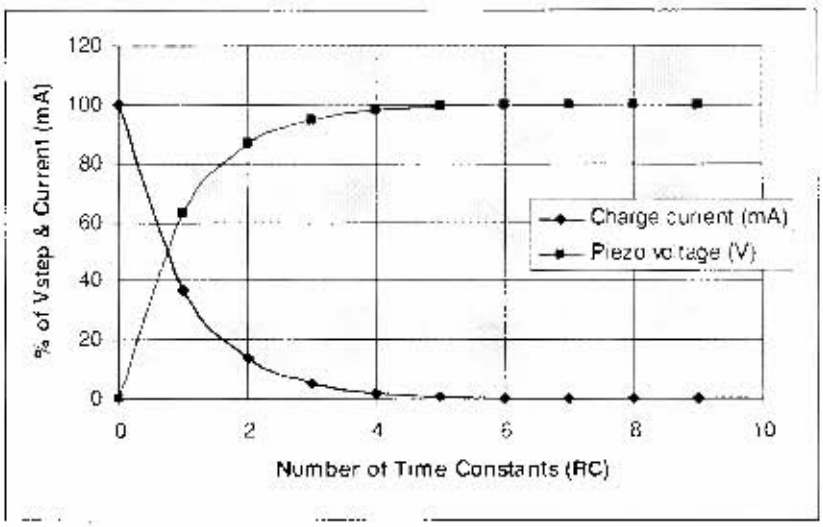


Figure 47: Typical piezo voltage and current as a function of RC time constant.

One part of the time constant is already known; the capacitance of the piezo is 0.35uF. One can therefore investigate the relationship between the charging, or actuation time and the current-limiting series resistor, as in figure 48 below.

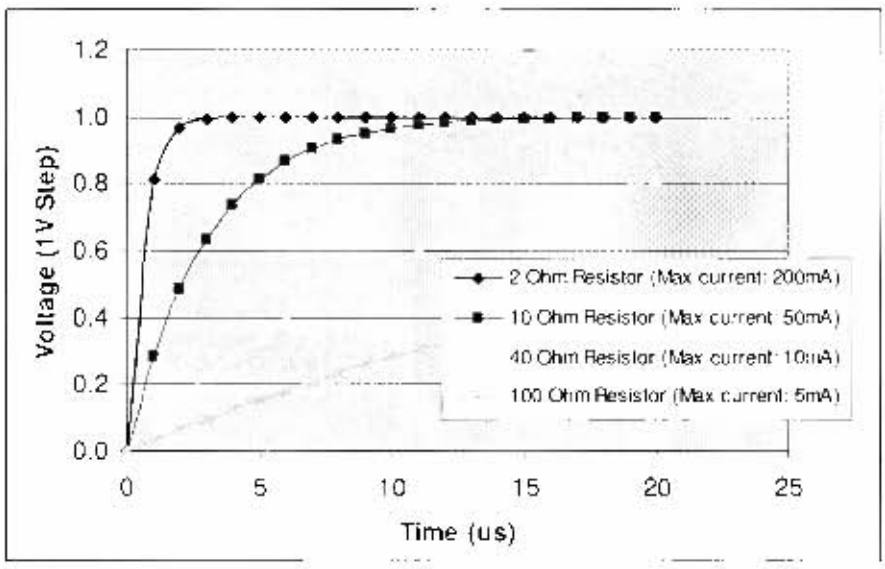


Figure 48: Capacitor charging time as a function of series resistor (C=0.35µF)

Decreasing the resistance of the series resistor results in a faster charging time, however draws more current. A 40Ω resistor will give a reasonable 60µs charge time (to 0.99V of 1V step) while only drawing 10mA. With a 5V step, the actuation will take place in 300µs.

However, the current output of the DAC V_{out} pin is limited to 5mA, as is typical for precision DAC's. Hence, in order to be able to source additional current, a current buffer is required between the DAC's V_{out} pin and the piezo. A precision rail-to rail operational amplifier was used for this purpose (LT1006CN8), connected in a negative feedback voltage-follower configuration.

4.4.3.4 Total shifting time

In summary, the time taken to shift the mirror in phase stepping depends on the communication speed, DAC settling time and the piezo actuation time (related to the current which the controller is able to supply). Figure 49 shows the sequence in which these processes occur.

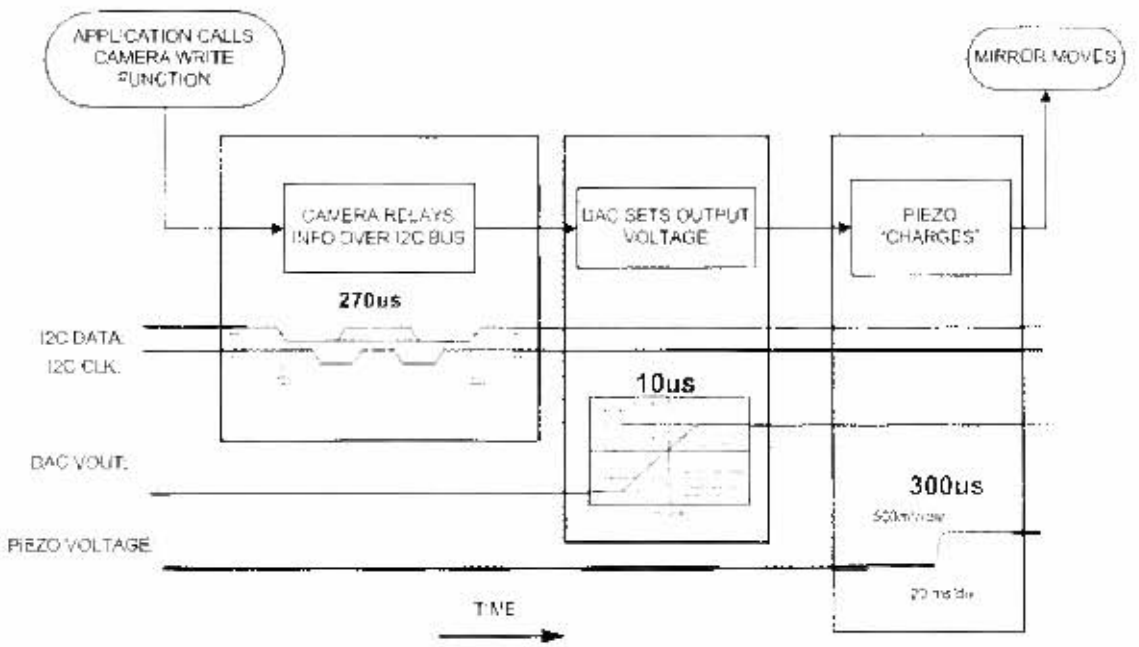


Figure 49: Timing diagram for complete piezo communication

The level of impact that the design of the controller board can have on the first two of these processes is limited; the communication time is determined by the camera's I^2C bus frequency and the DAC settling time is a function of the IC selected. However, the speed at which the piezo is actuated depends on the amount of current which is supplied by the controller board. Using a series resistor of 40Ω, the piezo should achieve a full shift within 300µs.

In total then, the shifting time should be the sum of the three delays.

$t_{total} = 270\mu s + 10\mu s + 300\mu s = 580\mu s$

This should not significantly impede the video processing, as new frames are captured at 40ms intervals, and 580µs is an order of magnitude faster than the general guideline of 5ms which was initially specified.

4.4.4 Controller board PCB design

The solution was built and tested on breadboard before moving to veraboard and finally PCB. The free version of Eagle PCB Layout Editor [43] was used to do the PCB layout. UCT's Electrical Engineering Department has PCB-building capabilities, and these were used to make to PCB board. The circuit diagram is shown in figure 50 below. Figures 51 and 52 on the following page illustrate the PCB layout and the board's location respectively.

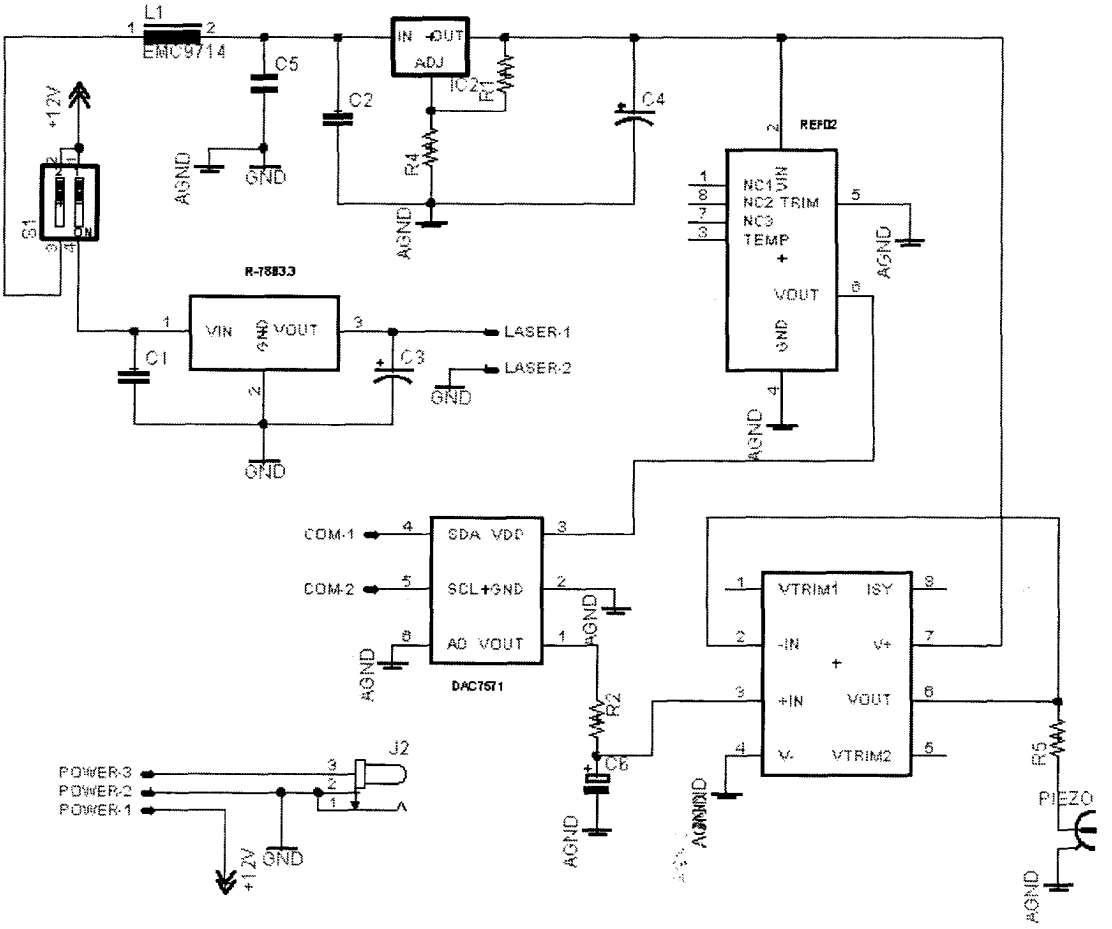


Figure 50: Controller board circuit diagram

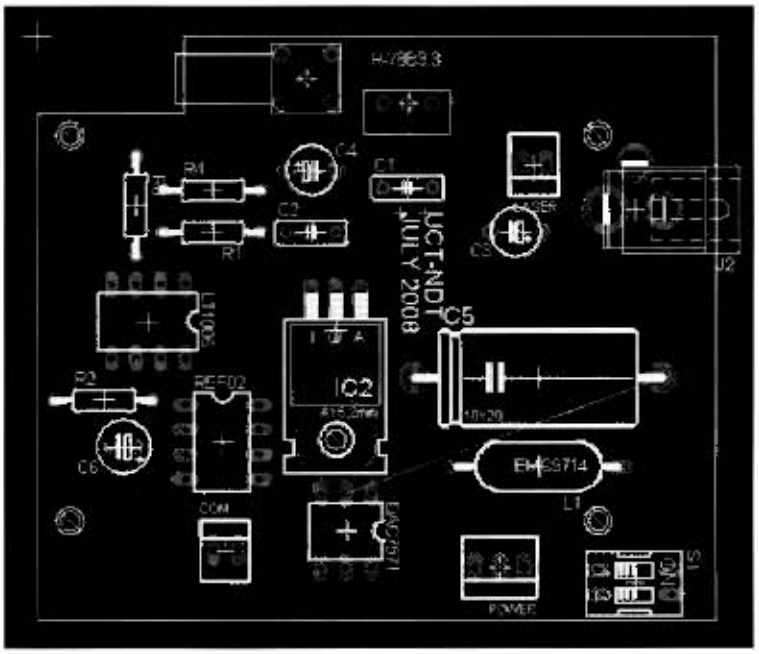


Figure 51: Controller board PCB Layout (bottom)

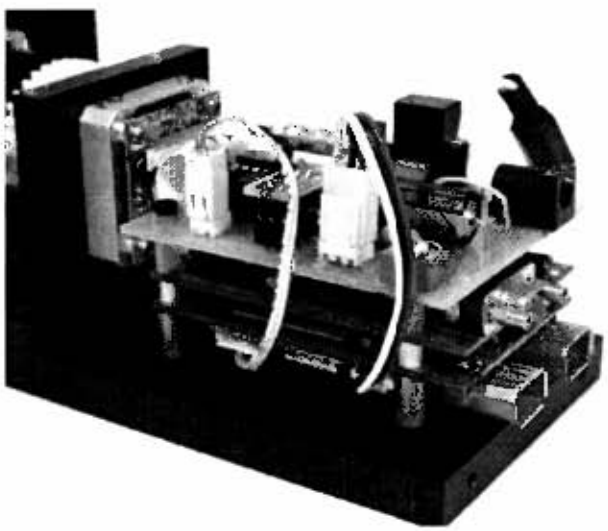


Figure 52: Peripheral controller board stacked on top of two camera boards

4.4.5 Cost

The cost of the controller board is approximately R600, which is significantly less than the PCI controller used for benchmarking. Where possible, it uses parts and connectors that easily obtainable, and stocked by the UCT Electrical Engineering Department. In some cases however, specialist parts are required which can be purchased from the UK through RS Components. The complete parts list and cost is given in table 7 on the following page, which cost R6 400.

Table 7: Parts list for controller board

Component		Serial number/value	Qty	Total Cost
Ordered parts				
60mm × 80mm PCB		Printed at UCT	1	±R50.00
Power connector[1]		Molex 35507-0300	1	R1.84
Power crimp terminals[1]		Molex 50212-8100	10	R29.10
COM port connector[1]		Molex 51021-0800	1	R5.60
COM port crimp terminals[1]		Molex 50079-8100	10	R17.12
M2.5 10mm M/F Standoff[1]		24331K-ND	8	R25.60
M2.5 5mm M/M Standoff[1]		24420K-ND	4	R8.61
3.3V DC-DC Converter[2]		Recom R-78B3.3-1.0	1	R106.47
I^2C DA Converter[2]		T.I. DAC7571IDBVT	1	R34.34
Piezo coaxial connector[2] (M)		LEMO FFA.00.250.CTAC29	1	R102.16
Piezo coaxial connector[2] (F)		LEMO EPL.00.250.DTN	1	R169.91
Precision rail-to-rail op-amp[2]		LT1006CN8	1	R37.31
Precision voltage reference[2]		REF02AP	1	R38.49
UCT Electrical Engineering Stores components[3]				
Power connector		3-Pin Molex	1	R5
Laser & DAC connectors		2-Pin Molex	2	R5
Power jack		2.1mm	1	R10
DIP switch		2-Pin	1	R8
Inductor		1mH	1	R10
Voltage regulator		LM317	1	R10
Resistors	R1	1.2K	1	R0.50
	R2	1K	1	R0.50
	R3	22Ω	1	R0.50
	R4	4.7K	1	R0.50
Capacitors	C1, C2	220nF	2	R0.50
	C3, C4	10uF	2	R0.50
	C5	1000uF	1	R0.50
	C6	1uF	1	R0.50
Total				**R596.55**

Notes:

1. Parts ordered from www.digikey.com
2. Parts ordered from RS-Components
3. Prices from the UCT stores are estimates.
4. Where Rand-Dollar conversions were required, $1 = R8 was used.

4.4.6 Performance

Overall the board performed as expected, performing reliable digital to analog conversions while powering the laser. The DAC resolution and noise level performance gave the piezo similar performance to that of a commercially available controller. Figures 53 and 54 demonstrate the phase-stepping voltages and noise level of the piezo controller respectively, both measured on an oscilloscope.

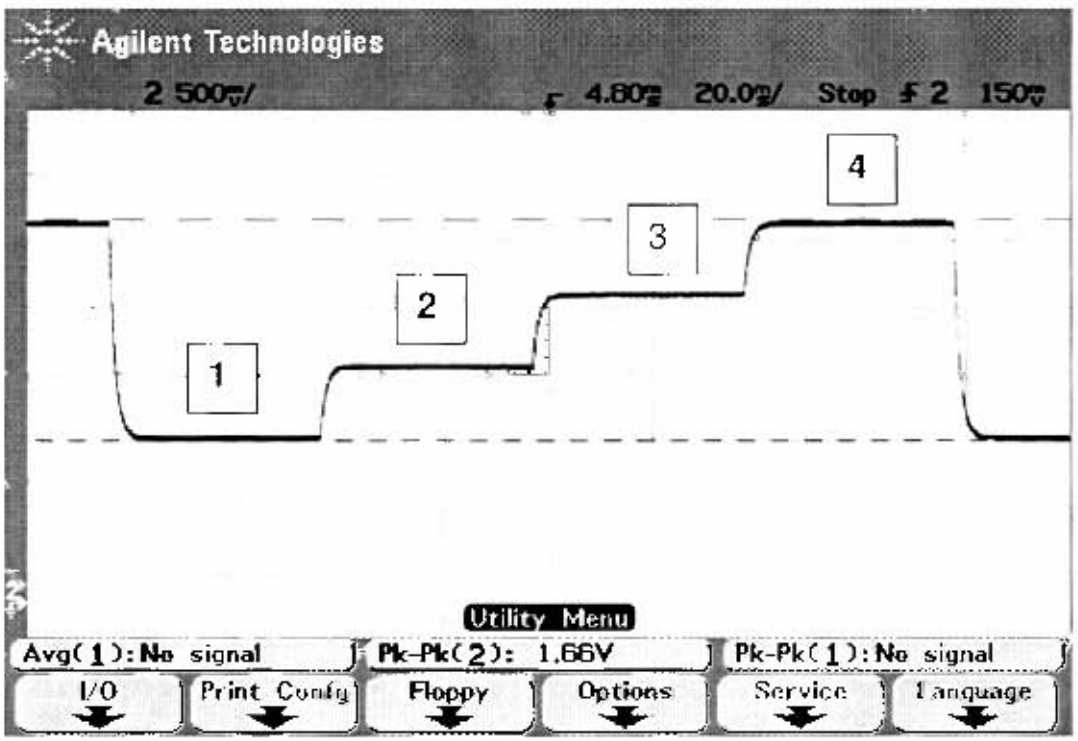


Figure 53: Phase-stepping voltage during image capture process

Figure 53 clearly shows how the piezo actuator's voltage is stepped during image capture. The labels marked 1 to 4 show where images are captured, in between these discrete levels, the piezo voltage is stepped. The step is not perfectly square because the piezo is effectively a capacitor and requires time to charge. The full step takes approximately 4ms to complete, which is not as fast as initially hoped, but adequate nonetheless. A 5ms pause was inserted in the *Inspector 2.0* code to account for this.

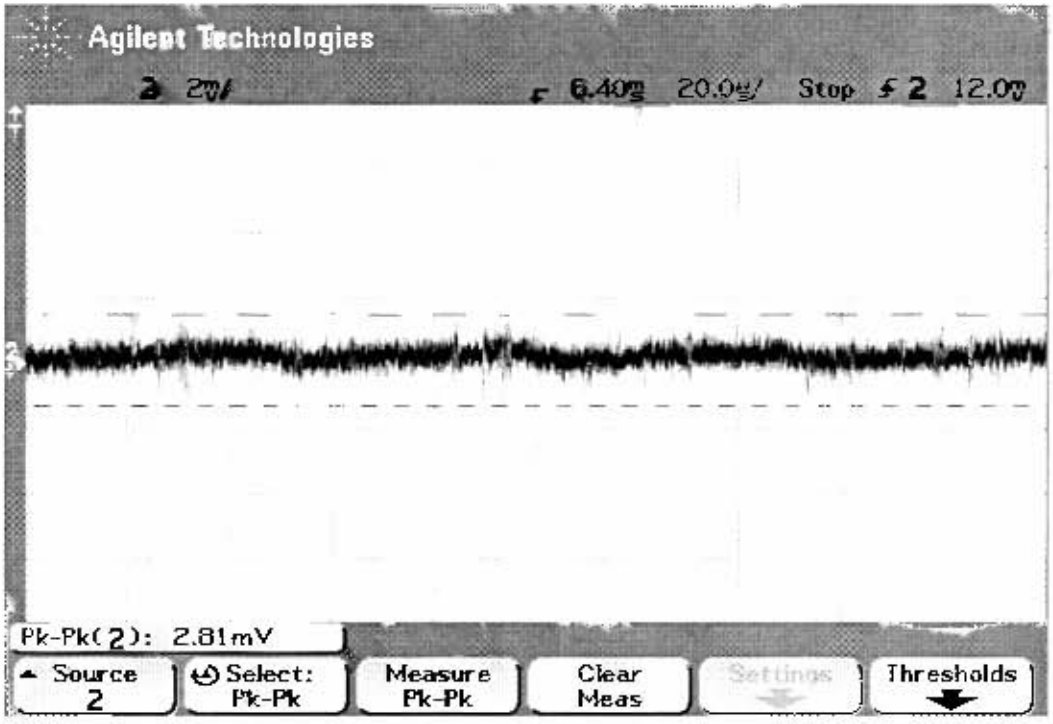


Figure 54: DAC output voltage

Figure 54 shows the AC component of the peripheral controller board's piezo voltage output. Output noise will affect the mirror's position, but there is a limit to this; the mass of the mirror has the effect of damping out high frequency noise. The noise is mostly high frequency (the time scale here is 20µs). The peak-to-peak noise is 2.8mV, although if one were to eliminate the noise above 1kHz, it appears that this noise would be closer to 1mV, which is similar to the resolution of the system.

While noise of 2.8mV is greater than the 1mV resolution, the noise component is still better than that of the PCI card used for benchmarking, which claims 5mV noise. Taking into account, however, that the piezo used here has a higher displacement-voltage ratio, the overall displacement noise is slightly worse than the benchmarked system: 0.37ηm compared to 0.2ηm.

Table 8 illustrates the differences between the benchmarked controller and the custom designed one. The new design achieves similar displacement noise and resolution specifications to the benchmarked system, but costs an order of magnitude less.

Table 8: Comparison between new and benchmarked piezo controllers

Specification		Benchmarked controller	New controller
DAC resolution		14 bit	12 bit
Range	Voltage	150V	5V
	Displacement	6μm	600ηm
Resolution	Voltage	2.3mV	1.2mV
	Displacement	0.1ηm	0.1ηm
Noise	Voltage	5mV	2mV
	Displacement	0.2ηm	0.37ηm
Cost		**R6 400**	**R600**

One of the main differences between this custom-made controller and commercial piezo controllers is that the custom-made one only operates over a 5V range, while others can operate up to 500V. However, as only a 5V range is required, 500V is surplus to requirements.

4.5 Custom software

4.5.1 Application design

The *Inspector 2.0* application was written using *Visual C++*. It uses the *PixeLINK* SDK for image capture, which has no restriction on deployment licensing.

The application uses the recommended document/view architecture of MFC (Microsoft Foundation Classes), which is an implementation framework for the widely accepted MVC (Model-View-Controller) design pattern [44].

There are three distinct operational modes:

- 'Video' mode displays unaltered video, and can be used during the setup of specimens and testing apparatus, e.g. to adjust the lens focus and aperture.
- 'Shear' mode performs intensity-based shearography, displaying the results of this process.
- 'Phase' mode performs phase-stepped shearography by phase-stepping one of the interferometer mirrors in synchronization with the image capture.

Custom multithreading was implemented to perform image processing tasks. Using multi-threading requires that the image acquisition code, contained in the worker thread, be written in such a way that it can be stopped by the user-interface thread. This can be difficult as the two different threads of execution will not be synchronised. The recommended means of synchronising threads in MFC is with 'Mutex' objects. Mutex [45] is an abbreviation of 'mutually exclusive'; they are used to allow different thread access to shared resources. Only one thread can 'hold' a particular Mutex at a time, and other threads must wait for the Mutex to be released before it can 'hold' the Mutex. Mutex's were used in the *Inspector 2.0* code to synchronize the UI and worker threads.

The flowchart on the following page (figure 55) shows how synchronization was used to ensure that the worker thread was not stopped while accessing the camera. Synchronization elements are shown in either red or green. Once the user presses 'Play', the new thread is created. Thereafter, the worker thread constantly checks whether the user has pressed 'Stop' or 'Pause' in the UI thread. If 'Stop' is pressed, the worker thread terminates, and complete control returns to the UI thread. When processing, the UI thread is idle and as such is able to respond to user actions.

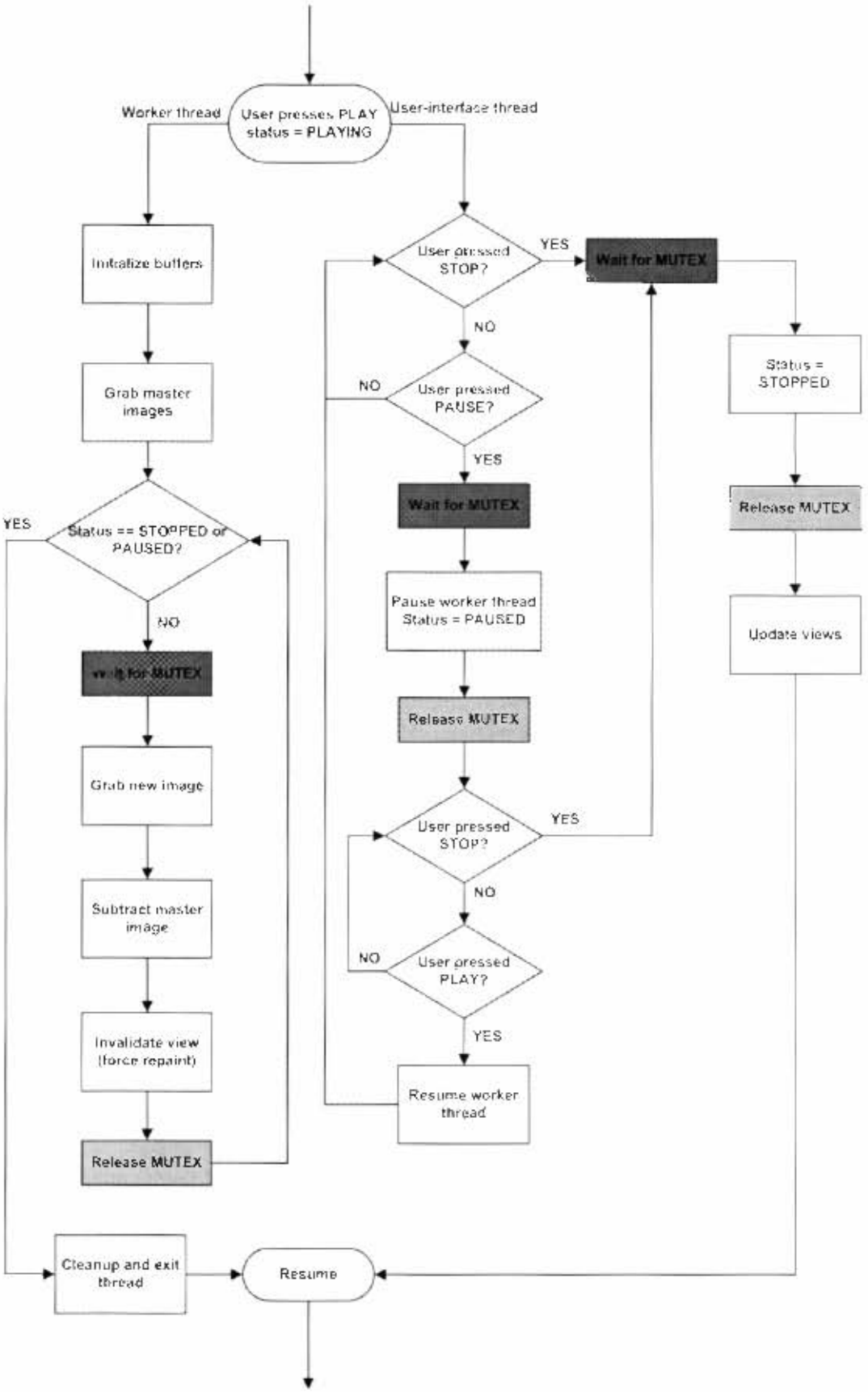


Figure 55: Play, Pause, Stop flowchart with multithreading

All of the image processing takes place in two functions, both global functions contained in the worker thread. One function, *workerThread*, performs most of the initialization and processing, while the other function, *getCompositeImage*, is used only in phase-stepping mode to simplify the *workerThread* code. The following schematic diagram (figure 56) and discussion shows the detailed operation of the worker thread once it has been created by user pressing 'Play'.

After initialization (which may include grabbing reference images), the thread enters a 'while' loop, continually grabbing and processing frames in a manner that is dependent on the operational mode. Only one buffer is used for display, *pImageDisplay*, which all operational modes share. Once this buffer has been written to, a call to *view->Invalidate()* forces the view to repaint the new image. The use of synchronization objects in display also ensures that the view does not repaint itself while being written to. The thread constantly checks whether 'Stop' has been pressed, and will exit the 'while' loop and terminate itself once this occurs.

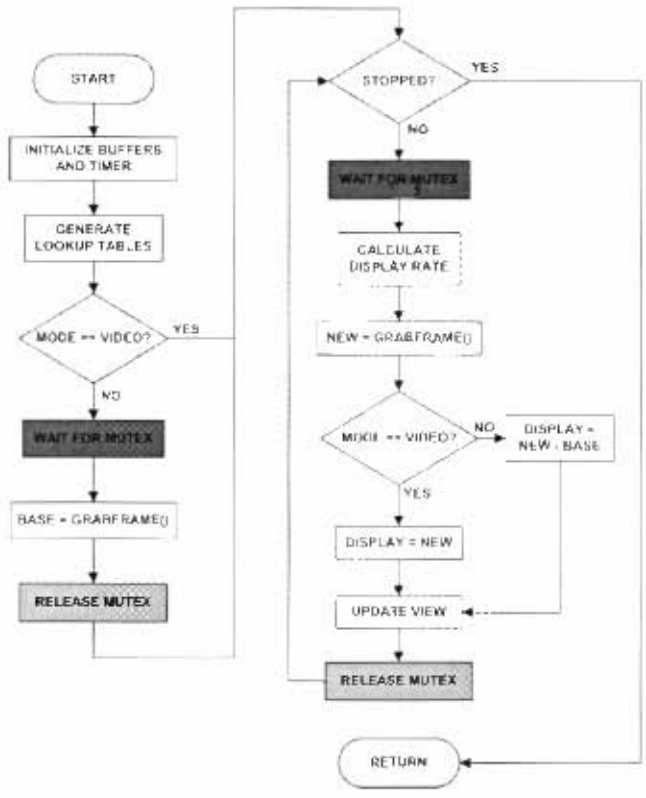


Figure 56: Worker thread flowchart

The function *workerThread* makes use of the PC's high performance timer to measure the time taken for each image loop, and calculates the rate at which new images are displayed.

4.5.1.1 Discussion of image processing routines

The following sections discuss some of details of how the new image processing routines were implemented, and contain code snippets to illustrate these concepts. *Other software concepts such as asynchronous image capture are also discussed.*

The image processing routines which are discussed here are implementations of the theory of shearography which is discussed in detail in the literature review.

Note that virtually all image-processing in shearography performs operations on corresponding pixels from different images rather than operations that involve the pixels within the same image, thus the spatial position of a particular pixel is not relevant to the calculations that must be performed on it. Taking this into account, most of the image-processing routines in the *Inspector 2.0* application treat images as one-dimensional arrays instead of two-dimensional arrays. This is made possible by the way in which C++ references multi-dimensional arrays; 2D arrays are actually special cases of 1D arrays with special look-up rules. As the array is stored contiguously, a 1D reference may be used where convenient. Hence, a 'for' loop that may have looked been declared like this:

```
for (int i = 0; i < sizeX; i++) {
      for (int j = 0; j < sizeY; j++)
```

Will now be declared using a single line:

```
for (int i = 0; i < sizeX * sizeY; i++)
```

This method has three advantages:

- Confusion involving different indices for width and height is avoided.
- Code is more compact.
- Execution will be slightly faster as one counter is maintained instead of two, and the special 'behind-the-scenes' 2D indexing is no longer necessary.

4.5.1.1.1 Shear mode

Capturing in shear mode requires that every new image acquired subtracts the base image (or master image) in order to determine the differences. Images are represented internally as an array of pixel values, each 8 bits (0-255), with black corresponding to 0 and 255 to white. When one subtracts one image from another, the result of some pixel subtractions could be negative. As it is impossible for negative pixel values to be displayed, one has to adjust the resulting image for display.

Technically, the result of the subtraction is a number between -255 and 255. One way to adjust this image for display is to linearly map the range [-255, 255] to [0, 255]. However, this would mean the pixels in the new and old image that were equal (that subtracted to 0) would be mapped to 127 or 128, which is grey. This is counterintuitive, as in shearography one would expect the result to be black.

Instead, the recommended method is simply to use the absolute value of the result, thus [-255, 255] will be mapped to [0, 255]. This way, pixels with equal intensities will stay black when mapped. The code for this process is shown here:

```
pCamera->CamGrabFrame(pImageNew, sizeX*sizeY);
for (int i = 0; i < sizeX * sizeY; i++)
      pImageDisplay[i] = abs(pImageNew[i] - pImageBase[i]);
```

However, in practice it is found that the resulting image is very dark, and fringes are difficult to differentiate. In order to remedy this, the image must be brightened and contrast-adjusted using lookup tables.

The new application uses the simplest and fastest lookup table possible; an array of 8-bit integers. This lookup table is generated before capture starts (and whenever the brightness or contrast sliders are moved), and uses the value of the brightness and contrast sliders to generate itself. The brightness lookup table linearly maps the range [*brightness*, 255] to [0, 255], where *brightness* is an integer in the range [0, 255]. This code snippet shows the brightness lookup table being generated:

```
unsigned short int LUTbrightdark[256];
int fullScale;
for(int i=0; i<255; i++) {
      fullScale = 255*(i)/(256-sliderBrightness);

      //clip to 0-255 range
      if (fullScale < 0)
            LUTbrightdark[i] = 0;
      else if (fullScale > 255)
            LUTbrightdark[i] = 255;
      else
            LUTbrightdark[i] = fullScale;
}
```

The contrast lookup table linearly maps the range [*contrast*, (255-*contrast*)] to [0, 255], where *contrast* is an integer in the range [0, 127]. The contrast table is generated in a very similar manner to the brightness table, replacing the line,

```
fullScale = 255*(i)/(256-sliderBrightness);
```

with,

```
fullScale=255*(i-sliderContrast)/(255-2*sliderContrast);
```

Both of these lookup tables are used just before display, in the view's *OnDraw* function. The display image is first passed through the brightness lookup table followed by the contrast table:

```
if (mode == SHEAR) {
   for (int i = 0; i < sizeX * sizeY; i++)
      displayBuffer[i]=LUTcontrast[LUTbrightdark[pImageDisplay[i]]];
}
```

4.5.1.1.2 Phase mode

There are two different phase modes implemented in the *Inspector 2.0* application; interlaced and non-interlaced, both of which implement the 'four-bucket' data reduction algorithm. These two modes are described in further detail in the following sections. In both instances, phase images are captured using the same algorithm as described in the literature review in section 2.1.4.

The composite images are analogous to the captured frames in shear mode – a composite image is captured when processing first starts, (*pImageBase*), then every new image (*pImageNew*) subtracts the first composite image to get the display image (*pImageDisplay*).

4.5.1.2.1 Arctan lookup tables

Calculating the composite image is the most resource-intensive process involved in deriving new phase images. Previous versions UCT's existing shearography application used to perform the arctan calculation for every pixel. A simplified code snippet would look something like this:

```
double phase;
for (int i = 0; i < sizeX * sizeY; i++) {
  phase = atan2(pImage3[i]-pImage1[i], pImage4[i] - pImage2[i]);
  pImageOut[i] = U8((phase+PI)*255/(2*PI));
}
```

Note that the function *atan2(Δy, Δx)* is similar to *atan(Δy/Δx)*, but atan2 takes 2 arguments instead of 1, and calculates the resulting angle over the full 0-360° range taking into account the signs of Δy and Δx and does not include any π ambiguities.

The result of the *atan2* calculation is an angle between $-\pi$ and π. This needs to be rescaled to the [0, 255] range, by adding π and multiplying by $255/2\pi$. This use of *atan* and double precision floating point calculations makes the process slow.

In an effort to improve this the speed at which phase images are calculated, *Inspector 2.0* pre-calculates all the possible results of the *atan2* calculation before processing starts, as follows:

Every composite image pixel is the non-linear combination of the 4 other pixels, each between [0, 255]. Having four inputs to calculate every output would be complicated – a four-dimensional lookup table would be required. Instead, we use the fact that the *atan2* function requires two inputs, Δy, Δx, each in between -255 and 255. These can be calculated every time, and their arctan only once and the result stored. This requires only a two-dimensional lookup table. In addition, the lookup table could perform the rescaling so that the adjusted [0, 255] value is returned instead of [$-\pi$, π].

In other words, for every value of $(I_3(x_i, y_i) - I_1(x_i, y_i))$ and every value of $(I_4(x_i, y_i) - I_2(x_i, y_i))$, we calculate the value of *arctan* $(I_3(x_i, y_i) - I_1(x_i, y_i))/(I_4(x_i, y_i) - I_2(x_i, y_i))$. The following code snippet shows the generation of the lookup table. Note that we must add 255 to the counters to ensure that only positive indices are used:

```
U8 LUTatan[512][512];
double phase;
for(int i=-255; i<255; i++) {
  for(int j=-255; j<255; j++) {
      phase = atan2(j, i);          //phase -> [-pi pi]
      LUTatan[i+255][j+255] = U8((phase+PI)*255/(2*PI)); //->[0 255]
  }
}
```

This lookup table is then used during image capture, and looks very similar to the way the previous *Inspector* worked. We must add 255 to map Δy and Δx from [-255, 255] to [0, 510] so as to ensure that only positive indices are used.

```
int dy, dx;
for (int i = 0; i < sizeX * sizeY; i++) {
  dy = pImage3[i]-pImage1[i]+255;  //dy -> [0, 510]
  dx = pImage4[i]-pImage2[i]+255;   //dx -> [0, 510]
  pImageOut[i] = LUTatan[dy][dx];
}
```

4.5.1.2.2 Image subtraction

Image subtraction in phase mode is different to that in shear mode. In phase mode, we are interested in the 2π discontinuities introduced by phase-stepping. Again, the result of the subtraction of two images is an integer between [-255 and 255]. Instead of taking the absolute value, we want negative pixels to 'wrap around' to the other end of the scale; -1 should be mapped to 255 and -255 should be mapped back to 1. Conveniently, this is the default behaviour when one casts a signed short integer to an unsigned short integer in C++. This is very simple:

```
for (int i = 0; i < sizeX * sizeY; i++) {
  pImageDisplay[i]  = (U8) (pImageNew[i] - pImageBase[i]);
```

Here, `pImageDisplay` is an unsigned 8-bit integer, and the result of the subtraction, `pImageNew[i] - pImageBase[i]` is being cast to an unsigned number.

4.5.1.2.3 Non-interlaced mode

Non-interlaced mode requires that four images are captured for every display image. The following schematic diagram shows how this mode was implemented. Note that the `GrabCompositeImage` label describes a function that is executed from the 'Main' function. Once the reference composite image has been acquired, a while loop is entered where a new composite image is acquired and subtracted from the reference image in each iteration. Note that this is a simplified schematic representation (figure 57) and does not include synchronization or multithreading elements.

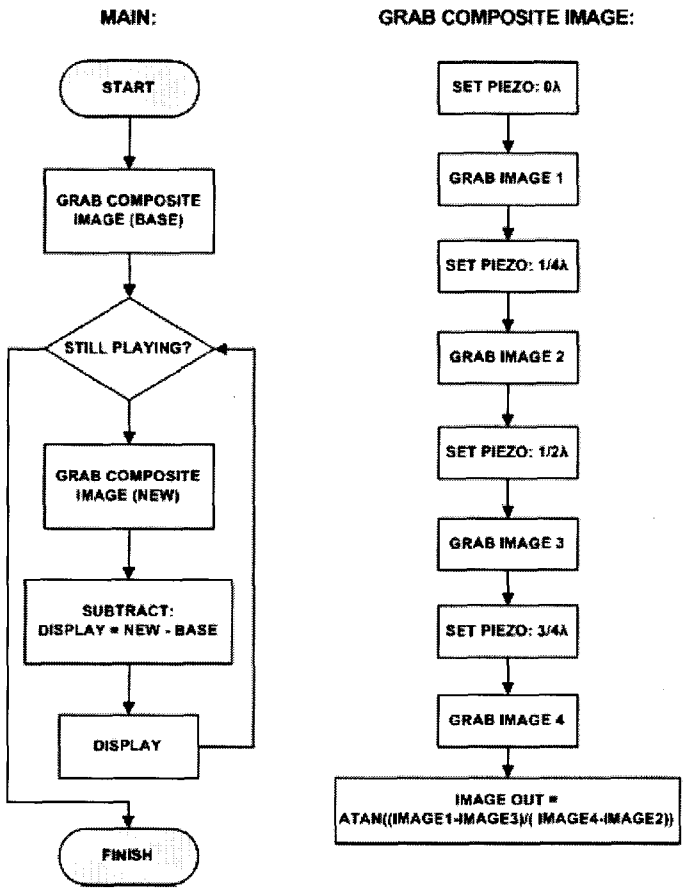


Figure 57: Non-interlaced phase-stepping processing

This is the simplest mode to implement, but also runs slowly. The best possible display rate will be a quarter of the best capture rate. A display rate of 6.2fps was achieved, which is as fast as possible given that the video capture rate was 25fps. One of the reasons that a higher display rate is preferred is because it implies that the piezo is stepping between positions faster. This gives less time for the specimen to have moved in between piezo steps. In the process of capturing a composite image, it is important that the state of the specimen should remain as constant as possible, as this assumption is made in the derivation of the phase-step algorithms.

4.5.1.2.4 Interlaced processing

Interlaced processing gives one display image for every new image that is captured, resulting in higher display rates than non-interlaced processing. Each display image is still comprised of 'base' and 'new' composite images, but pointers to the previous three captured frames are stored so that they can be reused after the next phase-step.

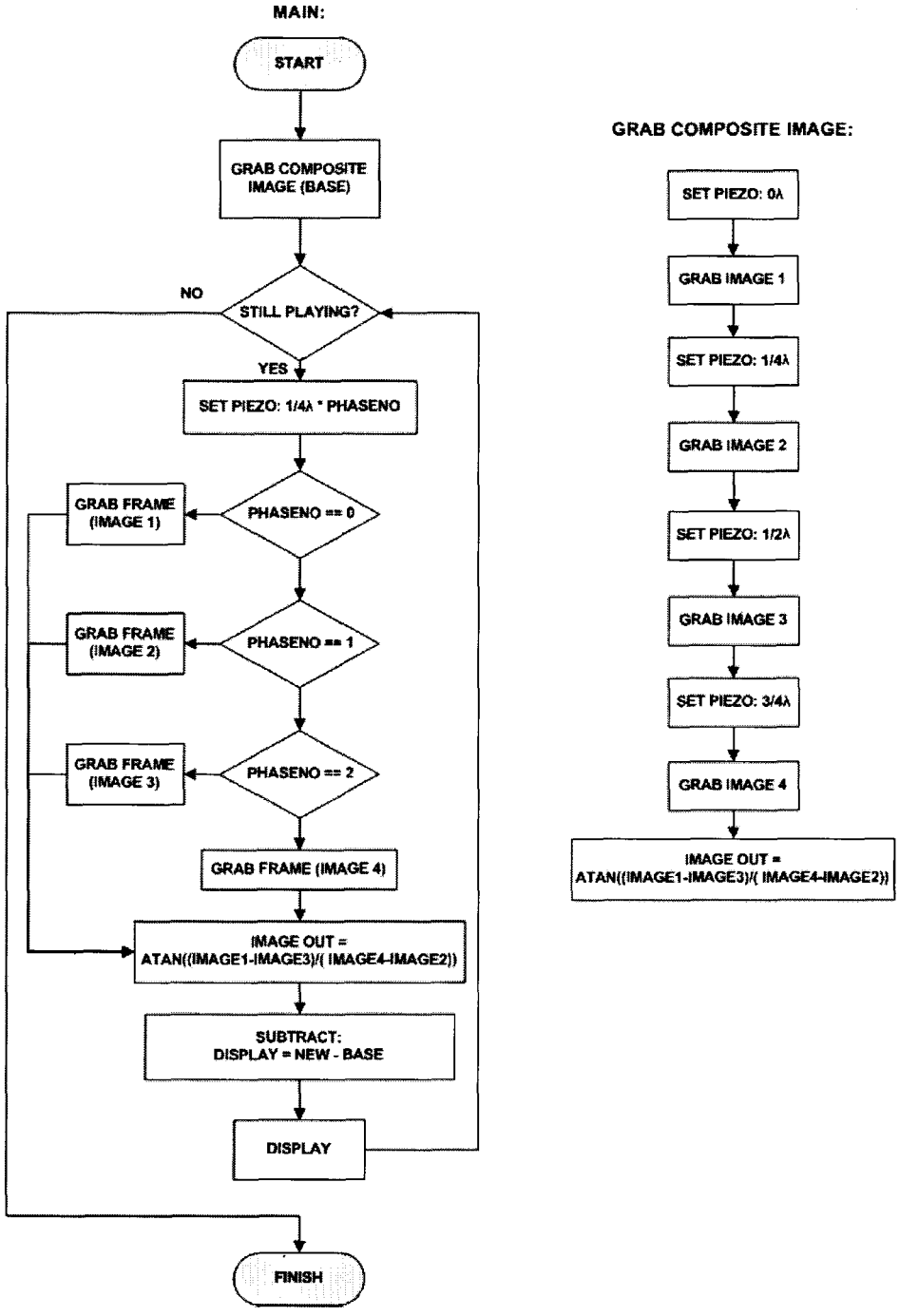


Figure 58: Interlaced phase-stepping process

The schematic figure 58 on the previous page describes the image capture procedure. The `GrabCompositeImage` function is still used, but only to acquire the reference image. Thereafter, process of interlaced processing updates only one frame buffer per iteration, but still performs the arctan lookup each iteration. One can see that the piezo is shifted only once per image capture loop, meaning that it will take four image display loops for it to have stepped through the four positions.

Thus it may take marginally longer for the piezo to step through all voltages in interlaced compared with non-interlaced modes. However, with the fast execution of the arctan lookup, the difference in negligible: 6fps in non-interlaced and 22fps in interlaced mode.

The following code snippet shows how interlaced processing was implemented:

```
int phaseNumber = (frameCounter-1)%4;
pCamera->CamSetPiezo(piezoVoltages[phaseNumber]);

//update relevant buffer
if (phaseNumber == 0)
  pCamera->CamGrabFrame(pImage1, sizeX*sizeY);
else if (phaseNumber == 1)
  pCamera->CamGrabFrame(pImage2, sizeX*sizeY);
else if (phaseNumber == 2)
  pCamera->CamGrabFrame(pImage3, sizeX*sizeY);
else
  pCamera->CamGrabFrame(pImage4, sizeX*sizeY);

//perform atan lookup and image subtraction
for (int i = 0; i < sizeX * sizeY; i++) {
  dy = pImage3[i]-pImage1[i]+255;  //dy -> [0, 510]
  dx = pImage4[i]-pImage2[i]+255;   //dx -> [0, 510]
  pImageNew [i] = LUTatan[dy][dx];
      pImageDisplay[i] = (U8) (pImageNew[i] - pImageBasePhase[i]);
}
```

4.5.1.2.5 Phase image filtering

The process of phase image filtering is an important part of the unwrapping process. Filtering removes sufficient noise from the image, allowing the phases to be unwrapped, and surface plots derived.

One cannot simply, however, low pass filter the phase image; this would blur the phase steps and destroy important information. The preferred process is to split the phase image back into its sine and cosine components (which are continuous), filter these images, and then recombine them using arctan. The smoothing process may be performed multiple times to smooth the image.

The *Inspector 2.0* application uses the *CImg* toolkit (which implements a Canny-Deriche filter) to blur the sine and cosine images. The following code snippet shows the use of the blurring filter in the phase-filtering function.

```
float* pSinData = new float[sizeX*sizeY];
float* pCosData = new float[sizeX*sizeY];

//split phase image into sine and cosine channels
double angle;
for (int i=0; i < sizeX*sizeY; i++) {
      angle = pImagePhase[i]*PI/127.5;
      pSinData[i] = (float) sin(angle);
      pCosData[i] = (float) cos(angle);
}

//need to use CImg lib here, for one thing - smoothing (or blurring)
CImg<double> sinData(pSinData, sizeX, sizeY);
CImg<double> cosData(pCosData, sizeX, sizeY);

//smooth pSinData and pCosData
for (i = 0; i < filterRepeats; i++) {
      sinData.blur(1,1,1);
      cosData.blur(1,1,1);
}
```

```
//copy the buffers out from the CImg objects
for (i=0; i < sizeX*sizeY; i++) {
      pSinData[i] = (float) sinData[i];
      pCosData[i] = (float) cosData[i];
}

//combine sin & cos back into pImageDisplay.
//pImagePhase remains unchanged in this entire routine.
double tanVal;
for (i = 0; i < sizeX * sizeY; i++) {
      tanVal = atan2(pSinData[i], pCosData[i]);
      pImageDisplay[i] = U8((tanVal)*255/(2*PI));
}
```

4.5.1.1.3 Synchronous vs. Asynchronous capture

In general, digital cameras can operate in one of two modes; synchronous or asynchronous capture. In synchronous mode, the camera will return a captured frame that is only taken after the call to grab the frame was made. Thus, a function call to grab a camera frame in synchronous mode will typically take 1/25 of a second for a 25fps camera. This delay is a disadvantage in that it will result in slower display rates; the delay in grabbing a frame will be in series with the processing delay, which (depending on the processing mode) is often of a similar magnitude, effectively halving the display rate.

Asynchronous capture on the other hand, will return a frame as soon as the call to grab a frame is made, giving faster display rates. Asynchronous capture can mean different things in different camera libraries. For instance in the MIL environment, an asynchronous grab call will immediately return an image buffer that is still being written to. This is dangerous behaviour, and usually requires the use of 'double buffering'; the process of alternating grabbing into two different buffers, while returning (processing) the other buffer. The following code snippet shows the use of double buffering:

Initialization:

```
MIL_ID MilImage[2];
for (int i = 0; i < 2; i++)
    MbufAlloc2d(MilSystem, sizeX, sizeY, M_DEF_IMAGE_TYPE,
      M_IMAGE+M_GRAB+M_PROC+M_OFF_BOARD, &MilImage[i]);
n = 0;
```

Frame-by-frame usage (image returned for processing in `pFrame`)

```
MdigGrab(MilDigitizer, MilImage[n]);
MbufGet2d(MilImage[1-n], 0L, 0L,  sizeX, sizeY, pFrame);
n = 1-n;
```

The *PixeLINK* SDK operates in a similar manner except that it performs double or multiple buffering behind the scenes and will not return frames that are being written to. It also does not offer the option of using synchronous mode; although this may be simulated using a delay (the length of the capture process) after capture is initiated. Even with the use of double buffering, there is still some unexpected behaviour; the frame that is returned in `pFrame` was not the most recently acquired frame. In some

applications this may not be important, but in phase-stepped shearography, where the mirror must be synchronised with the frames, it becomes critical. Consider the following code snippet; the mirror *should* be shifted *before* a frame is grabbed.

```
pCamera->CamSetPiezo(piezoVoltages[0]);
pCamera->CamGrabFrame(pImage1, sizeX* sizeY);
```

In synchronous mode, the frame that is returned would have been taken after the mirror was shifted (in the first line above), as expected. However, in asynchronous mode, there is no guarantee that the returned frame would have been taken after the mirror was shifted. It would most likely be a frame that was taken previous to the mirror being shifted, although if there is any delay in setting the piezo voltage, the captured frame may be a frame from after (or during) the mirror shift. Bearing in mind however, that phase-shifting operates in a circular loop with phase-steps in-between, it is not always important for the mirror and frame capture to be synchronized; whether capture takes place before or after the mirror is shifted does not affect the fringes developed. The difference between these two situations is that the fringes will end up being 90° out of phase from each other.

In summary, while synchronous capture will not have any phase ambiguity by guaranteeing that the image returned was captured after the image capture call was made, the application will run substantially slower. Asynchronous mode will run faster, but is open to phase ambiguity, which can result in 'flicker' between different phases.

4.5.1.2 Loose-coupling of camera and application code

In an effort to standardize the code base that is used for different sets of hardware (for instance the new *PixeLINK* camera and the existing system's *Pulnix* and *Matrox* functions), all the camera functions were moved to an abstract base class, from which child classes were then derived. Figure 59 shows this concept:

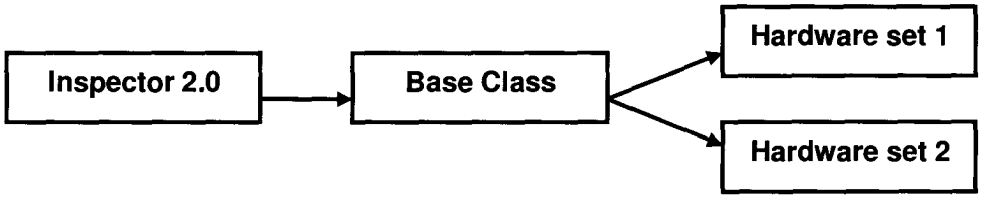


Figure 59: The use of 'loose-coupling'

The *Inspector 2.0* code uses polymorphism to access the camera specific code via an abstract base class. The base class acts as a well-defined interface for the *Inspector 2.0* application to access hardware resources. This 'loose-coupling' of vendor camera code and application code allows the use of completely different cameras without extensive modifications to the application code; all that is required is for a new class to be derived and included for the new vendor's camera functions. There are significant advantages to using this approach: as a demonstration, a new class was derived for the laboratory's camera, framegrabber and piezo controller. The new *Inspector 2.0* application was extended to be able to control the old system's hardware within an hour.

The abstract base class encapsulates all of the functionality that is required by the *Inspector 2.0* application. The header file for the base class which defines the interface for connecting to external devices, *CameraCtrl.h* is given here.

```
class CameraCtrl  {
public:
      //Constructed at startup
      CameraCtrl(){}
      virtual ~CameraCtrl(){};

      //Initialization occurs every time play is pressed
      virtual int CamInitialize() = 0;
      //Camera is uninitialized on when playing stops
      virtual void CamUninitialize() = 0;
      virtual int CamStartStreaming() = 0;
      virtual void CamStopStreaming() = 0;
      virtual int CamGrabFrame(U8* pFrame, int bufferSize) = 0;
      virtual int CamSetExposure(double value) = 0;
      virtual int CamSetGain(double value) = 0;
      virtual int CamSetGamma(double value) = 0;
      virtual int CamGetImageSize(int& x, int& y) = 0;
      virtual int CamSetPiezo(int voltage) = 0;

      int piezoSteps;   //piezo controller resolution (1023,4095,etc)
      USHORT defaultPiezoVoltages[5];     //pre-calibrated voltages
      int piezoDelay;               //predetermined settling time
      char* uniqueFileName;   //used to write config info to file
};
```

In an implementation class, all or most of the functions above would contain camera-specific code for performing a particular function. The names of the functions in the implementation class are fairly self-explanatory, and describe the process to be performed by the function.

As an example of how these functions are implemented, two examples of the `CamGrabFrame(U8* pFrame, int bufferSize)` function are offered here. The first is taken from the implementation class for the new *PixeLINK* camera, and the second is taken from the other class that was implemented to control the existing hardware employed by UCT's NDT laboratory (via the Matrox framegrabber). Both the functions contain proprietary code from the camera/framegrabber vendor, but both grab a frame from the camera and return it in the same buffer; hence the calling application doesn't need to know the specifics of how the function is implemented.

```
int PxLCamera::CamGrabFrame(U8* pFrame, int bufferSize)
{
	if (hCamera == 0)
		return 1;
	PXL_RETURN_CODE rc;
	FRAME_DESC frameDesc;
	rc = PxLGetNextFrame(hCamera, bufferSize, (LPVOID)pFrame, &frameDesc);
	ASSERT(API_SUCCESS(rc));
	return 0;
}
```

```
int MatroxCamera::CamGrabFrame(U8* pFrame, int bufferSize)
{
	if (MilDigitizer == 0)
		return 1;
	MdigGrab(MilDigitizer, MilImage[n]);
	MbufGet2d(MilImage[1-n], 0L, 0L,  sizeX, sizeY, pFrame);
	n = 1-n;
	return 0;
}
```

An effort was made to allow the type of hardware (camera and piezo controller) to be specified at run-time. This would be desirable, as the exact same application could be used to control either set of hardware without any recompilation. However, this does have disadvantages; both sets of hardware libraries must be present at compile time and must be linked into the final executable as either could be used at run-time. One way of overcoming this limitation would be to compile each camera library and its *Inspector 2.0* interface class as dynamically-linked libraries, enabling run-time linking.

This was not implemented due to time constraints, and the fact that it is not strictly necessary; the current design requires that only one line of source code is changed and recompiled to switch between camera libraries.

Note that there is only one base class for two different external components, the camera and piezo controller. One may expect that these two components would be handled separately by *Inspector 2.0* through different base classes. The reason that this is not the case is that the new system accesses the piezo controller board via the camera (over the same cable), requiring the use of the same handle. The use of a single base class which can decide what type of hardware to connect to is more generic than two separate classes, as this solution is able to accommodate this situation. In the case where two separate, unrelated pieces of hardware are used for the camera and piezo controller, the base class can manage them separately, even sub-classing operation further to make their operation modular, as is the case with the solution derived to control UCT's existing system.

4.5.1.3 Other features of the *Inspector 2.0* implementation

Several additional features were added to the developed application:

- The image resolution is soft-coded (instead of explicitly specified) by querying the camera's internal registers before dynamic allocation of image storage arrays. This means that different resolution *PixeLINK* cameras can be used without any software changes or recompilation.

- The ability to 'hot-swap' between different processing modes while maintaining the video stream makes testing simpler.

- The application also allows for in-situ calibration of the piezo actuator. This is convenient as it eliminates the external calibration process, and any in-situ setup differences, such as the mirror-wobbling effect [46].

- The widely recognized *Subversion* [47] version control system is used to manage source code, track bugs, and manage release versions.

4.5.2 Graphical user interface

The GUI was implemented using floating control bars, which may be hidden and shown as necessary.

- Display rate and piezo operation indicator are shown in the status bar (figure 60). These show the user the current display rate in frames per second, and an indicator to show whether the I^2C DAC chip is returning success codes or not.

Figure 60: Frame rate and piezo operation indicators

- A control panel for camera-specific features such as exposure, gain and contrast. These offer further control, for instance altering the camera's on-board exposure time when the aperture setting is already at its maximum or minimum.

- A control panel for editing the piezo's phase-stepping voltages (figure 61). This panel includes control for testing the piezo calibration, via on-screen feedback (in shear mode), allowing in situ calibration of the piezo.

- The control dialog (figure 62) contains all of the main control options, and is opened by default. This dialog is used in the process of initializing, capturing and processing shearography images.

Figure 61: Phase calibration dialog

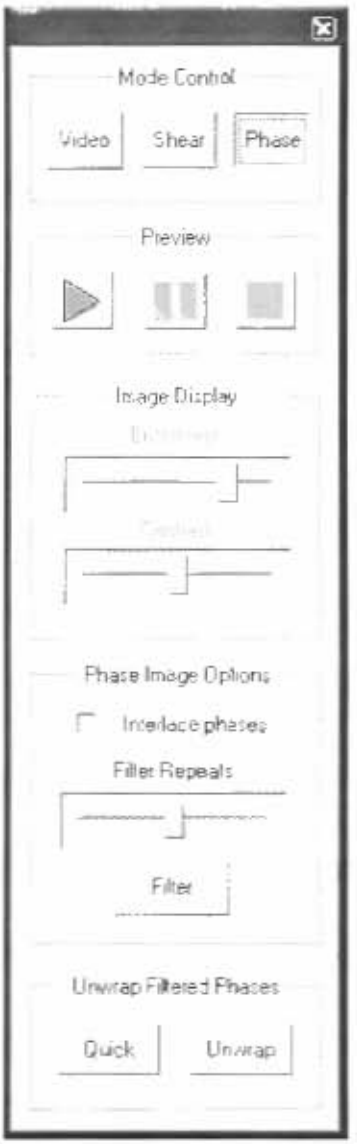


Figure 62: Control dialog

4.5.3 Installer

An installer has been written which automates the tasks installing the application software. The execution of the installer copies files and performs registration:

- Creates a program file in the Program Files folder, and copies necessary files.
- Registers the *Inspector 2.0* Application with the operating system.
- Installs camera drivers.
- Creates desktop shortcuts.

Figure 63: *Inspector 2.0* Installer

The installer was written in a free installer scripting language called *MakoMsi*, by Dennis Bareis [48]. It is a Microsoft Installer Package (*.msi), the type recommended by Microsoft. It makes use of a 'custom action' [49] to install the camera's drivers. The installer is capable of complete uninstall, as well as being convenient for detecting newer versions when updating an installation.

The installer contains all the necessary files internally as well as commands for installing the camera drivers. Upon execution, it copies the files and executes the necessary commands, as well as creating shortcuts. The files copied are listed in table 9, below:

Table 9: Installer files

File	Default destination	Function
Inspector.exe	C:/Program Files/Inspector[1]	*Inspector 2.0* application
Inspector.hlp	C:/Program Files/Inspector[1]	*Inspector 2.0* help files
unwrap.exe	C:/WINDOWS/system32[2]	Phase-unwrapping plug-in
unwrap.ctf	C:/WINDOWS/system32[2]	Phase-unwrapping plug-in
JPEGDLL32.dll	C:/WINDOWS/system32[2]	JPEG compression utility
PxLAPI40.dll	C:/WINDOWS/system32[2]	PixeLINK API 4.0
PxLAPI41.dll	C:/WINDOWS/system32[2]	PixeLINK API 4.1
PxLFW.sys	C:/WINDOWS/system32/drivers[2]	FireWire camera driver
PxLTF.ax	C:/WINDOWS/system32[2]	Permanent file
PxLUSB64.sys	C:/WINDOWS/system32/drivers[2]	USB camera driver
InstallHelper.exe	Temporary file[3]	PixeLINK configuration utility
PxLFW.inf	Temporary file[3]	Driver file
PxLUSB.inf	Temporary file[3]	Driver file
PxLUSB.sys	Temporary file[3]	Driver file
PxLUSBx64.cat	Temporary file[3]	Driver file
PxLFWx86.cat	Temporary file[3]	Driver file

Notes:

1. The destination directory may be modified from C:/Program Files to any user-specified option.
2. All the *.sys, *.dll and *.ax files are marked during the install as permanent, which means that they won't be removed when the application is uninstalled. This is recommended behaviour for files of these types.
3. The temporary files are used for installing the drivers, and are deleted when the installation completes.

The files marked as temporary in table 9 above are used to install the camera drivers with the help of a PixeLINK executable, *InstallHelper.exe*, which is designed to simplify a custom install. This executable is run by the installer with the following command-line arguments:

```
InstallHelper.exe DRIVERFILE=PxLFW.sys; INFFILE=PxLFW.inf;
MODELNAME=FIREWIRE_CAMERA_RELEASE_4; SUPPORTDIR="C:/Program
Files/Inspector"; VENDORNAME=OEM;
```

Note that the camera should not be connected before or during the installer run. It is possible that the drivers will not install properly in this case and the camera will not be recognised. The installer should also be run with administrator privileges.

4.5.4 Manual installation

The installation can still be performed manually in the event that the installer fails. The simplest way to install all the camera files is to install the free *PixeLINK* sample application, *PixeLINK Capture OEM*. This installer will perform all of the necessary operations for installing the camera. Then, one needs to copy across the *Inspector 2.0* files to the target PC. These files are the first four files in the preceding table; *Inspector.exe*, *Inspector.hlp*, *unwrap.exe* and *unwrap.ctf*.

4.5.5 *MATLAB* dependency

The *Inspector 2.0* application makes use of a phase unwrapping plug-in called *Unwrap* [5], which requires that *MATLAB* runtime environment is installed. Once the development and compiler licenses for *MATLAB* have been purchased, redistribution of the MCR (*MATLAB Component Runtime*) is free and unlimited, so this dependency does not impact the cost of distribution, even though the licences are costly.

4.6 Cost

The total cost of the shearography system can be broken down into two components; fixed (table 10) and variable costs (table 11). Fixed costs include the cost of development licenses and software development kits. The variable costs are those of everything that goes into the shearography head, including camera, piezo, and the actual box, as well as the cost of any runtime licenses used. The cost calculated here is not the total cost, as it does not include all the small fasteners and other minor accessories, but the main components of the system. This section aims to make a comparison between the cost of UCT's existing system and the new one.

The main cost reductions in variable costs were achieved were through the elimination of a framegrabber card, proprietary piezo controller, runtime licences and industrial PC.

Table 10: Comparison of variable costs

Component (Existing vs. new system)	Cost reduction
Camera (Camera Link vs. FireWire)	42%
Framegrabber (Solios vs. none)	100%
Piezo controller (PCI card vs. custom)	91%
Piezo actuator (Packaged vs. basic)	80%
Lens (same)	0%
Beamsplitter (same)	0%
Mirrors (same)	0%
Laser (same)	0%
Workshop parts (solid vs. sheet metal)	0%
Runtime licences (MIL vs. no licenses)	100%
PC (Industrial PC vs. equivalent laptop)	60%
Total	**52%[1]**

Notes: Value-weighted reduction reflects true variable cost reduction

Table 11: Comparison between fixed costs

Component (Existing vs. new system)	Cost reduction
Imaging library (MIL vs. PixeLINK SDK)	91%
Visual Studio (same)	0%
MATLAB License	0%
MATLAB Compiler	0%
MATLAB Image Processing Toolbox	0%
Total	**39%**[1]

Notes: Value-weighted reduction reflects true fixed cost reduction.

As previously mentioned, both UCT's existing shearography head the low-cost version have a dependency on MATLAB. As such, the cost reduction in this regard is 0%. Also, both systems use Visual Studio as a development environment, giving a 0% cost reduction. The only cost reduction in fixed costs was due to the elimination of the proprietary 3rd party imaging library (MIL) dependency.

4.7 List of main specifications

Table 12: List of specifications

Specification			Value
Camera resolution			1320 × 1040 pixels
PC Interface			FireWire (1394a)
Frame rate[1]	Video		25fps
	Simple shear		25fps
	Phase-stepped	Non-interlaced	6.2fps
		Interlaced	25fps
Supported image formats			*.bmp
Shear angle			-15° to 15°
Inspection area[2]			225 × 165mm
Maximum working distance[3]			±3m
Laser			660nm, 100mW
Overall dimensions			113 × 74 × 224mm
Power consumption	Camera		3.4W
	Laser		<2W
	DAC & PZT		<0.5W
	Total		<6W
Minimum PC/laptop requirements			32-bit Windows 2000, XP or Vista, 1.5GHz processor, 512MB RAM, 50MB hard drive space, VGA Screen
Cost			R39 000

Notes:

1. Measured using a 2.8GHz Pentium 4 with 512MB of RAM.
2. At working distance of 1.5m
3. Depends on specimen reflectivity

4.8 Summary

A new shearography system, which achieves a significant overall cost reduction compared to UCT's existing shearography system, was designed and built. This cost reduction was made possible by:

- A more cost effective digital camera, using a generic FireWire port instead of a framegrabber-based interface.
- A higher level of integration, making maximum use of the camera's extended features and a custom-designed piezo controller board
- Eliminating dependencies on 3rd party software packages which require licensing.

The end product of this design process can be seen in figures 64 to 66, namely the shearography head itself and a screenshot of the *Inspector 2.0* Digital Shearography application, which was written in Microsoft Visual C++.

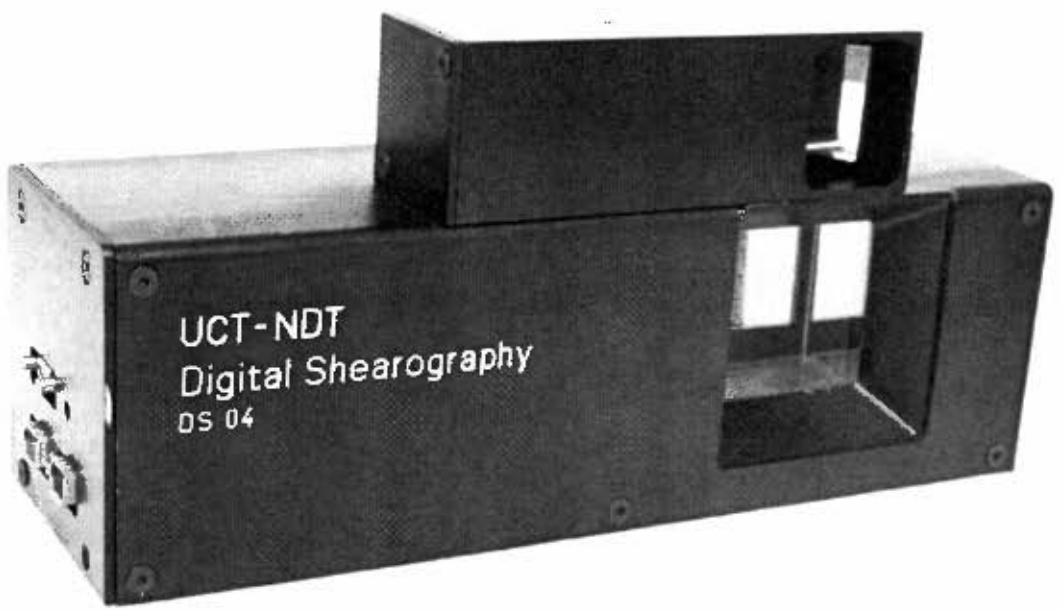


Figure 64: Front view of shearography head

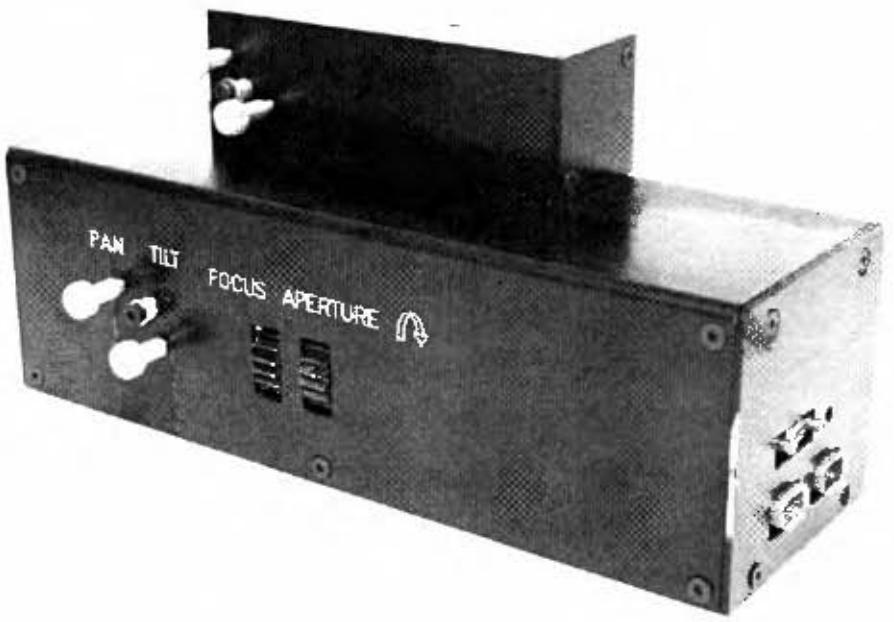


Figure 65: Rear view of shearography head

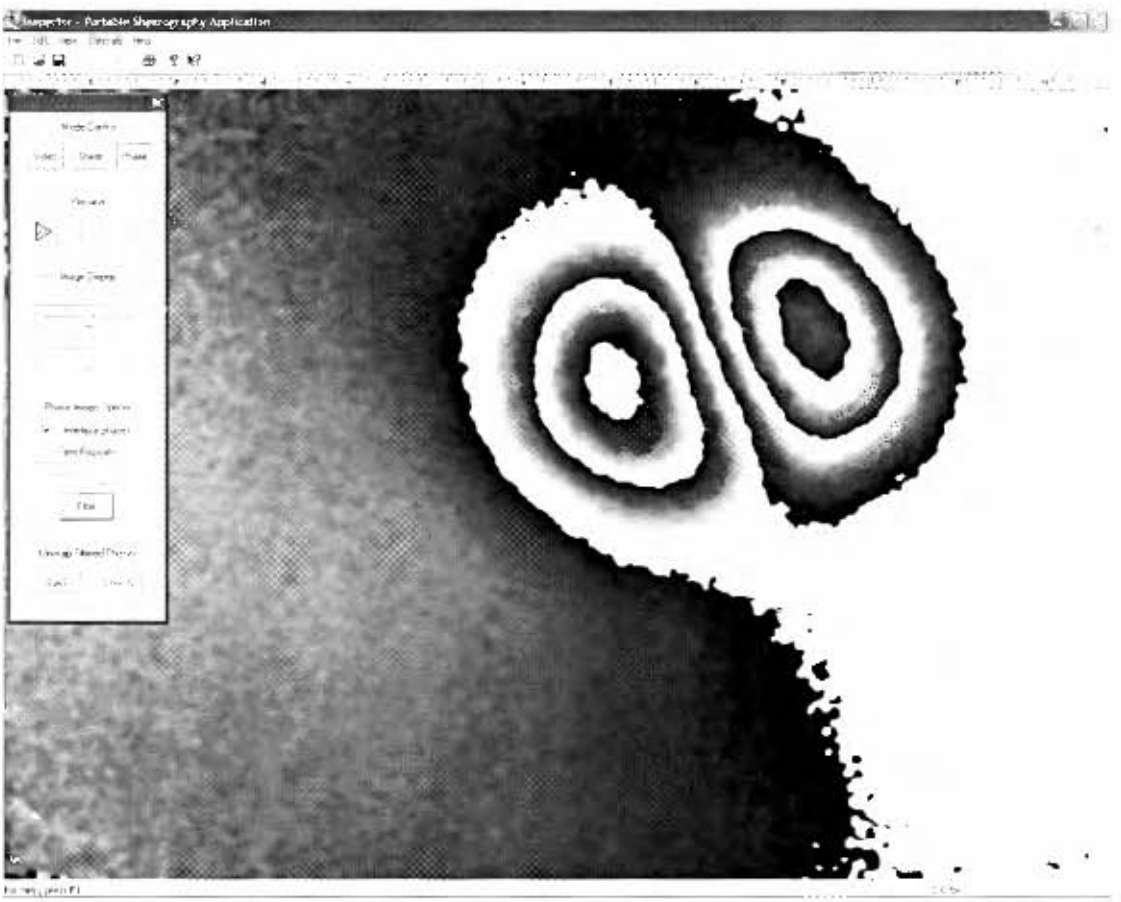


Figure 66: Screenshot of the *Inspector 2.0* application

5. TESTING AND RESULTS

Testing was performed on different aerospace components in order to obtain proof-of-concept for the new system. Extensive quantitative studies were not performed, as these lie outside the scope of this dissertation. The testing procedure here was carried out on a desktop PC, however a similar test was also carried out using a laptop.

5.1 Procedure

The testing procedure that was followed is based on the procedure described in a previous quantitative study performed by Vincent Musonda [50]. Testing was performed on a section of Oryx helicopter rotor blade, which comprises a honeycomb core surrounded by Kevlar-reinforced plastics with an aluminium-reinforced leading edge. This type of component is an example of a high value aerospace component that may be tested in industry using shearography.

Artificial 'flaws' were machined in the specimen to different depths in nine different places, as shown in figure 67. The flaws are equally spaced 140mm apart from each other. For the sake of brevity, only 6 of the 9 flaws were tested. Table 13 describes the details of each flaw.

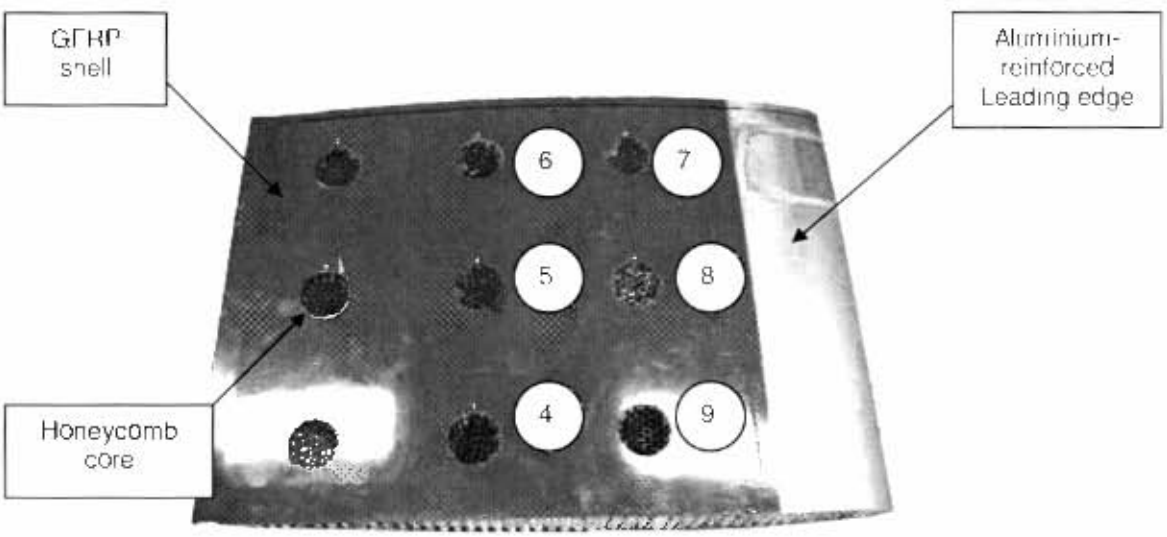


Figure 67: Section of Oryx helicopter blade used as testing specimen

Table 13: Description of artificial flaws in specimen

Flaw number	Diameter of flaw (mm)	Depth of flaw (mm)
4	44	13
5	44	16
6	44	30
7	44	43
8	44	57
9	44	64

The tests were performed using the new *Inspector 2.0* application, loaded on a 2.8GHz Pentium 4 with 512MB of RAM. The tests were performed on the NDT laboratory's vibration isolation table for convenience. The specimen was held in place using two magnetic clamps for the duration of the tests (figure 68), and an infra-red lamp was used to stress the specimen, using a timer to ensure that the heating period remained constant for every test. The specimen was heated in each case for a period of 1 second, with the lamp positioned behind the specimen at a distance of 250mm.

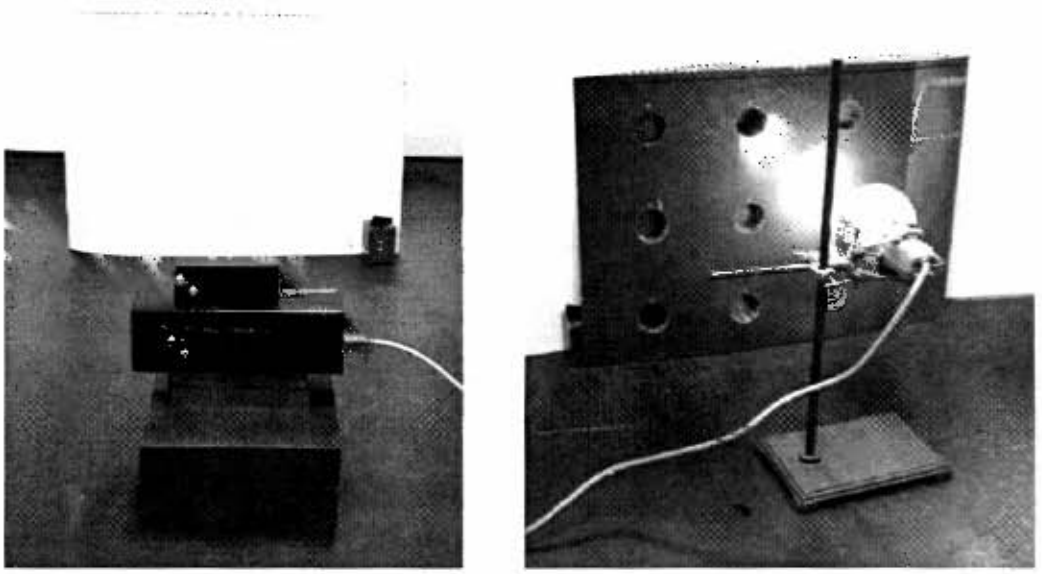


Figure 68: Specimen and shearography head (left) with infra-red lamp (right)

The shearography head was positioned 400mm away from the front surface of the test specimen. A small image shear (equivalent to approximately 50mm on the specimen) was introduced using the shear mirror. Owing to the different sizes of the flaws, different amounts of cooling were required in some circumstances to obtain interference fringes.

In each case, the image processing was initiated 1s after the heating was stopped. For each of the flaws, a sheared image, a phase-stepped image and an unwrapped surface were derived and are presented here. The piezo actuator that was used for this process was the more cost-effective, unpackaged piezo with custom housing.

5.2 Results

The results were good; in each case the new system was able to detect the flaw in both shear and phase-stepped mode. The fringes acquired in phase-stepped mode were sufficiently clear for them to be filtered and unwrapped using the *MATLAB*-compiled unwrapping plug-in.

Figures 69 to 74 illustrate results of the testing. In each case, the simple intensity-based shearography image is shown in the top-left, followed by the corresponding phase-stepped image on its right, followed downwards and right-to-left by the filtered phase-stepped image, unwrapped image, gradient plot and finally surface plot.

5.2.1 Flaw no.4 (cooled for 20s)

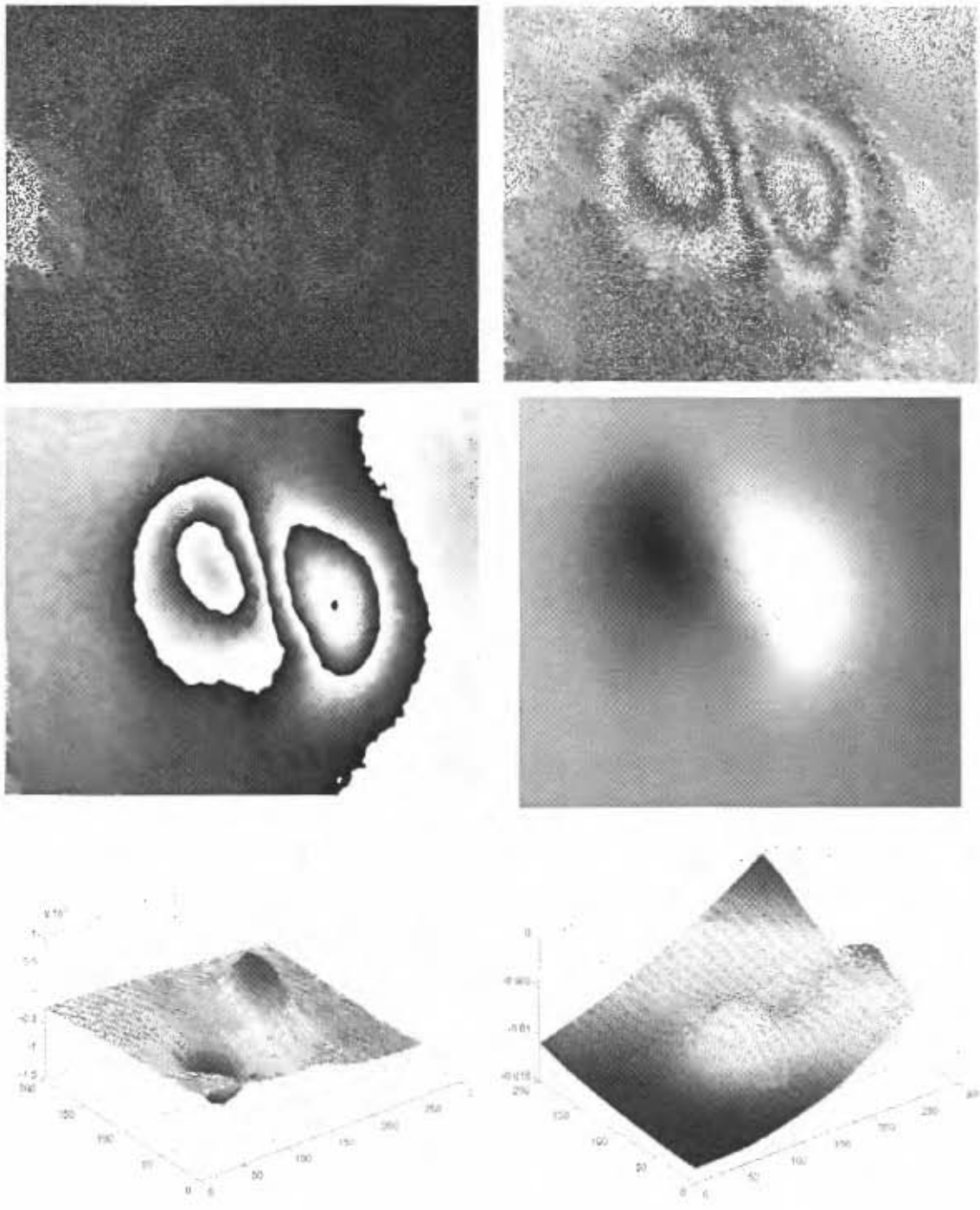


Figure 69: Testing of flaw 4

5.2.2 Flaw no.5 (cooled for 20s)

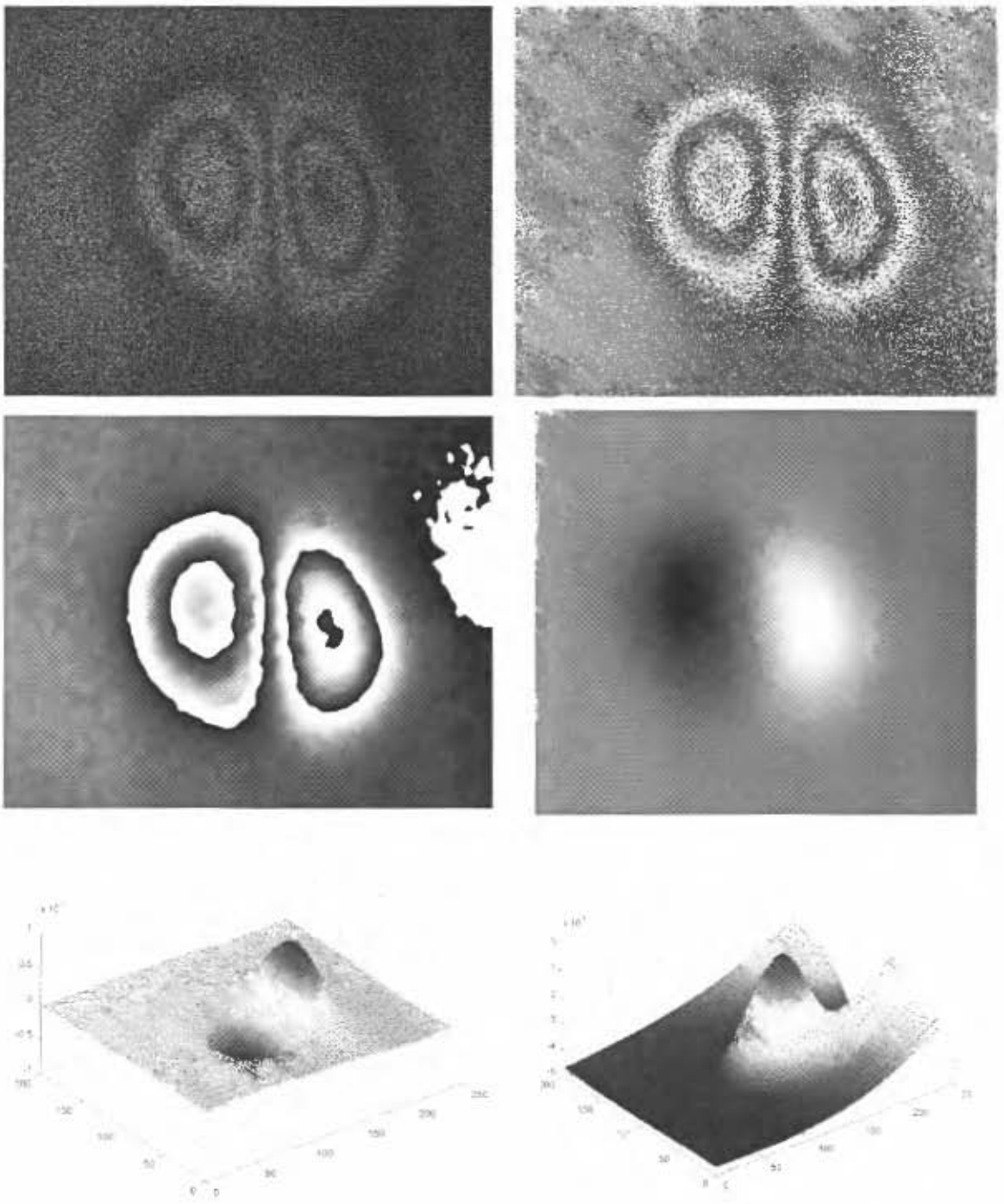


Figure 70: Testing of flaw 5

5.2.3 Flaw no.6 (cooled for 20s)

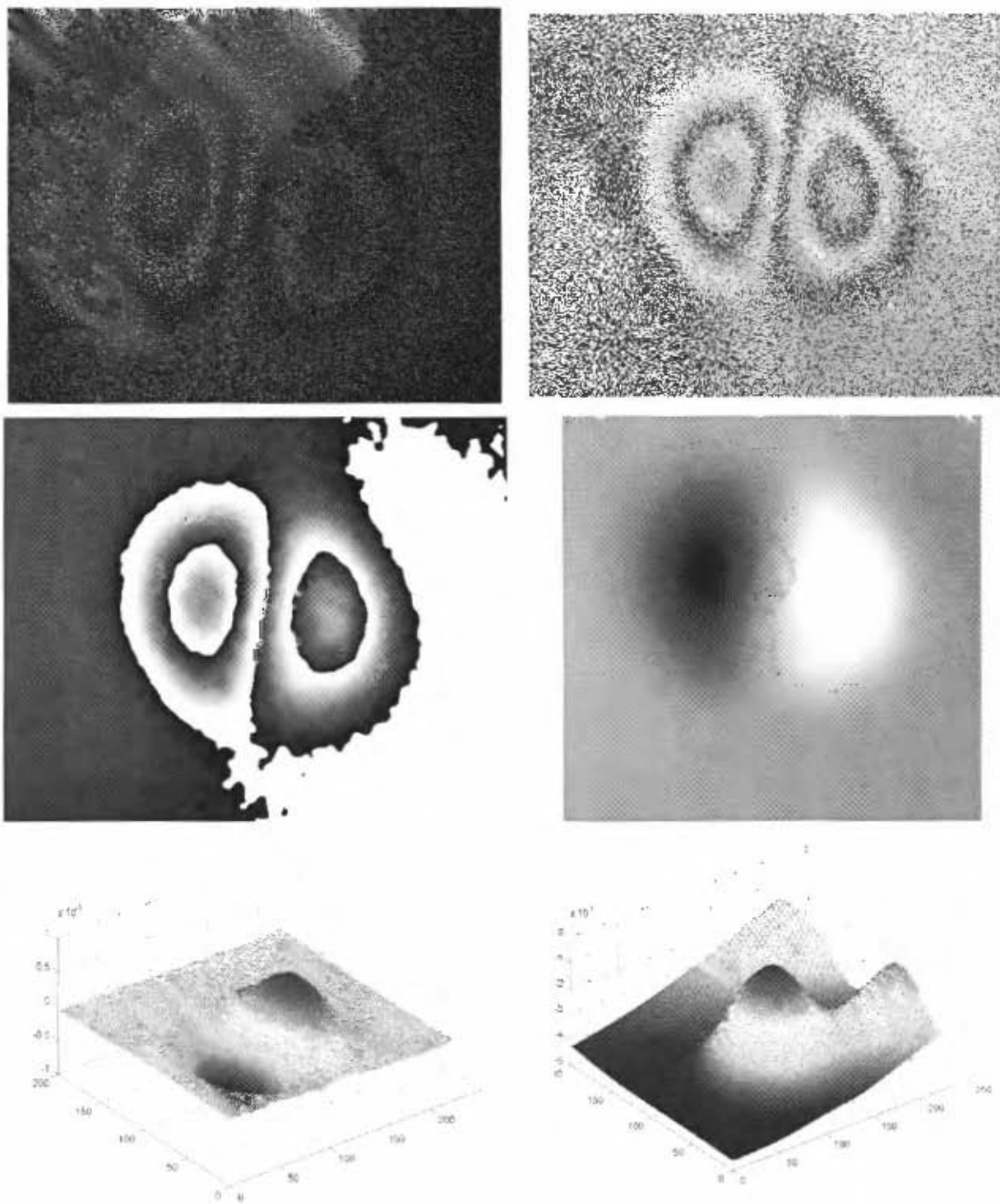


Figure 71: Testing of flaw 6

5.2.4 Flaw no.7 (cooled for 30s)

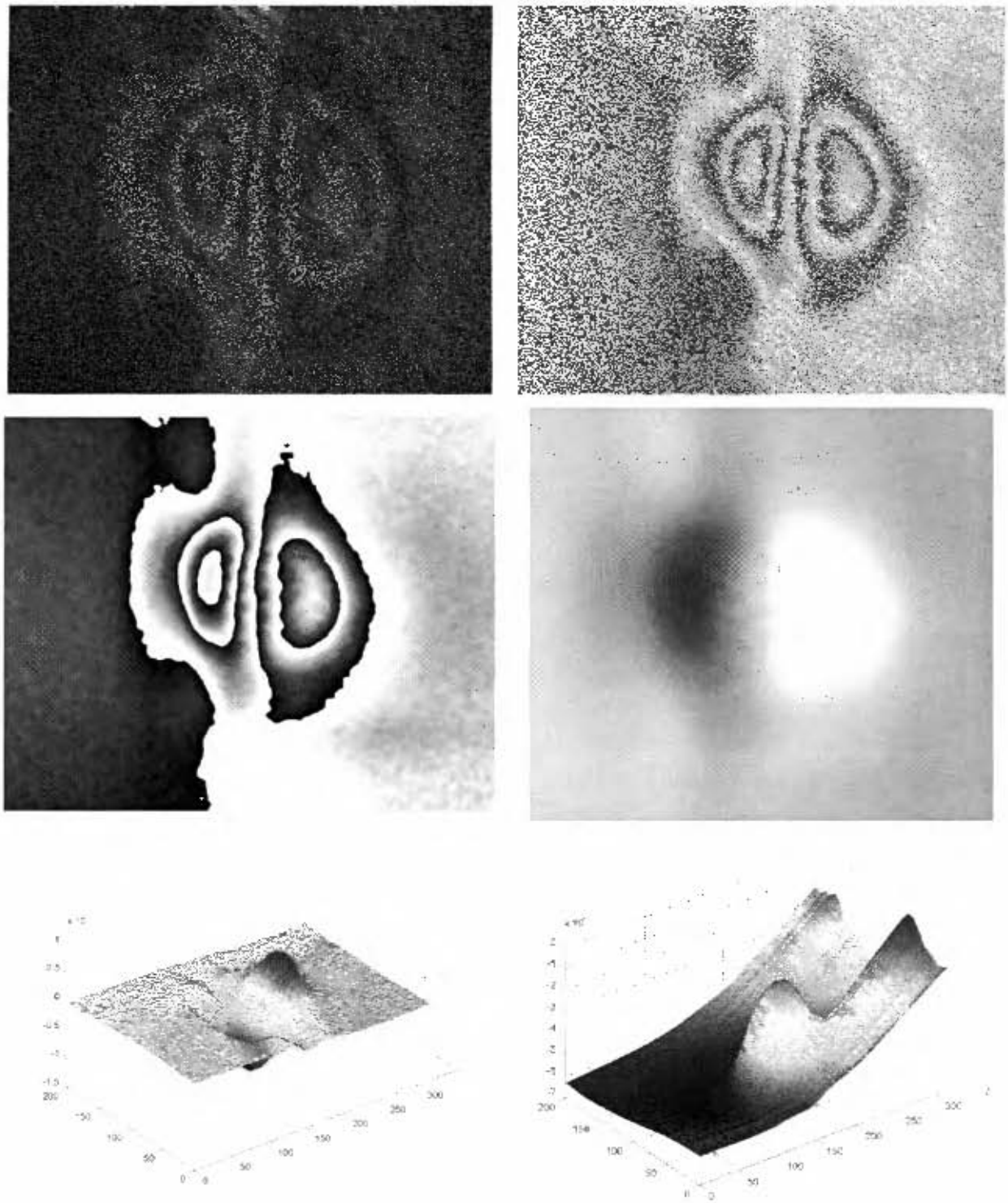


Figure 72: Testing of flaw 7

5.2.5 Flaw no.8 (cooled for 60s)

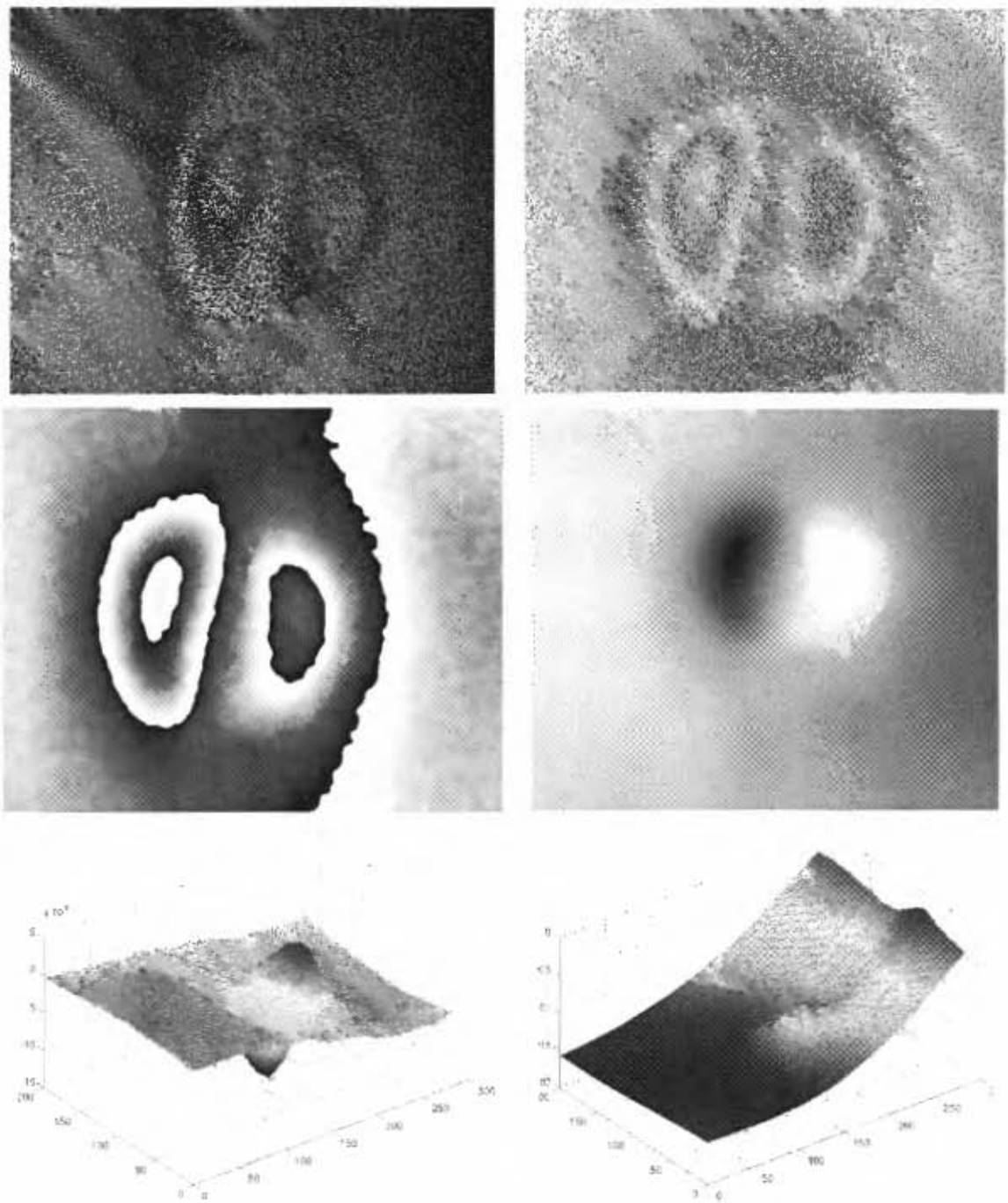


Figure 73: Testing of flaw 8

5.2.6 Flaw no.9 (cooled for 60s)

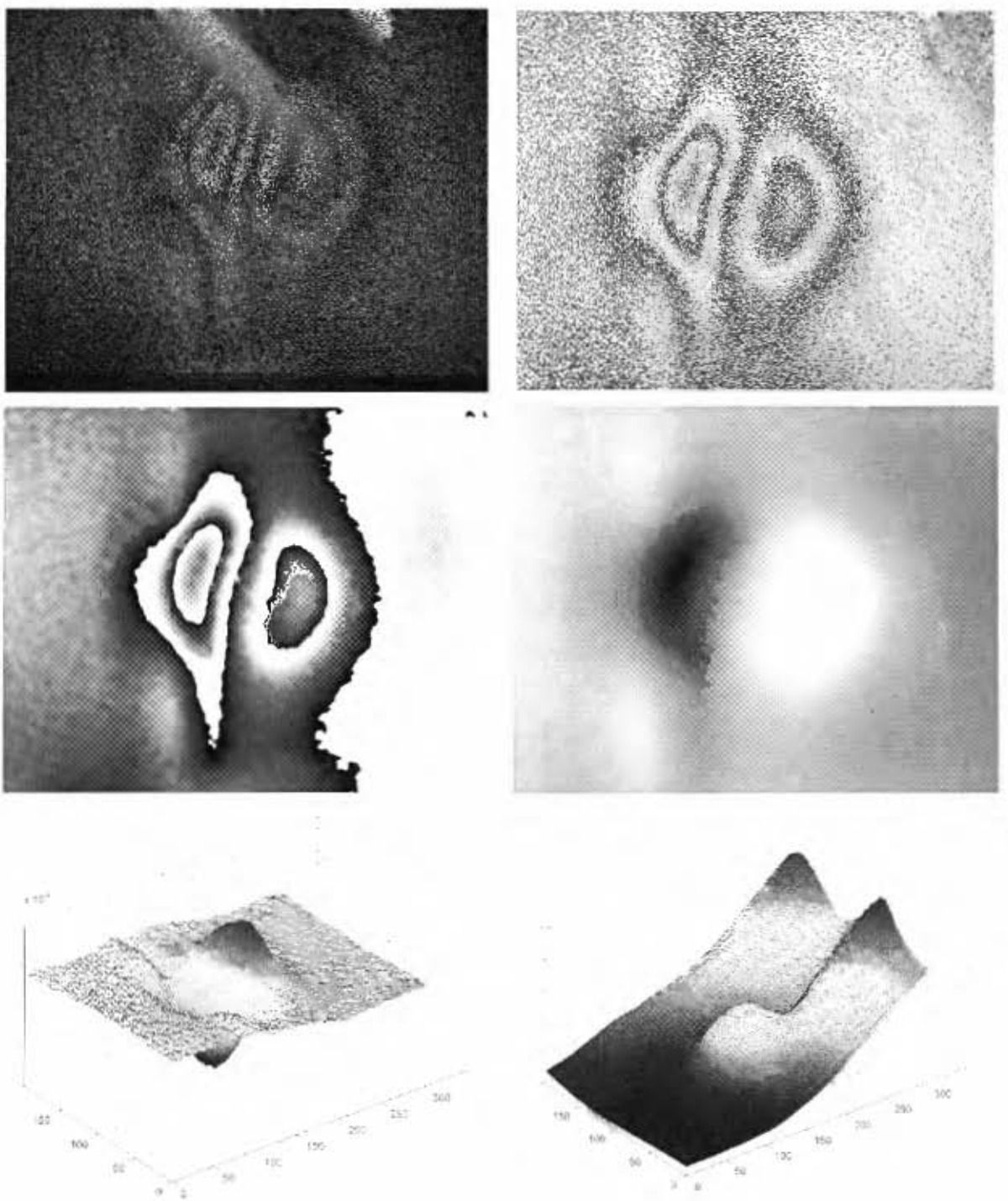


Figure 74: Testing of flaw 9

6. DISCUSSION

6.1 Benchmarking: image clarity and quality

The new system was capable of acquiring phase-stepped images which were of sufficient quality to be phase-unwrapped. It was also deemed important however, to compare the quality of the images acquired to those acquired using UCT's existing system on the same specimen and under the same conditions, to determine whether the new system had adversely affected the quality of the results obtained.

It was found that, in general, the clarity of fringes developed by UCT's existing system were slightly better than the new system. The difference was visible in both conventional intensity-based shearography as well as phase-stepped shearography as a gradual increase in noise level and deterioration of fringe quality over time. Figure 75 shows the different in fringe quality; these images were captured simultaneously on the same specimen. The fringes developed by the existing system are clearer than those developed by the new low-cost system.

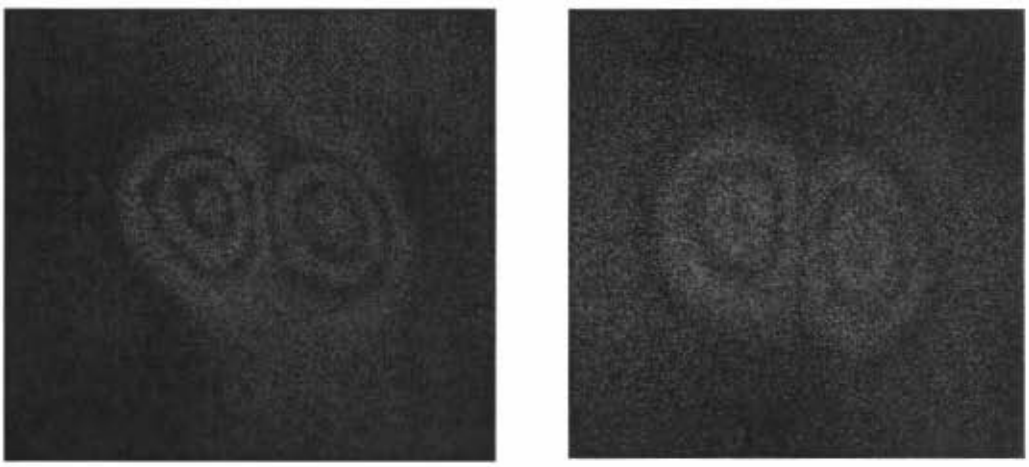


Figure 75: Difference in fringe quality between low-cost system (right) and existing system (left)

There could be many reasons for this, from the way that *Inspector 2.0* processes images, to the camera's CMOS instead of CCD sensor, to controller noise, etc. In order to determine the exact cause, a process of elimination was followed:

A new camera implementation class for the existing system's piezo controller and camera was written, so that the new *Inspector 2.0* application could control the equipment used by UCT existing system. It was found that better fringes were still developed in the new application even when using the existing system's hardware, eliminating the possibility that the decreased fringe clarity was software-related; it could be deduced that the problem must be hardware-related.

It was suspected at this stage that the decrease in fringe quality was due to the piezo controller noise moving the mirror over time. In order to test that this was the case, the piezo and controller were disconnected and turned off, and tests were performed using simple conventional shearography. Surprisingly, it was found that the problem of pixel intensity drift was still present. This eliminated the possibility that the drift was being caused by the piezo or controller.

It is unlikely that the quality difference is due to the optics used, as these are essentially the same. It is expected that the pixel intensity drift is caused by the type of image sensor used: CMOS sensors are generally noisier than CCDs. The existing shearography head uses a CCD camera whereas the low-cost version uses CMOS.

6.2 Lack of complete piezo range

It was discovered during testing that the piezo could not shift the mirror by a full step, due to the limited range of the digital-to-analogue converter. This was not expected, as the DAC controller board had been designed to allow full shifting.

This lack of piezo range was discovered when calibrating the piezo using the 'shear' mode calibration dialog: by stressing a specimen and allowing full fringes to develop in 'shear' mode, and then manually setting the piezo voltage, the effect of the mirror's movement on the developed fringes can be observed. Recall that a mirror shift of ½λ will result in a phase-shift of 1λ for the laser light. A full mirror step of ½λ wavelength should give the same fringe pattern as the 0λ setting, and this is used to calibrate the piezo.

The mirror should be able to shift 0λ, $^{1}/_{8}\lambda$, ¼ λ and $^{3}/_{8}\lambda$ increments of the laser's wavelength. These fractions correspond to distances 0ηm, 82.5ηm, 165ηm and 247.5ηm respectively (using a 660ηm laser). This was not achievable as the piezo could only shift approximately 0.4λ (264ηm), even though with its 5V range and a

displacement-to-voltage ratio of 120nm/V, the controller board should be able to shift the piezo by 600nm, which is significantly more than necessary.

The reason for this lack of range was unclear; according to the design calculations made, the piezo should be able to shift the mirror fully. The piezo controller's voltage was measured on an oscilloscope to confirm that it was behaving as expected. It is suspected that the lack of range has to do with non-linearity of the piezo at low voltages: the full piezo range is 18µm over 150V, which means that only a small fraction of the piezo's displacement range is being used.

However, the inability of the piezo controller to shift the mirror half a wavelength is not necessarily a problem in normal phase-shifting. Bearing in mind that the full ½λ wavelength phase shift is not required in phase-stepped mode (only 0λ, $^1/_8$λ, ¼ λ and $^3/_8$λ are required), the lack of range is not a problem in practice. An alternative method to calibrate the piezo, using only phase-steps that are actually used in phase-stepping, is to ensure that phase-steps 0λ and ¼λ cause the fringes to invert their intensity. This is a more approximate method, but it was found to offer reasonable results; the phase-stepped diagrams do not appear to suffer from a lack a full greyscale range (usually an indication that the piezo is not stepping the full range). Figure 76 illustrates this calibration method; the figure on the right is a shear image with the piezo set at position 0λ, whereas the image on the left is taken with a phase step of ¼λ. The inversion of the fringes shows correct calibration.

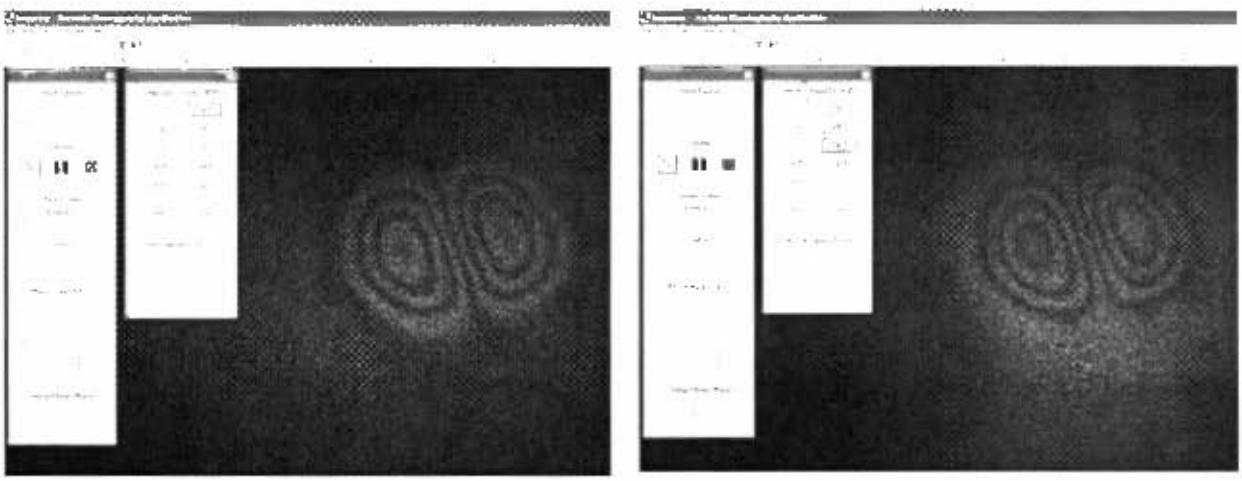


Figure 76: Comparison of phase steps 1 and 3 (should have inverted fringes)

6.3 Phase-stepped image flicker

It was found that under certain circumstances, the phase-stepped images would exhibit 'flicker' when running the application in debug mode, where consecutive phase images would be out of phase with each other, causing the display to flash between lighter and darker interferograms. The fact that the display flickers between interferograms that are out of phase with each other indicates that there is a loss of synchronization between the piezo and the image capture or that the piezo is failing to shift in certain circumstances.

It is difficult to illustrate the phase flicker, as it is temporal in nature, but figure 77 illustrates two frames that may occur consecutively in the video stream during phase-stepped shearography in which phase-flickering is occurring. Notice that the flaw that the fringes describe is essentially the same in each case, but the images are out of phase with each other.

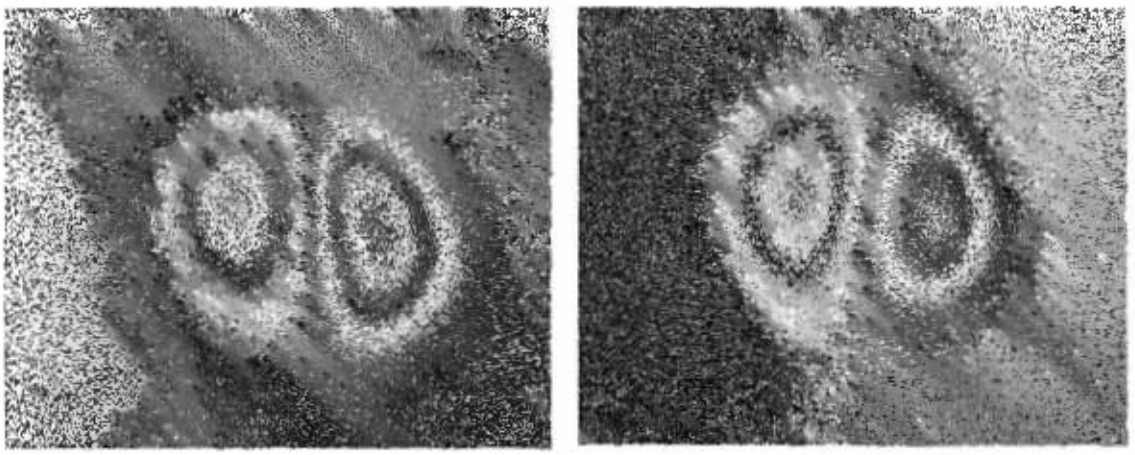


Figure 77: Illustration of phase flicker

Initially it was suspected that this problem was due to the custom-designed piezo housing 'sticking' on certain phase shifts, but this possibility was effectively eliminated when it was noticed that both of the different piezo's exhibited the same characteristics. Another possibility was that the controller was failing to set the piezo voltage successfully due to a serial communication error. However, by analyzing the controller board's return codes, it was established that it was always returning success codes, indicating that there was no communication error. Also, by analysing the controller's voltage output on an oscilloscope, it was found that the piezo's voltage was always being set correctly.

Knowing that the piezo was in fact shifting correctly, it was realized that the problem must be due to a loss of synchronization between capture and shifting. As both of these processes are controlled from the same thread in the *Inspector 2.0* application, it was unlikely that the synchronization problem was due to a program flow or logic error.

After much debugging it was realized that the synchronization problem probably had to do with the way that the camera was returning frame buffers; the camera was operating in asynchronous mode (and in fact the *PixeLINK* API did not offer a synchronous mode). The function call to grab a frame was returning immediately with a frame that was not being written to (this was tested by comparing the state of the captured frame over a period of 100ms). The fact that the function returned immediately with a constant frame dictated that the returned frame had been captured prior to the function call being made. Communication with the camera manufacturers, *PixeLINK*, confirmed that this was probably the case.

This has important implications for shearography; it is important that the piezo and frame capture are synchronized. Two solutions to this problem were proposed:

- Simulate synchronous operation of the camera by inserting 'sleep' statements into the *Inspector 2.0* code after the piezo has been shifted and before the capture function is called. A 45ms pause should ensure that the returned frame was captured after the piezo was shifted. This would make the application run more slowly, but also eliminate phase flicker.
- Leave the application as is; phase-stepped shearography operates in a circular loop with phase shifts interspersed with image capture calls. If the phase shift occurs before or after the frame is captured does not matter as long as:
    - The phase shift does not occur *while* the frame capture occurs.
    - The phase shift always occurs either before or after image capture, but not alternating between the two, as this is when phase-flicker will occur.

It was decided to use the second solution for two reasons:

- The *Inspector 2.0* application only exhibits phase-flicker in certain circumstances, viz. when operating in debug mode (this causes the

application to run more slowly and unevenly), and when opening and closing other windows or performing other operations while also running *Inspector 2.0* in phase mode. By performing testing using only a release version, the problem can be minimized.

- This solution maintains the fast frame rates (6.2fps in non-interlaced mode and 25fps in interlaced mode) for phase-stepped mode which allows detection of flaws in more heat conductive materials.

On slower PC's, image flicker (as opposed to phase flicker) may be a result of slow screen drawing speeds. The *Inspector 2.0* application uses Device Independent Bitmaps (DIB's) and the *StretchDIBits* function, which is a fast mechanism for drawing to screen, although not as fast as some others; more advanced direct screen access technologies such as OpenGL and DirectX would be able to draw to screen faster, but are also more complicated to implement.

7. CONCLUDING REMARKS

Section 5 illustrated that the new shearography system is capable of detecting flaws in composite materials using both simple intensity-based as well as phase-stepped shearography. This was achieved while also achieving a cost reduction in excess of 50%.

It is hoped that the new design, through its low-cost approach and user-friendly operation, will make shearography more accessible to a wider range of industries that may take advantage of the NDT capabilities it offers. The capability of the new shearography unit to operate from a laptop is an important advantage, as it increases the portability and thereby utility of shearography.

The main cost-reducing drive was to eliminate the need for PCI cards and external controllers that are used in existing shearography systems. This was achieved by making maximum use of the camera's extended features, performing the function of the two PCI cards in the shearography head itself. The use of this integrated design has two major benefits:

1. The system can now operate from a laptop (using an auxiliary power supply or battery). This makes the system truly portable, and reduces the cost further by eliminating the need for a ruggedized PC.

2. The shearography head can now operate from only one cable, which is pluggable. This makes setting up the head for testing simple, and installation on a target PC is "plug-and-play".

Many of the new design ideas used in the mechanical construction of the shearography head's housing were found to be inferior to the original design:

1. The use of sheet-metal did not offer significant a cost reduction, and made assembly more difficult.

2. The use of spur gears instead of worm gears makes the adjustment of the focus and aperture clumsy and liable to get stuck.

3. The use of grooves for component location in the old design was overlooked, and made reassembly of the new system open to misalignment problems.

However, the use of an unpackaged piezo in a custom designed housing was found to offer significant cost reduction while not adversely affecting the quality of phase images. The design of the peripheral controller was successful, but there are many areas in which its performance may be improved, especially by increasing the range of the DAC output voltage to allow full phase-stepping.

The use of a FireWire camera was found to offer reasonable results, though the effect of using a CMOS sensor instead of CCD should be investigated further.

A custom shearography application, *Inspector 2.0*, was written using *Visual C++*. The image-processing routines were implemented using basic types wherever possible and optimised for speed. An installer package was also written. These efforts resulted in the following improvements:

1. The new application does not use proprietary imaging libraries, except for the camera's image acquisition library. Instead it uses basic types, and makes limited use of the open-source *CImg* image processing toolkit.

2. The phase-stepping display rate was increased from approximately 4fps in UCT existing system to 25fps, giving the user a better indication of the material's transient response.

3. Installation is simple using the preconfigured installer.

The main objective, to decrease the cost of a shearography system, was achieved, but there is still room for further improvement in this regard. Decreasing the variable costs of the entire system further might entail using moulded parts instead of machined ones and plate beamsplitters. Rewriting the phase-unwrapping program in C++ instead of *MATLAB* would also offer a large reduction in fixed costs.

While the new system offers improved performance in many areas, there are aspects which did not perform as expected and should be reconsidered or redesigned. The next section discusses these aspects and makes some recommendations for future work.

8. RECOMMENDATIONS

8.1 Mechanical design

8.1.1 Head construction

It was found that the advantages of using sheet-metal do not significantly outweigh the disadvantages of machining the box out of solid aluminium. The solid box is stronger, easier to assemble (it does not require end plates or plate-locks) and uses fewer components

During the course of the project, the possibility of injection-moulding these parts from polyurethane arose. This is an ideal solution, as it would eliminate the cost of a large piece of aluminium but still provide the strength of a solid box.

8.1.2 Lens adjustment

Using spur gears to adjust the aperture and focus was found to be cumbersome. The protrusion of the gears outside the box made fitting the u-cover over the assembly difficult. The reason that spur gears were used was to decrease cost, but the existing system's worm gears were subsequently moulded successfully by the NDT laboratory, thereby eliminating the problem of their cost and sourcing. The use of worm gears is also preferable because it allows the box to be completely dust-proofed. Dust-proofing is important in industrial settings, but is difficult to achieve with spur gears, as clearance must be left for the gears teeth to rotate

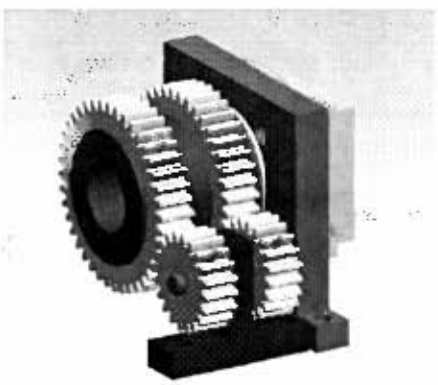


Figure 78: Spur gear adjustment of focus and aperture

8.1.3 Alignment of shear mirror

The housing of the shear mirror was not centred relative to the beamsplitter in the new shearography head, in order that the shear adjustment pivot would be centred. This proved unnecessary, and mounting the shear mirror would be simpler if the mirror plate was centred.

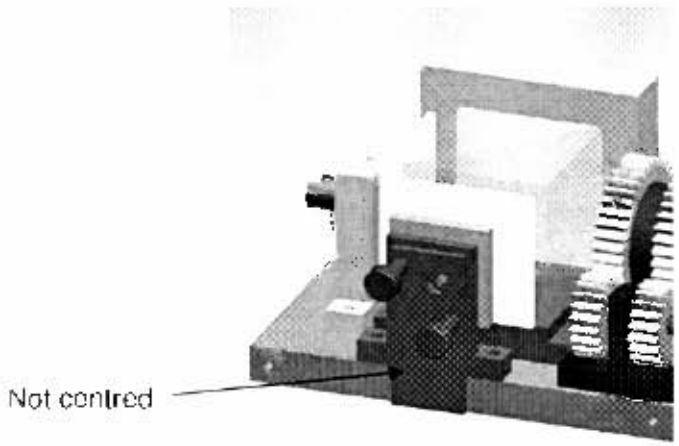


Figure 79: Alignment of shear mirror plate

8.1.4 Piezo mounting design

The housing of the unpackaged piezo was found to work well, illustrating proof-of-concept. However, the housing is clumsy in that it requires the mirror to be removed every time the housing is disassembled. Also, the housing is bigger than necessary; a more compact housing should be designed.

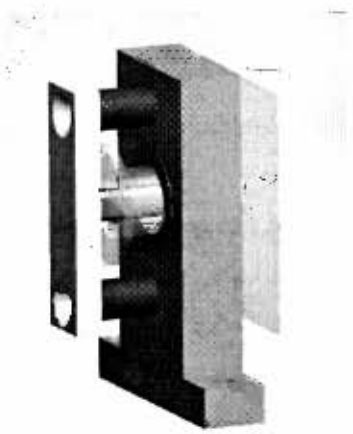


Figure 80: Design of piezo housing

8.1.5 Alignment grooves

The existing system uses alignment grooves in the base plate of the shearography head to align components. Alignment is important, as misalignment can result in the camera picture not aligning perfectly with the laser illumination. The use of alignment grooves was overlooked in the low-cost design, but should be re-implemented in future designs.

8.1.6 Using a beamsplitter plate instead of cube

Beamsplitters come in two main types: plate and cube beamsplitters. Cubes have traditionally been used in the past, even though plate beamsplitters have also been shown to work. However, plates are significantly more cost-effective than using cubes, $31 compared to $229 for similar sizes. Using a plate beamsplitter instead would offer further cost reduction, but this would require some redesign of the head layout, and consideration should also be made of the slight decrease in image quality using plate beamsplitters due to internal 'ghost' images.

8.2 Peripheral controller design

8.2.1 Adjusting reference voltage

Future design iterations should consider revising the reference voltage design for the DAC. The reference voltage chip, the REF02, is suitable, but should be able to be trimmed using a potentiometer.

8.2.2 Increase piezo range

There are several ways in which the range, or amount of travel, available when phase shifting may be increased. It may be achieved either by:

- Increasing the voltage range of the piezo controller board. The voltage range is limited by its supply voltage which is derived from the FireWire bus. Typically this voltage is 12V, but significant filtering and line regulation is required to smooth the high frequency digital noise on the power bus, resulting in a 5V range. Significantly increasing the voltage range would require complete redesign of the controller board using either:

  - An external power supply at a higher voltage, and not using the FireWire bus power.
  - A DC-DC converter to boost the supply voltage to the desired level. This may also require the use of an external power supply and a large amount of signal filtering as DC-DC converters usually superimpose significant switching noise on their output voltage.
- Using a piezo with a greater displacement-voltage ratio. These actuators are less cost-effective and their displacement is more sensitive to supply voltage noise, but would be a simple solution to the lack of displacement range.

The other way to obtain full phase-stepping using the existing controller is to use a piezo with a greater displacement/voltage ratio. These are slightly less cost-effective, longer and with greater noise sensitivity, but would probably solve the problem of incomplete phase-steps.

8.2.3 Improve controller noise

The displacement noise on the signal of the piezo controller is not quite as good as the specification of the system that was used as a benchmark (0.37ηm compared to 0.2ηm). While this is not necessarily a problem, ideally one would want to eliminate all of this noise, as it may affect the piezo displacement, and result in a deterioration of the fringes. However, the fact that the power supply from the camera is extremely noisy (approximately 400mV peak-to-peak) makes this very difficult. Further electronic noise suppression and filtering may result in better noise characteristics.

8.3 Software considerations

8.3.1 Further investigate phase flicker

Under certain circumstances, the phase-stepped images are found to 'flicker' between different phases. This effect should be further investigated by testing the application on different PCs.

8.3.2 Optimize phase image filtering

The *Inspector 2.0* application's phase image filtering routine uses a 2D Canny-Deriche filter provided by the *CImg* toolkit, which was found to be fairly slow. There are three ways in which this process may be speeded up in future:

1. Use sine, cosine and arctan lookup tables for splitting and recombining phase images; taking the sine and cosine of the image as double precision floating point numbers is time consuming, as is recombining it with arctan. The same arctan lookup table used in the phase mode may be used instead, as well as implementing sine and cosine lookup tables. However, these operations are only performed once per filtering cycle, so the overall performance improvement would be marginal, whereas the blurring process is performed up to 50 times per cycle.

2. Implement Gaussian filtering; *CImg* has support for image convolutions using special kernels, but not Gaussian filtering. Gaussian filtering has computational benefits, as the Gaussian kernel has the special property of being linearly separable. This allows two 1D passes instead of one 2D pass, which is computationally superior.

3. Region-of-interest processing would decrease the total area to be smoothed, which would decrease the computational time in an approximately quadratic manner. As region-of-interest processing is already implemented in the unwrapping application, it would make sense to only filter the region-of-interest as sections of the filtered image are discarded anyway.

8.3.3 Implement saving/opening routines for more image formats

The current implementation uses the *CImg* toolkit for loading and saving images. Natively, *CImg* only supports bitmaps (*.bmp), but can use other plug-ins, such as *ImageMagick* to save other formats. Alternatively, Microsoft's GDI+ library can be used to read a wide variety of image formats.

8.3.4 Further investigation of Visual Studio Installer

The installer program used to generate the *Inspector 2.0* installer was *MakeMsi*. This scripting language is highly configurable, but is not as simple or intuitive to use as the Visual Studio Installer program. *MakeMsi* was used because it supports custom actions simply (required to install camera drivers), whereas Visual Studio has a more complicated process for custom actions. In the long term however, Visual Studio Installer would be a better solution because of its user-friendliness.

8.4 Recommendations for future work

The current design of the shearography head is extensible as it offers opportunities for the use of other peripheral devices on the same I^2C bus that the DAC uses, as well as simpler control using the four GPIO pins. These devices could easily be software-controlled over the same FireWire cable as the camera.

8.4.1 Software-implemented laser on/off switch

The ability to turn the laser on and off from the *Inspector* application may be useful in certain situations. This could be achieved using a GPIO pin and transistor switch. One could also use an electronically-triggered shutter to 'strobe' the laser light. This method has been shown to be effective in reducing the effects of vibration in shearography [51].

8.4.2 Increasing the maximum inspection area

The utility of shearography can be limited by the maximum inspection area achievable and the fact that the head must be repositioned every time the inspection area must be changed. Using better lasers to uniformly illuminate a larger surface and higher resolution cameras will allow larger areas to be inspected at one time. Another way of inspecting larger areas at one time is to use an active head positioning system; servo motors to control the head position (in two degrees of freedom, as in figure 81) via the same FireWire cable would allow different areas to be inspected without the need for repositioning the camera head, and would be a step towards automated inspection.

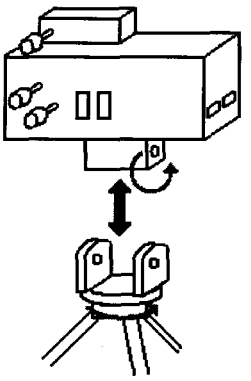


Figure 81: Automatic head positioning using servo motors (2 DOF)

8.4.3 Remote adjustment of lens, aperture, laser and shear mirrors

The use of gears to adjust the camera focus and aperture is not ergonomic; the user needs visual feedback to correctly adjust focus & aperture, making it difficult to adjust these controls on the shearography head while craning to see the screen. Allowing software control of these features via the use of servo motors would eliminate this problem. A similar problem exists with the adjustment of the shear and laser mirrors which could also be solved using servo motors.

8.4.4 Inclusion of infra-red lamp in shearography software

Currently, the infra-red lamp that is used to stress the specimen is controlled separately from the *Inspector 2.0* application. It would be desirable for the heating lamp to be controllable from the *Inspector 2.0* application, as this would eliminate problems that have been experienced in the past using timers and control circuits for the lamp [52].

8.4.5 Rewrite *Unwrap* in C++ using *CImg*

The phase-unwrapping utility was developed in *MATLAB* because of its high-level functions and ease of use. However, distributing the compiled application requires that the MCR is also distributed. This is a 150MB file, which makes distribution more difficult, and the requisite *MATLAB* license is costly.

It would be desirable for the *Unwrap* utility to be rewritten in C++. The *CImg* toolkit, which was used during this dissertation, contains most of the image-processing functions required, as well as tools for producing and rendering 3D surfaces (required for surface visualization).

This toolkit is free and would provide faster unwrapping options. Using fast Gaussian filtering and rapid phase-unwrapping, it may even be possible to watch real-time 3D surface plots being rendered (at a slower frame-rates than pure video). Another development that may make this possible would be the use of region-of-interest processing instead of processing the entire image as is currently the case.

8.4.6 Investigate new forms of laser illumination

The quality of interference images acquired is dependent on the stability of the laser illumination provided. It was found during this dissertation that the laser light available from the TECRL modules used is not always of the highest quality, and often needs to be tuned to avoid mode hopping and multiple frequency bands.

Also, these modules are only capable of inspecting small areas at a time. Further investigation should be undertaken using arrays of less powerful laser diodes to illuminate larger areas more uniformly and with greater stability.

9. REFERENCES

[1] D Findeis, J Gryzagoridis, A Andhee, "Comparison of Normal and Phase-Stepping Shearographic NDE", Non-Destructive Testing Laboratory, University of Cape Town, pp11, 2005

[2] HA Aebischer, S Waldner, "A Simple and Effective Method for Filtering Speckle-Interferometric Phase Fringe Patterns", Optics Communication 162, 1999

[3] J Gryzagoridis, D Findeis, "Benchmarking Shearographic NDT for Composites", Proceedings of 46th Annual British Conference on NDT", pp250 2007.

[4] D Findeis, J Gryzagoridis, "A Comparison of the Capabilities of Portable Shearography and Portable Electronic Speckle Pattern Interferometry", Non-Destructive Testing Laboratory, University of Cape Town, pp1, 2004

[5] B Pitman, MSc Project, "Phase Unwrapping of Phase-Stepped Interference Images", Non-Destructive Testing Laboratory, University of Cape Town, 2007

[6] 1394 Trade Association, http://www.1394ta.org/index.html, Last Accessed: 14 April 2008

[7] I^2C Bus Specification, http://www-us2.semiconductors.philips.com/acrobat/literature/9398/39340011_21.pdf, Last Accessed: 25 September 2008

[8] LabVIEW website, http://www.ni.com/labview/, Last Accessed: 12 April 2008

[9] Matrox Imaging Library, http://www.matrox.com/imaging/home.htm, Last Accessed: 13 August 2008

[10] J Gryzagoridis, D Findeis, "Benchmarking Shearographic NDT for Composites", Proceedings of 46th Annual British Conference on NDT", pp249 2007.

[11] O. Erne, T. Waltz, A. Ettemeyer, "Composite Structural Integrity NDT with Automatic Shearography Measurements", American Society for Non-Destructive Testing, 2000

[12] D Findeis, J Gryzagoridis, E Xaba, D Reid-Rowland, "Aircraft tires inspection

using portable shearography and electronic speckle pattern interferometry" Non-Destructive Testing Laboratory, University of Cape Town, pp1, 2000

[13] D Findeis, J Gryzagoridis, W Bopape, "Impact damage detection on composites using electronic speckle pattern interferometry", Non-Destructive Testing Laboratory, University of Cape Town, pp7, 2005

[14] Y.Y. Hung, H.P. Ho, "Shearography: An optical measurement technique and applications", Department of Manufacturing Engineering and Engineering Management, City University of Hong Kong, 2005

[15] RTI HG7000 Interferometry Computing System, http://www.pcholo.com/hg7000.htm, Last Accessed: 2 April 2008

[16] Michael Kalms, Werner Jueptner, "Mobile shearography", BIAS - Bremen Institute of Applied Beam Technology, Germany, 2005

[17] Portable Shearography System Q-810, http://www.dantecdynamics.com/Default.aspx?ID=673, Last Accessed: 21 April 2008

[18] R Jones, C Wykes, "Holographic and Speckle Interferometry", 2nd Edition. Cambridge University Press, 1989.

[19] Chiayu Ai, James Wyant, "Effect of piezoelectric transducer nonlinearity of phase shift interferometry", Optical Fiber Measurement Symposium Proceedings, 1987.

[20] AM Maas, PM Somers, "Two Dimensional Convolution Applied to Phase-stepped Shearography", Optics and Lasers 26, pp351-360, 1997

[21] Canny Edge Detector, http://en.wikipedia.org/wiki/Canny_edge_detector, Last Accessed: 3 October 2007

[22] Noe Alcalá Ochoa, A.A. Silva-Moreno, "Fringes demodulation in time-averaged digital shearography using genetic algorithms", Centro de Investigaciones en Optica, Mexico, 2005

[23] J.M. Huntley, "Noise-immune phase-unwrapping algorithm", Appl. Opt. 28, 3268-3270, 1989

[24] Hugo Krynauw, BSc Thesis, "Laser diode cooling using a thermo-electric cooler", Non-Destructive Testing Laboratory, University of Cape Town, 2006

[25] IEEE 1394 interface, Technical Specifications,

http://en.wikipedia.org/wiki/FireWire, Last Accessed: 23 July 2008

[26] IEEE 1394 interface, Comparison to USB, http://en.wikipedia.org/wiki/FireWire Last Accessed: 23 July 2008

[27] Powered USB, http://en.wikipedia.org/wiki/PoweredUSB, Last Accessed: 1 June 2008

[28] Michelson, A. A.; Morley, E. W. "On the Relative Motion of the Earth and the Luminiferous Ether". American Journal of Science 34 (203): 333–345, 1887

[29] Edmund Optics, World's Largest Inventory of Optical Components, www.edmudoptics.com, Last Accessed: 10 September 2008

[30] Mark Murphy, Mel Conway, Gary Casey, "Lens drivers propel high-resolution cell phone cameras", available at: http://www2.electronicproducts.com/Lens_drivers_propel_high-res_cell_phone_cameras-article-facnanalog-oct2007-html.aspx, Last Accessed: 10 May 2008

[31] Basic Designs of Piezoelectric Positioning Drives/Systems, http://www.physikinstrumente.com/en/products/prdetail.php?sortnr=400800.10 Last Accessed: 13 May 2008

[32] C. Webb, D. Jones, "Handbook of Laser Technology and Applications", Published by Taylor and Francis, pp. 1826, , 2004

[33] Kim, Seong Jong; Kang, Young June; Hong, Dong Pyo; Kim, Kyung Suk; Park, Nak Kyu; Ryu, Weon Jae; Choi, Man Young "Precision displacement measurement using a modulating ESPI", Proceedings of the SPIE, Volume 6719, pp. 671908, 2007

[34] CMU 1394 Digital Camera Driver, http://www.cs.cmu.edu/~iwan/1394/, Last Accessed: 23 April 2008

[35] C++ Template Image Processing Toolkit, http://cimg.sourceforge.net/, Last Accessed: 11 September 2008

[36] ImageMagick: Open source image processing and conversion suitehttp://www.imagemagick.org/script/index.php, Last Accessed: 10 September 2008

[37] Dantec Dynamics, "Laser optical measurement systems and sensors", http://www.dantecdynamics.com Last Accessed: 18 November 2008

[38] A Andhee, "A Novel Compact Shearographic NDT System", MSc Dissertation, Non-Destructive Testing Laboratory, University of Cape Town, 2005

[39] Physik Instrumente, The World of Nano and Micro Positioning, http://www.physikinstrumente.com/, Last Accessed: 13 May 2008

[40] PiezoMechanik, Innovative Motion Control, http://www.piezomechanik.com/en/home/introduction/index.html?1, Last Accessed: 11 May 2008

[41] PiezoSystemJena, Equipment for Nano-Positioning, http://www.piezojena.com/, Last Accessed: 11 May 2008

[42] Mad City Labs, Nano-Positioners, http://www.madcitylabs.com/, Last Accessed: 11 May 2008

[43] Eagle PCB Layout Editor, http://www.cadsoft.de/freeware.htm, Last Accessed: 11 June 2008

[44] S. Burbeck, "Applications Programming in Smalltalk-80™: How to use Model-View-Controller (MVC)", 1987

[45] Mutex Class – MSDN Documentation, http://msdn.microsoft.com/en-us/library/system.threading.mutex(VS.71).aspx, Last Accessed: 2 September 2008

[46] A. Dávila, J.E.A. Landgrave, G. Garnica, "In situ calibration of a Michelson type, speckle-shearing interferometer: Wobbling mirror effect", Centro de Investigaciones en Óptica, México, 2006

[47] Subversion, an open-source version control system, http://subversion.tigris.org/, Last accessed: 2 February 2008

[48] Dennis Baries' MakeMsi Windows Installer Toolkit, http://dennisbareis.com/makemsi.htm, Last Accessed: 4 August 2008

[49] Windows Installer: Custom Action Table, http://msdn.microsoft.com/en-us/library/aa368062.aspx, Last Accessed 23 July 2008

[50] Vincent Musonda, MSc dissertation, "Comparative study of non-destructive testing methods for defect detection on aircraft", Non-Destructive Testing Laboratory, University of Cape Town, Experimental Techniques, pp56 , 2007

[51] R. Pezzoni, R. Krupka, "Laser-Shearography for Non-destructive Testing of Large Area Composite Helicopter Structures", Proceedings of 15th World

Conference on Non-Destructive Testing, 2000

[52] D Smith, "Development of a Heating Mechanism for Digital Shearography", Non-Destructive Testing Laboratory, University of Cape Town, 2008

APPENDIX A: OPERATION MANUALS

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. Inspector 2.0 application

1.1 Overview

Shearography is an optical, non-contacting and full-field Non-Destructive Testing (NDT) technique used in numerous industries. The Inspector 2.0 Shearography System is a portable inspection system used to detect flaws, cracks, debonds and delaminations in a wide range of materials. It is a complete system, consisting of:

1. The shearography head, containing a digital camera, diode laser, speckle-shearing interferometer and processing board.
2. A tripod-stand for fine-positioning of the shearography head.
3. A FireWire-enabled laptop or PC, loaded with the Inspector 2.0 Digital Shearography application.
4. A 6-pin to 6-pin FireWire cable and/or either a 6-pin to 4-pin FireWire cable or a mains-powered FireWire laptop adaptor.

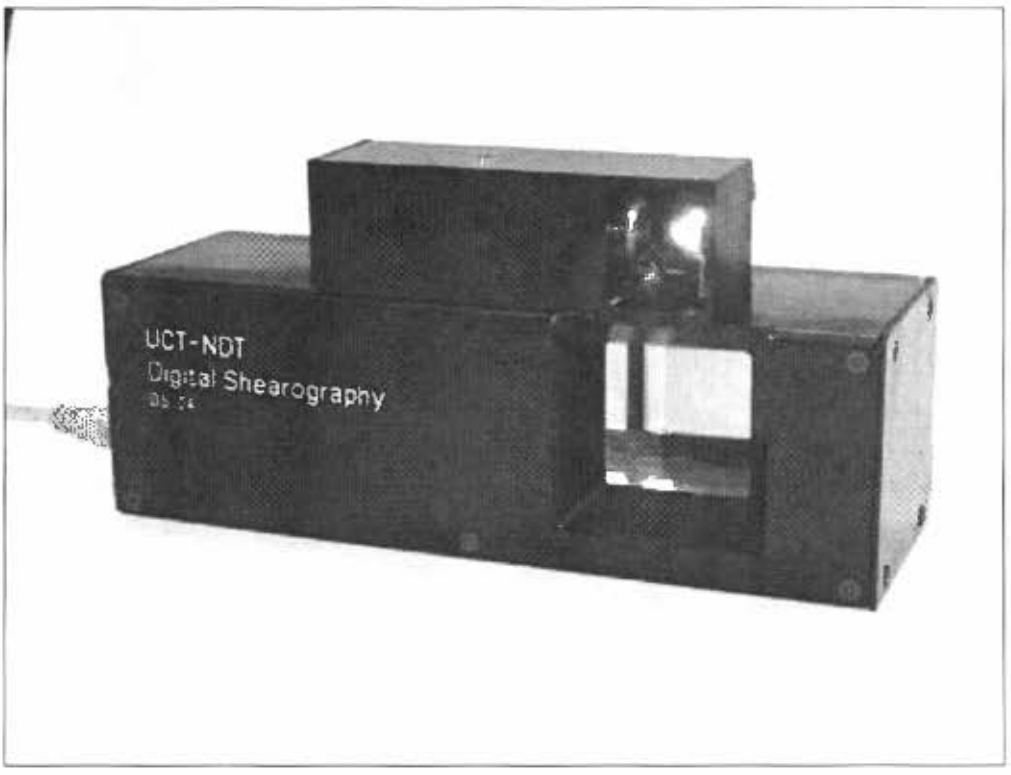


Figure 1: Shearography head

1.2. Installation

Note:

Please install the Inspector 2.0 software before connecting the shearography head, as the camera drivers must be installed before connection. If the camera is inadvertently connected first, disconnect it and restart the PC. Proceed by installing the Inspector 2.0 application.

1.2.1 Installing the Inspector 2.0 application

If the installer does not run upon insertion of the installation disc, browse to the disk, and double click the installer package, *Inspector.msi* (or *setup.exe*) which will install all of the necessary files and perform the necessary registration.

Figure 2: Inspector 2.0 installer dialog

The application has the following minimum requirements: 32-bit Windows 2000, XP or Vista, 1.5GHz processor, 512MB RAM, 50MB hard drive space, 1 × FireWire 400 port and VGA Screen.

The application may be uninstalled by following: *Control Panel->Add/Remove programs->Inspector 2.0.*

1.2.2 Installing MATLAB Runtime Component (MCR)

The Inspector 2.0 application's phase unwrapping plug-in uses the MATLAB MCR. Please also install this application when installing Inspector 2.0. Browse to the installation CD, and double-click *MATLAB Component Runtime 7.5.msi*. This installation may take a few minutes.

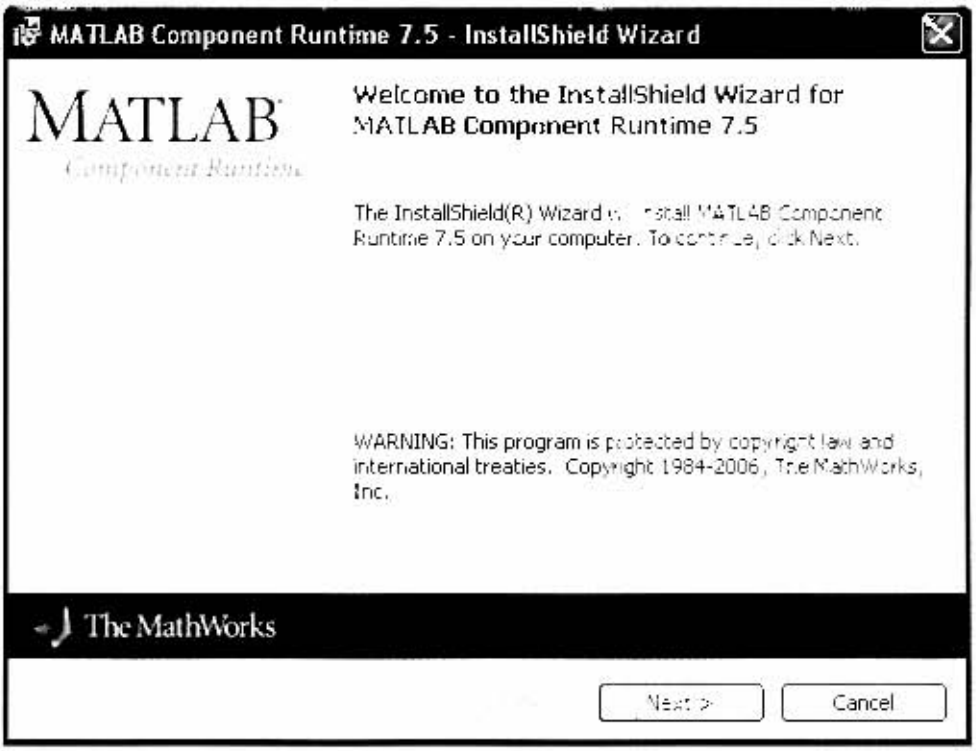


Figure 3: MATLAB MCR installer dialog

Notes:

1. Ensure that the MCR installed is the same MCR that is distributed with Inspector 2.0, as the versions of the compiled Inspector 2.0 MATLAB application and the MCR must match.
2. If the installer complains that the .NET framework is not installed, click 'OK' to continue, as the .NET framework is not required by Inspector 2.0.

1.3. Connecting the shearography head

The shearography head connects to the target PC using a FireWire cable. The shearography head has two FireWire connectors, either of which may be used.

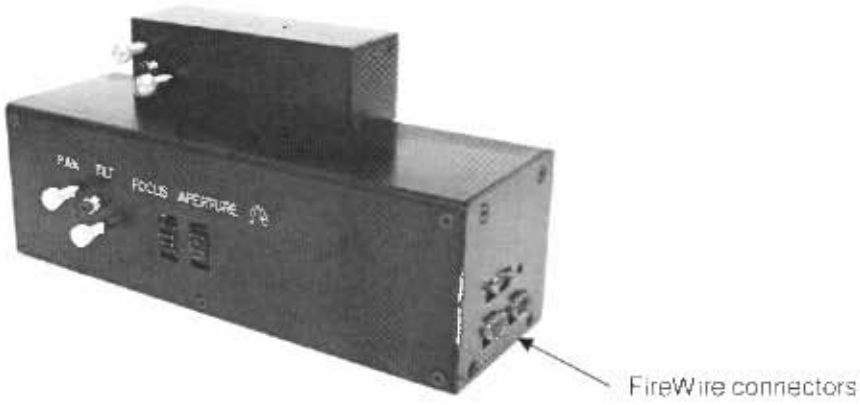


Figure 4: Rear view of shearography head

The shearography head may connect to either a PC or laptop, the instructions for which are slightly different, owing to the fact that most laptops do not supply power over their FireWire ports:

Connecting to a PC:

Use a 6-pin to 6-pin FireWire cable to connect the head to the PC.

Connecting to a laptop:

This requires additional power to be supplied, which may be achieved in one of two ways:

Using a laptop adaptor kit:

Setup the adapter kit as specified by the manufacturer, which will probably involve connecting it to the mains. Connect the shearography head to the adaptor using a 6-pin to 6-pin FireWire cable.

Using an external power supply:

The shearography head has a Φ2.1mm DC power connector to supply 12VDC to the head, via either battery or other means. Once powered, use a 4-pin to 6-pin FireWire cable to connect the head to the laptop.

The shearography head LED indicator light will flash orange during startup and turn solid green when successfully initialized.

Note: The head will draw ±6W under normal operation.

1.4. Running Inspector 2.0

The installer should have installed two shortcuts on the desktop, called *Inspector 2.0* (the main shearography application) and *Phase Unwrapper* (the phase-unwrapping plug-in). Double-click the Inspector 2.0 icon. Alternatively, the application should typically be contained in the *C:\Program Files\Inspector* directory. The following screen (figure 5) should appear.

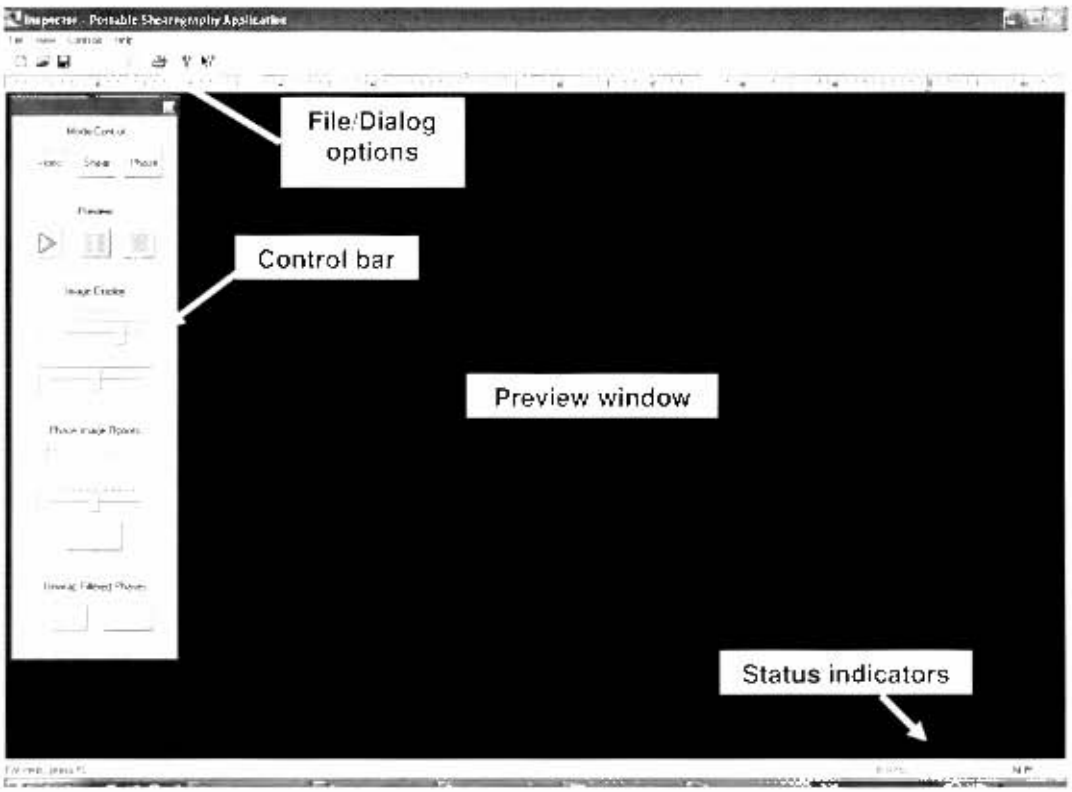


Figure 5: Screenshot of Inspector 2.0 application

In order to test that the application detects the shearography head correctly, press the ▷| button ensuring that the shearography head is connected and its indicator LED is showing solid green. This should start video streaming into the preview window. Note however, that the aperture and focus should be adjusted for image clarity, which is described in the following section.

If the application gives the error message, '*There is no camera connected!*', see the section entitled *Troubleshooting*. If the screen appears still black, but there was no error message, it is likely that the camera is streaming video but it is either too dark or the aperture setting is closed. Continue on to the next section.

1.5. Adjusting focus, aperture, image shear and laser direction

The shearography head does not perform automatic focusing and aperture adjustment, and as such these should be adjusted manually using the adjustment wheels at the back of the shearography head. In order to determine the best settings; press ▶| with the Inspector 2.0 application in *Video* mode. Adjust the focus and aperture to get the best picture quality.

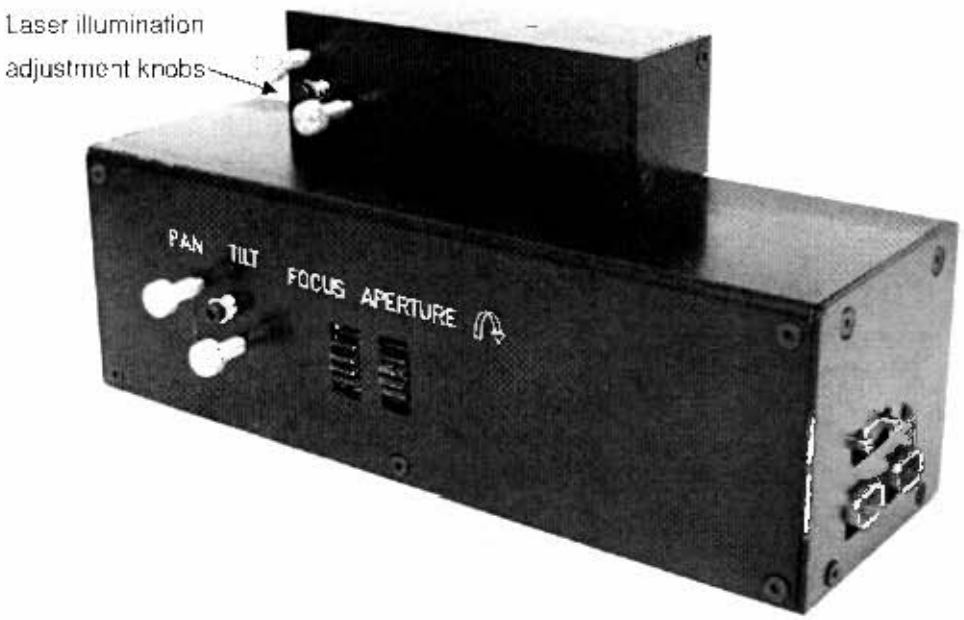


Figure 6: Description of shearography head controls

Next, adjust the shear mirror position using the control knobs labelled '*PAN*' and '*TILT*' on the shearography head. This should adjust the amount of shear occurring between the two image beams, with '*PAN*' shearing horizontally and '*TILT*' shearing vertically. As a rough guideline, shearography requires a shear magnitude of 50mm between the two beams (measured at the reflecting surface).

Finally, adjust the laser illumination direction using the two adjustment knobs situated as shown in the image above. The brightest part of the laser illumination should coincide with the centre of the video images in the Inspector 2.0 application.

1.6. Performing inspections

Shearography requires that the specimen is stressed during the inspection process. The stressing of the specimen reveals anomalies in the way that the specimen deforms, which serves as the means to identify flaws. The means by which stressing is achieved is unimportant; it may be either thermal, mechanical or vibratory.

In Inspector 2.0, inspections may be performed in either *Shear* or *Phase* modes. Both store a copy of the first frame that was grabbed, and use it to compare with the new frames that are captured, to reveal flaws as fringe patterns. Hence, while it is important that the relative positions of the shearography head and specimen do not change during the recording process, it is important that the stress-state of the specimen does change to reveal the way in which it deforms.

The following procedure may be following to test for flaws:

1. Put the application in *Video* mode, and press ▶. Ensure that the camera is focussed and images are reasonably bright. Press ■.
2. Put the application in *Shear* mode. Press ▶, and then stress the specimen using whatever method available. Adjust the *Brightness* and *Contrast*. If a flaw is present, fringe patterns should develop on screen, such as in figure 7.

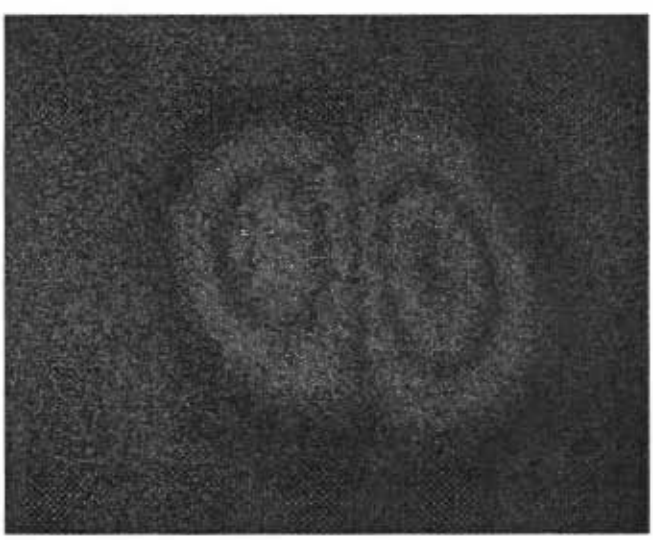


Figure 7: Example fringe pattern

3. Press ▮▮ or ■ to save the image if necessary.

4. Put the application in *Phase* mode. Press ▶| and stress the specimen again (or just resume by pressing ▶ if ▮▮ was pressed in step 3). Again fringe patterns should develop if there is a flaw present. This fringe pattern will look slightly different, as in figure 8.

Figure 8: Example phase-stepped fringe pattern

5. Press ▮▮ or ■ to save the image if necessary.
6. Filter the phase stepped image by pressing the *Filter* button. This may take a few seconds. The resulting image should look similar to figure 9.

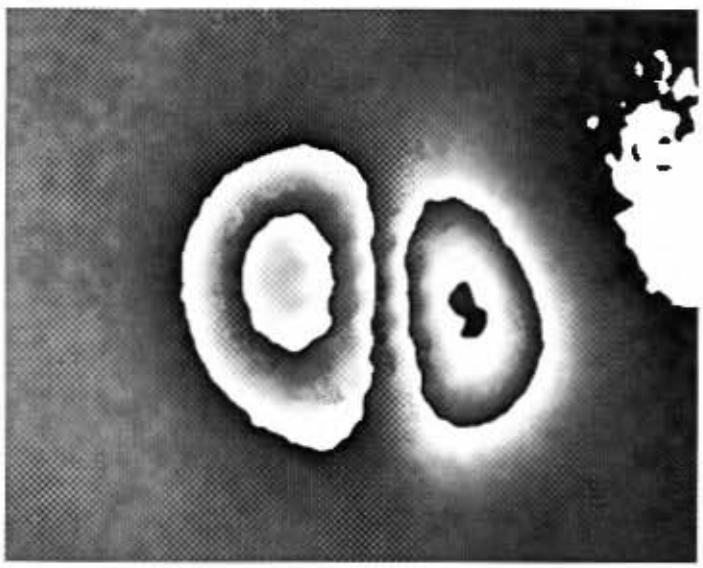


Figure 9: Example filtered phase image

7. Press ▮▮ or ■ to save the image if necessary.

1.7. Inspector 2.0 control bars and functions

Figure 10 shows the Inspector 2.0 menu bar. The '*File*', '*View*', and '*Help*' menus provide the standard functionality:

- Opening and saving images (in *.bmp file format)
- Printing images using the standard Windows Printer dialog
- Showing and hiding toolbars
- Getting context-sensitive help

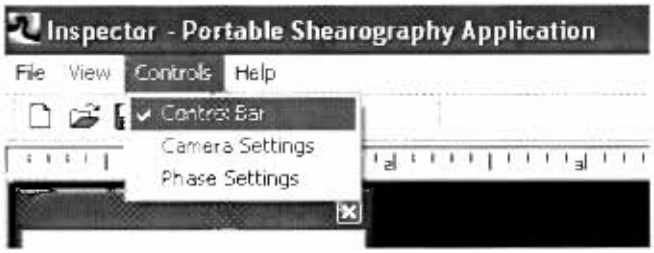


Figure 10: Inspector 2.0 menu bar

The '*Controls*' tab controls the visibility of certain application-specific toolbars. The '*Control Bar*' is opened by default, and may be hidden if needed. The functions of these toolbars are discussed in greater detail in the following sections.

1.7.1 Control bar

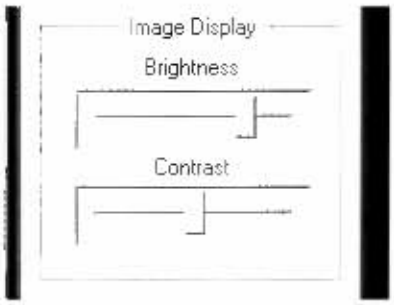
The '*Mode Control*' toggle buttons alter the way in which video is displayed, while the '*Preview*' buttons control the state of the video preview.

In '*Video*' mode, the video is streamed directly from the camera to the application's Preview Window. This mode is used to test the setup of the camera's focus and aperture settings, as well as the image shear and laser illumination settings.

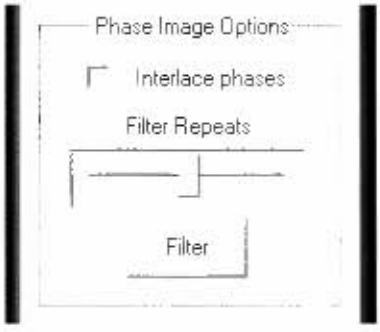
The '*Shear*' mode is used to perform intensity-based shearography. Every new video image is subtracted from the first captured image to determine the phase change of the reflected light.

'*Phase*' mode performs phase-stepped shearography by shifting the mirror by fractions of the laser's wavelength in synchronization with image capture. Typically, this reduces noise and gives clearer results.

The ▶ (play) button initiates video previewing. While playing, most other buttons will be disabled until previewing is stopped by pressing either ▮▮ (pause) or ■ (stop) buttons. The ▮▮ button will pause the video preview, but not discard the originally grabbed images, and as such is convenient for saving images and then continuing with processing from where it left off. To restart processing, discarding the original base images, press ■ followed by ▶. Note that the user may switch between *Video*, *Shear* and *Phase* modes without first stopping or pausing the video feed.



The '*Image Display*' sliders are used to adjust the clarity of images in '*Shear*' mode. Typically, these images are dark, and should be brightness and contrast-adjusted for optimum clarity. Note that these sliders do not affect the camera settings or the way in which images are captured from the camera. They perform brightness and contrast adjustment just before the image is displayed.



The '*Phase Image Options*' contain options that are only available in '*Phase*' mode. The checkbox, '*Interlace phases*' specifies whether or not the interlaced processing algorithm should be used to obtain phase stepped images. Interlaced Phases will give a higher frame-rate with better dynamic response compared to ordinary processing, but may be less accurate in certain circumstances. The default is ordinary processing.

The '*Filter*' button will perform low-pass filtering on the phase-stepped image in the display window, and replace the displayed image with the filtered image. The '*Filter Repeats*' slider determines the level of filtering to be applied; more filtering will give a smoother image. Figure 11 shows the difference between a phase-stepped image which has not been filtered, and one which has been filtered.

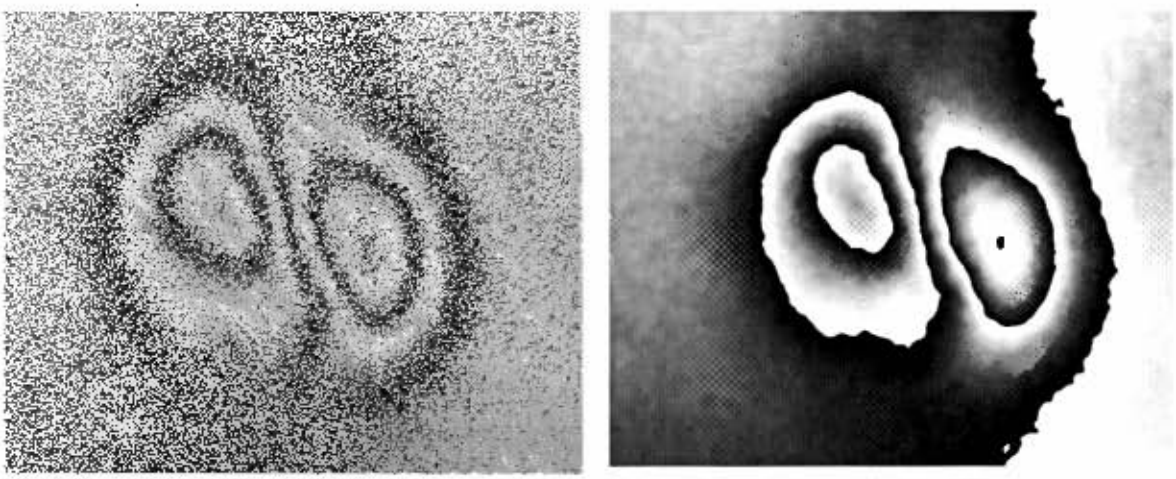
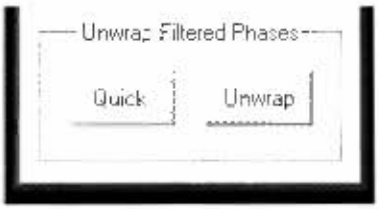


Figure 11: Difference between unfiltered and filtered images

The final options on the 'Control Bar' are the unwrapping options. These two buttons both perform phase-unwrapping or demodulation on filtered phased stepped images, such as the image above-right. Phase-unwrapping reconstructs the surface displacement from the phases, rendering a 3D plot of the surface.



There are two options; namely *Quick* and *Unwrap*. The *Unwrap* button launches another GUI for controlling the unwrapping process. The GUI contains numerous related settings, and gives options for where the output images should be saved. The *Quick* button performs phase-unwrapping without opening a GUI, using the default settings, and is typically faster than using the GUI. In this case, the output images will be saved to a folder in the project directory, but will also be opened for viewing after processing.

An example of the *Phase Unwrapper* GUI is shown on the following page in figures 12 and 13. Please see the documentation of the *Phase Unwrapper* for further information regarding its operation.

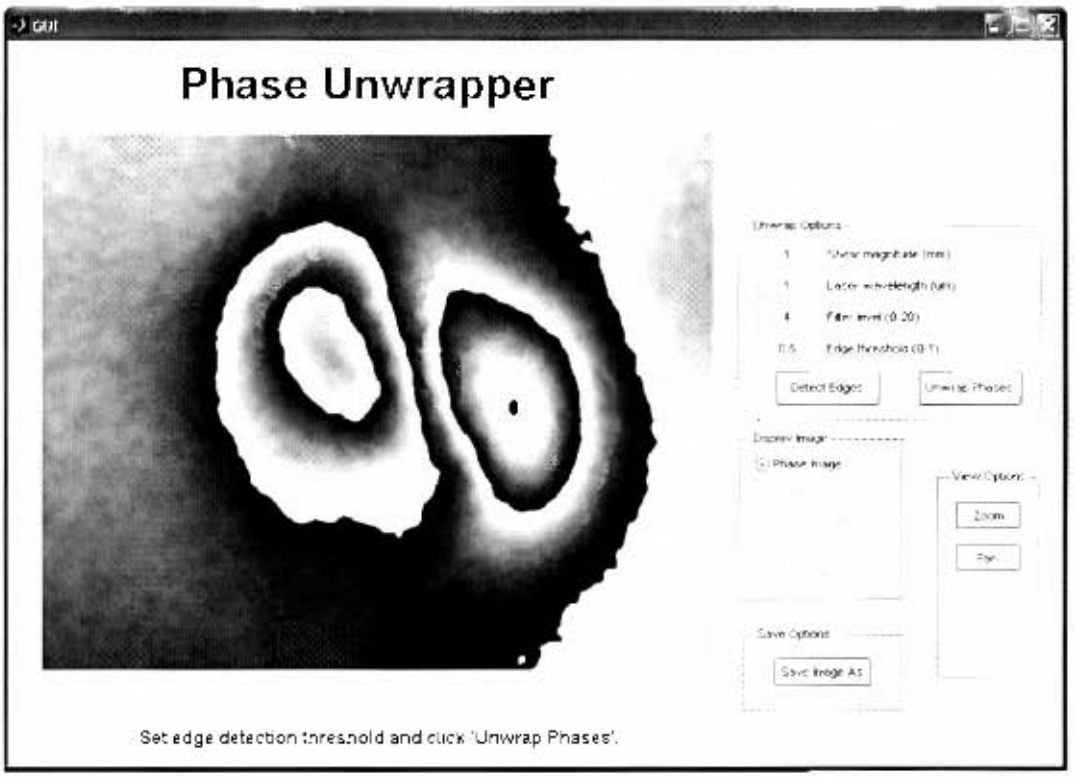


Figure 12: Screenshot of the Phase Unwrapper application

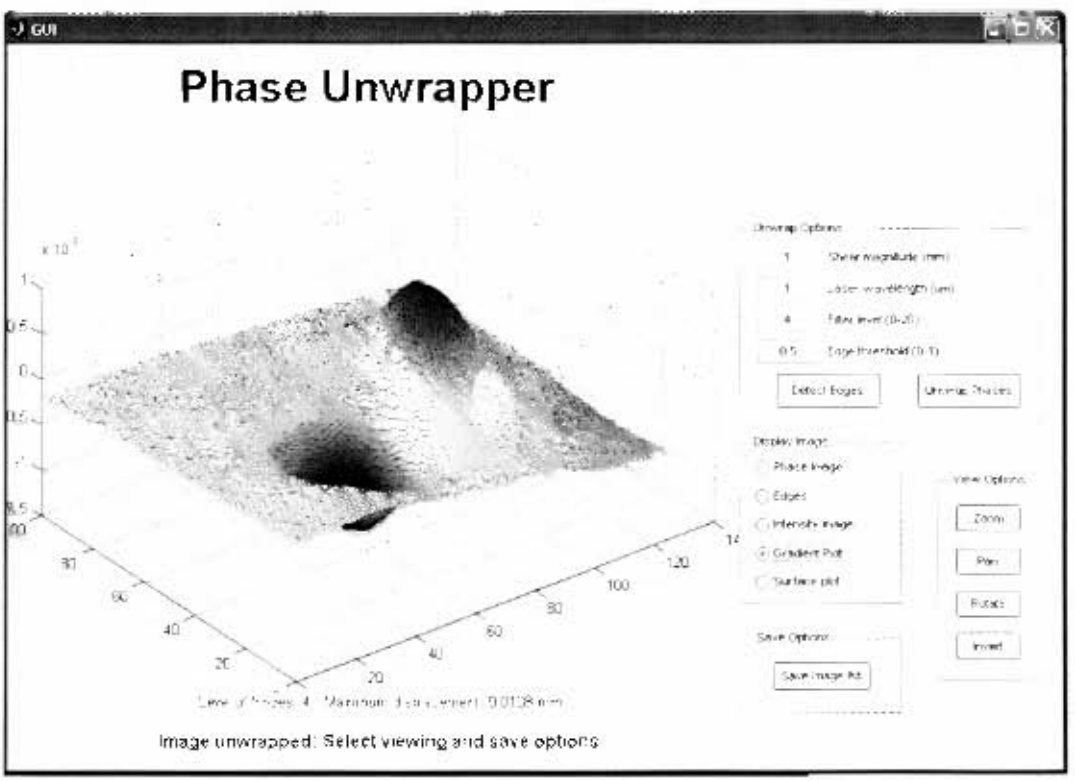


Figure 13: Example of unwrapped image

1.7.2 Camera settings

The camera settings dialog offers extended control over the camera's onboard settings, such as Exposure, Gain and Contrast. These can be modified to obtain better image quality. Note that adjusting the aperture is a better way of brightening images than increasing the exposure time, as increasing the exposure time will slow down operation.

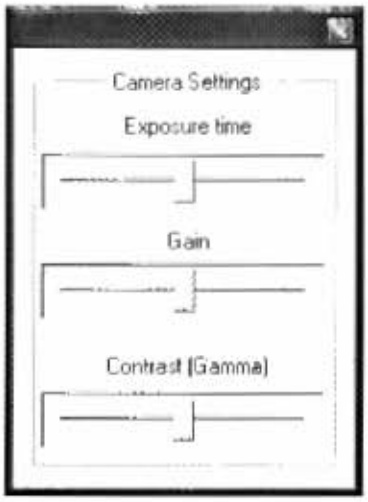


Figure 14: Camera Settings dialog

1.7.3 Status indicators

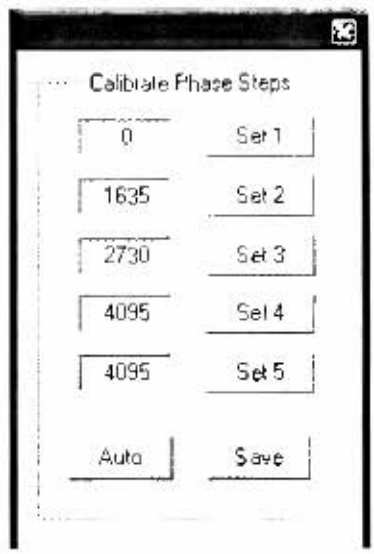
The status indicators is the lower right corner of the *Inspector* application offer useful feedback. The first indicator shows the display rate of frames in the application in frames per-second. The other indicator shows whether or not the piezo actuator is operating normally or not.

Figure 15: Status bar indicators

If operating normally, is should say *Piezo normal*, otherwise it will read *Piezo error*. Note that this indicator is only active in *Phase* mode.

1.7.4 Piezo settings

The piezo is used to actuate the mirror in *Phase* mode. It is important that the phase steps of the piezo match the laser's wavelength. The piezo is actuated using a voltage proportional to the phase step required. The piezo settings dialog can be used to test the piezo's calibration or for recalibration. The first five edit boxes and push buttons, *Set 1-5*, correspond to voltage steps of 0λ, ¼λ, ½ λ, ¾ λ and 1λ where λ is the wavelength of the laser. By pushing the button *Set 1*, the piezo will be actuated to the voltage specified in the neighbouring edit box.

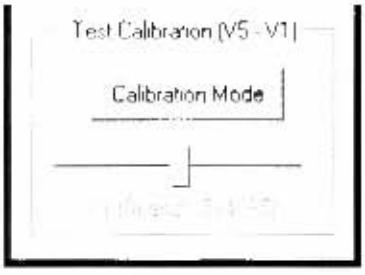


To test the calibration, put the application in *Shear* mode. Stress the specimen, and then press ▷|. Allow fringes to develop, and preferably acquire a permanent set. Then, using the piezo settings dialog, ensure that either: by pressing *Set 1* and *Set 5* alternately, the fringe pattern does not change, or by pressing *Set 1* and *Set 3* alternately, the fringe pattern will invert (white fringes will change to black).

If this is not the case, alter the edit boxes for these voltages so that this does occur. Once completed, press the *Auto* button to auto calculate that phase voltages for the other voltage steps (this approach will fill in the voltages based on a linear scale between Set 1 and Set 5, hence will not work if the calibration was performed using Set 1 and Set 3).

Once completed, press Save to save these settings for later use in the *Phase* mode. These settings will be reloaded again on startup.

The *Calibration Mode* toggle button is analogous to the *Video*, *Shear* and *Phase* modes. This mode can also be used to test the calibration of the piezo, but is still in beta development.



To use this mode, press the *Calibration Mode* toggle button, and stress the specimen. Press  and allow fringes to develop, preferably acquiring a permanent set. The *Brightness* and *Contrast* sliders may be used to adjust picture quality. Now, use the Calibration slider, situated under the Calibration Mode button to test the calibration of the piezo. This mode will actuate the piezo between Set 1 and Set 5, and subtract the resulting images. If properly calibrated, there should be no difference between *Set 1* and *Set 5*, and hence the output should be black. If not calibrated properly, there will be some semblance of fringe patterns. Use the slider to adjust the *Set 5* voltage so as to obtain a black output. Once obtained, transfer this value of *Set 5* to the edit boxes above, and *Save*.

1.8. Specifications

Table 1: Specifications

Specification			Value
Camera resolution			1320 × 1040 pixels
PC Interface			FireWire (1394a)
Frame rate[1]	Video		25fps
	Simple intensity-based		25fps
	Phase-stepped	Non-interlaced	6.2fps
		Interlaced	25fps
Supported image formats			*.bmp
Shear angle			-15° to 15°
Inspection area[2]			225 × 165mm
Maximum working distance[3]			±3m
Laser			660nm, 100mW
Overall dimensions			113 × 74 × 224mm
Power consumption	Camera		3.4W
	Laser		<2W
	DAC & PZT		<0.5W
	Total		<6W
Minimum PC/laptop requirements			32-bit Windows 2000, XP or Vista, 1.5GHz processor, 512MB RAM, 50MB hard drive space, VGA Screen
Cost			R39 000

Notes:

1. Measured using a 2.8GHz Pentium 4 with 512MB of RAM.
2. At working distance of 1.5m
3. Depends on specimen reflectivity

1.9 Troubleshooting

1.9.1 Installer fails

The installer requires administrator privileges. Make sure you are logged in as an administrator.

1.9.2 Application does not detect camera

This may be due to several reasons:

1. The camera is not connected.
2. The drivers were installed while the camera was connected. In this case, disconnect the camera, wait 5 seconds and try again. In this does not work, try rebooting the PC. Finally, try reinstalling the application.
3. The camera is connected to a 4-pin FireWire port and has no external power. When connecting to a 4-pin port, power should be supplied either using a FireWire laptop adaptor kit (4-pin to 6-pin) or an external 12V source via the 2.1mm coaxial connector.
4. Try restarting the Inspector 2.0 application.
5. If the problem persists, contact support.

1.9.3 Laser does not turn on or piezo does not respond.

This may be due to the DIP switch setting on-board the controller board. Remove the front cover plate, and the DIP switches will be accessible (the whole cover does not need to be removed). There is a switch for each of the piezo (channel 2) and laser (channel 1). They should both be set in the 'ON' position.

1.9.4 Phase-stepped images do not show clear fringes

This may be due to incorrect piezoelectric actuator calibration. Try using the software calibration mechanism provided (go to *Controls->Phase Settings*). Put the camera into shear mode, and start stressing the specimen. Press  , allowing fringes to acquire a permanent set. Then, using the *Set* buttons, adjust the piezo voltages such that either *Step 1* and *Step 5* give the same fringe pattern, or such that *Step 1* and *Step 3* give inverted fringes. Press *Save* to save these settings.

1.9.5 LED indicator codes

The camera's LED light often gives valuable information regarding the state of the camera. The indicator codes may be summarised as follows.

LED Indicator codes

Table 2: LED Codes

LED Effect	Description
Flashing orange	The camera is initializing. This happens first.
Solid orange	The camera is loading the FFC parameters from memory. This process can take up to 20 seconds.
Flashing green	The camera is performing a lengthy operation or is streaming video data.
Solid green	The camera is ready for operation
Flashing red	The camera has issued a warning on the latest command received
Solid red	The camera has experienced an unrecoverable error. Go to Technical Support if this happens.

1.9.6 Phase-unwrapping utility does not open

This component requires that *MATLAB Runtime Component* is installed. First ensure that this is the case. If it is installed, check that the phase-unwrapping utility is also installed; after running the installer, there should be a shortcut on the desktop *Phase Unwrapper*, otherwise look in *C:WINDOWS* for *unwrap.exe* & *unwrap.ctf*. Double click the .exe file, and the unwrapping utility will open. If it doesn't, look for a folder called *unwrap_mcr* in the same directory and delete it. Double click the *.exe file again and it should open. You can load and save images from there.

2. Phase unwrapping utility

2.1 Overview

Unwrap is a MATLAB-compiled phase-unwrapping utility that is used to reconstruct a specimen's surface displacement from phase-stepped shearography or ESPI images. The utility can operate in stand-alone mode, or act as a plug-in to the *Inspector* shearography application.

2.2 Installation

The application relies on MATLAB Component Runtime (MCR). As such, the MCR must first be installed on the target PC. Install this environment by double-clicking on the *MATLAB Component Runtime 7.5.msi* file contained in the distribution. This installation may take a few minutes.

Notes:

3. Ensure that the MCR installed is the same MCR that compiled the *Unwrap* application; the versions of the compiled *Unwrap* application and the MCR must match.
4. If the installer complains that the .NET framework is not installed, click 'OK' to continue, as the .NET framework is not required by Inspector 2.0.

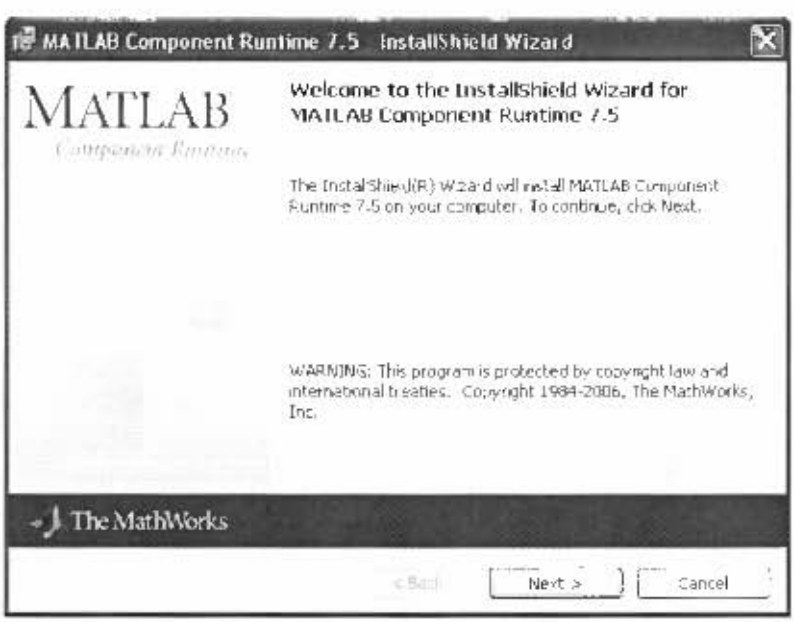


Figure 16: MATLAB MCR installer dialog

Next, copy the *Unwrap* application to the target PC's *C:\WINDOWS* directory (or any other directory in the operating system's PATH environment variable). The *Unwrap* application consists of two files, *unwrap.exe* and *unwrap.ctf*. Both of these files should be placed in the same directory. This completes the installation.

2.3 Testing the installation

To test that the application is installed properly double-click on the *unwrap.exe* file in the *C:\WINDOWS* directory. A command prompt should open the first time that the program runs, as in figure 17 below:

Figure 17: CTF archive extraction

Subsequently, the GUI will open (this may take a few moments):

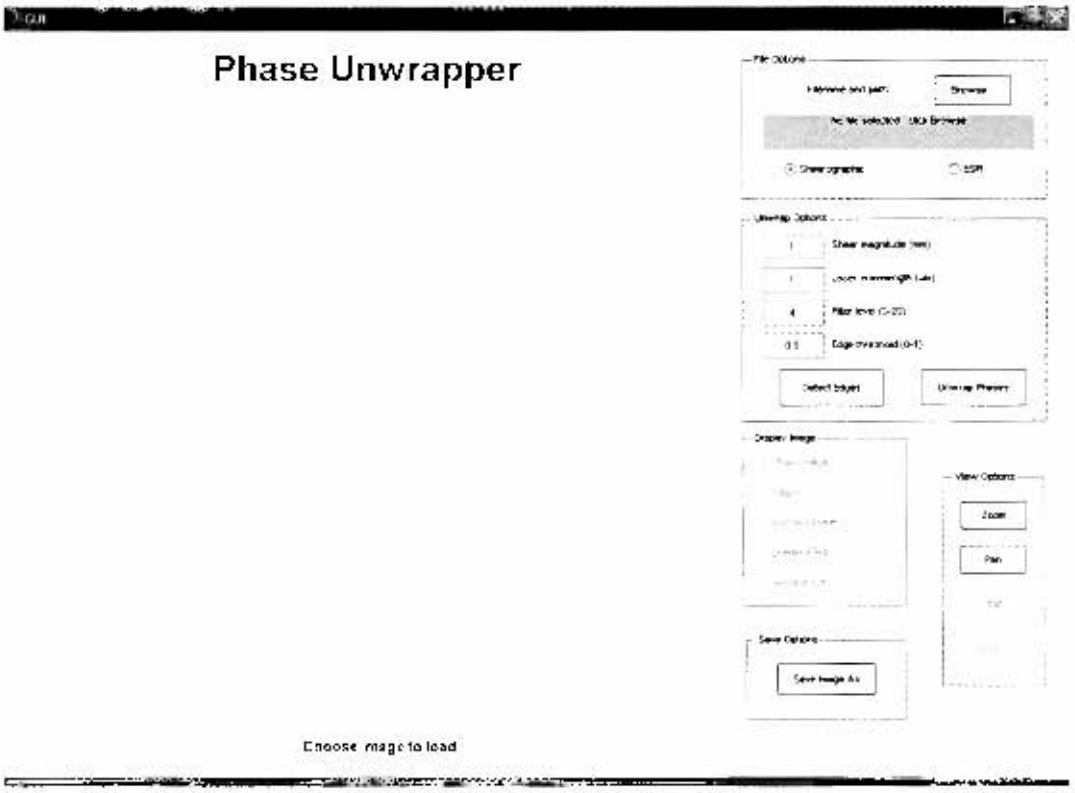


Figure 18: Unwrap screenshot

2.4 Using *Unwrap*

The process for loading an image to unwrap depends on whether the application was launched from the *Inspector* application or by double-clicking. *Inspector* will automatically open the image, whereas when double-clicking, the image should be found using the *Browse* button. Once a filtered phase image has been loaded, the typical process for performing phase-unwrapping on an image using the *Unwrap* utility is as follows:

2.4.1 Selecting the image region to process

The loading of an image will automatically turn the cursor into a set of crosshairs for selecting the image region to process. Click two different points on the source image to select a region. The new region will be the sub-region defined by the diagonal between the two points selected, as follows in figure 19:

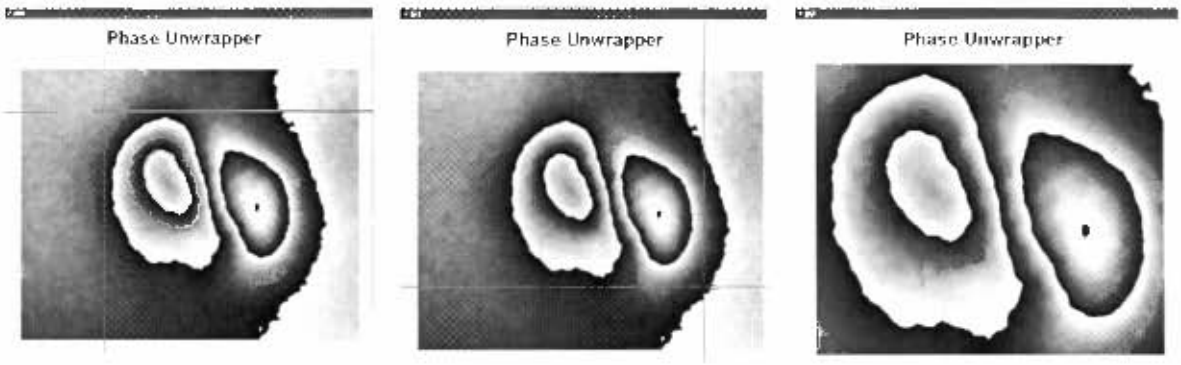


Figure 19: Selecting an image region using crosshairs

2.4.2 Setting the unwrap parameters (figure 20)

In most cases, the default settings will work, and this section can be skipped. In some cases however, they should be modified. The first two options, the *Shear magnitude* and *Laser wavelength* refer to the capture parameters. This information is used to determine the correct scale for the resulting 3D plots. *Shear magnitude* refers to the amount of image shear, measured at the specimen.

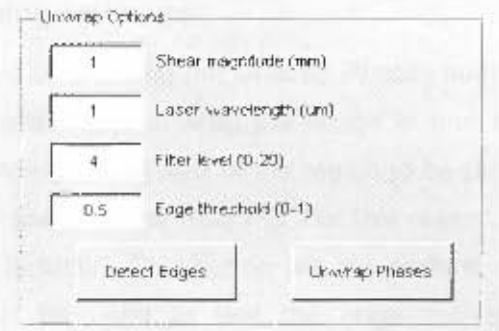


Figure 20: unwrap parameters

The next two options are unwrapping parameters; filter level refers to the amount of filtering that will be applied to the unwrapped image. Filtering is typically necessary to smooth the noise around phase fringes. The filter level is specified as an integer between 0 and 20.

Edge threshold refers to the sensitivity with which to detect phases in the phase image. The detection of phases is a critical part of unwrapping images successfully. The threshold is represented as a decimal number between 0 and 1, with 0 being very sensitive and 1 being highly insensitive. The following figures show how the *Edge threshold* effects the detection of edges. These are the edges detected for the region selected in the section entitled '*Selecting an image region to process*'.

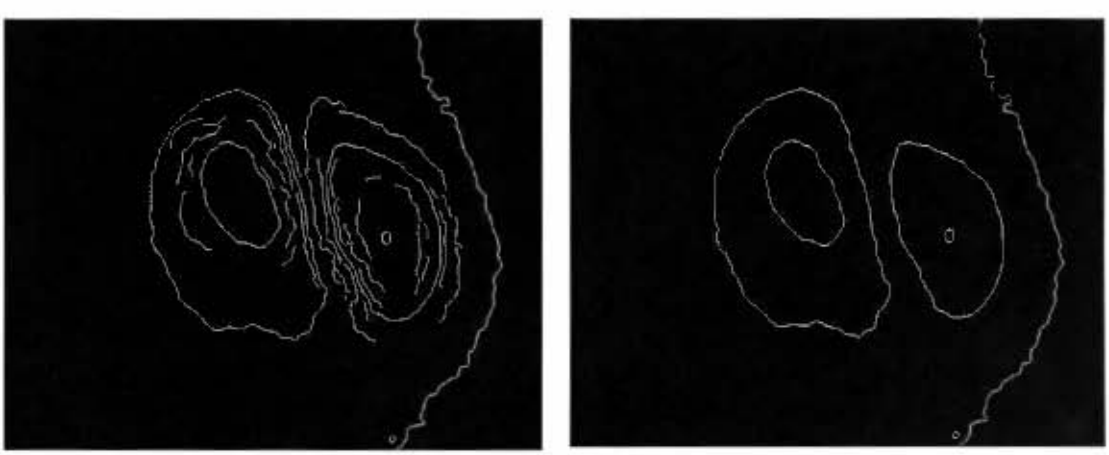


Figure 21: Edge threshold = 0.1 (left), Edge threshold = 0.5 (right)

In figure 21, the figure on the left was captured with an *Edge threshold* value that was too low (sensitive) because it has detected phase-fringes that weren't in the source image. The figure on the right matches the phase-fringes, and will give accurate results.

2.4.3 Unwrapping an image

An image is unwrapped by pressing the *Unwrap Phases* button. Pressing this button will perform edge detection and unwrap the image in one step, and may take up several minutes depending on the size of the region to be processed, the number of phase-fringes and the speed of the host PC. For this reason, another button, called *Detect Edges* is also included. This button will not perform unwrapping, only edge detection, thus it can be used to test the edge threshold parameter before unwrapping the entire image. Note that not every phase-stepped image can be successfully unwrapped; the image must have been filtered, and must have clearly defined fringes. Images that do not meet these conditions may give erroneous results. Once unwrapped, the unwrapped intensity image will be displayed:

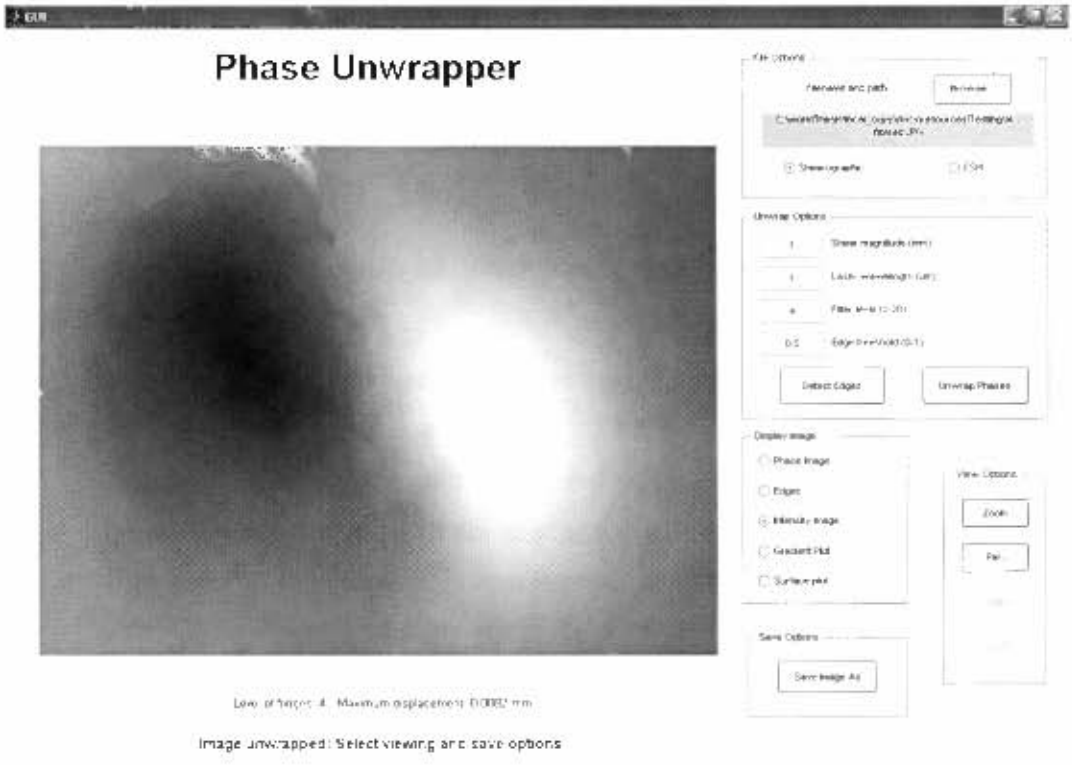


Figure 22: Unwrapped intensity image

2.4.4 Viewing and saving options

There are several options available for viewing and saving unwrapped images, as follows in figure 23:

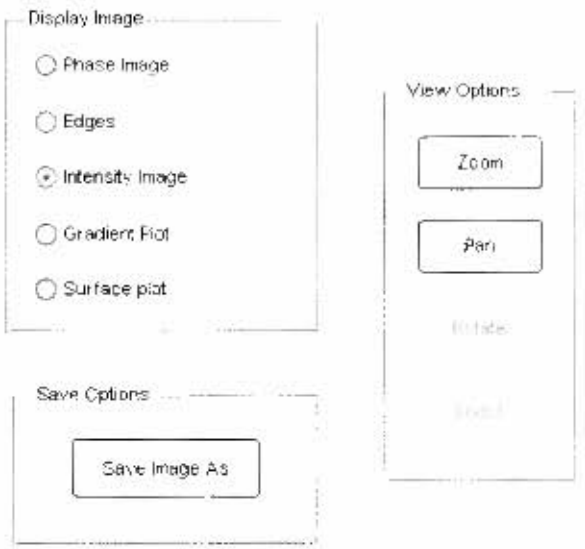


Figure 23: Image viewing options

The options under 'Display Image' all change the type of image that is displayed:

- *Phase Image* is the source image.
- *Edges* is a binary image, showing where edges were detected.
- *Intensity Image* is the unwrapping image, displayed as a 2D intensity image.
- *Gradient Plot* is a 3D plot of the *Intensity Image*. In ESPI this option will be unavailable, in shearography this is the surface's gradient profile.
- *Surface Plot* is a 3D plot of the specimen's surface displacement.

The *View options* allow the user to Pan, Zoom, Rotate or Invert the displayed images. Note that *Rotate* and *Invert* will only be available for 3D plots. The *Save Image As* button can be used to save the currently displayed image. If, after saving, the image disappears from screen, don't worry, as it can be retrieved by pressing the corresponding '*Display image*' radio button.

The following screenshot (figure 24) is an example of the 3D rendered gradient plot for a shearography image. This surface may rotated, zoomed, panned or inverted using the view options

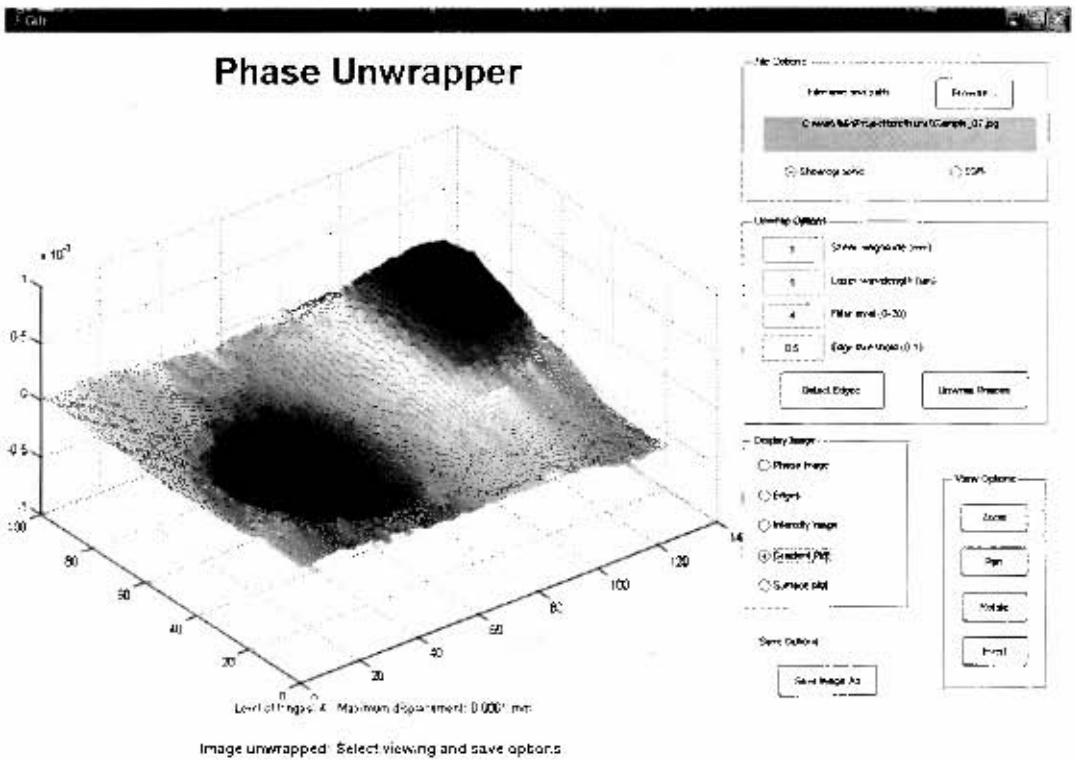


Figure 24: Rendered 3D gradient profile

2.5 How it works

As mentioned previously, *Unwrap* can act either as a plug-in to *Inspector* or as a stand-alone application, but both instances use the same *unwrap.exe* file. The behaviour for these two instances is different; the distinction is in the command-line arguments given: *unwrap.exe* expects three command-line arguments, in order.

1st argument:
Should be either 'GUI' or 'CLI' (Graphical User Interface or Command-Line Interface). This will determine whether the GUI is opened or the default unwrapping parameters are used in a faster command-line operation.

2nd argument:
Should be '0' or '1'. '0' will make the *Unwrap* assume that the input image is from shearography, whereas '1' will make it assume that the image is ESPI.

3rd argument:
The path to the image file to be unwrapped.

As an example, executing the following command-line will open the *Unwrap* in GUI mode, treating the input image, *test.jpg*, as shearography:

```
unwrap.exe  GUI  0  test.jpg
```

2.5.1 Stand-alone mode

The application will open in stand-alone mode when called without any (or incorrect) command-line arguments. This is what happens when the application is double-clicked. Because the application does not know what image to load, the '*Browse for file*' button will be visible.

2.5.2 Acting as a plug-in

When *Inspector* calls *Unwrap*, it first saves the display image as *input_shear.jpg*. It then calls *Unwrap* via the command-line, using the *system* command, as shown in the following code snippet:

```
fileName = "input_shear.jpg"; //doesn't really matter what its called
imagetype = 0; //imagetype = 0 -> shearography; imagetype = 1 -> ESPI
MbufExport(fileName, M_JPEG_LOSSY, pApp->MilColour); //save image
char buffer [50];
sprintf (buffer, "unwrap.exe GUI %d %s", imagetype, fileName);
system(buffer); //execute statement in buffer
```

This code snippet will execute the following command on the command-line:

```
unwrap.exe GUI 0 input_shear.jpg
```

The operating system will look for a file called *unwrap.exe* in the OS path. If not found (it should be found because *C:\WINDOWS* is always on the search path, and *unwrap.exe* should be installed there), a console window will flash with the message:

```
'unwrap.exe' is not recognized as an internal or external command, operable
program or batch file
```

When found, this command will cause *Unwrap* to load the image automatically and treat it as a shearography image.

2.6 Troubleshooting

2.6.1 Updating an installation

Note that when updating an installation of *unwrap*, both file *unwrap.exe* and unwrap.ctf must be replaced, as must the directory *unwrap_mcr*.

Executing the *unwrap* application for the first time results in the message in Figure 2 being displayed, namely that the CTF archive is being extracted from the *unwrap.ctf* file. The contents of this folder are extracted to a folder named *unwrap_mcr* in the same directory. The contents of this folder will then be used every subsequent time that the application is run.

If a new installation tries to reference an old *unwrap_mcr* folder, an error will occur, thus this folder should be deleted whenever an installation is updated.

In fact, this folder may always be deleted as it will simply be re-extracted the next time that the application is executed.

2.6.2 *mclmcrrtxx.dll* error message

If the following error message (figure 25) appears, it is an indication that either:

Figure 25: Error message

- The MATLAB Component Runtime is not installed.
- The incorrect version of the MCR is installed.

2.6.3 Identifying erroneous results

The user should be aware that not all images can be successfully unwrapped using the *unwrap* application, and as such should be able to identify erroneous results. The most typical problem using unwrap is that the fringes are not clearly formed, and hence the fringes are not continuous when performing the edge detection.

This error will give an output intensity image that has abrupt colour discontinuities corresponding to the positions of the fringes. This problem may, in some cases, be solved by increasing the edge detection sensitivity, so that continuous fringes are obtained. In other cases, it may be necessary to recapture the images using the *Inspector* application. The following two figures show how an image with discontinuous fringes can be successfully processed by adjusting the edge detection settings.

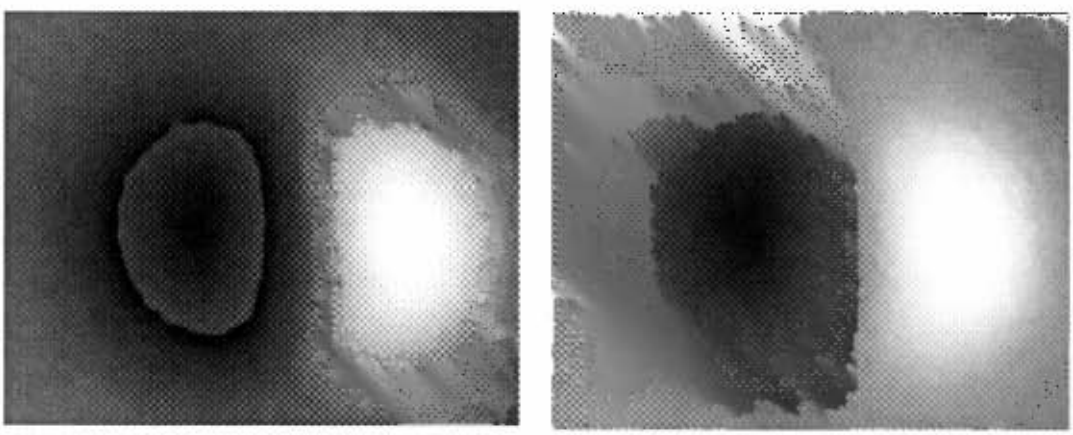
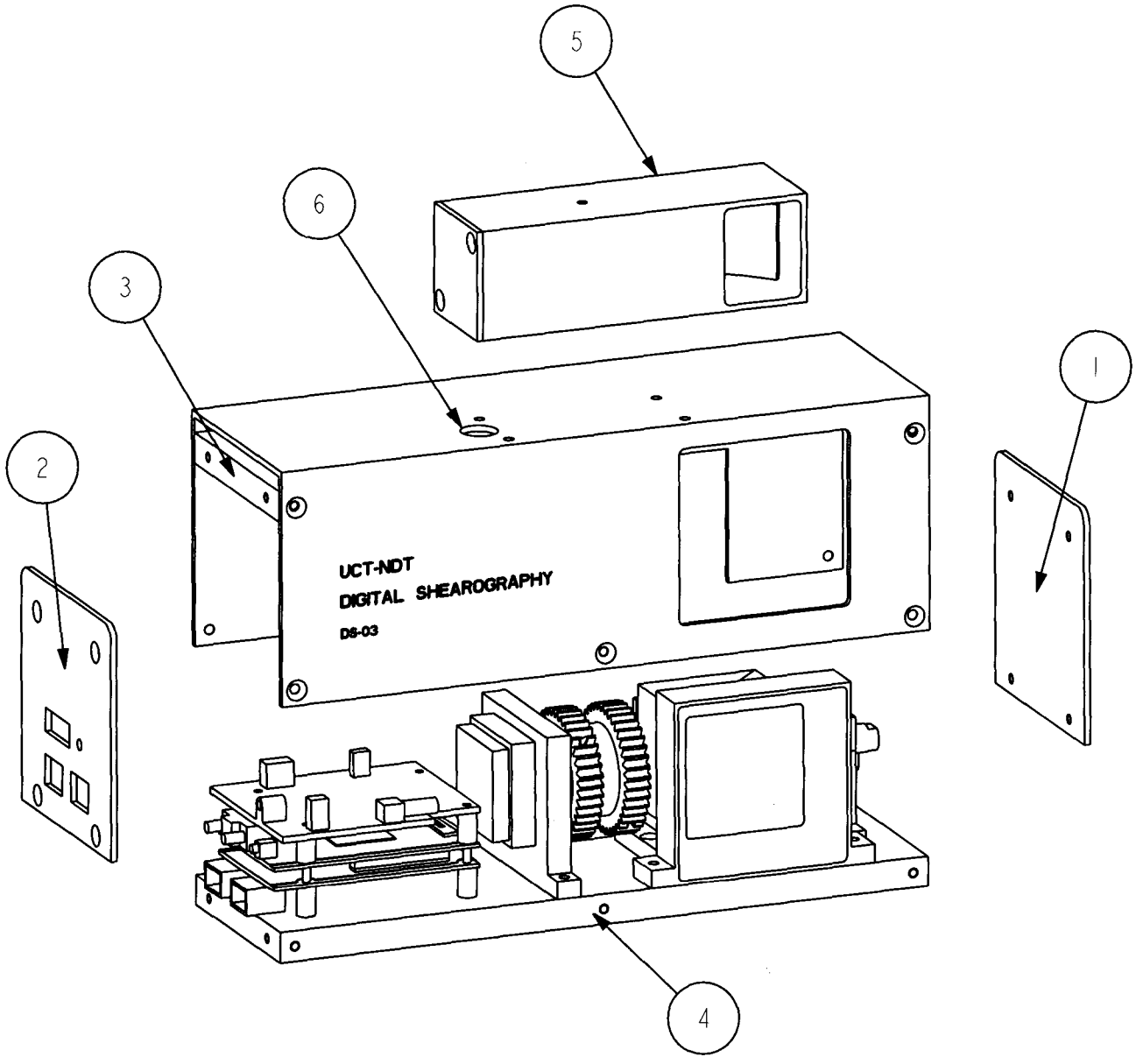


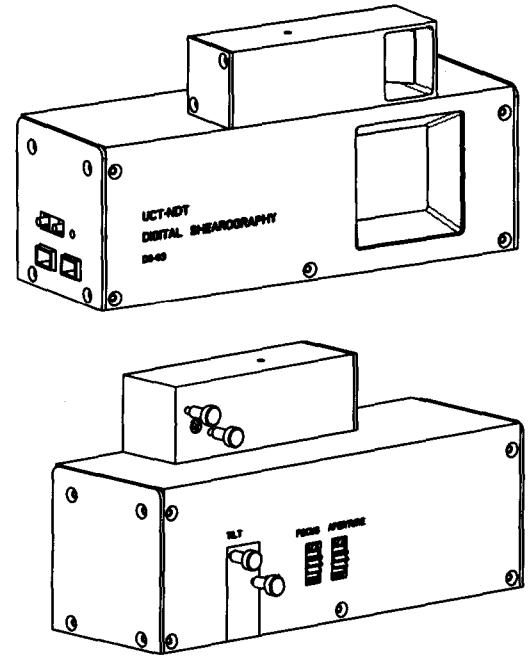
Figure 26: Erroneous result (left), correct result (right)

The figure on the left has a clear discontinuity in the left lobe, which is as a result of an incomplete fringe. Plotting this image in 3D will give a surface displacement profile that is incorrect. This image was unwrapped with an edge detection threshold of 0.6. The image on the right was unwrapped from the same source image, using an edge detection threshold of 0.5. The increased sensitivity gave a continuous intensity image, which is a sign that the image was unwrapped successfully.

APPENDIX B: MANUFACTURING DRAWINGS

No	Title	Quantity	Type
1	1. Final assembly	1	A
2	1.1. Internal assembly	1	A
3	1.1.1. Camera assembly	1	A
4	1.1.1.1. Camera head mount	1	P
5	1.1.1.2. Focus shaft housing	1	P
6	1.1.1.3. Lens focus shaft	1	P
7	1.1.1.4. Bush	2	P
8	1.1.1.5. Lens gear	2	P
9	1.1.1.6. Lens pinion	2	P
10	1.1.2. Beam splitter assembly	1	A
11	1.1.2.1. Beam collector	1	P
12	1.1.2.2. Beam splitter cube mount	1	P
13	1.1.3. Shear assembly	1	A
14	1.1.3.1. Shear mirror mount	1	P
15	1.1.3.2. Shear mirror plate	1	P
16	1.1.3.3. Shear adjustment shaft	3	P
17	1.1.4. Piezo assembly	1	A
18	1.1.4.1. Piezo mount	1	P
19	1.1.4.2. Piezo tip adaptor	1	P
20	1.1.5. Camera board mount	1	P
21	1.2. Laser assembly	1	A
22	1.2.1. Laser housing	1	P
23	1.2.2. Laser mirror wedge	1	P
24	1.2.3. Laser cover rear	1	P
25	1.3. U-Cover	1	P
26	1.4. Cover plate	2	P
27	1.5. Cover plate lock	2	P



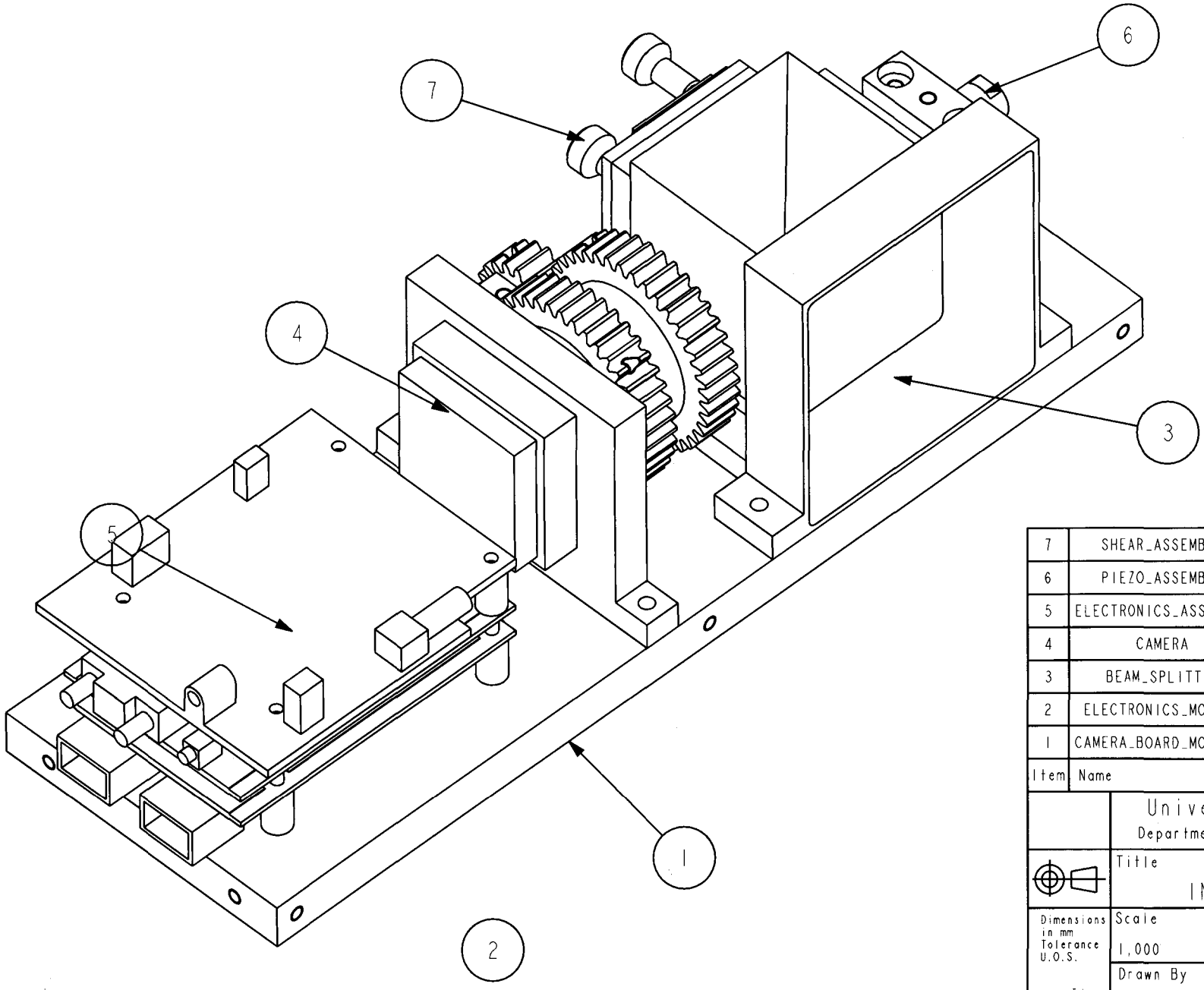


Item	Name	Qty	Material
6	U_COVER	1	ALUMINIUM
5	LASER_ASSEMBLY	1	ASSEMBLY
4	INTERNAL_ASSEMBLY	1	ASSEMBLY
3	COVER_PLATE_LOCK	2	ALUMINIUM
2	COVER_PLATE_CUTOUT	1	ALUMINIUM
1	COVER_PLATE	1	ALUMINIUM

University of Cape Town
Department of Mechanical Engineering

Title
FINAL ASSEMBLY

Dimensions in mm Tolerance U.O.S.	Scale	Date	Sheet	of
	0,500	27 MAY 2008	1	27
	Drawn By		Drawing Number	
0.1	B PITMAN		1	

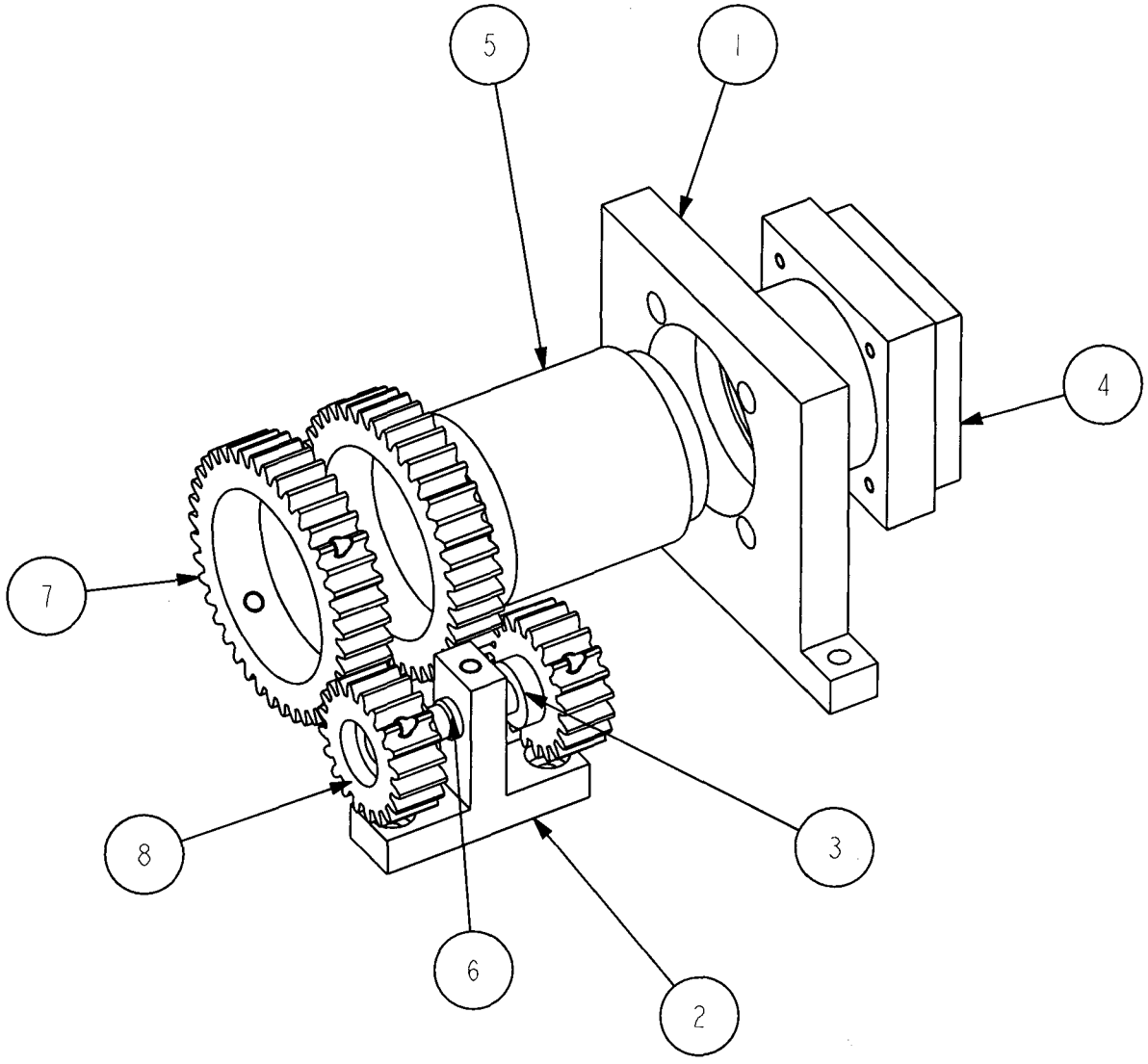


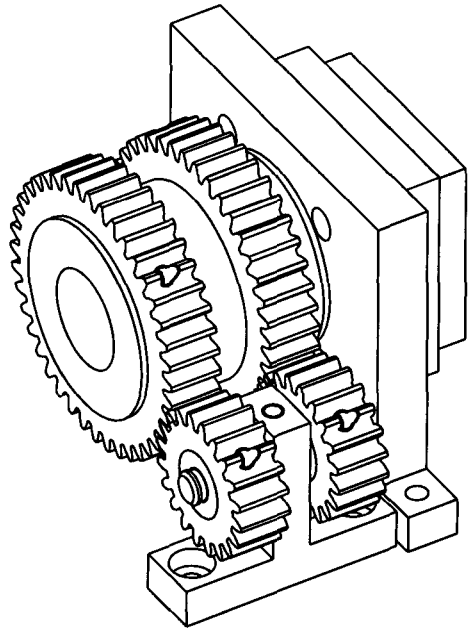
Item	Name	Qty	Material
7	SHEAR_ASSEMBLY	1	ASSEMBLY
6	PIEZO_ASSEMBLY	1	ASSEMBLY
5	ELECTRONICS_ASSEMBLY	1	ASSEMBLY
4	CAMERA	1	ASSEMBLY
3	BEAM_SPLITTER	1	ASSEMBLY
2	ELECTRONICS_MOUNTS	4	NO_MATERIAL
1	CAMERA_BOARD_MOUNT_2	1	ALUMINIUM

University of Cape Town
Department of Mechanical Engineering

Title
INTERNAL ASSEMBLY

Dimensions in mm Tolerance U.O.S.	Scale	Date	Sheet	of
0.1	1,000	27 MAY 2008	2	27
	Drawn By: B PITMAN		Drawing Number: 1.1	



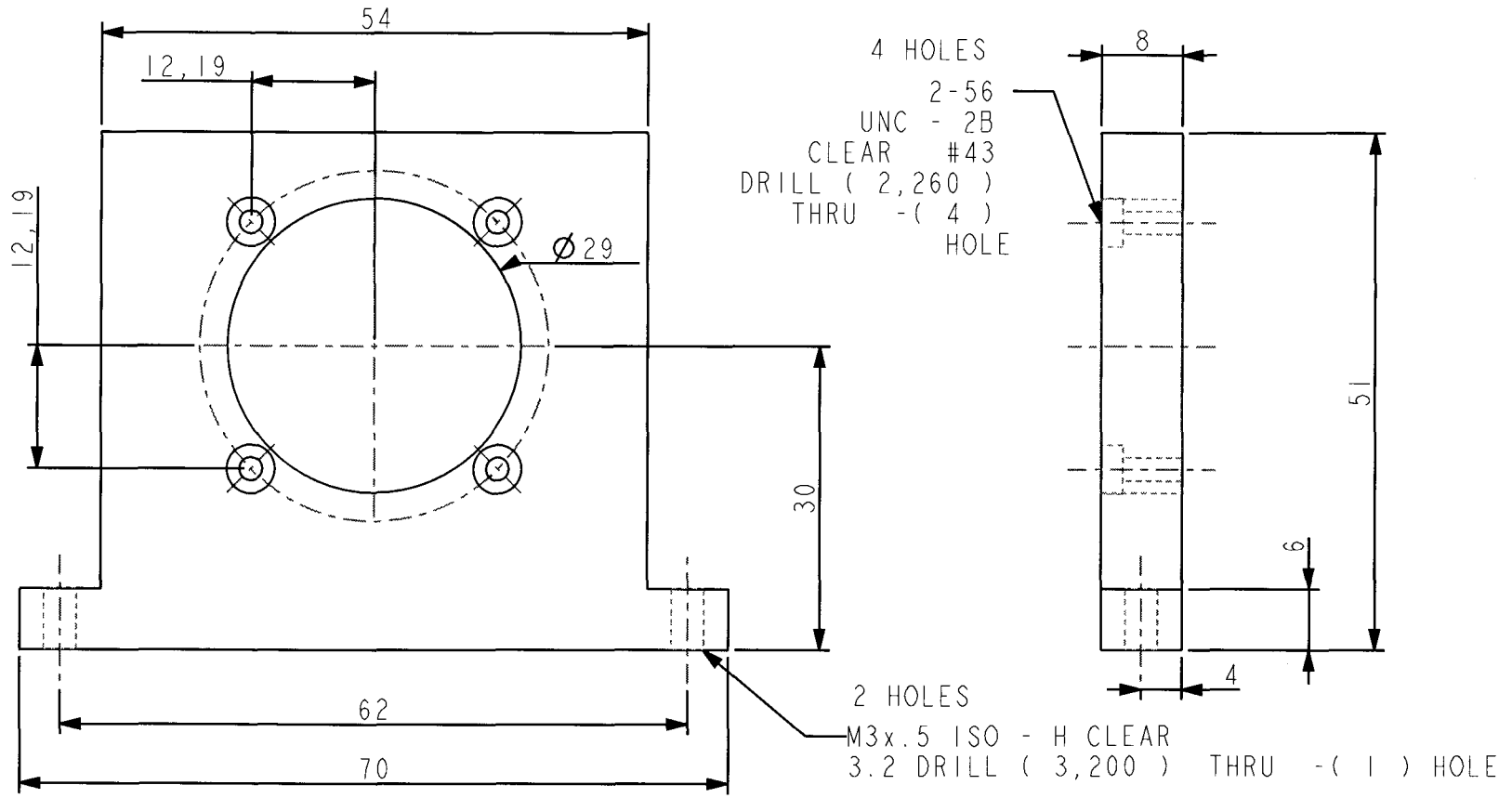


Item	Name	Qty	Material
8	LENS_PINION	2	OEM (MODIFIED)
7	LENS_GEAR	2	OEM (MODIFIED)
6	LENS_FOCUS_SHAFT	1	STEEL_LC
5	LENS	1	NO_MATERIAL
4	CAMERA_HEAD	1	NO_MATERIAL
3	BUSH	2	BRASS
2	FOCUS_SHAFT_HOUSING	1	ALUMINIUM
1	CAMERA_HEAD_MOUNT_2	1	ALUMINIUM

University of Cape Town
Department of Mechanical Engineering

Title
CAMERA ASSEMBLY

Dimensions in mm Tolerance U.O.S.	Scale	Date	Sheet	of
0.1	1,000	27 MAY 2008	3	27
	Drawn By: B PITMAN		Drawing Number: 1.1.1	

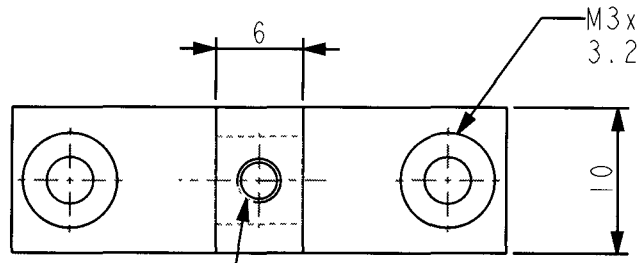


	ALUMINIUM	1	10MM SHEET
Item	Material	Qty	Remarks

University of Cape Town
Department of Mechanical Engineering

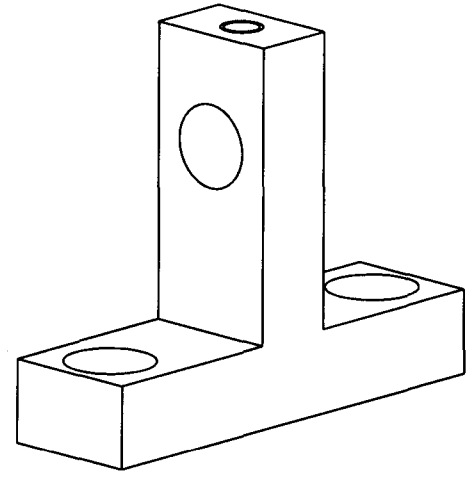
Title
CAMERA HEAD MOUNT 2

Dimensions in mm Tolerance U.O.S.	Scale	Date	Sheet	of
0.1	1,500	27 MAY 2008	4	27
	Drawn By		Drawing Number	
	B PITMAN		1.1.1.1	

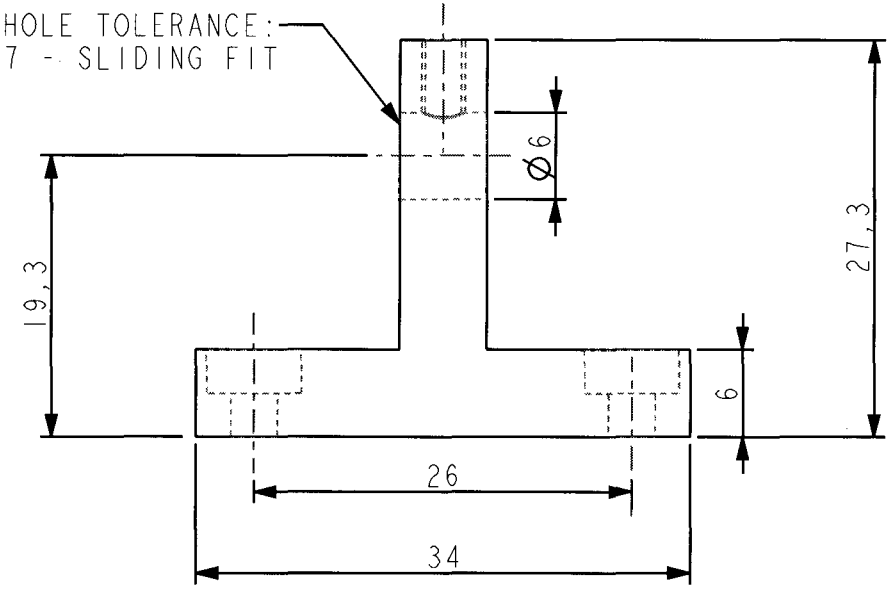


M3x.5 ISO - H CLEAR
3.2 DRILL (3,200) THRU -(1) HOLE

M3x.5
ISO - H TAP
↧ 6,000 2.5 DRILL
(2,500) ↧ 8,000
-(1) HOLE



HOLE TOLERANCE:
H7 - SLIDING FIT

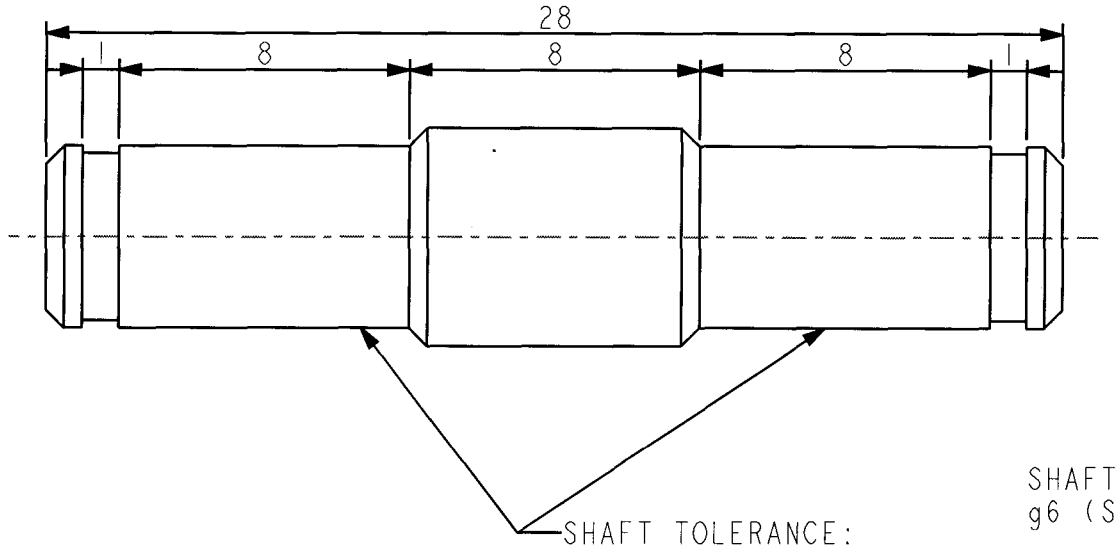


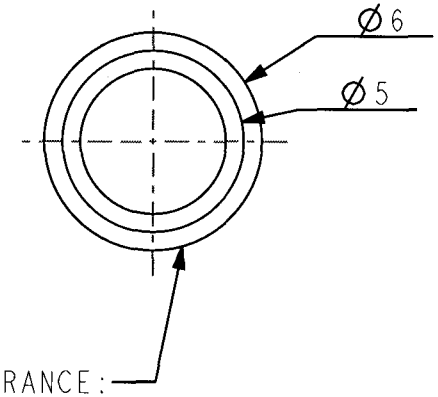
	ALUMINIUM	1	10MM SHEET
Item	Material	Qty	Remarks

University of Cape Town
Department of Mechanical Engineering

Title
FOCUS SHAFT HOUSING

Dimensions in mm Tolerance U.O.S.	Scale	Date	Sheet	of
0.1	2,000	27 MAY 2008	5	27
	Drawn By		Drawing Number	
	B PITMAN		1.1.1.2	





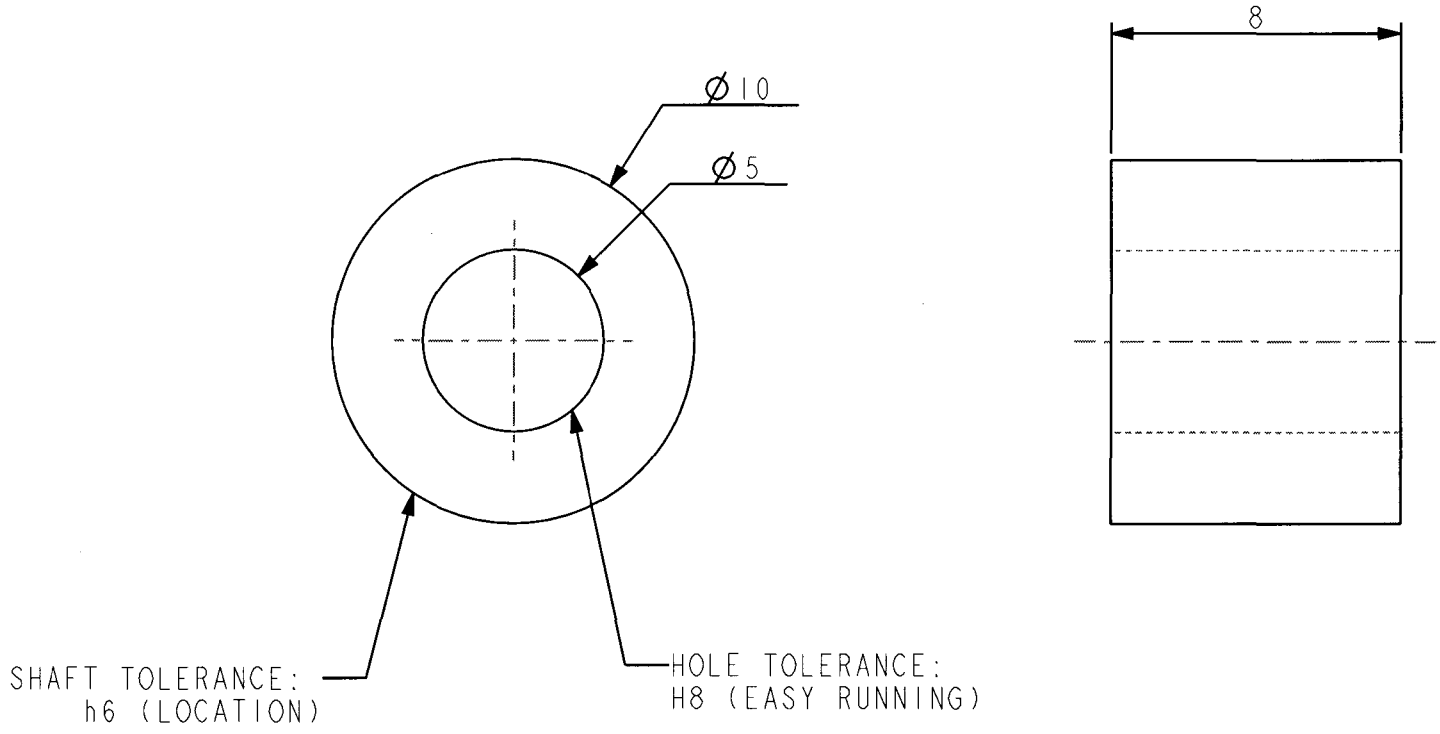
NOTE:
ALL CHAMFERS ARE
0.5mm * 0.5mm

	AXLE/SILVER STEEL	1	TURNED
Item	Material	Qty	Remarks

University of Cape Town
Department of Mechanical Engineering

Title
LENS FOCUS SHAFT

Dimensions in mm Tolerance U.O.S.	Scale	Date	Sheet	of
	5,000	27 MAY 2008	6	27
	Drawn By		Drawing Number	
0:1	B PITMAN		1.1.1.3	

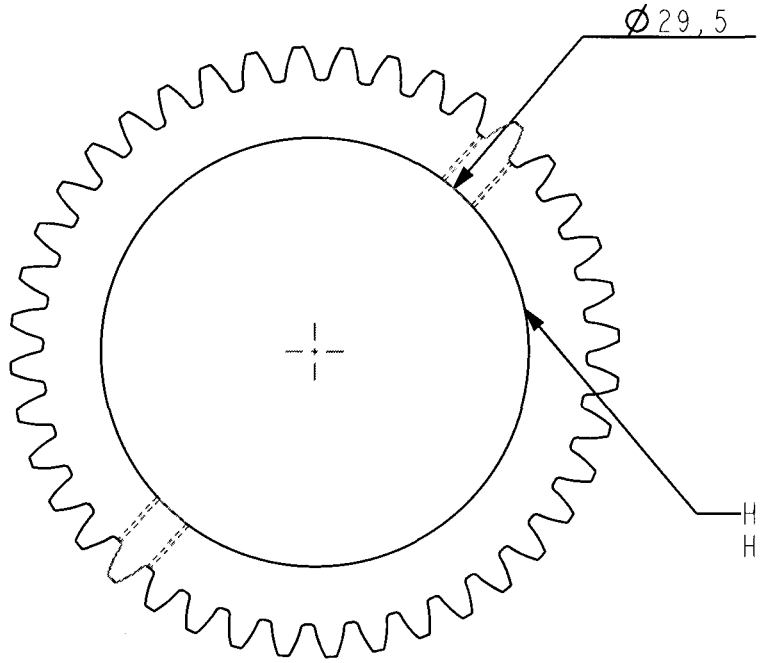


	BRASS	2	TURNED (OR OEM)
Item	Material	Qty	Remarks

University of Cape Town
Department of Mechanical Engineering

Title: BUSH

Dimensions in mm Tolerance U.O.S.	Scale	Date	Sheet	of
	5,000	27 MAY 2008	7	27
	Drawn By		Drawing Number	
0.1	B PITMAN		1.1.1.4	



2 GRUB SCREWS
M3x.5 ISO - H TAP ↧ 7,000
2.5 DRILL (2,500) ↧ 10,000 -(2) HOLE

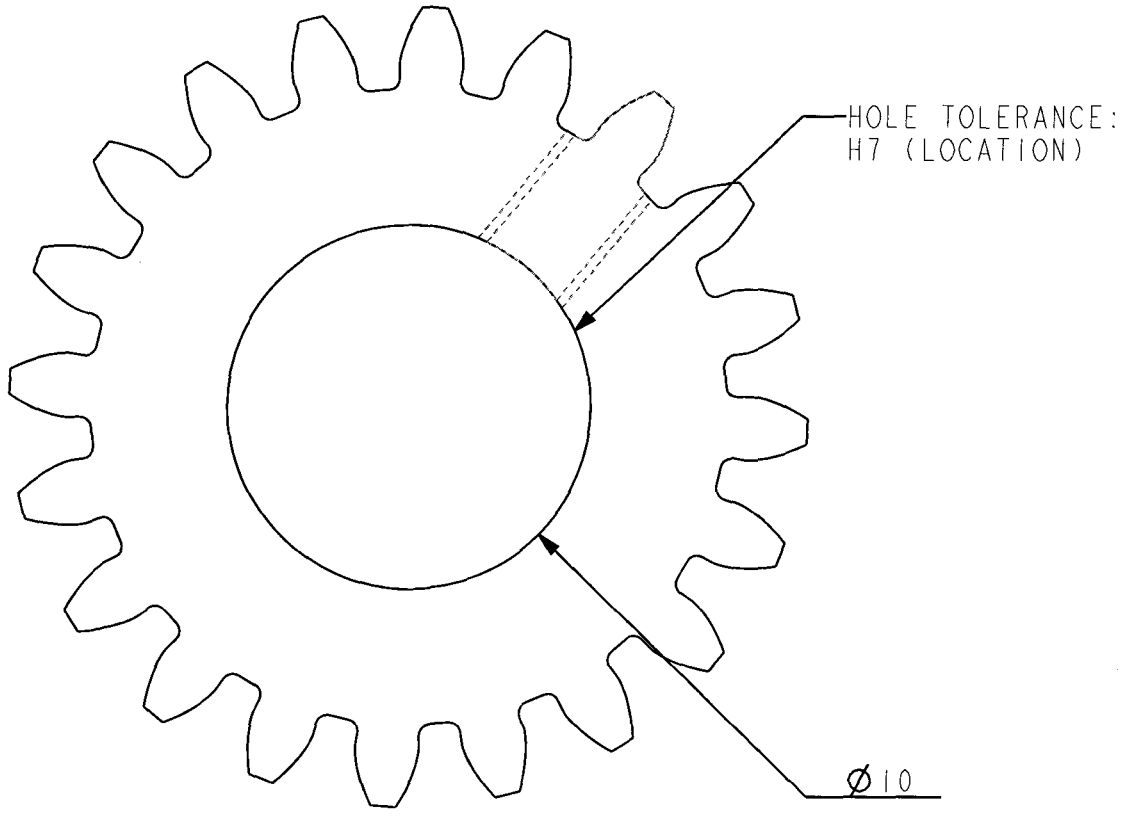
TO DO:
MILL to 8mm THICKNESS
REBORE TO 29.5mm INTERNAL DIAMETER
DRILL AND TAP HOLE FOR SETSCREWS

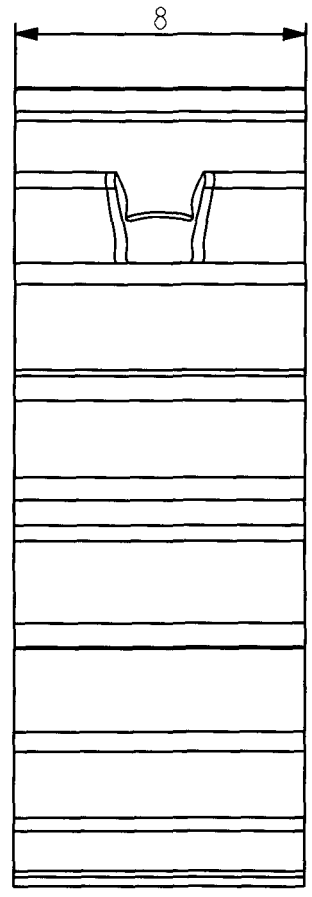
	STEEL	2	OEM PART
Item	Material	Qty	Remarks

University of Cape Town
Department of Mechanical Engineering

Title: LENS GEAR

Dimensions in mm Tolerance U.O.S.	Scale	Date	Sheet	of
	2,000	27 MAY 2008	16	27
	Drawn By		Drawing Number	
±0.1	B PITMAN		1.1.1.5	





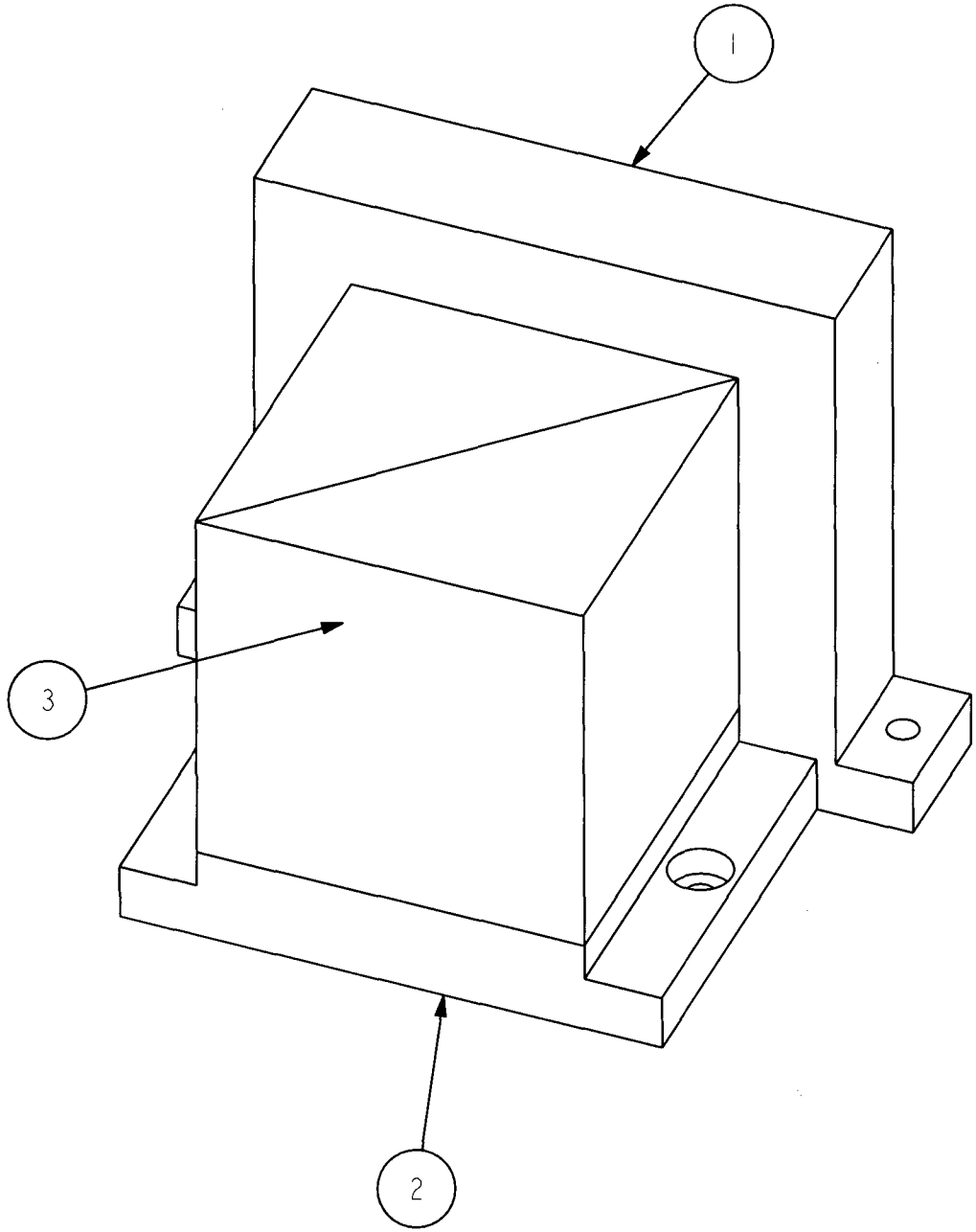
TO DO:
MILL TO 8MM THICKNESS
REBORE TO 10MM INTERNAL DIAMETER
DRILL AND TAP HOLE FOR SETSCREW

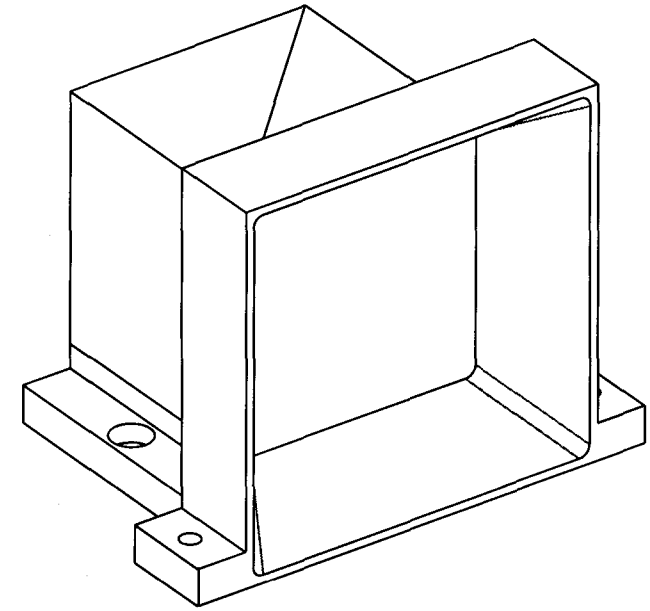
	STEEL	2	OEM PART
Item	Material	Qty	Remarks

University of Cape Town
Department of Mechanical Engineering

Title
LENS PINION

Dimensions in mm Tolerance U.O.S.	Scale	Date	Sheet	of
	5,000	27 MAY 2008	17	27
	Drawn By		Drawing Number	
-0.1	B PITMAN		1.1.1.6	





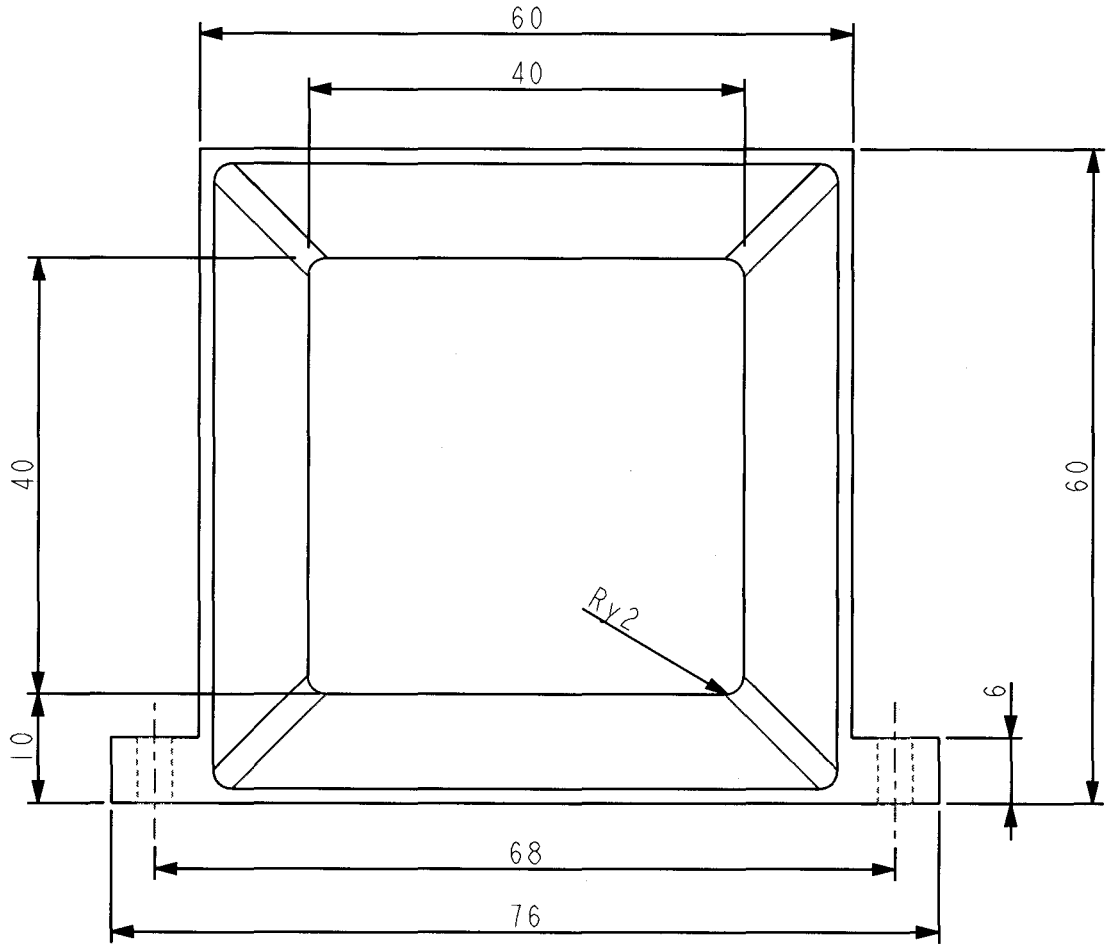
SCALE 1,000

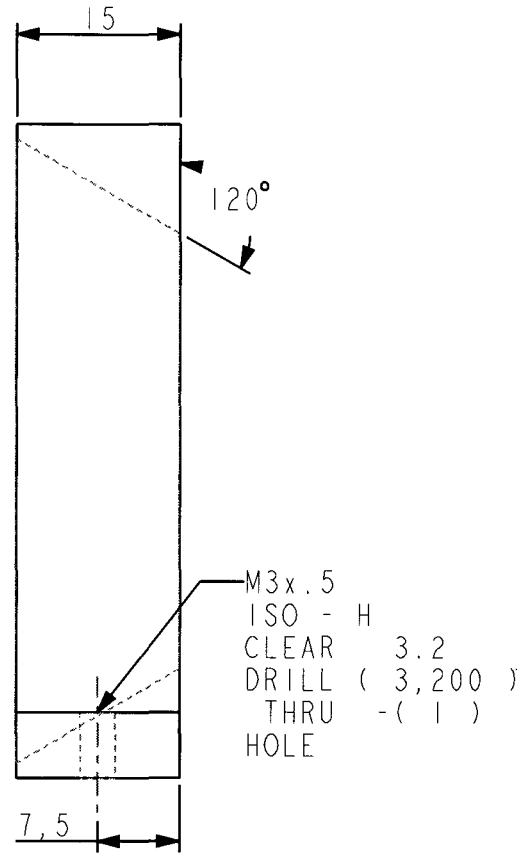
Item	Name	Qty	Material
3	BEAM_SPLITTER_CUBE	2	ALUMINIUM
2	BEAM_SPLITTER_CUBE_MOUNT	1	OEM PART
1	BEAM_COLLECTOR	1	ALUMINIUM

University of Cape Town
Department of Mechanical Engineering

Title
BEAM SPLITTER ASSEMBLY

Dimensions in mm Tolerance U.O.S.	Scale	Date	Sheet	of
0.1	1,500	27 MAY 2008	10	27
	Drawn By: B PITMAN		Drawing Number: 1.1.2	



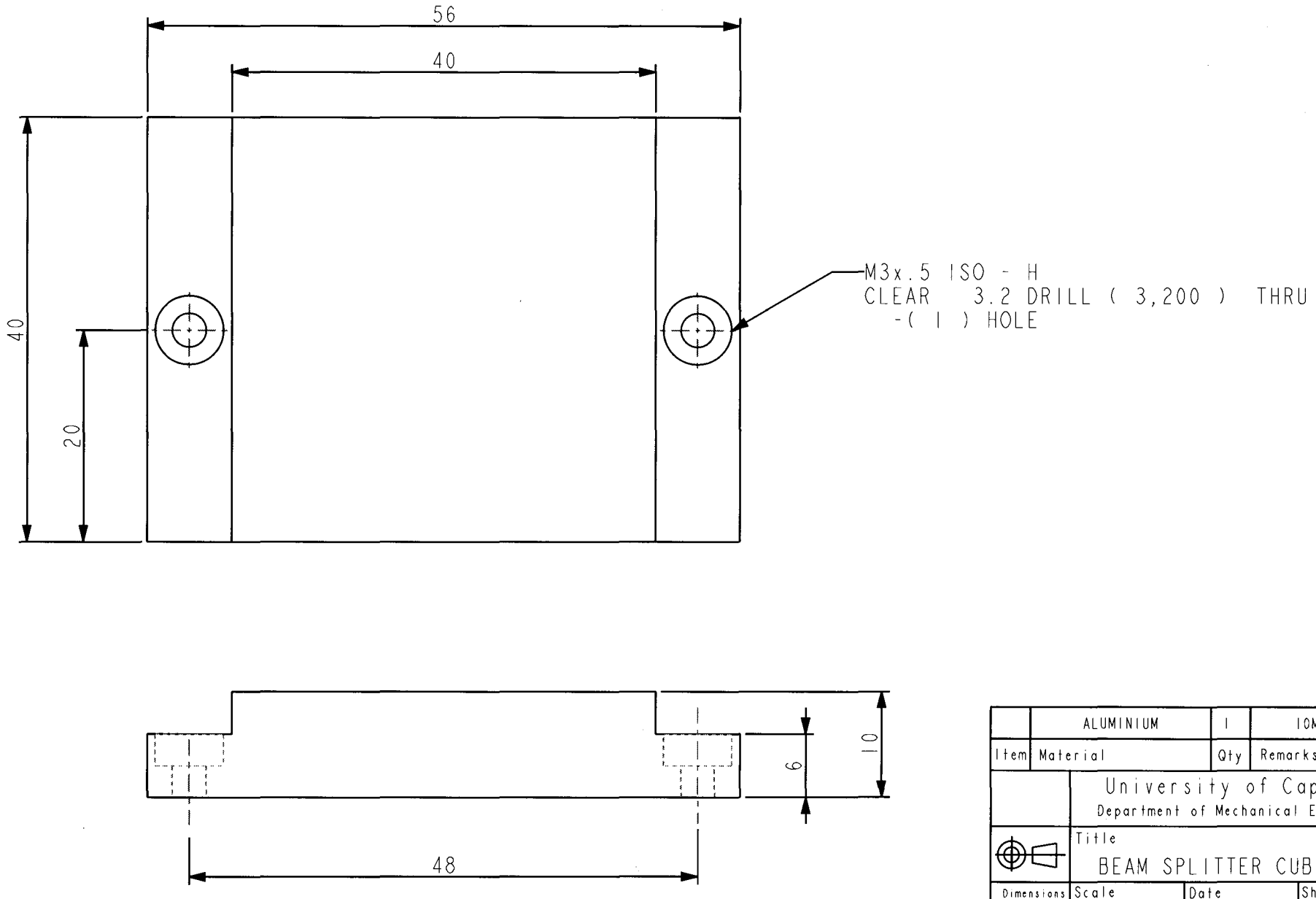


	ALUMINIUM	1	
Item	Material	Qty	Remarks

University of Cape Town
Department of Mechanical Engineering

Title
BEAM COLLECTOR

Dimensions in mm Tolerance U.O.S.	Scale	Date	Sheet	of
	1,500	27 MAY 2008	11	27
	Drawn By		Drawing Number	
	B PITMAN		1.1.2.1	

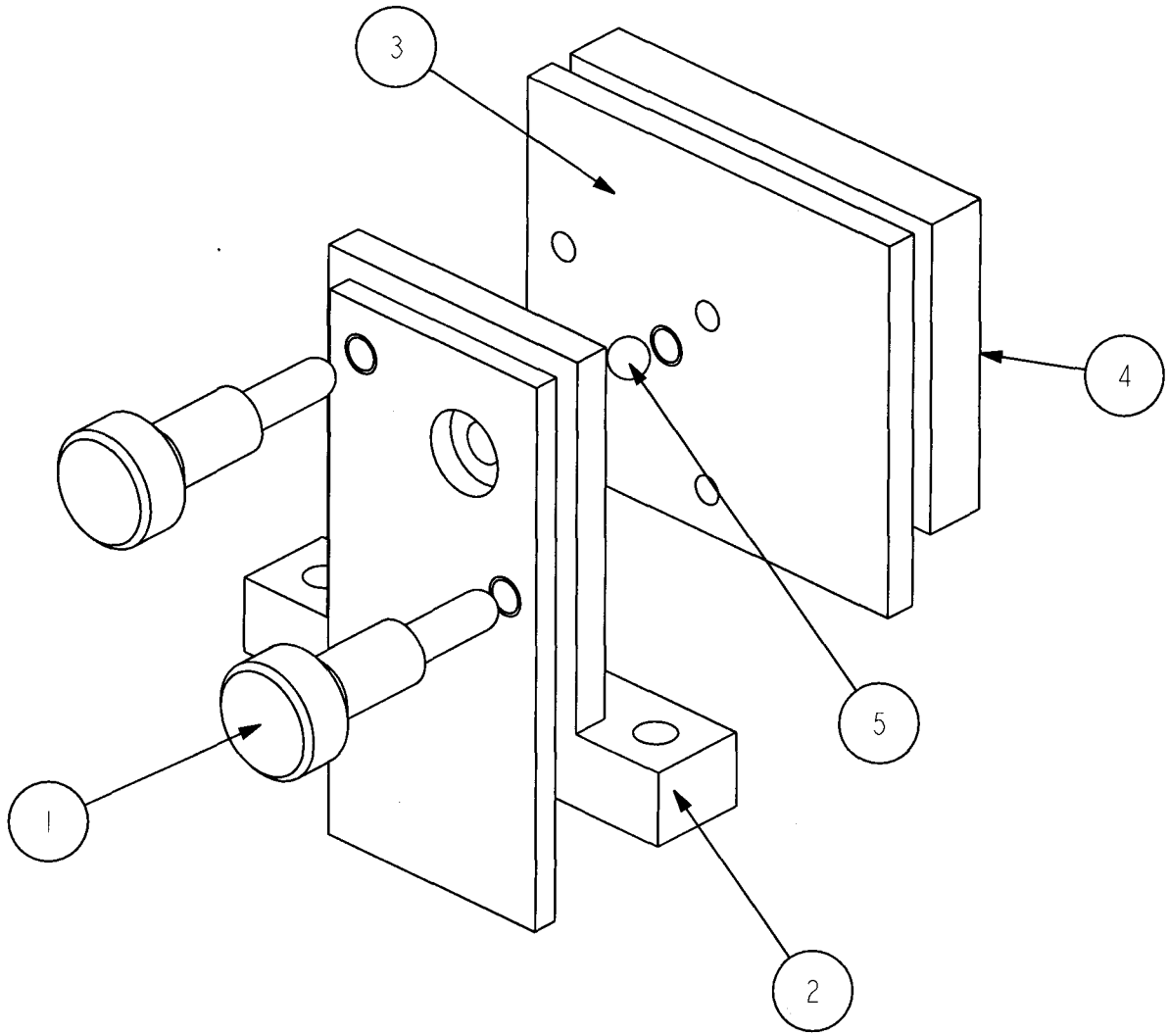


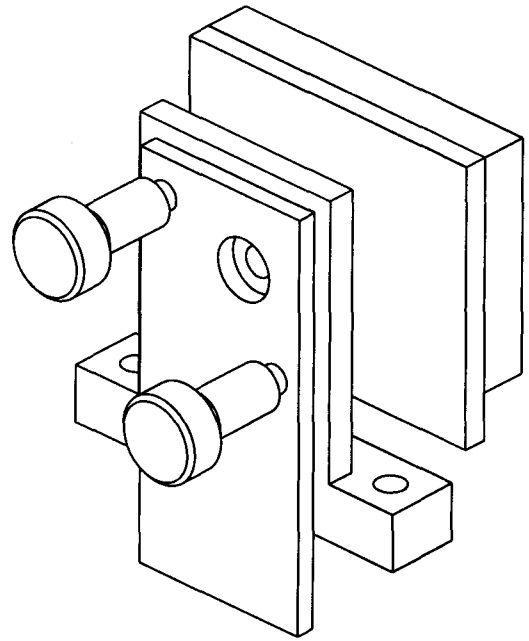
	ALUMINIUM	1	10MM SHEET
Item	Material	Qty	Remarks

University of Cape Town
Department of Mechanical Engineering

Title
BEAM SPLITTER CUBE MOUNT

Dimensions in mm Tolerance U.O.S.	Scale	Date	Sheet	of
	2,000	27 MAY 2008	12	27
	Drawn By		Drawing Number	
0.1	B PITMAN		1.1.2.2	





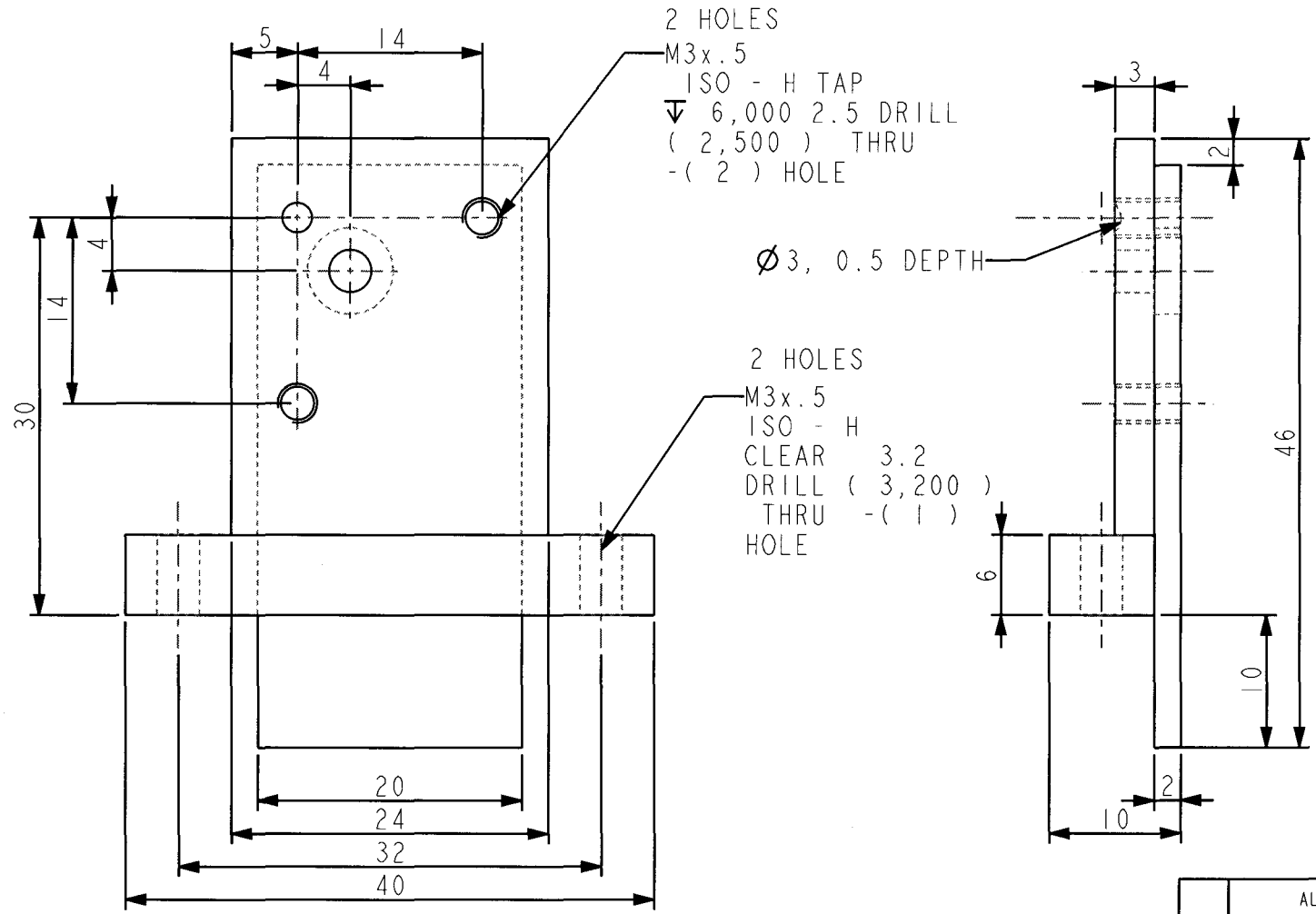
SCALE 1,500

Item	Name	Qty	Material
5	SPHERE	1	STEEL_LC
4	MIRROR	1	NO_MATERIAL
3	SHEAR_MIRROR_PLATE	1	ALUMINIUM
2	SHEAR_MIRROR_MOUNT	1	ALUMINIUM
1	SHEAR_ADJUSTMENT_SHAFT	2	ALUMINIUM

University of Cape Town
Department of Mechanical Engineering

Title
SHEAR ASSEMBLY

Dimensions in mm Tolerance U.O.S.	Scale	Date	Sheet	of
0.1	2,000	27 MAY 2008	13	27
	Drawn By: B PITMAN		Drawing Number: 1.1.3	



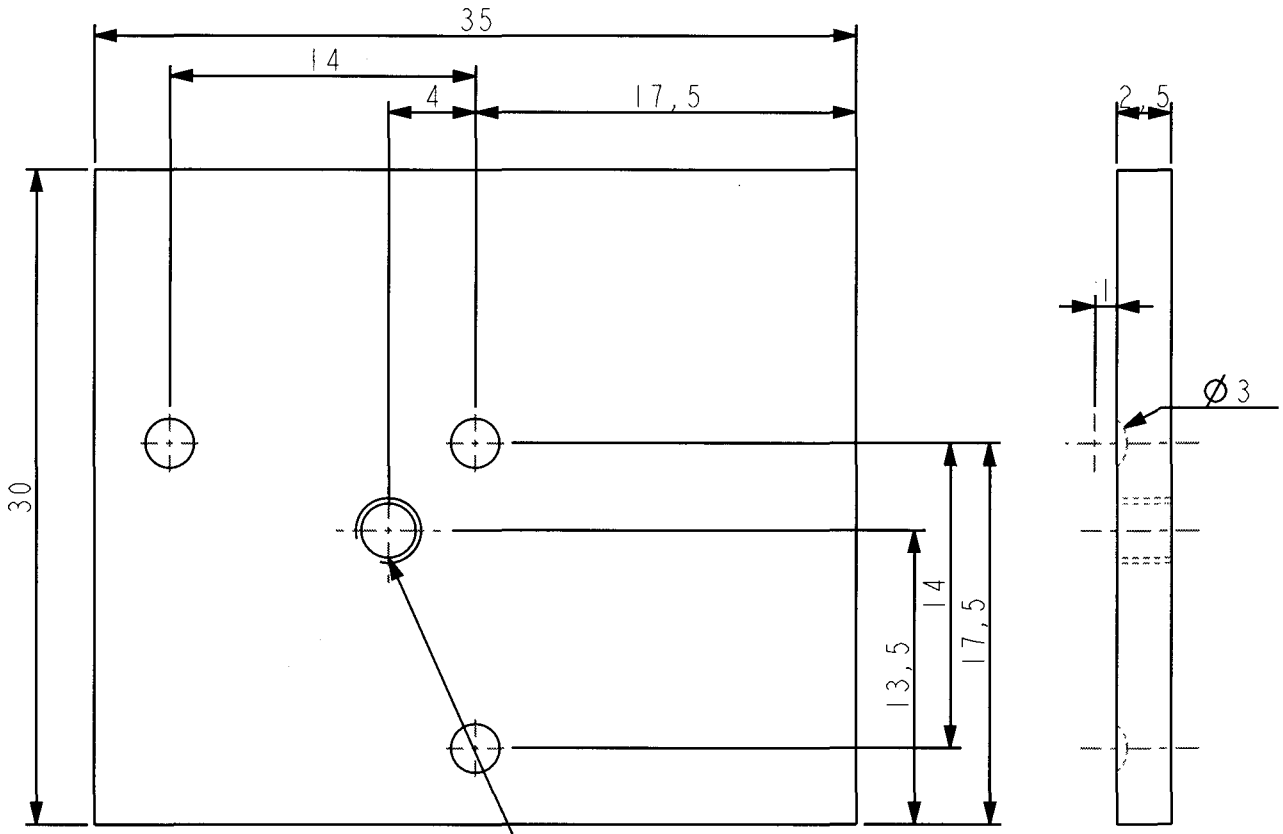
	ALUMINIUM	1	10MM SHEET
Item	Material	Qty	Remarks

University of Cape Town
Department of Mechanical Engineering

Title
SHEAR MIRROR MOUNT

Dimensions in mm Tolerance U.O.S.	Scale	Date	Sheet	of
0.1	2,000	27 MAY 2008	14	27
	Drawn By: B PITMAN		Drawing Number: 1.1.3.1	

NOTE:
SPHERICAL HOLES ARE USED FOR
LOCATING A 3MM BALL BEARING
ON WHICH MIRROR PLATE PIVOTS

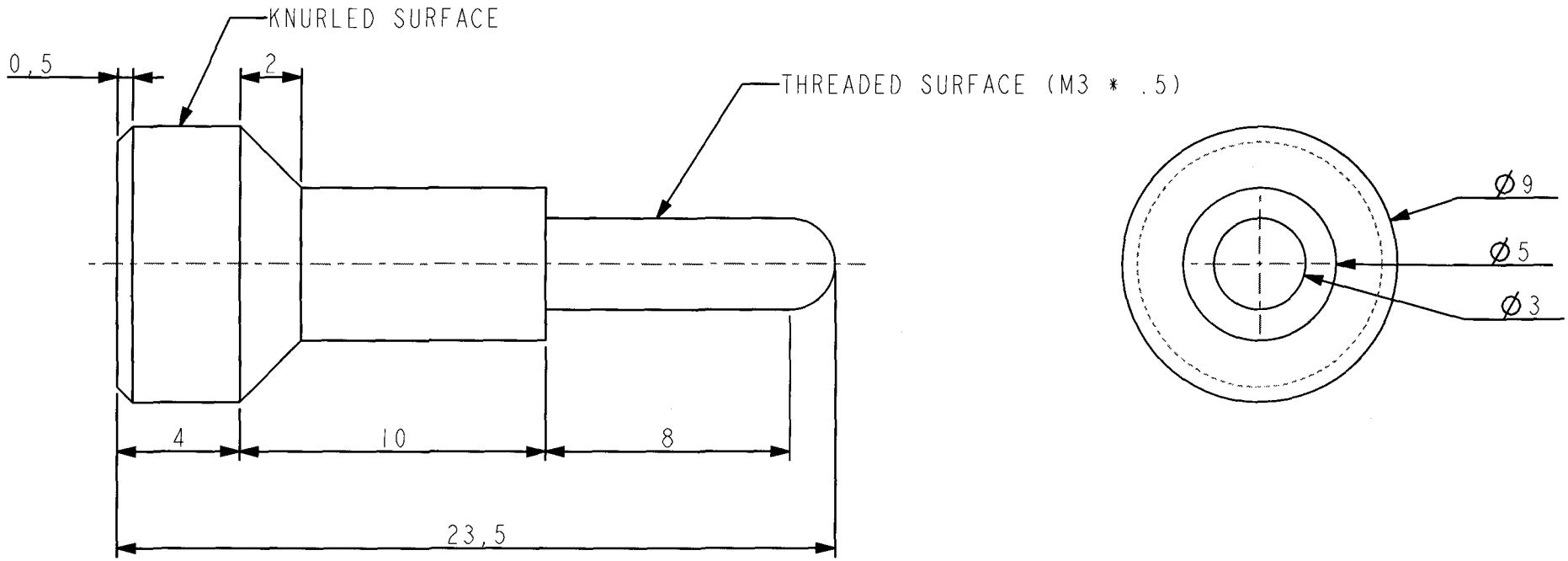


	ALUMINIUM	1	10MM SHEET
Item	Material	Qty	Remarks

University of Cape Town
Department of Mechanical Engineering

Title
SHEAR MIRROR PLATE

Dimensions in mm Tolerance U.O.S.	Scale	Date	Sheet	of
±0.1	3,000	27 MAY 2008	15	26
	Drawn By		Drawing Number	
	B PITMAN		1.1.3.2	



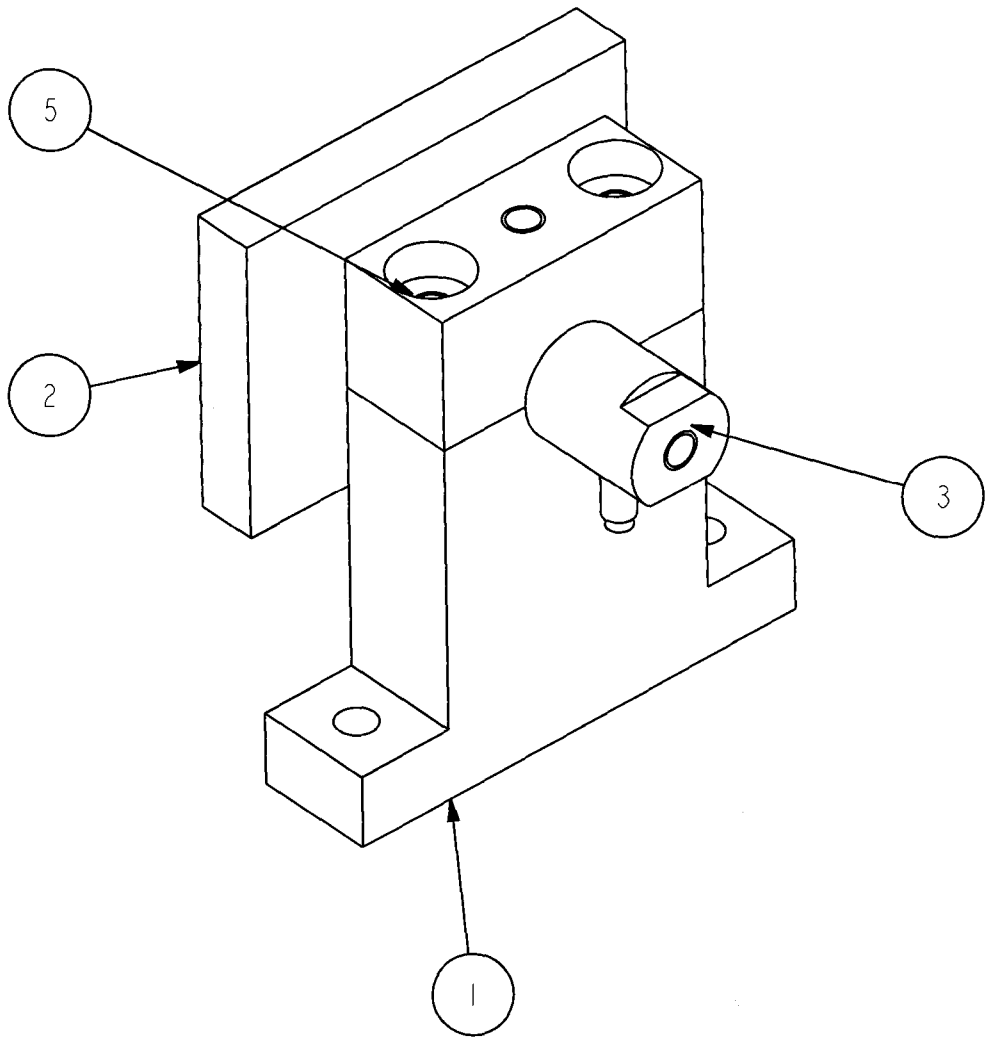
NOTE:
ALL CHAMFERS ARE 45°

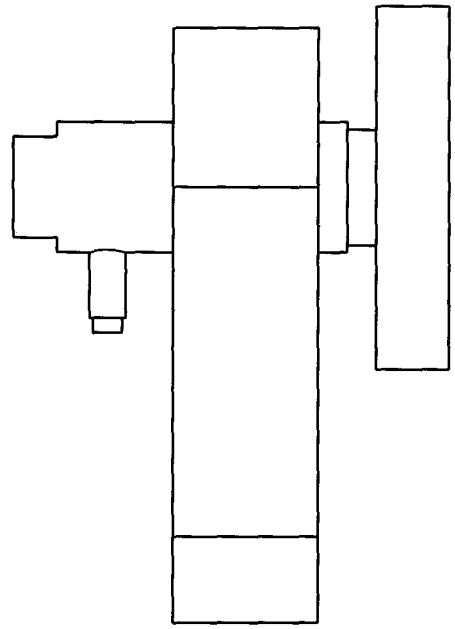
	ALUMINIUM	4	TURNED
Item	Material	Qty	Remarks

University of Cape Town
Department of Mechanical Engineering

Title
SHEAR ADJUSTMENT SHAFT

Dimensions in mm Tolerance U.O.S.	Scale	Date	Sheet	of
	5,000	27 MAY 2008	16	27
	Drawn By		Drawing Number	
0.1	B PITMAN		1.1.3.3	



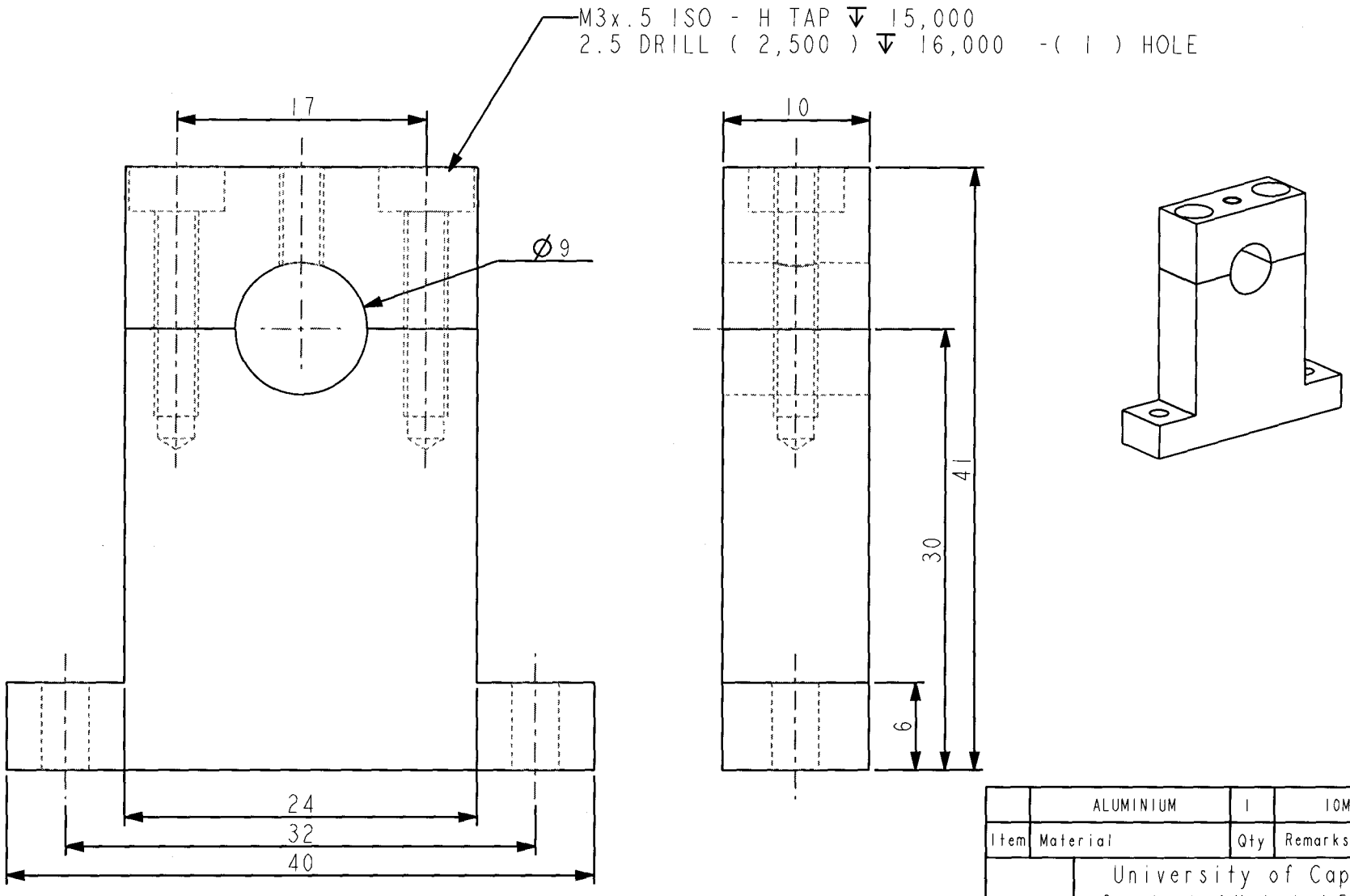


Item	Name	Qty	Material
5	PIEZO_TIP_ADAPTOR	1	STEEL_LC
4	PIEZO_MOUNT_TOP	1	STEEL_LC
3	PIEZO_MIRROR_SHIFTER_2	1	NO_MATERIAL
2	MIRROR	1	NO_MATERIAL
1	PIEZO_MOUNT	1	ALUMINIUM

University of Cape Town
Department of Mechanical Engineering

Title
PIEZO ASSEMBLY

Dimensions in mm Tolerance U.O.S.	Scale	Date	Sheet	of
0.1	2,000	27 MAY 2008	17	27
	Drawn By: B PITMAN		Drawing Number: 1.1.4	

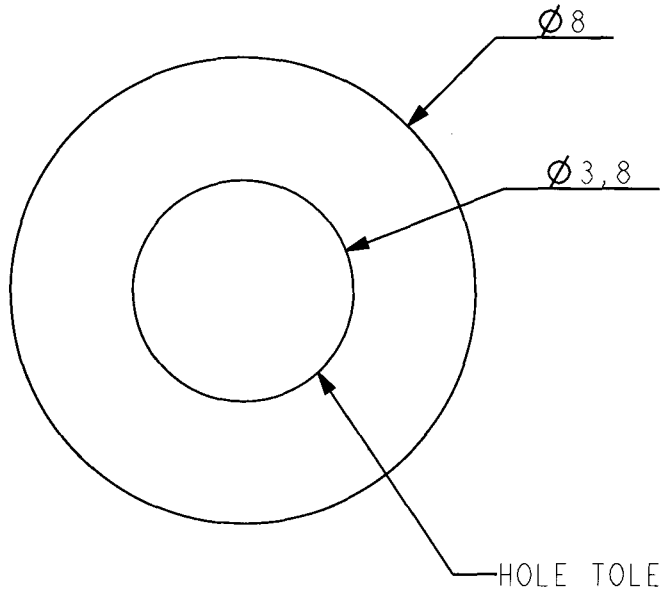


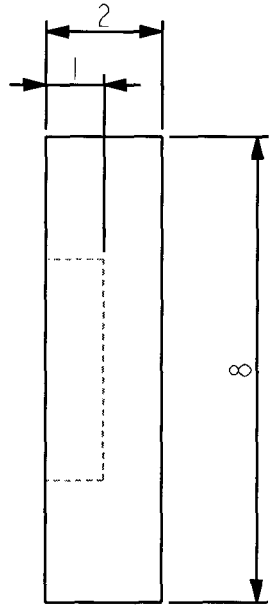
	ALUMINIUM	1	10MM SHEET
Item	Material	Qty	Remarks

University of Cape Town
Department of Mechanical Engineering

Title: PIEZO MOUNT

Dimensions in mm Tolerance U.O.S.	Scale	Date	Sheet	of
	1,000	27 MAY 2008	18	27
-0.1	Drawn By: B PITMAN		Drawing Number:	1.1.4.18



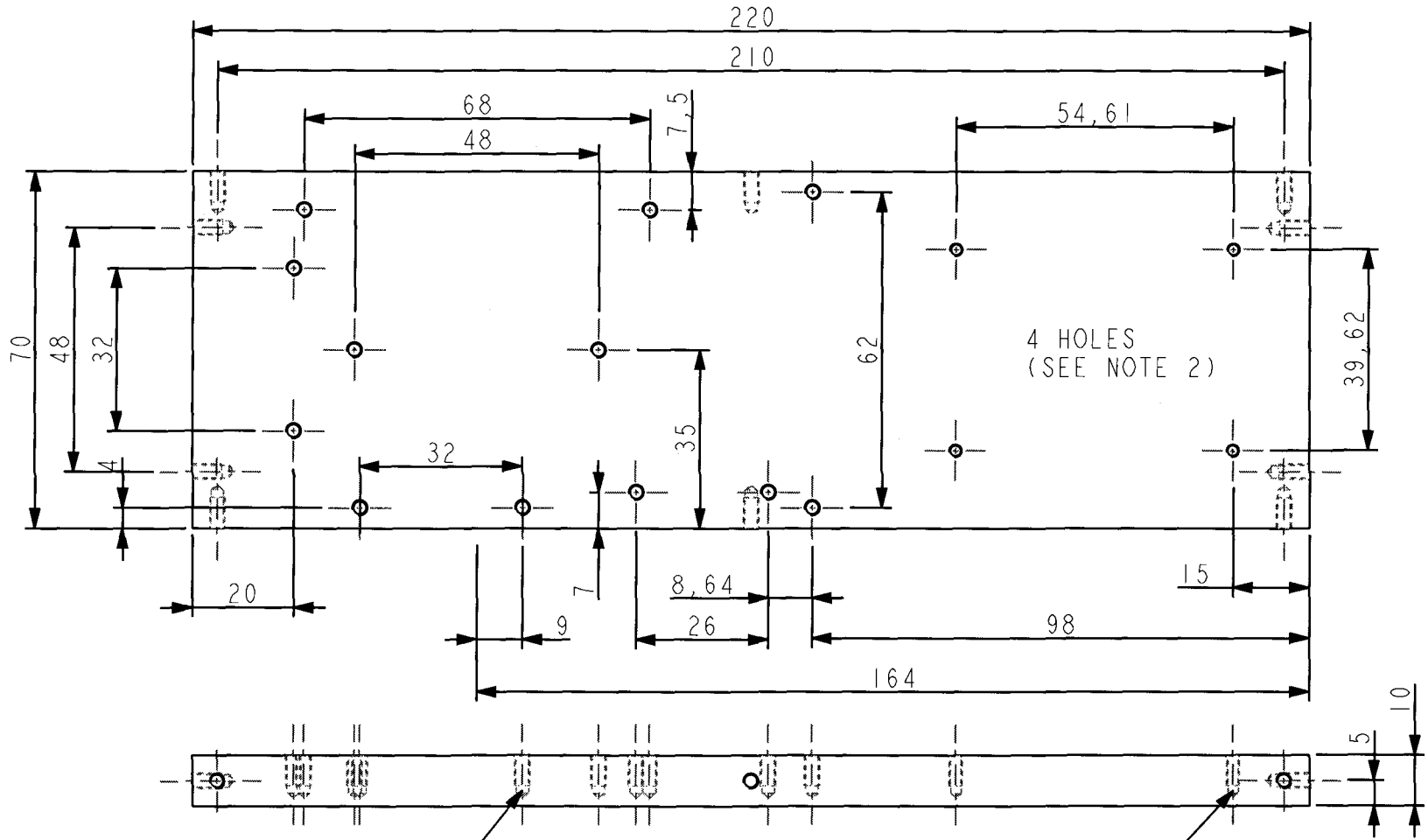


	STEEL	1	MAGNETIC
Item	Material	Qty	Remarks

University of Cape Town
Department of Mechanical Engineering

Title
PIEZO TIP ADAPTOR

Dimensions in mm Tolerance U.O.S.	Scale	Date	Sheet	of
	8,000	27 MAY 2008	19	27
	Drawn By		Drawing Number	
0.1	B PITMAN		1.1.4.2	



NOTE 1:
ALL HOLES ARE M3 BY .5 (AS ABOVE)
UNLESS OTHERWISE INDICATED

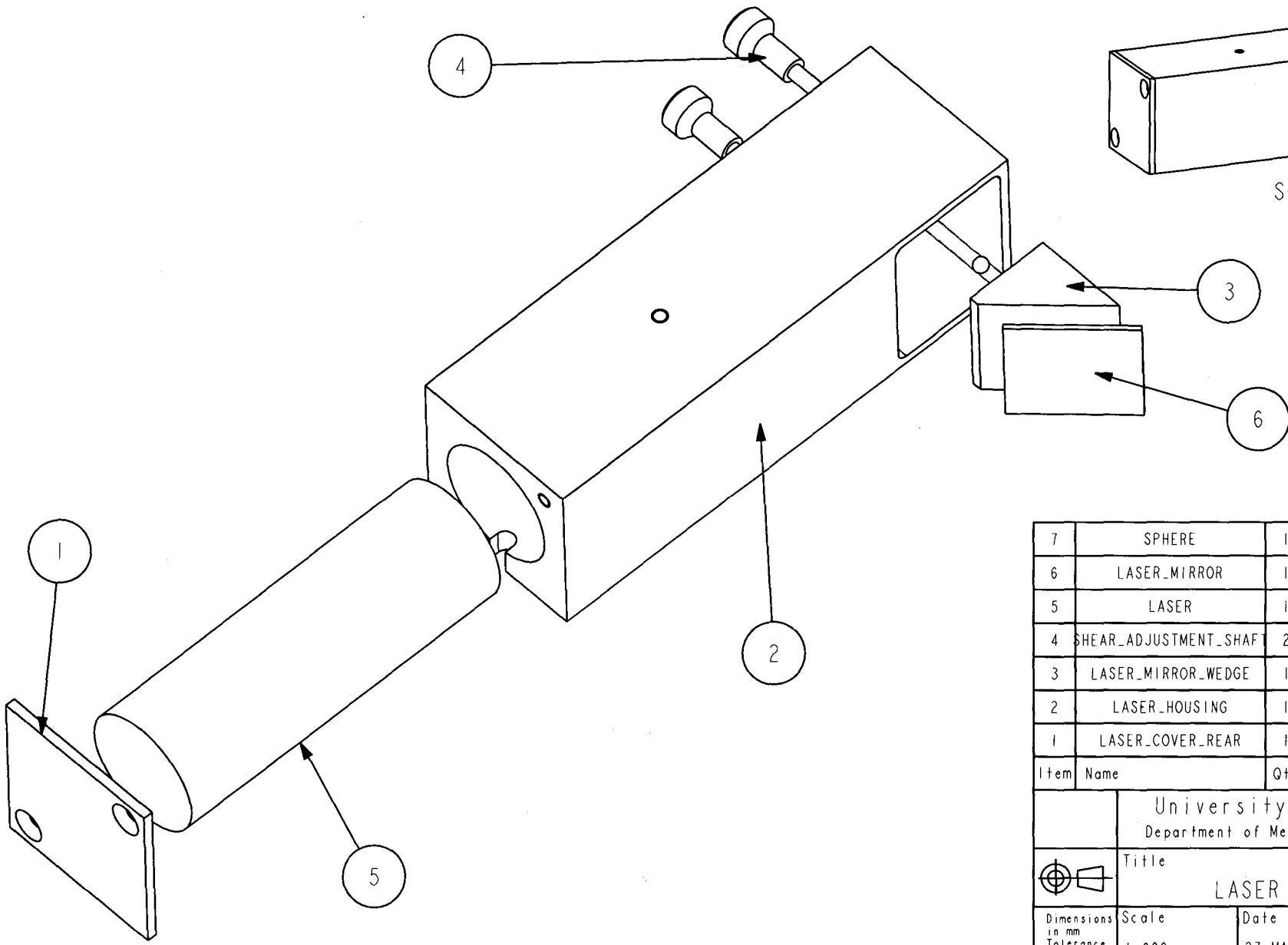
NOTE 2:
THESE 4 HOLES ARE M2.5 * .45
(AS ABOVE)

	ALUMINIUM	1	10MM SHEET
Item	Material	Qty	Remarks

University of Cape Town
Department of Mechanical Engineering

Title
CAMERA BOARD MOUNT

Dimensions in mm Tolerance U.O.S.	Scale	Date	Sheet	of
0.1	0,800	27 MAY 2008	20	27
	Drawn By		Drawing Number	
	B PITMAN		1.1.5	



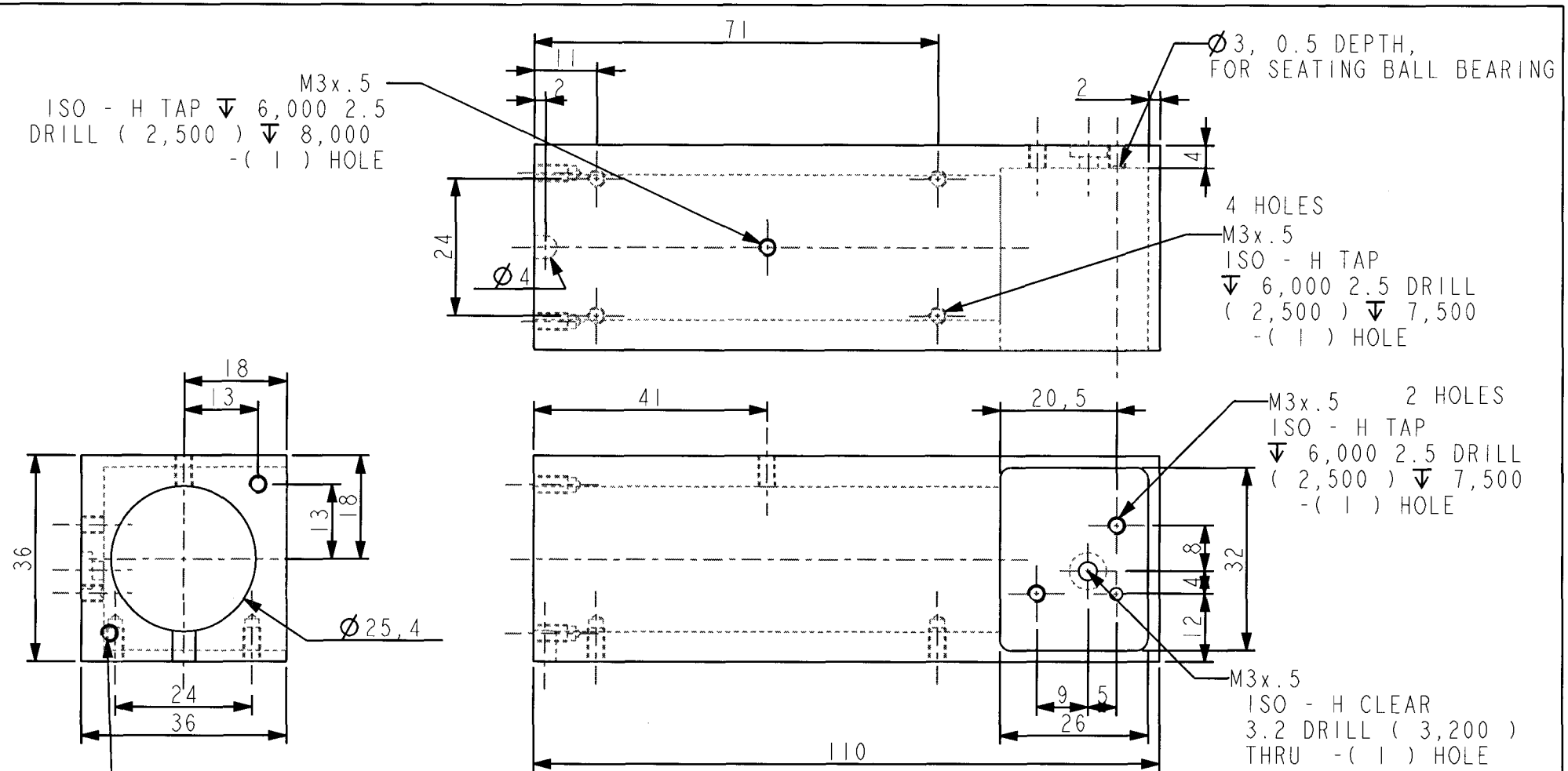
SCALE 0,500

Item	Name	Qty	Material
7	SPHERE	1	STEEL_LC
6	LASER_MIRROR	1	NO_MATERIAL
5	LASER	1	NO_MATERIAL
4	SHEAR_ADJUSTMENT_SHAFT	2	ALUMINIUM
3	LASER_MIRROR_WEDGE	1	ALUMINIUM
2	LASER_HOUSING	1	ALUMINIUM
1	LASER_COVER_REAR	1	ALUMINIUM

University of Cape Town
Department of Mechanical Engineering

Title
LASER ASSEMBLY

Dimensions in mm Tolerance U.O.S.	Scale	Date	Sheet	of
0.1	1,000	27 MAY 2008	21	27
	Drawn By: B PITMAN		Drawing Number: 1.2	

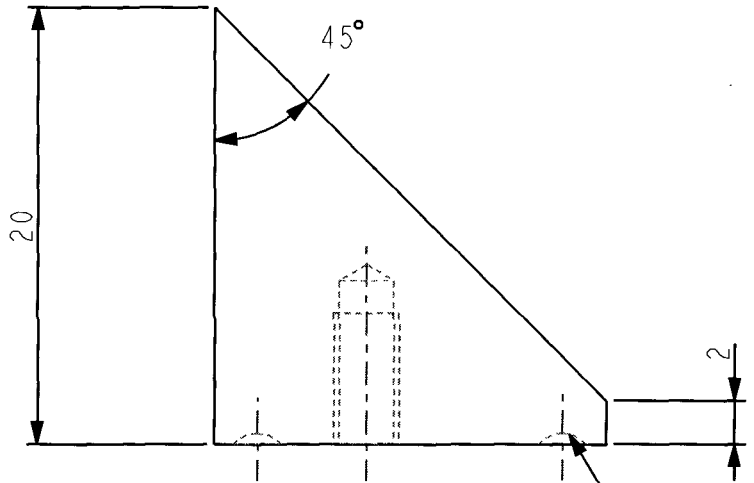


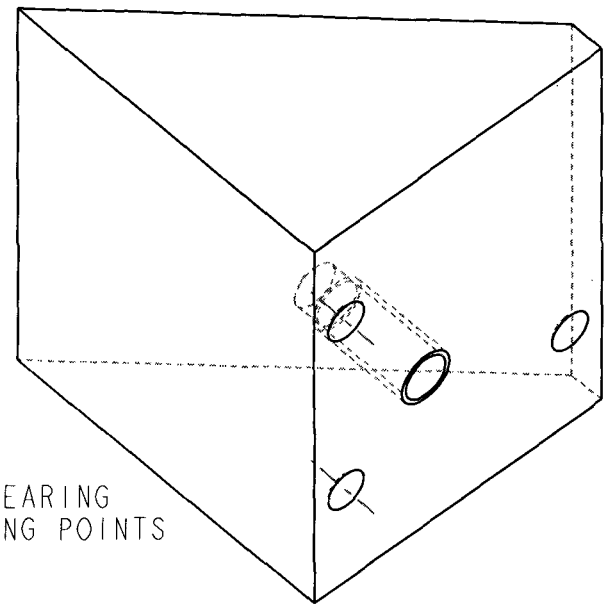
	ALUMINIUM	1	
Item	Material	Qty	Remarks

University of Cape Town
Department of Mechanical Engineering

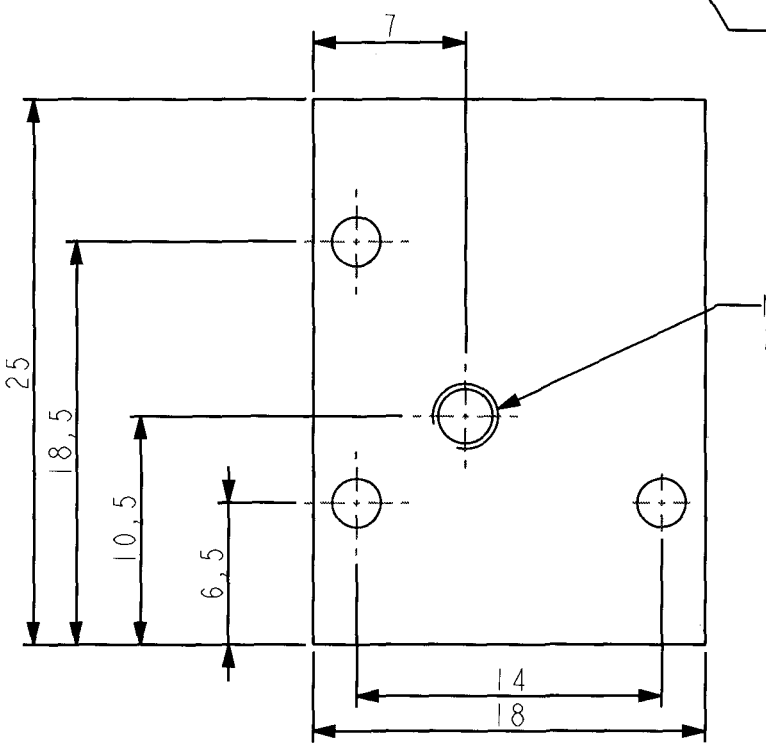
Title
LASER HOUSING

Dimensions in mm Tolerance U.O.S.	Scale	Date	Sheet	of
0.1	1,000	27 MAY 2008	22	27
	Drawn By		Drawing Number	
	B PITMAN		1.2.1	





Ø3, 0.5 DEPTH,
FOR SEATING BALL BEARING
(3 IDENTICAL SEATING POINTS
ON THIS PART)



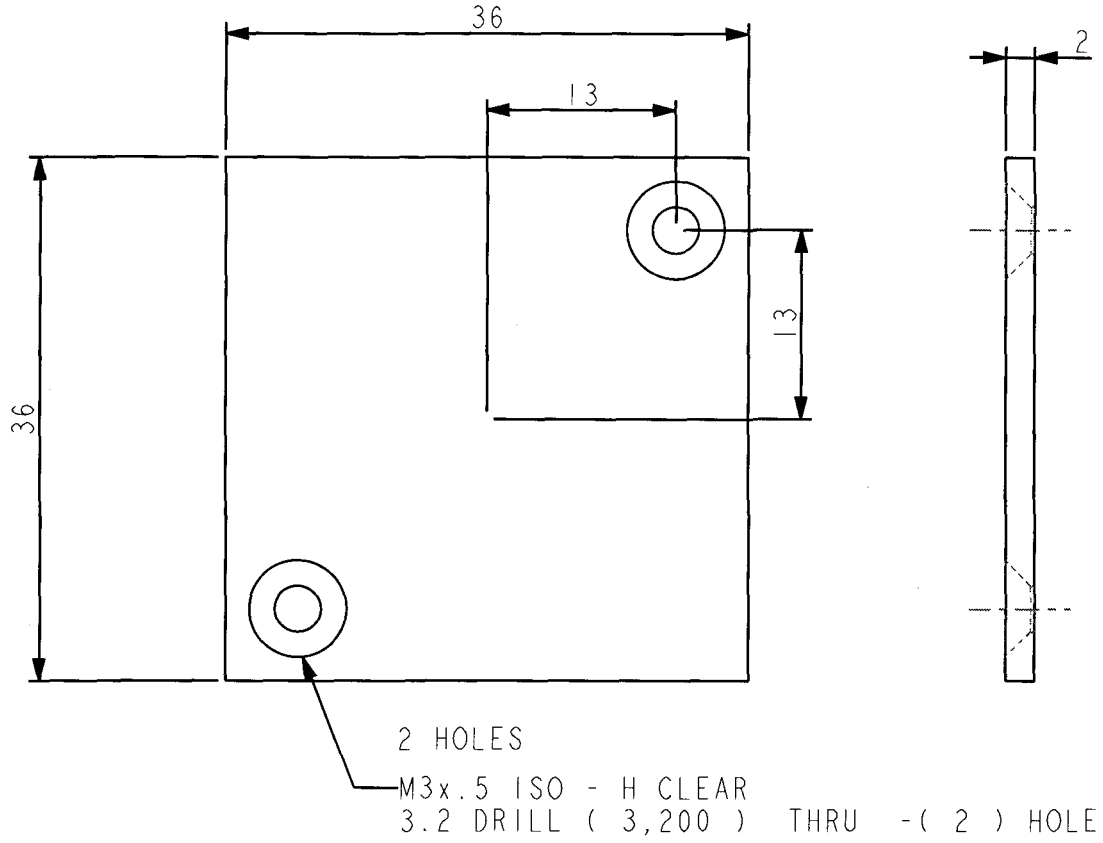
M3x.5 ISO - H TAP ↧ 6,000
2.5 DRILL (2,500) ↧ 7,500 -(1) HOLE

	ALUMINIUM	1	
Item	Material	Qty	Remarks

University of Cape Town
Department of Mechanical Engineering

Title
LASER MIRROR WEDGE

Dimensions in mm Tolerance U.O.S.	Scale	Date	Sheet	of
± 0.1	3,000	27 MAY 2008	23	27
	Drawn By		Drawing Number	
	B PITMAN		1.2.2	

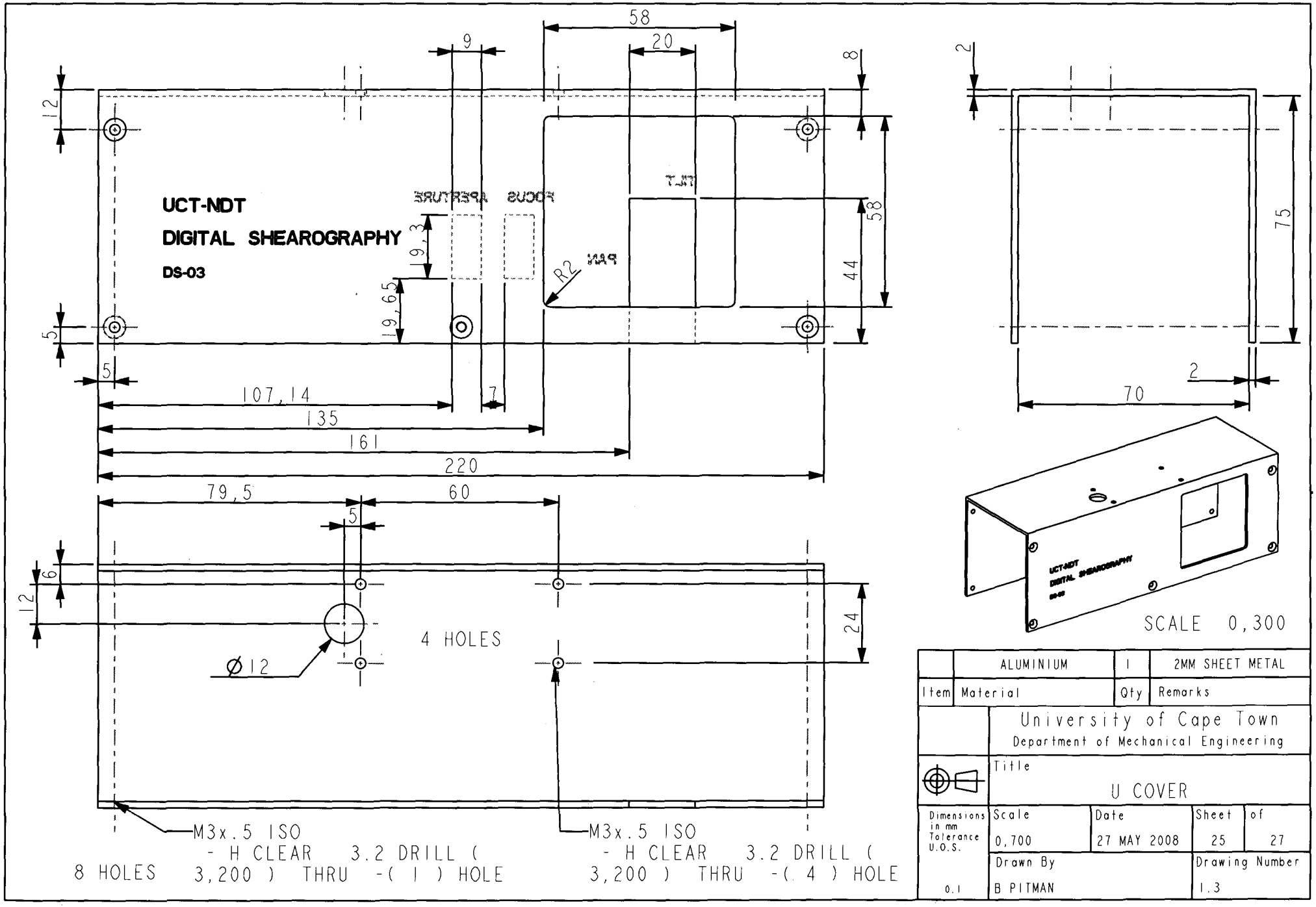


	ALUMINIUM	1	2MM SHEET
Item	Material	Qty	Remarks

University of Cape Town
Department of Mechanical Engineering

Title
LASER COVER REAR

Dimensions in mm Tolerance U.O.S.	Scale	Date	Sheet	of
	2,000	27 MAY 2008	24	27
	Drawn By		Drawing Number	
0.1	B PITMAN		1.2.3	

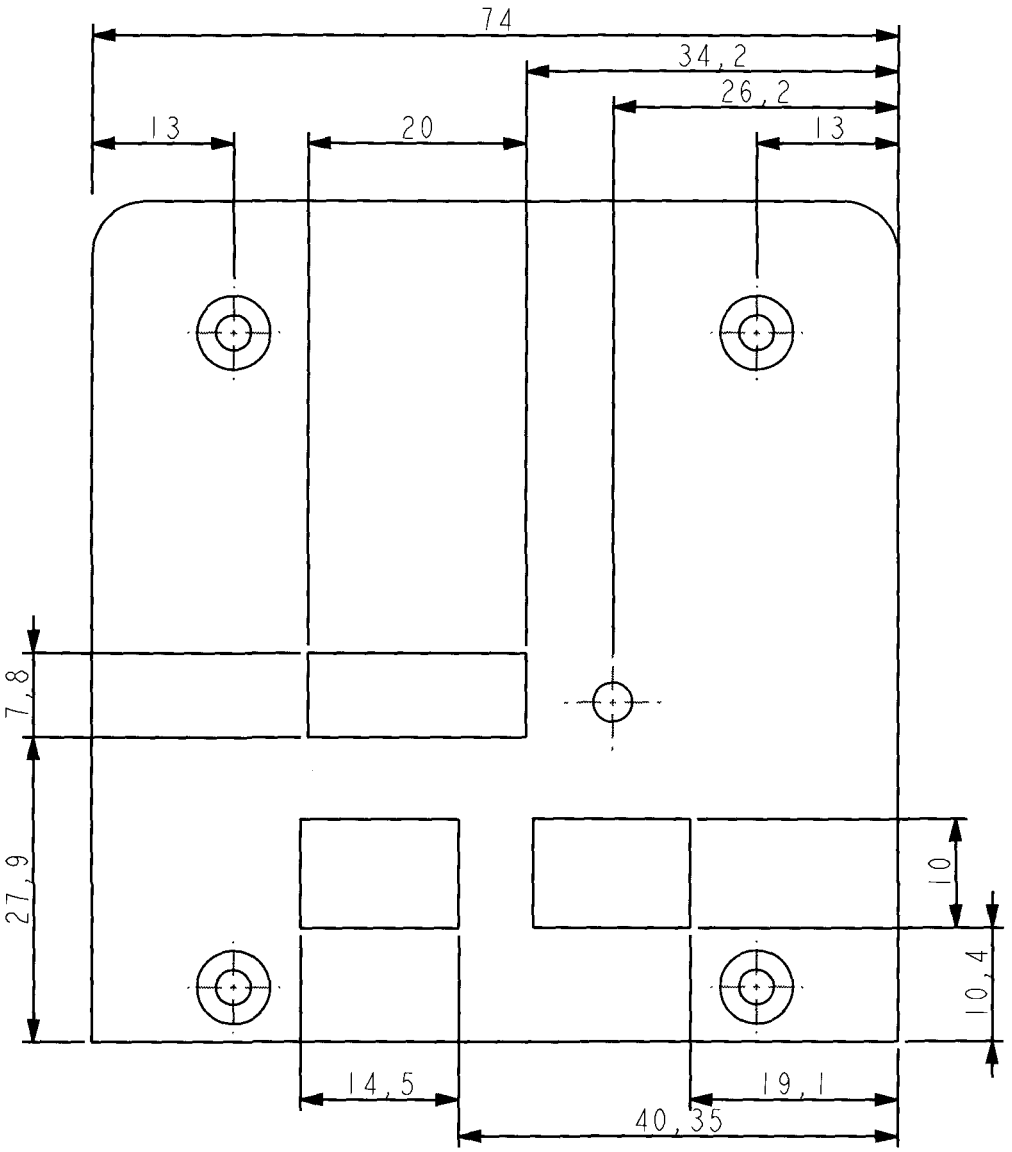


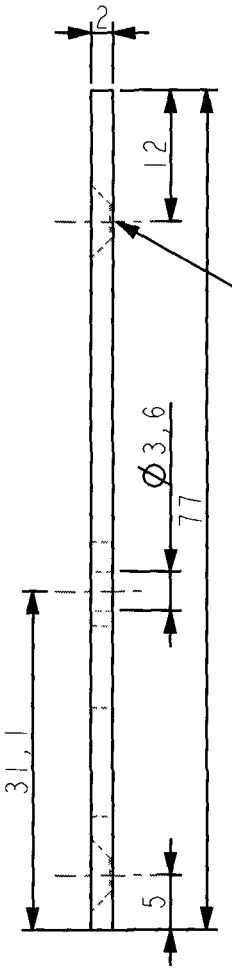
	ALUMINIUM	1	2MM SHEET METAL
Item	Material	Qty	Remarks

University of Cape Town
Department of Mechanical Engineering

Title: U COVER

Dimensions in mm Tolerance U.O.S.	Scale	Date	Sheet	of
0.1	0,700	27 MAY 2008	25	27
	Drawn By: B PITMAN		Drawing Number: 1.3	





4 HOLES

M3x.5
ISO - H CLEAR
3.2 DRILL (3,200)
THRU -(1) HOLE

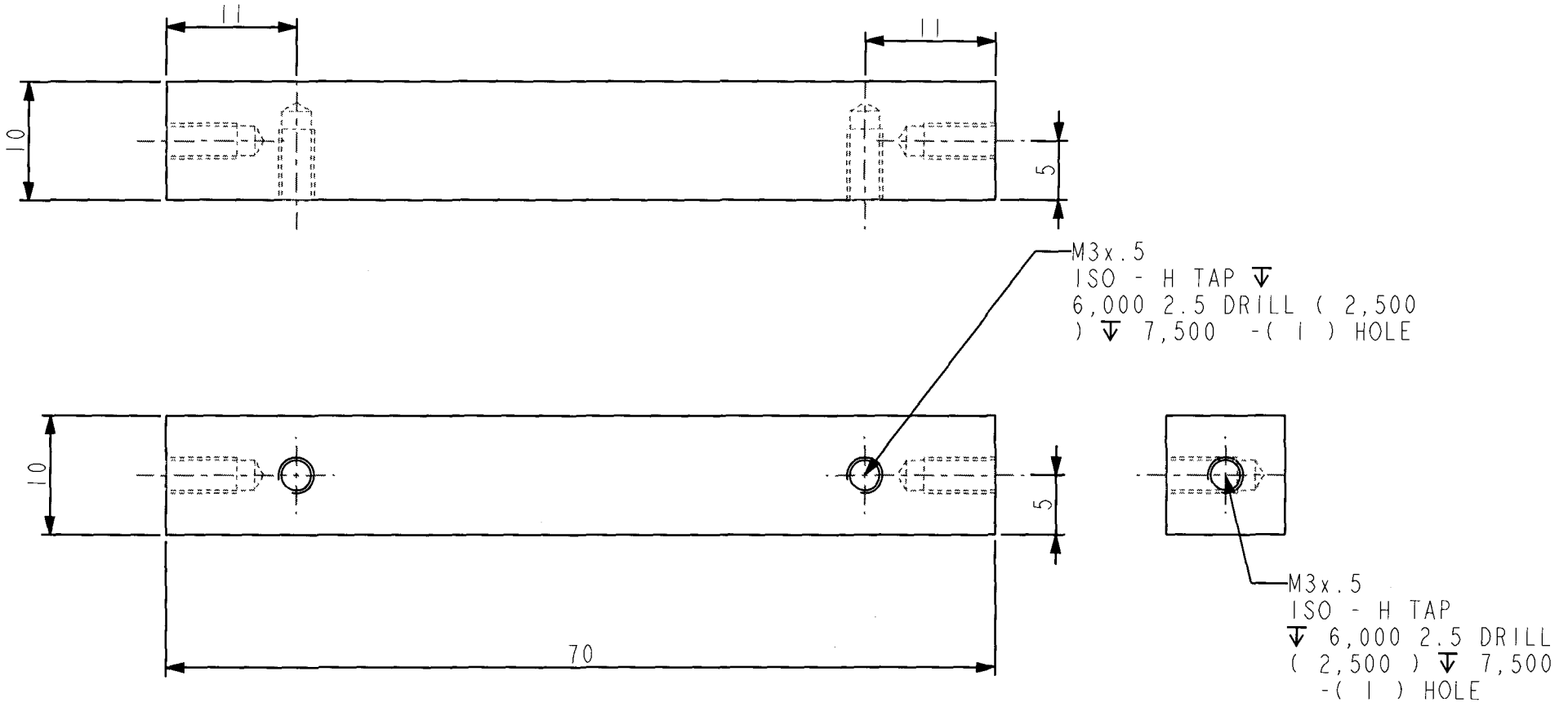
NOTES:
CUTOUTS SHOULD ONLY BE PERFORMED ON 1 OF THE 2 COVER PLATES
TWO LOWER CUTOUTS HAVE SAME HEIGHT AND WIDTH

	ALUMINIUM	2	2MM SHEET METAL
Item	Material	Qty	Remarks

University of Cape Town
Department of Mechanical Engineering

Title
COVER PLATE & CUTOUT

Dimensions in mm Tolerance U.O.S.	Scale	Date	Sheet	of
	1,500	27 MAY 2008	26	27
	Drawn By		Drawing Number	
0.1	B PITMAN		1.4	



	ALUMINIUM	2	10MM SHEET
Item	Material	Qty	Remarks

University of Cape Town
Department of Mechanical Engineering

Title
COVER PLATE LOCK

Dimensions in mm Tolerance U.O.S.	Scale	Date	Sheet	of
	2,000	27 MAY 2008	27	27
	Drawn By		Drawing Number	
0.1	B PITMAN		1.5	