Quantitative and Automatic Analysis of Interferometric Fringe Data Using Carrier Fringe and FFT Techniques

A thesis submitted for the degree of Doctor of Philosophy in the Department of Engineering, University of Warwick, UK, by

C. Quan, B.Sc., M.Sc.

September 1992

Contents

List of Figures

List of Tables

Acknowledgements

I am indebted to my supervisor, Dr. P. J. Bryanston-Cross, for his encouragement, technical guidance and inspiration; to Dr. D. G. Chetwynd and Dr. S. T. Smith for their help and fruitful discussion of many problems; to Dr. T. R. Judge for his work on fringe analysis software; to technicians in the Microengineering and Metrology Centre, University of Warwick, for their collaboration; to Mr. R. Dale-Jones for his help in reading the draft thesis.

The project is funded by my government and The British Council, and I am grateful for the associated financial support.

Declaration

This thesis is presented according to the regulations of the degree of Doctor of philosophy. The work described in this thesis is the result of my own investigations except where references are made to the work of others. It has not been submitted in any previous application for any degree.

Abstract

Computerised analysis of optical fringe pattern is a rapidly developing approach to extract quantitative data from phase encoded intensity distribution. This thesis describes results of investigations of quantitative and automatic analysis of interference fringe data using carrier fringe and FFT techniques. Several automatic and semiautomatic fringe analysis algorithms that enable the reduction of fringe patterns to quantitative data have been reviewed and illustrated with some examples. A fresh holographic contouring approach by the movement of object beams through fibre optics is described. The use of fibre optics provides a simple method for obtaining contouring fringes in the holographic system.

A carrier fringe technique for measuring surface deformation is described and verified by experiments. A theoretical analysis of the carrier fringe technique is given. The effects of carrier frequency on holographic fringe data has been investigated with computer generated holograms. In contrast to conventional holography and fringe analysis, this holographic system based on fibre optics and automatic spatial carrier fringe analysis technique. The FFT approach is used to process the interferograms. An excellent correlation between the theoretical deformation profile and that suggested by the technique is given. The accuracy of the measurement for a centrally loaded aluminum disk is 0.05μm.

The design and construction of a computerised photoelastic stress measurement system is discussed. This full field, fully automated photoelastic stress measurement system is a new approach to photoelastic fringe analysis. Linear carrier fringes generated using quartz wedge are superimposed on fringes formed by the stressed model. The resultant fringes pattern is then captured using a CCD camera and stored in a digital frame buffer. A FFT method has been used to process the complete photoelastic fringe image over the whole surface of the model. The whole principal stress difference field has been calculated and plotted from one single video frame.

List of publications based on the thesis

1. C. Quan, P. J. Bryanston-Cross, "Double-source holographic contouring using fibre optics", *Optics & Laser Technology* Vol. 22, No. 4, 255-259(1990).

2. C. Quan, T. R. Judge and P. J. Bryanston-Cross, "Holographic measurement of deformation using carrier fringe and FFT techniques", *Proc. SPIE*, Vol. 1507, 463-275(1991).

3. T. R. Judge, C. Quan and P. J. Bryanston-Cross, "Holographic deformation measurement by Fourier transform technique with automatic phase unwrapping", *Optical Engineering*, Vol. 31, No. 3, (533-543)1992.

4. C. Quan, P. J. Bryanston-Cross and T. R. Judge, "Photoelasticity stress analysis using carrier fringe and FFT techniques", accepted for publication in *Optics and Lasers in Engineering*, September 1992.

List of reports based on the thesis

1. C. Quan, T. R. Judge and P. J. Bryanston-Cross, "Holographic deformation measurements using fibre optics and automatic spatial carrier fringe pattern analysis", Report No. 8, Control and Instrument Systems Centre, Engineering Department, April, 1990.

2. C. Quan and P. J. Bryanston-Cross, "Automated & high accuracy for measuring holographic interference fringe patterns", Report No. 13, Control and Instrument Systems Centre, Engineering Department, February, 1991.

Seminar presented

1. C. Quan, "Holographic deformation measurements using carrier fringe and FFT techniques", Control and Instrument Systems Centre, University of Warwick, February, 1991.

Chapter 1

General Introduction

Since its invention three decades ago, holographic interferometry has proven to be a powerful non-destructive testing tool for the measurement of displacement and its derivatives [1]. This technique can be used for measuring changes in the shape of objects with rough surfaces with very high precision. With the aid of automatic fringe analysis, holographic interferometry is now extensively used in the industrial environment in the following four applications areas: (1) non-destructive testing of materials and parts [2]; (2) static and dynamic construction optimisation [3]; (3) measuring outlines and contouring [4, 5, 6]; (4) examination of particle image velocimetry [7, 8].

In chapter 2, holography and holographic interferometry is reviewed. Attention is drawn to the importance of holography contributing to a better presentation of the results of measurement, and to the formation of holographic fringes. The developments and applications of holographic interferometry have been discussed, and prospects of more rapid development have also been mentioned, such as X-ray holography [9, 10].

There are two important aspects of holographic interferometry. The first is the setting up and optimisation of the equipment needed to do the experiment. The theory of holographic interferometry in reference to the description of the set up and its effect on the formation of fringe patterns have been well established in

the last 25 years [1, 11, 12, 13, 14, 15, 16]. Another important theoretical aspect of holographic interferometry is the evaluation and interpretation of the interferograms. Chapter 3 presents a discussion of the "state of the art" technology for fringe analysis. The automation of analysis of interference patterns has progressed considerably during the last ten years. Automatic evaluation requires a computer to detect the fringe patterns. There are four main fringe analysis methods which have been developed: (1) fringe counting and skeleton [17, 18]; (2) the heterodyne method which can be used in real-time or in the double-exposure measurement [19, 20]; (3) the phase-stepping method has been shown to be reliable over the last few years [21, 22, 23, 24, 25]; (4) the Fourier transform method which is based on the carrier fringe and fast Fourier transform techniques [26, 27, 28].

The main contribution of the thesis is to develop carrier fringe and FFT techniques in the field of holographic deformation measurement and photoelasticity stress analysis. The carrier fringes are generated using a fibre optic holographic system. In the case of photoelasticity, carrier fringes are produced by putting a birefringent wedge in a polariscope. The theoretical analysis of the techniques and their applications are discussed in the later chapters. In chapter 3, examples of both 1-D and 2-D Fourier transform fringe analysis are given.

The holographic set-ups should be designed to be as flexible and simple as possible. Therefore, they can be adapted to any given task. There is a gradually increasing employment of fibre optics in holographic set-ups [29, 30, 31, 32]. Chapter 4 describes the fibre optic and its application to holography. The use of fibre optics allows laser light to be delivered via single-mode fibre optics for both the object beam and the reference beam. This has many advantages over a conventional holographic system, such as more flexible arrangements, and less optical equipment(mirrors, beam splitters, spatial filters etc.). A holographic system based on fibre optics was developed for holographic contouring and holographic deformation measurement using carrier fringe and FFT techniques. In a later chapter, the use of this holographic set up in an automated system for the analysis of holographic interferograms will be discussed.

The problem of contouring objects has been approached from both the viewpoint of coherent optics and incoherent optics. In the case of coherent optics, a considerable amount of interest has been shown in the development of contouring techniques using holographic interferometry. Among them are the two wave-length technique [33, 34], the immersion technique [35] and two source technique [36, 37]. In the case of incoherent optics, Moire contouring is currently used in some surface measurement applications [38]. Chapter 5 describes different surface contouring techniques and their advantages and disadvantages. The aim of chapter 5 is to present a fresh holographic contouring approach based on fibre optics and automatic fringe analysis. The novel feature of the work is the introduction of fibre optics into holographic contouring. The holographic contouring system was developed to obtain contour maps of a 3-D diffuse object. The strategy is implemented in three steps. First, a set of interference fringes modulating the 3-D variations of the surfaces of the objects is produced. These fringes are created by translating fibre optics illuminating the object surface. The second step concerns the reconstruction of the hologram and image capture by CCD camera. The third step is devoted to the demodulation of these fringes by the FFT technique, to extract a reconstructed object with contour lines superimposed on it. Advantages of the technique presented here are that it produces a very good fringe visibility, and the contour interval is easily varied by precise control of the translating fibre optics. The phase of the fringe pattern can be determined to a much higher degree of accuracy.

In holographic interferometry, it is often difficult to identify the sign of a displacement normal to the surface of an object. For years, this has been the major obstacle to automatic data acquisition of holographically recorded displacements. Recently, the carrier fringe technique has been developed to overcome the problem of phase ambiguities [39, 40, 41, 42, 43, 44, 45]. Chapter 6 describes the theory of the carrier fringe technique, and describes different methods of generating carrier fringes. The effects of carrier frequency on the holographic fringe field has been investigated, with several examples, by computer generated holograms.

Chapter 7 demonstrates the applications of the carrier fringe and FFT techniques to holographic deformation measurements. The techniques are based on the principle of holographic interferometry: that is, first of all recording an object on a holographic plate before applying a mechanical load to the object and then making a second exposure after applying a mechanical load to the object. A special optical setup based on fibre optics has been developed to evaluate this type of hologram, producing hyperbolic interference fringes from which one can determine the displacement of a point of interest. Experiments are performed with a static loaded disk and cantilever beam. Nearly straight and equally spaced fringes are produced by a lateral displacement of the fibre optics illuminating the object. The fringes can be used as carrier fringes which are superimposed on the deformation fringes. The complicated deformation fringes resulting from double-exposure holography can be reduced to a simple linear fringe pattern. This facilitates the use of the automatic fringe analysis method. In the data reduction procedure, the fast Fourier transform technique is used to process the interferograms. It is shown that this method is a successful technique for extracting phase information from fringe patterns. A new phase unwrapping technique is introduced based on the tile level and minimum spanning tree strategy. This has been proved to be an effective technique that can deal with any fringe field; the experimental results are well correlated with the theoretical predictions. The technique described in chapter 7 proves to be simple and reliable. It is a new way of generating carriers in holographic interferometry measurement, i.e. using fibre optics. It also introduces a novel fringe data reduction method using FFT and automatic phase unwrapping techniques. These techniques are an effective way of investigating out-of-plane and in-plane displacement measurements.

Holographic interferometry, photoelasticity, and Moire techniques all contain information in the form of an optical fringe pattern. Reduction of these fringe patterns to quantitative data for each of the above techniques differs only by the generation of carrier fringes and the interpretation of the optical phase [46]. Chapter 8 discusses the application of the carrier fringe and FFT technique to

photoelastic fringe patterns. When polarised light is passed through a stressed transparent material, interference patterns or fringes are formed. These patterns provide immediate qualitative information about the general stress distribution, positions of stress concentrations and areas of low stress. At points where the fringes are most numerous and closely spaced, the stress is high; at points they are widely spaced or absent, the stress is low. The interpretation of fringe patterns is usually done by hand and the final results of stress distribution depends on the skill of the analyst.

The technique developed in this chapter presents a new way of analying photoelastic fringe patters. A major feature of the technique is that a linear carrier fringe pattern generated by a quartz wedge is superimposed on the stressed model fringes. The resultant fringe patterns are captured by a CCD camera, and stored in a digital frame buffer. Therefore, a large amount of data can be accessed frequently and a computer connected to the store has direct access to the information at every point in the image. The complete photoelastic fringe pattern is processed over the whole surface of the model by using a fast Fourier transform approach. Finally, the whole principle stress difference field is given by a 3-D mesh plot. The technique provides an effective method of failure investigation, and produces valuable information leading to successful design.

Chapter 2

Optical Interference, Holography and Holographic Interferometry

2.1 The General Principle of the Interference of Light

2.1.1 Interference and the Properties of Light

The interference phenomenon results from the superposition of two or more beams of light. The superposition of beams of monochromatic light always gives rise to interference.

The first man successfully to demonstrate the interference of light, and thus establish its wave character, was Thomas Young [47]. Figure 2.1 shows the arrangement of Young's experiment. Light was first allowed to pass through a pinhole S and then, at a considerable distance away, through two pinholes S_1 and S_2 separated by a distance **d**. The two sets of spherical waves emerging from the two holes interfered with each other.

The light diffracted through the two pinholes S_1 and S_2 in screen B spreads out toward the screen C. Interference of the two spherical waves produces a variation in irradiance on screen C. According to Young and Fresnel [1, 47], it is due to the

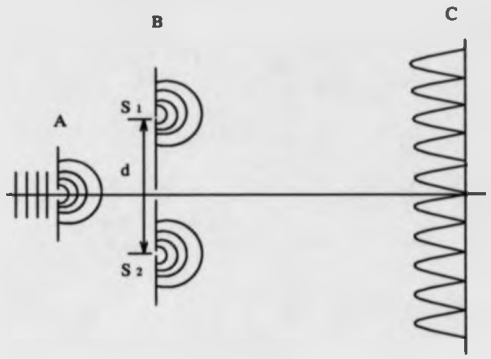


Figure 2.1: Experimental arrangement for Young's two-pinhole experiment.

summation of the light waves with account taken of their phases. The phases are determined by the distances from the pinhole S_1 and S_2 to the relevant points of screen C.

In accordance with the principle of the electromagnetic theory of light, a representation of the electric component of a monochromatic light wave can be written as [48]

$$\vec{E}(x,y,z,t) = \vec{a}(x,y,z)e^{-i[\phi(x,y,z)+\omega t]} \tag{2.1}$$

where $\vec{E}(x,y,z,t)$ is the electric field intensity vector, $\vec{a}(x,y,z)$ is the amplitude vector of a wave, x,y,z are the spatial coordinates of the wave, and t is the travelling time of the wave. Equation 2.1 represents a monochromatic plane wave with phase $\phi(x,y,z)$ angular frequency $\omega = 2\pi f$.

The irradiance of light is proportional to the square of the amplitude of the electric field, therefore, the response of the detector can be expressed by

$$I = \varepsilon c < |\vec{E}(x,y,z,t)|^2 > \tag{2.2}$$

where I is the intensity of the light, the brackets $\langle\ \rangle$ indicate averaging in time. $|\vec{E}(x,y,z,t)|^2 = \vec{E}(x,y,z,t)\vec{E}^*(x,y,z,t)$, $\vec{E}^*(x,y,z,t)$ is the complex conjugate

of $\vec{E}(x,y,z,t)$. ε is the permittivity of the medium, c is the velocity of light in a vacuum.

Evaluating the real part of Eq. 2.1 and substituting into Eq. 2.2 gives

$$I = \varepsilon c < a^2(x,y,z)\cos^2[\phi(x,y,z) + \omega t] > \tag{2.3}$$

The average value of I during the time t is given by

$$I = \frac{\varepsilon c}{t} a^2(x,y,z) \int_{-t/2}^{t/2} \{1 + \cos\ 2[\phi(x,y,z) + \omega t]\} dt \tag{2.4}$$

Since the integration time t is greater than the period of oscillations T of light,

$$\int_{-t/2}^{t/2} \{\cos\ 2[\phi(x,y,z) + \omega t]\} dt \doteq 0$$

therefore, we obtain

$$I = \frac{\varepsilon c}{t} a^2(x,y,z) \tag{2.5}$$

Equation 2.5 indicates that irradiance is proportional to the square of the amplitude of electric field. By omitting the constant factor $\frac{\varepsilon c}{t}$, the intensity of the light can be defined as

$$I = [a(x,y,z)]^2 \tag{2.6}$$

The complex quantity $\vec{E}(x,y,z,t)$ can be represented in the form of two multipliers, one of which depends only on the time , and the other only on coordinates of the given point

$$\begin{aligned} \vec{E}(x,y,z,t) &= \vec{a}(x,y,z) e^{i\phi(x,y,z)} e^{-i\omega t} \\ &= \vec{A}(x,y,z) e^{-i\omega t} \end{aligned} \tag{2.7}$$

where the quantity $\vec{A}(x, y, z)$ is the complex vector of the amplitude of the wave. Therefore, the light intensity is given by

$$I = \vec{E}\vec{E}^* = EE^* = \vec{A}\vec{A}^* = AA^* \tag{2.8}$$

where * stands for a complex conjugate quantity.

2.1.2 Interference of Two Coherent Waves

(1) Interference of two coherent waves [49]

Two spherical waves that propagate from the point source O_1 and O_2 (see Fig 2.2) can be expressed as

$$\vec{A}_1(\vec{r}_1) = \vec{a}_1(\vec{r}_1)e^{-i(\delta_1 - \vec{k}\vec{r}_1)} \tag{2.9}$$

$$\vec{A}_2(\vec{r}_2) = \vec{a}_2(\vec{r}_2)e^{-i(\delta_2 - \vec{k}\vec{r}_2)} \tag{2.10}$$

where $\vec{r}_1$ and $\vec{r}_2$ are the position vectors of the point P relative to O_1 and O_2

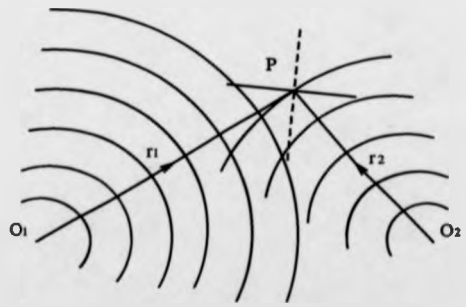


Figure 2.2: Interference of two spherical waves.

The total intensity at P is

$$I = a_1^2(\vec{r}_1) + a_2^2(\vec{r}_2) + 2\vec{a}_1(\vec{r}_1)\vec{a}_2(\vec{r}_2)cos[\vec{k}(\vec{r}_1 - \vec{r}_2) - (\delta_1 - \delta_2)] \quad (2.11)$$

Assuming that the initial phase difference $\delta_1 - \delta_2 = 0$ or remaining constant, a phase difference at P is given by

$$\delta = \vec{k}(\vec{r}_1 - \vec{r}_2) \quad (2.12)$$

where the term $\vec{r}_1$ and $\vec{r}_2$ are the radii of the spherical wavefronts overlapping at P. The intensity in the region surrounding O_1 and O_2 will vary from point to point as $(\vec{r}_1 - \vec{r}_2)$ varies. Equation 2.12 defines a family of hyperboloids of revolution. If the waves are in phase at the emitter, $\delta_1 - \delta_2 = 0$, constructive interference and destructive interference occur respectively when

$$\vec{k}(\vec{r}_1 - \vec{r}_2) = 2m\pi \qquad \text{constructive interference} \quad (2.13)$$

$$\vec{k}(\vec{r}_1 - \vec{r}_2) = (2m + 1)\pi \quad \text{destructive interference} \quad (2.14)$$

where $m = 0, \pm 1, \pm 2.....$, m is the fringe order.

(2) Interference of two plane waves [49]

Considering two sources O_1 and O_2 in Fig. 2.2 emitting monochromatic waves of the same frequency in a homogeneous medium with a separation is much greater than wavelength, and the point of observation P far enough away from the sources, in this case, the wavefronts at P will be planes. Figure 2.3 shows the superposition of two plane waves of identical frequency characterised by the wave vectors $\vec{k}_1$ and $\vec{k}_2$.

Two wave fronts of the amplitude can be written in the form of the vectors

$$\vec{A}_1 = \vec{a}_1(\vec{r})e^{-i(\delta_1 - \vec{r}\vec{k}_1)} \quad (2.15)$$

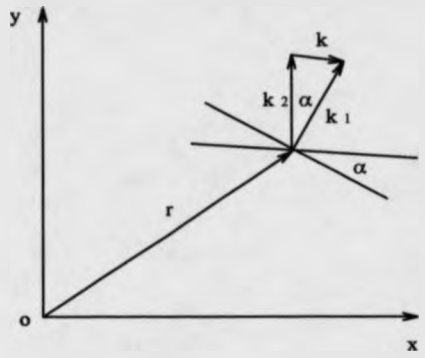


Figure 2.3: Superposition of two plane waves of identical frequency.

$$\vec{A}_2 = \vec{a}_2(\vec{r})e^{-i(\delta_2 - \vec{r}\vec{k}_1)} \tag{2.16}$$

In accordance with the principle of superposition, the complex amplitude intensity of the two waves is then

$$I = a_1^2 + a_2^2 + 2a_1a_2cos[\vec{r}(\vec{k}_1 - \vec{k}_2) - (\delta_1 - \delta_2)] \tag{2.17}$$

At various points in space, the resultant intensity becomes maximum and minimum depending on the phase δ

$$\delta = \vec{r}(\vec{k}_1 - \vec{k}_2) - (\delta_1 - \delta_2) = 2m\pi \qquad \text{maximum} \tag{2.18}$$

$$\delta = \vec{r}(\vec{k}_1 - \vec{k}_2) - (\delta_1 - \delta_2) = (2m+1)\pi \qquad \text{minimum} \tag{2.19}$$

If the initial phase difference $\delta_1 - \delta_2$ is constant or 0, the maximum fringe intensity occurs when

$$\vec{r}(\vec{k}_1 - \vec{k}_2) = 2m\pi \tag{2.20}$$

2.2 Optical Interferometry

The subject of optical interferometry makes use of the two kinds of light interference. The first type of interference is by division of wave front, such as in Young's double-slit experiment. The second is interference by division of amplitude, such as in Twyman-Green and Michelson interferometer.

In the case of division of wave front, two or more portions of a wave front are combined. The relative phases of the portions determine whether they interfere constructively or destructively, and the visibility of fringes produced by division of wave front will be related to the spatial coherence distance produced by the source of light. Figure 2.4 shows the interference with two point sources as an example of division of wave front [50].

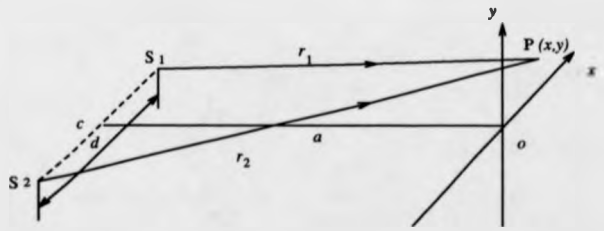


Figure 2.4: Illustration interference with two point sources.

Suppose the pattern is observed over a plane xoy, normal to the perpendicular bisector co of S_1S_2, and with the x-axis parallel to S_1S_2. The separation of the two point sources is d. a is the distance between the point source and observing screen. For a point $P(x,y)$ in the plane of observation, the optical path difference Δr is given by

$$\Delta r = r_2 - r_1 = \frac{2xd}{r_2 + r_1}$$

In practice, d is much smaller than a, and $r_2 + r_1 \doteq 2a$, therefore, the optical path difference Δr is

$$\Delta r = \frac{xd}{a}$$

The corresponding phase difference δ is given by

$$\delta = \frac{2\pi}{\lambda}\frac{xd}{a} \tag{2.21}$$

The maximum intensity occurs when

$$x = \frac{ma\lambda}{d} \quad |m| = 0, 1, 2..... \tag{2.22}$$

and the minimum of intensity occurs when

$$x = \frac{ma\lambda}{d} \quad |m| = 1/2, 3/2, 5/2..... \tag{2.23}$$

The interference pattern on the screen consists of bright and dark bands as shown in Figs. 2.5 and 2.6

In division of amplitude, a wave at one point is divided into two or more parts by a beam splitter, a time delay is introduced into each part, and the parts are recombined, resulting in an interference pattern. A simplified diagram of the Michelson interferometer is shown in Fig. 2.7 [47].

Figure 2.5: Young's fringe patterns.

The constructive interference and destructive interference can be written as

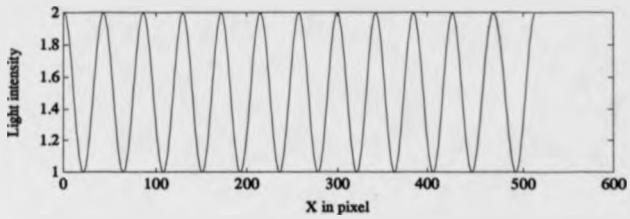


Figure 2.6: Intensity distribution of Young's fringes.

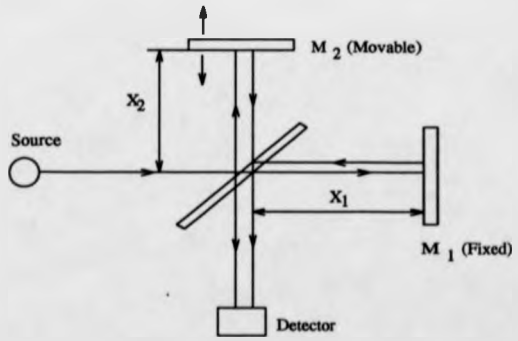


Figure 2.7: The Michelson interferometer.

$$\delta = 2(x_2 - x_1) = m\lambda \tag{2.24}$$

and

$$\delta = 2(x_2 - x_1) = (m + 1/2)\lambda \tag{2.25}$$

where $m = 0, 1, 2, 3.....$

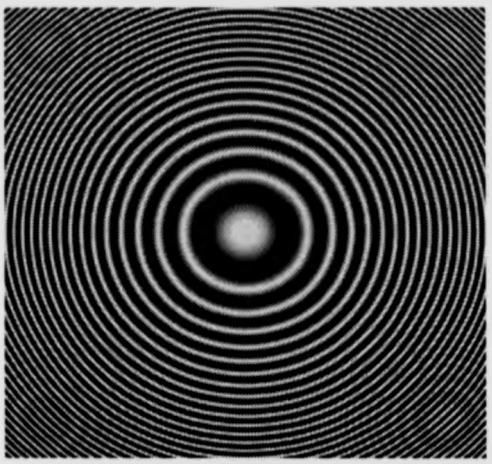


Figure 2.8: A ring pattern observed with a Michelson interferometer (image size 512×512).

The interference fringe pattern observed in a Michelson interferometer is shown in Fig. 2.8 [51]. Figure 2.9 shows a Twyman-Green interferometer with CCD camera controlled by a microcomputer. The interference fringe pattern can be displayed on the monitor. This type of interferometer is widely used in industrial real-time measurements. A linear fringe pattern observed by such an interferometer is shown in Fig. 2.10. Figure 2.11 is the linear fringe pattern modulated by a soldering iron which shows the temperature change around the edge of the soldering iron.

Newton's rings are another example of division of amplitude. They are formed

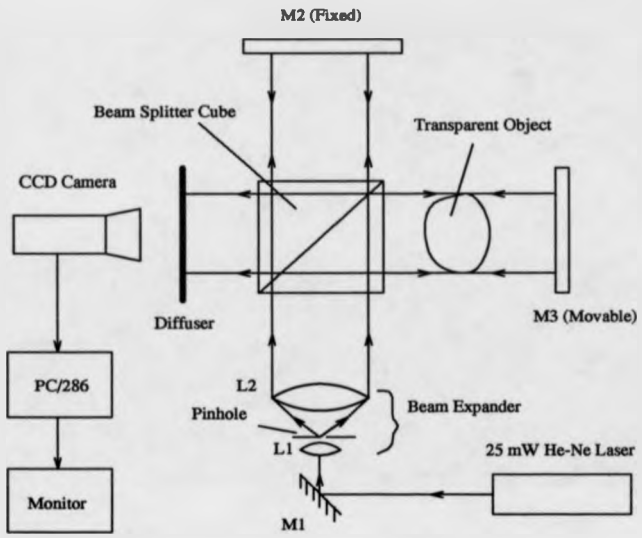


Figure 2.9: Twyman-Green interferometer and computer system.

Figure 2.10: A linear fringe pattern observed with a Twyman-Green interferometer (image size 512 × 512).

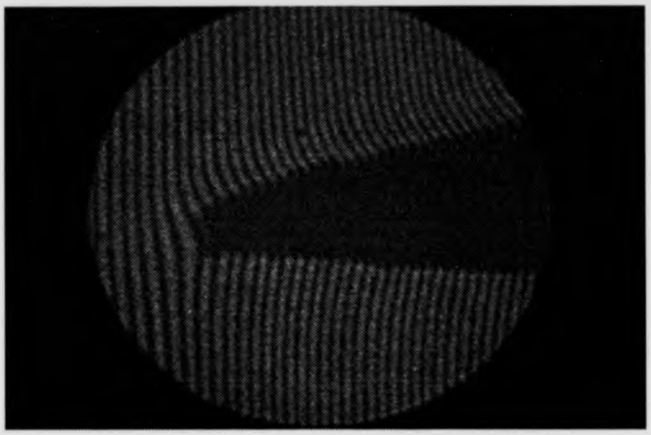


Figure 2.11: A linear fringe pattern modulated by a soldering iron (image size 512 × 512).

when a spherical surface and flat surface are put together as shown in Fig. 2.12.

Newton's rings pattern can be observed using reflected light for a dark spot in the middle and using transmitted light for a bright spot as shown in Fig. 2.13 and Fig. 2.14.

In optical interferometry, by knowing the shape of one of the interfering fronts, the shape of the second one can be determined according to the form of the fringe pattern. The displacement of the optical elements of the interference setup can be determined by investigating the displacements in the shape of the interference pattern.

The thrust in the development of optical interferometry at the present time is the combination of electronics, microcomputers and interferometers for rapid acquisition and analysis of interferometric data [52]. Holographic, speckle and moire interferometry has developed based on the principles of optical interferometry.

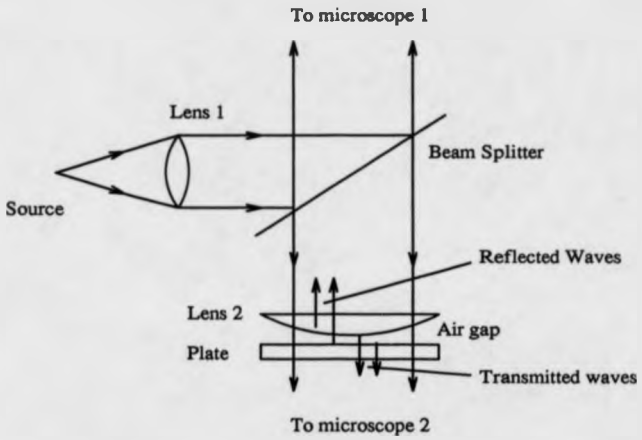


Figure 2.12: Experimental setup for the observation of Newton's rings

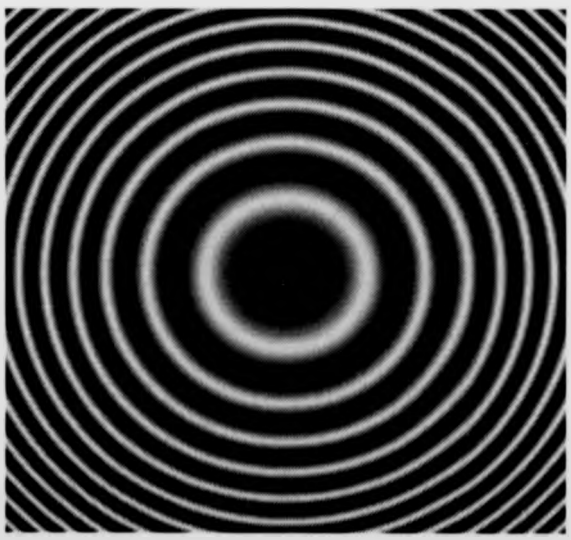


Figure 2.13: Newton's rings observed using reflected light (image size 512×512).

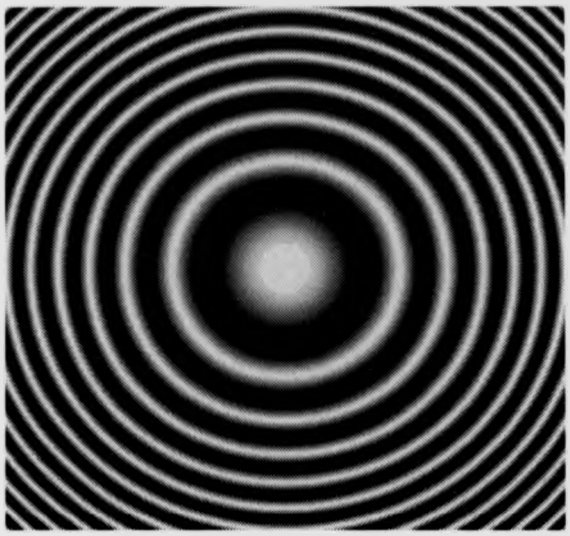


Figure 2.14: Newton's rings observed using transmitted light (image size 512 × 512).

2.3 Holography

2.3.1 Brief History of Holography

Holography is a technique for recording and reconstructing three-dimensional images. This technique is based on recording the distribution of intensity in an interference pattern formed by an object wave and a reference wave coherent with it. Holographic techniques have great potential in the field of modern optical measurements [2, 52, 53].

Holography was first invented by Gabor [11, 12, 13] to improve the resolving power of electron microscopes. Gabor suggested a two step photographic imaging process in which the hologram was recorded with electron wavelengths and reconstructed with wavelengths in the visible region. The Gabor-type hologram has two major problems: (1) The object blocks most of the light to the film so that exposing the film properly is difficult; (2) the virtual image is inconveniently

located in front of the real image when reconstructing the image. However, in the early 1960s, Leith and Upatnieks [14, 15, 16] demonstrated a new method for optical holography based on communication theory techniques. With the advent of the laser, very intensive coherent quasi-monochromatic beams of light became available. Using the laser, progress in optical holography has been explosive. Since 1962, many holographic techniques have emerged, such as colour holography [54], composite holography [55] and white light holography [56]. The application of holographic techniques to the areas of optical data processing, computing, and data storage is receiving much attention [57]. Additional applications of holography may be foreseen in the area of photography for purposes of utilising the three dimensional nature of such imaging systems. Holographic interferometry shows promise in the areas of deformation and vibration analysis [58, 59], visualisation of combustion flame [60], stress analysis and nondestructive testing and turbulence studies [2]. It is already clear that new applications of holography are emerging.

Figure 2.15 displays a typical fringe pattern obtained by the double-exposure method on a silicon wafer under a central loading. This deformation fringe pattern can be modulated by adding the linear carrier fringes as shown in Fig. 2.16, which is also obtained by a double-exposure technique.

2.3.2 Features of Optical Holography

Unlike photographic negatives, holography is an interference method of recording the light waves diffracted by an object illuminated with coherent light. In a first step, the diffracted waves are caused to interfere with a phase-related reference wave. If the waves are highly coherent, the relative phase between object and reference wave remains constant in time producing an observable effect on the intensity distribution of the resulting pattern. The photographic record of this pattern contains sufficient information about both phase and amplitude of the diffracted waves to permit their reconstruction. In a second step, when the holo-

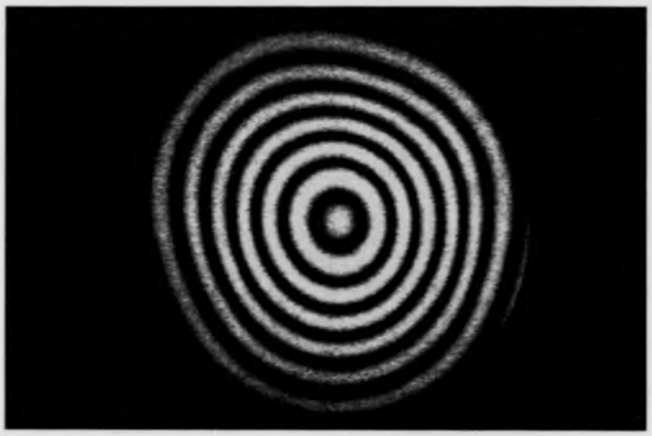


Figure 2.15: Example of double-exposure hologram showing the deformation of a silicon wafer under central loading (image size 512 × 512).

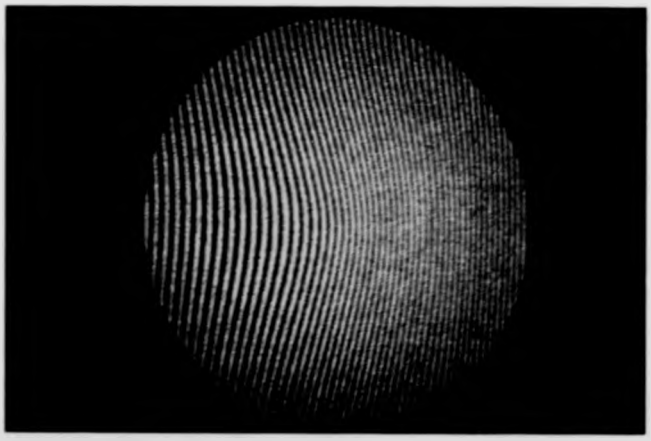


Figure 2.16: An double-exposure hologram representing the deformation fringes modulated by the carriers (image size 512 × 512).

gram is illuminated with the reference wave wavefront reconstruction takes place. So the features of the hologram are as follows:

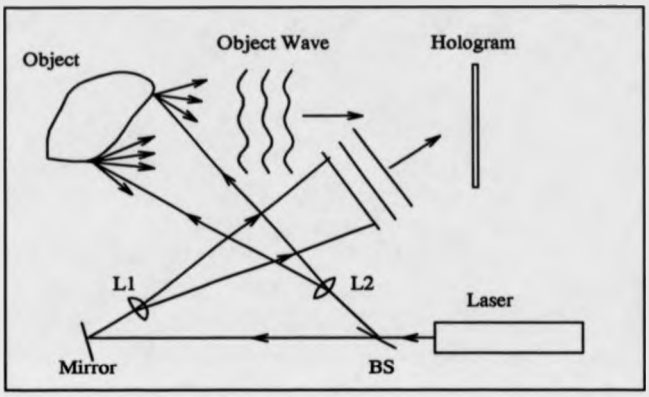
- Three dimensional: The hologram can contain three dimensional information of the object. This is very different from photographic recording which stores an image by photographic density variation alone, disregarding the phase which in practice is the most important information carrier.
- Free of destroy: Any part of the hologram can be used to reconstruct a full three-dimensional image of the original object, but with limited field of vision.
- The actual recording medium is usually a photographic plate, and to store the increased amount of information, the hologram is formed on a high resolution emulsion(1500 lines/mm) with long exposure times in consequence.
- The optical system is very simple, it can be made independent of lenses by using fibre optics.

2.3.3 Fringe Formation in Holography

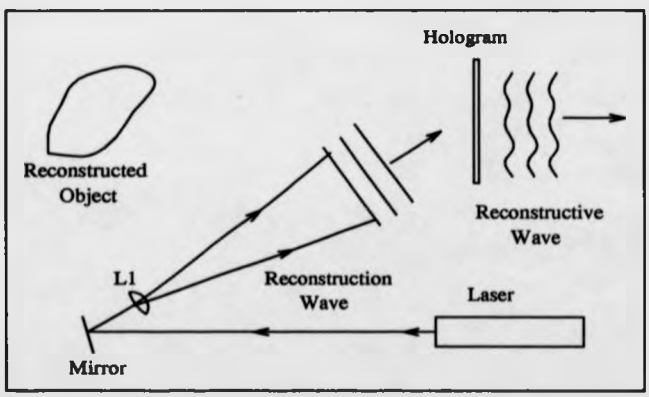
Holography is rather different from the general photographic method, the holographic plate records both the phase and amplitude information which provides a complete description of the light wave. The basic technique of hologram formation is to divide the coherent light coming from a laser into two beams, one to illuminate an object and one to act as a reference. An experimental arrangement which accomplishes hologram formation is shown in Fig. 2.17.

Because a hologram is an interference pattern formed by an object wave and a reference wave, the complex amplitudes of the object and reference waves in the plane of the hologram can be expressed as

$$A_0 = a_0 e^{i\phi_0} \tag{2.26}$$



(a) Hologram formation



(b) Hologram reconstruction

Figure 2.17: Experimental arrangement for hologram formation.

and

$$A_r = a_r e^{i\phi_r} \tag{2.27}$$

where a_0 and a_r are the amplitudes of the object and reference waves, ϕ_0 and ϕ_r are the phases of the object and reference waves.

If these waves are coherent, the two waves interference at the photographic plate and the intensity becomes

$$\begin{aligned} I(x,y) &= |A_0 + A_r|^2 \\ &= a_0^2 + a_r^2 + a_0 a_r e^{i(\phi_0 - \phi_r)} + a_0 a_r e^{-i(\phi_0 - \phi_r)} \end{aligned} \tag{2.28}$$

Two distinct forms of holography can be defined from condition in Eq. 2.28. If $\phi_0 = \phi_r$, then the two waves are traveling along a common axis resulting in the Gabor system of 'in line' holography. When $\phi_0 \neq \phi_r$, the waves move along a separate axis, as in 'off-axis' holography first described by Leith and Upatnieks [14].

The response of photographic materials to exposure to light is normally represented by a curve (known as the Hurter and Driffield, or H-D curve) in which the optical density of the material, after it has been developed and fixed, is plotted against the logarithm of the exposure given to it. Considering the exposure characteristic of photographic emulsion, given by H-D curve [61], the amplitude transmission T can be expressed by

$$T = T_0 + \beta(a_0^2 + a_r^2) + \beta a_0 a_r e^{i(\phi_0 - \phi_r)} + \beta a_0 a_r e^{-i(\phi_0 - \phi_r)} \tag{2.29}$$

where T_0 is a constant field transmission level on the photographic emulsion, β is the coefficient which determines the slope of the straight section in the H-D curve.

Upon illuminating the hologram with the reference wave $A_r = a_r e^{i\phi_r}$, the

distribution of complex amplitudes after its plane is

$$TA_r = [T_0 + \beta(a_0^2 + a_r^2)]a_r + \beta a_0 a_r^2 e^{i\phi_0} + \beta a_0 a_r^2 e^{-i(\phi_0 - 2\phi_r)} \quad (2.30)$$

On the right hand-side of Eq. 2.30, there are three terms. The first term equals the complex amplitude of the reference wave with an accuracy up to the multiplier $T_0+\beta(a_0^2+a_r^2)$ and corresponds to a zero order wave. The second term is identical to the original object wave except for the unimportant multiplicative constants. This wave is traveling in the same direction as the original object wave. The third term is the phase conjugate to the wave $e^{i\phi_0}$. The conjugate wave produces an image lying on the opposite side of the hologram to the illuminating source.

2.3.4 Applications of Holography

Holography has produced a wide variety of possible applications. It has become the powerful measurement tool in scientific research and industrial production. From the practical viewpoint, holographic interferometry appears to be one of the main method in use.

Holographic interferometry is an important diagnostic tool which makes dramatically visible the flow of air through wind tunnels, display the vibration patterns of mechanical components and structures. With the aid of computer fringe analysis, holographic technique can be applied to measure deformation of object down to submicron meter level using carrier fringe and FFT approach [62]. Holographic interferometry is also applied to surface contouring [4, 5, 6].

2.3.4.1 Flow Distribution Measurement by Holography

Holography can be used to measure the particle size and flow distribution very effectively, it is a reasonable method to obtain the quantitative information of the whole flow field [8, 63, 64]. If a high power pulse laser is used as the light source the high speed occurrences can be visualised with a resolution of micrometer. It can be capable of producing a three-dimensional quantitative picture of the flow

by using a scatter plate to produce a diffuse background of illumination.

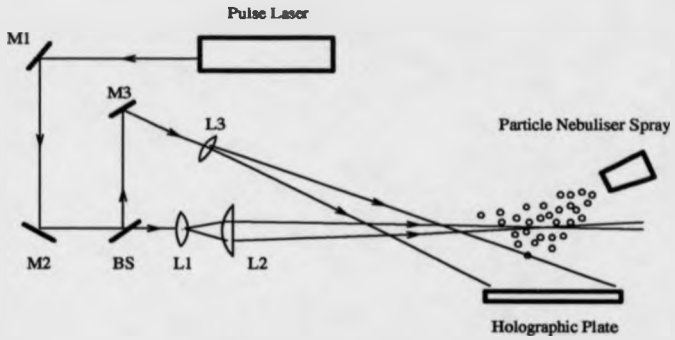


Figure 2.18: Experimental arrangement for recording holograms of particle image velocity.

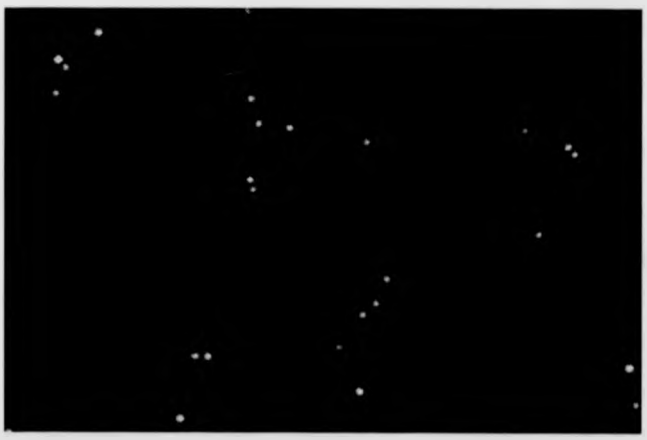


Figure 2.19: Reconstructed holographic particle image obtained by CCD camera using diffraction limited lens (image size 512 × 512).

Figure 2.18 shows the experimental system for recording holograms of particle image velocity. A ruby double pulse laser beam has been divided into two parts. A fraction greater than 99% has been used for the sample beam illumination and passed through a negative spherical and then a cylindrical lens to form a

Figure 2.20: Youngs fringe pattern of holographic particle image obtained by defocus of diffraction limited lens (image size 512 × 512).

light sheet of approximately 50mm height and 2mm wide. The remaining part of the beam was path length matched and used as the holographic reference beam, illuminating the holographic plate directly at the Brewster's angle. Figure 2.19 shows the holographic particle image and Fig. 2.20 presents the Youngs fringes generated in the defocused image. Note that the turbulence of the individual particles can be seen as a variation of the fringe direction.

2.3.4.2 Vibration Studies by Holography

Vibration studies may be aided by the holographic process using one of two important techniques [65, 66, 67]. One of these is known as the time-average method, in which a single exposure hologram is used to record the surface of vibrating object. The other technique is a direct application of real-time holographic interferometry, instead of using a continuous beam for object illumination, a chopped laser beam is employed.

If a hologram is made of an object that is being steadily vibrated, the resulting hologram will have fringes in the image, which indicate the exact amount of

vibration at every point on the object's surface.

The dual reference approach has been applied to the vibration analysis [68]. This new approach to vibration analysis has three advantages: (1) All the information is stored in a single holographic plate and hence vibration data can be recorded and then analysed. (2) The novel optical switching method is employed to record the hologram with the beam sources close together in space. This produces a continuously variable phase difference between the two images of the object. (3) The phase step can be chosen freely at the reconstruction stage and any number of fringe views can be obtained at the required phase step.

2.3.4.3 Holography in Inspection and Testing

The basic inspection instrument with some additional facilities can easily be adapted to non-destructive testing. The principle is to record differences in the response of perfect and imperfect components when subjected to small physical changes, which may result from changes of stress or temperature. Such a system designed specially for the checking of limited panels and motor tires was described by Grant and Brown [69].

A noteworthy advance in the use of direct holographic methods in engineering metrology was provided by Archbold, Burch and Ennos [70] in their instrument designed for the comparison of internal-combustion engine cylinder bores. The potential advantage of the system over existing methods lay in its ability rapidly to show variations at any surface point, thus providing a more than was possible with the point by point check then in use.

2.3.4.4 X-Ray Holography

The use of holographic process as microscope was part of D. Gabor's original motivation for inventing holography [11, 12]. Following D. Gabor's invention, Baez [9] made another step that by recording a hologram with x-rays and reconstructing the image with visible light. At the time Baez made x-ray hologram,

there were no coherent x-ray power of the x-ray tubes and no spatial resolution of photographic emulsion. It was not until the 1980's the undulator x-ray sources on storage rings and high resolution polymer resists as recording media had been used, a lensless microscope of resolution superior to that of the optical microscope had been obtained.

In the experimental schemes for x-ray holography, the x-ray hologram is reconstructed using the visible light in order to get magnification. In this case, aberrations are then introduced which must be considered in seeking to achieve good resolution. Figure2.21 shows two type of x-ray hologram recording and reconstruction [10]

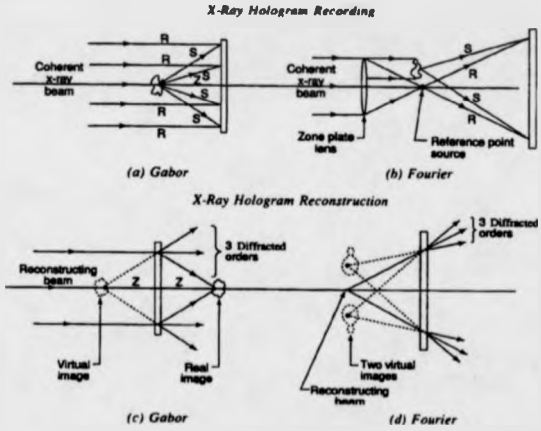


Figure 2.21: Experiment arrangement for x-ray holography; (a) Gabor type x-ray holography; (b) Fourier transform x-ray holography (Howells 1990)

X-ray holography is essentially a diffraction technique and the physics of the interaction of the beam with sample. The distinctive feature is the use of the reference beam to determine the phases of the diffraction pattern. For present-day x-ray imaging experiments, the resolution is several hundred Å and the wavelength 10-50Å, the numerical aperture is in the region 0.01-0.1Å and image is a two-dimensional projection.

In the future, further improvements to the x-ray holography are expected and further applications to biological problems are underway.

2.4 Holographic Interferometry

Holographic interferometry is a powerful technique for measuring deformation, deflection, and vibration. Unlike conventional interferometry, holographic interferometry can be used to make measurements on Lambertian surfaces. The hologram permits a three-dimensional surface to be recorded and investigated from different view points. Holographic interferometry allows an object to be compared with itself between undeformed and deformed states. The comparison produces a fringe pattern depicting surface displacements of the object deformation. It has the desirable feature of being sensitive whole-field, and non-contacting.

Conventional interferometry has limited applications, because wavefronts can only be shaped by optically polished surfaces of the exact form. The components used in conventional measuring interferometers have been restricted to flats. This situation is completely changed by the hologram. The main difference between the holographic interferometry and the conventional interferometry are as follows:

- Holographic interferometry allows measurements to be made on transparent and reflecting objects of arbitrary shape and surface condition.
- With this technique, a complex object can be examined interferometrically from many different view points. This is very useful for observations of three-dimensional particles in holographic particle image velocimetry [71].
- An object can be interferometrically examined at two different times, therefore, one can detect with wavelength accuracy any changes undergone by an object over a period of time.
- The technique does not impose strict requirements on the quality of optical components, which are so severe for conventional interferometers.

There are three different types of holographic interferometry. Three types are called dynamic time average holographic interferometry, static double-exposure holographic interferometry and real-time holographic interferometry.

2.4.1 Dynamic Time Average Holographic Interferometry

Powell and Stetson [65, 66] invented the time-average holographic interferometry and used it for vibration analysis. In dynamic time average holographic interferometry, the object moves continuously during the exposure time. The method for forming a time average hologram of a vibrating surface is identical to that used for forming a hologram when a surface is stationary. The hologram records a time average of the interference pattern intensity produced by the time dependent object complex amplitude at the hologram plane.

If the object surface vibrating angular frequency is ω, amplitude is A_0, along the z direction vibrating, thus the object surface vibrating may be written as

$$U_0 = A_0 exp(irz\cos\omega\ t) \tag{2.31}$$

Equation 2.31 represents the instantaneous vibrating state of the object. If a plane absorption hologram is exposed and processed so that it is a linear recording of the object wave. Illumination by the original reference wave reconstructs the time averaged original object wave given by

$$\bar{U} = \frac{1}{T}\int_0^T U_0 dt = \frac{A_0}{T}\int_0^T exp(irz\cos\omega\ t)dt \tag{2.32}$$

Its amplitude transmittance is proportional to the light intensity $\bar{I}$.

$$\bar{I} = |\bar{U}|^2 \propto |A_0|^2\left|\frac{1}{T}\int_0^T exp(irz\cos\omega\ t)dt\right|^2 \tag{2.33}$$

According to Bessel function, then,

$$exp(ix\cos\theta) = J_0(x) + 2\sum_{n=0}^{\infty} J_n(x)\cos n\theta \tag{2.34}$$

When n=0, Eq. 2.33 can be written as

$$\bar{I} \propto |A_0|^2 J_0^2(rz) \tag{2.35}$$

where, J_0 is the zero-order Bessel function of argument rz. r is the coefficient related to the illumination and viewing direction. When the hologram is reconstructed, incident light angle is θ_r, the viewing angle is θ_v, then, Eq. 2.35 becomes

$$\bar{I} \propto |A_0|^2 J_0^2[rz(\cos\theta_r + \cos\theta_v)] \tag{2.36}$$

Equation 2.36 tells us that the dark fringe in the observed interference pattern of a sinusoidally vibrating surface correspond to the zero of the J_0^2 function, and the bright fringes correspond to the maxima of the function. The fall-off of intensity limits the number of fringes which have sufficient visibility to be seen by an observer. The method has been applied to the study of variety of vibrating objects and can produce greater detail about the antinodal region than previous techniques. An example of vibration hologram generated by computer is shown in Fig. 2.22.

2.4.2 Real Time Holographic Interferometry

The principle of the real-time holographic interferometry is seen as Fig. 2.23. In this method, a hologram is initially exposured with the object in some given strain state. It is necessary within the test to remove the photographic plate from its holder for processing and then return it exactly to its previous location. If the object is still in the initial strain state, and all other conditions are the same, no fringe pattern should be produced. If the strain state is changed, an interference fringe pattern will be generated in the virtual image. Therefore any change in the object produces fringes in real-time.

Several factors have to be considered in the difficulty of real-time holographic

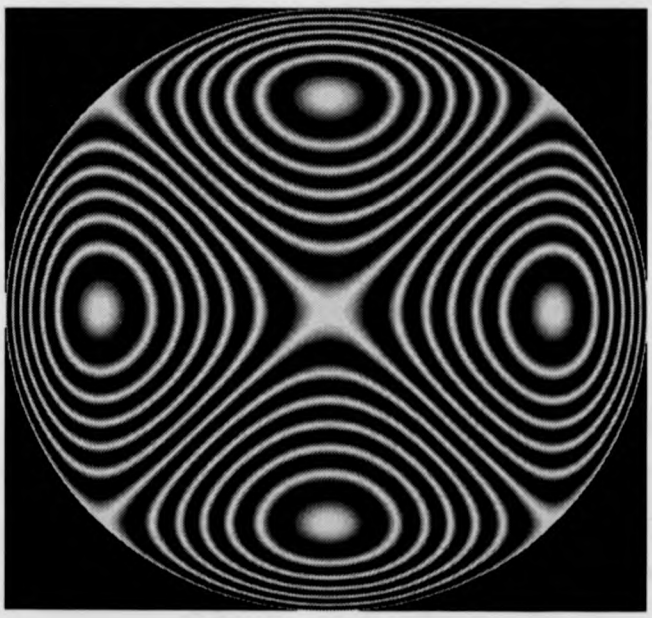


Figure 2.22: An example of vibration hologram generated by computer (image size 512 × 512).

interferometry.

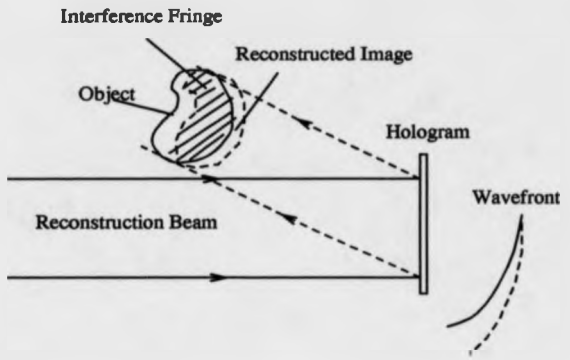


Figure 2.23: Real-time holographic interferometry.

- If the fringe pattern is to be observed or if it is to be characteristic of intended displacement, the object position and its illumination at the time when the processed hologram is replaced must be identical to that when the hologram was formed.
- The emulsion shrinkage accompanying the wet processing of photographic plates causes some unavoidable distortion of the observed interference fringes.
- Until the plate is thoroughly dry, continuing shrinkage may confuse attempts to register the hologram with the original reference beam.

2.4.3 Double-Exposure Interferometry

The double-exposure interferometry is most widely used way of constructing holographic fringe patterns which are used to determine surface deformations on a strained object [72] The method consists of taking a consecutive exposure of two holograms of a single object in different states on the same recording medium as shown in Fig. 2.24. Each exposure is made with the identical reference wave. In double-exposure holographic interferometry, two object wavefronts are recorded

and reconstructed simultaneously and an interferogram is thus obtained that corresponds to the changes of the object during the time between the two exposures. In the case of investigation of dynamic events, the conditions are constantly changing, rapid exposures may be made with a pulsed laser. The double-exposure method carried out with pulsed lasers has been applied with great success to the interferometry of transient events [73]. One of the major advantage of double-exposure method is the determination of the displacement of points on the surface of a strained object. In contrast to real-time holographic interferometry, care no longer has to be taken to reposition the hologram accurately in which it was during the exposure. It is only necessary to ensure stability of the setup in the interval between exposures.

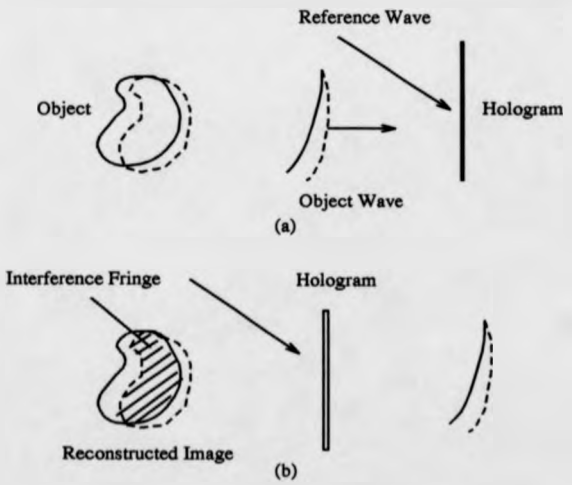


Figure 2.24: Static double-exposure method.

The formation of a fringe pattern in the double-exposure method may be examined in more details as the following. In double-exposure holographic interferometry, if both exposures are made with identical conditions of illumination and have the same duration, then the amplitudes of both reconstructed waves will be the same. The distribution of the illumination is

$$I_1(x,y) = a_{o1}^2 + a_r^2 + A_{o1}A_r^* + A_{o1}^*A_r \tag{2.37}$$

$$I_2(x,y) = a_{o2}^2 + a_r^2 + A_{o2}A_r^* + A_{o2}^*A_r \tag{2.38}$$

where * indicates complex conjugation; A_{oi} is the complex amplitude of the object wave during the i_{th} exposure; a_{oi} is the amplitude of the object wave during the i_{th} exposure; A_r is the complex amplitude of reference wave; a_r is the amplitude of reference wave.

Suppose that the waves being recorded have the same amplitude as the reference wave and differ from each other only in phase. The complex amplitude A of a lightwave from a point source under this condition is written as

$$A_{o1} = a_r exp[i(\phi_{o1} + \delta_1)] \tag{2.39}$$

$$A_{o2} = a_r exp[i(\phi_{o2} + \delta_2)] \tag{2.40}$$

$$A_r = a_r exp(i\phi_r) \tag{2.41}$$

where δ_1 and δ_2 are the additional constant phase, $\phi_{o1} = \frac{2\pi}{\lambda_{o1}} r_{o1}, \phi_{o2} = \frac{2\pi}{\lambda_{o2}} r_{o2}$, r_{oi} is the radial distance between the object and source during the i_{th} exposure. After double-exposure, the total intensity of the illumination is

$$I = 2a_r^2 + (a_{o1}^2 + a_{o2}^2) + (A_{o1} + A_{o2})A_r^* + (A_{o1}^* + A_{o2}^*)A_r \tag{2.42}$$

By illuminating the developed hologram with the reference wave A_r, the complex amplitudes of the waves reconstructed in the +1st order (the virtual image) will be

$$A_{+1} = A_r A_r^*(A_{o1} + A_{o2}) = a_r^2 a_{o1} exp[i(\phi_{o1} + \delta_1)] + a_r^2 a_{o2} exp[i(\phi_{o2} + \delta_2)] \quad (2.43)$$

The intensity of the reconstructed image is determined as

$$I = A_{+1} A_{+1}^* = a_r^4(a_{o1}^2 + a_{o2}^2) + cos[\frac{2\pi}{\lambda}(r_{o1} - r_{o2}) + (\delta_1 - \delta_2)] \quad (2.44)$$

The important quantity is the phase difference $\phi = \phi_{o1} - \phi_{o2} = \frac{2\pi}{\lambda}(r_{o1} - r_{o2})$. Assuming that $\delta_1 - \delta_2 = 0$, the conditions for the formation of the maximum brightness fringes are then governed by the equation

$$\frac{2\pi}{\lambda}(r_{o1} - r_{o2}) = 2n\pi \quad (2.45)$$

or

$$r_{o1} - r_{o2} = n\lambda \quad (2.46)$$

where n is an integer.

For the dark fringes the equation is given by

$$r_{o1} - r_{o2} = \frac{(2n+1)\lambda}{2} \quad (2.47)$$

Figure 2.25 shows a typical double-exposure hologram of the deflection of a cantilever beam. In order to identify the sign of the displacement, the object beam is shifted to generate the carrier fringes. Since the direction of the shifting is known, the sign of the deflection should be determined. Figure 2.26 shows a fringe pattern of carrier modulated by the deflection fringes using the double-exposure method.

It is clear that the double-exposure method gives a comparison between two states of the object in precisely the same way as does the single exposure method, however, the double-exposure method is easier because it avoids the realign-

Figure 2.25: Double-exposure hologram of a deflected cantilever beam with concentrated force (image size 512 × 512).

Figure 2.26: Double-exposure hologram of a deflection cantilever beam with carrier fringes (image size 512 × 512).

ment problem. Multi-exposure hologram [72] can be produced in the same manner, with each exposure representing an incremental change in the object. This method produces very sharp fringes and, hence, leads to accurate quantitative results.

2.5 Electronic Speckle Pattern Interferometry (ESPI)

The speckle effect is a result of illumination of a surface with coherent monochromatic light. The combination of the imaged speckles and reference beam produce a phase referenced speckle fringe pattern related to that object surface displacement. A standard television camera can be used to record the fringe pattern. The fringe pattern is then analysed by the computer.

Electronic speckle pattern interferometry is a real-time technique for many applications such as measurement of small displacements, displacement derivatives, vibration analysis, contouring and non-destructive testing[34, 74, 75, 76, 77] The technique was first demonstrated by Butters and Leendertz[74] . In ESPI, image processing is usually combined with phase-shifting techniques for vibration and deformation measurement[78, 79, 80].

The basic functions of ESPI system is shown in Fig. 2.27[77].

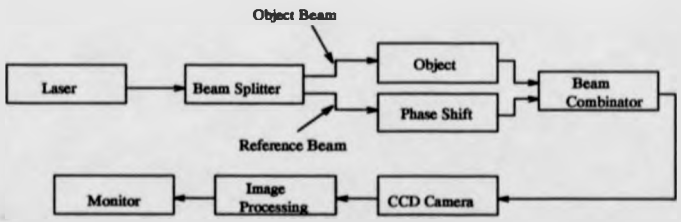


Figure 2.27: Basic functions of ESPI system (Vikhagen 1990).

In the ESPI system above, a laser is used as single coherent optical source to

produce the desired data. The laser beam is split into object and reference beams, each beam passing through its path to the recording plane. The light scattered from object is collected and imaged onto the recording medium. The reference beam is applied directly onto the medium without any modulation. The speckle fringe pattern is captured by a CCD camera and processed in a real-time. The resultant phase map can be displayed on the monitor. The conventional ESPI optical system is shown in Fig. 2.28.

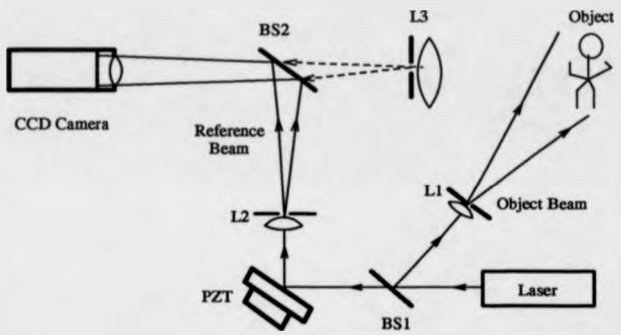


Figure 2.28: Conventional ESPI optical system.

There are a number of different optical systems which are appropriate for ESPI. However, the use of fibre optics in the ESPI system offers the advantages of convenience of alignment and flexibility of the system[34, 81]. Figure 2.29 shows a fibre optics ESPI system. In this system, optical fibres are used to deliver both object and reference beams. An optical fibre can be used to direct light through paths not possible with conventional optics. Using optical fibres, a hand-held ESPI system containing a laser diode and a CCD camera are possible.

In ESPI, the image processing is usually combined with phase shifting techniques. Several techniques have been applied for both vibration and deformation analysis[75, 79, 82]. In the ESPI system shown in Figs. 2.28 and 2.29, the face plate of CCD camera is located in the image plane of the speckle interferometer. The intensity of a given point in the image plane will be I_p where [79]

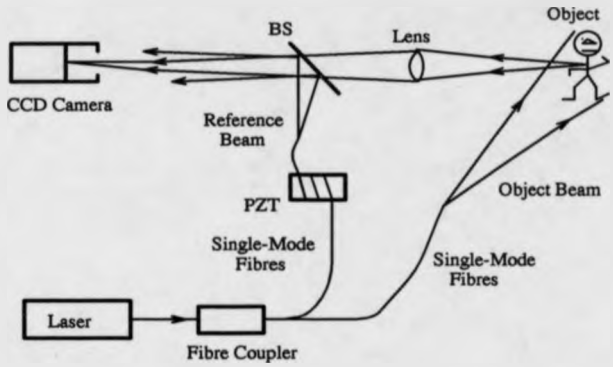


Figure 2.29: ESPI system based on the fibre optics (Sharp 1989).

$$I_p = I_r + I_o + 2\sqrt{I_r\ I_o}\ \cos(\phi_o - \phi_r) \tag{2.48}$$

where I_r and I_o are the reference and object intensity respectively, ϕ_r and ϕ_o are the reference and object phase respectively. The CCD camera should be able to resolve the frequency content of the speckle pattern determined by imaging optics. With the reference beam phase stepping carried out by the PZT, four interfering intensity can be recorded. In this case, two interfering intensities are recorded for the unloaded object and two for the loaded object. The reference phase is changed π from the first interfering intensity to the last intensity. These are given by[79]

$$I_{p1} = I_r + I_o + 2\sqrt{I_r\ I_o}\ \cos(\phi - \pi/2) \tag{2.49}$$

$$I_{p2} = I_r + I_o + 2\sqrt{I_r\ I_o}\ \cos(\phi) \tag{2.50}$$

$$I_{p3} = I_r + I_o + 2\sqrt{I_r\ I_o}\ \cos(\phi + \Delta\phi) \tag{2.51}$$

$$I_{p4} = I_r + I_o + 2\sqrt{I_r\ I_o}\ \cos(\phi + \Delta\phi + \pi/2) \tag{2.52}$$

where $\phi = \phi_o - \phi_r$ and $\Delta\phi = \frac{4\pi d}{\lambda}$, d is the displacement of the surface under

study. From these four interferograms, the phase of surface displacement $\Delta\phi$ can be computed by

$$\Delta\phi = 2 \arctan(\frac{\Delta I_1}{\Delta I_2}) \tag{2.53}$$

where

$$\Delta I_1 = I_{p3} - I_{p2} = -4 \sqrt{I_r I_o} \sin(\phi + \Delta\phi/2) \sin \Delta\phi/2 \tag{2.54}$$

$$\Delta I_2 = I_{p4} - I_{p1} = -4 \sqrt{I_r I_o} \sin(\phi + \Delta\phi/2) \cos \Delta\phi/2 \tag{2.55}$$

The video signal are proportional to the intensities and hence the subtracted signal V_{s1} and V_{s2} are proportional to ΔI_1 and ΔI_2 respectively. These signals have both positive and negative values. Since the negative signals are displyed as black areas by the monitor, to avoid the loss of signal, V_{s1} and V_{s2} are rectified before being displayed on the monitor. The brightness on the monitor is then proportional to $|V_{s1}|$ and $|V_{s2}|$, and the brightness B_1 and B_2 can be expressed as

$$B_1 = 4 K|\sqrt{I_r I_o} \sin(\phi + \Delta\phi/2) \sin(\Delta\phi)/2| \tag{2.56}$$

$$B_2 = 4 K|\sqrt{I_r I_o} \sin(\phi + \Delta\phi/2) \cos \Delta\phi/2| \tag{2.57}$$

where K is the proportionality constant. The brightness is maximum when

$$\Delta\phi = (2 m + 1) \pi \tag{2.58}$$

and minimum when

$$\Delta\phi = 2 m \pi \tag{2.59}$$

There is a fundamental difference in fringe formation between holography and ESPI. With holography, light from the distorted object actually interferes

with the unmodulated reference beam to form the inference fringes. The ESPI interferogram is created through the process of correlation among several video frames rather than wavefront interference.

Another difference between holography and ESPI is the image quality. With holography, the resolution of the recording medium (silver halide film) is approximately 1500 lines/mm. With ESPI, the recording medium is a CCD chip, the resolution is approximately 50 lines/mm. This affects the maximum displacement that the system can measure. In the case of holography, the image quality is much better than that of ESPI. Because there are many speckle noises in a specklegram, details of object's surface are often obscured.

A major disadvantage of holography is that the wet processing of the holographic data is need. This inherently puts a limit on the time of reconstructing holographic data and fringe analysis. On the other hand, ESPI is thousands of times faster than holographic technique, because the specklegram is produced at video rate. This enables speckle pattern correlation fringes to be observed instantaneously, which is often used for on-line inspection of production parts.

Both holography and ESPI have the ability in displacement and shape measurement. This is the one common feature that holography and ESPI shares.

With the advancement of CCD detector chip technology, the image quality of ESPI will be much improved. The improved image quality of ESPI combined with the speed and ease use will play an important role in future industrial measurement.

Chapter 3

Automated Algorithms for Measuring Interference Fringe Patterns

3.1 Introduction

The previous chapter described the optical interference, holography and holographic interferometry. In each case some details of the basic concepts of fringe formation and examples are discussed. This chapter presents state-of-the-art technology in the field of computer aided fringe analysis, with special emphasis on automated algorithms for data reduction. Several different phase-measurement algorithms for calculating the phase of a wavefront from interference fringe data are evaluated and compared. In these methods, the fringe patterns are automatically analysed using electronic and computer technology.

- The most obvious approach to fringe pattern analysis is the fringe tracking or fringe skeleton. However, this approach requires time consuming processing of data including interactive fringe numbering and polynomial fitting procedures to obtain a continuous phase map.
- In the spatial fringe scanning technique, the fringe data are directly taken

from the interferogram. No special devices are required to introduce the spatial carrier, and phases of fringes are calculated not by FFT but by simple correlation integrals.

- The phase-stepping technique is a powerful method for phase-measurement of interferometric fringe patterns. It requires a minimum of two phase stepped images to extract the phase information. High accuracy can be achieved by heterodyne techniques using a frequency shift of reference waves or by phase shifting of the reference waves during reconstruction.
- The Fourier-transform method requires an added spatial carrier frequency. It has the advantage that only one interference fringe pattern is needed during processing, and the interference fringes with a tilted wavefront are analysed with the automatic phase-unwrapping technique, and the background intensity variations and speckle noise can be reduced.

3.2 The Processing of the Interference Fringe Data

There are several kinds of interferometers for generating interferometric fringes. Therefore, because this chapter is concerned mainly with fringe pattern analysis, a commonly used interferometer, the Twyman-Green interferometer, is used to illustrate the formation of the fringe pattern and phase measurement. Figure 3.1 shows the Twyman-Green interferometer for measuring phase using a piezoelectric transducer (PZT) moving a mirror as the phase shifter and a CCD camera to capture and transmit data to a computer for the phase calculation.

The laser beam is expanded to a size matching that of the sample being tested. Part of the laser light is transmitted to the reference surface mounted on the PZT and part of the light is reflected by the beam splitter to the sample surface. Both of the beams are reflected back to the beam splitter where they are combined to form interference fringes. A CCD camera is used to capture interference fringe

patterns. In this case, phase modulation is induced by moving a mirror mounted on the PZT.

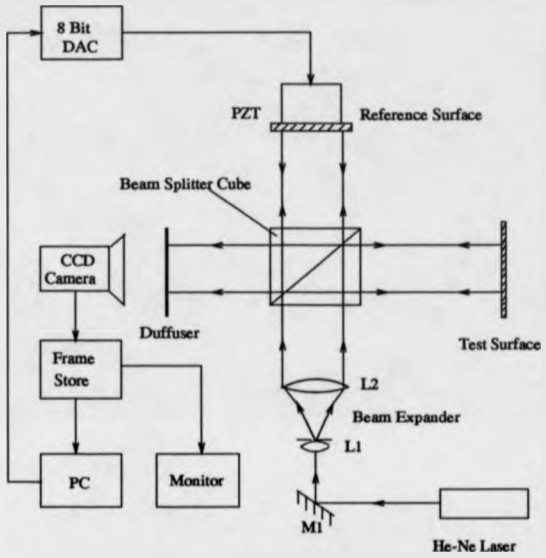


Figure 3.1: Twyman-Green interferometer for testing flat surfaces with PZT phase shift

Suppose the path length difference between the object beam and the reference beam is $\Delta L(x, y)$, the intensity of interference fringes may be expressed as follows,

$$I(x, y) = a(x, y) + b(x, y)\cos[\frac{2\pi}{\lambda}\Delta L(x, y)] \tag{3.1}$$

where λ is the wavelength, and the coefficients $a(x, y)$ and $b(x, y)$ represent bias irradiance and the contrast of fringes, respectively. The interference fringe pattern formed can be used to give a quantitative representation of the phase modulation which has occurred. Some efforts have been made to analyse fringe data by using digital image processing techniques [17].

3.3 The Methods for Phase Measurement of Interference Fringes

3.3.1 The Fringe Tracking Method

The fringe tracking or fringe skeleton method is an intensity identifying process involving binary quantisation, thinning and edge or peak detection. A number of fringe tracking methods have been proposed [18, 83]. Beker and his coworkers [84] suggested a fringe tracking method which is performed by digital filtering and by adaptive thresholding to produce a binary image, then followed by thinning and skeleton routines. Finally, a set of lines representing fringe position will remain by applying these procedures to all image data.

There are two main problems in fringe tracking. The first is that caused by high noise. The second one is created by images with large variations in local contrast. Most fringe tracking routines cannot cope with high noise and varying contrast fringe data without executive static noise operator guidance. High spatial frequency noise can be reduced by a low-pass filtering technique.

During the evaluation of a skeleton of a fringe pattern, it is difficult to distinguish a positive slope from a negative one, and also to consider points where the phase alteration changes sign. Therefore, sign ambiguities can arise. In the fringe tracking technique, interpolation between adjacent fringe positions is necessary, because the phase is determined on the skeleton lines only.

Robinson [85] described another fringe tracking method involving analysis of the fringe field along a limited set of parallel lines across the images. This fringe tracking method is especially suitable for fault detection in holographic nondestructive testing.

This method can be described by the following steps: 1) Counting the number of fringes appearing on several horizontal lines across the full width of the images. When the number of fringes counted reach a threshold on a given scan line, the fault is supposed to exist on that line; 2) Counting the number of fringes in a

short vertical scans until the number of fringes reaches a threshold. This will locate a ring pattern; 3) At the position of the ring pattern, counting the number of fringes along four short vectors angularly spaced by 45^0 until the fringe count in each direction reaches above a threshold, in this case, a fault is detected. By carrying out these procedures on all the image data, finally, each fault is detected.

An alternative fringe tracking technique described by Eichhorn *et al* [83] is to extract lines of equal phase from an interferogram. The method is based on the interpretation of an interferogram as an intensity profile with peaks, valleys and slopes. The electronic and speckle noise is reduced by time averaging and spatial averaging respectively. The regions in the image are classified by consideration of the characteristic neighbourhood relations on local grey level differences. The resulting binary line structures can be thinned to a skeleton of dark and bright fringes and therefore, lines of equal phase are provided by a skeleton of bright and dark fringes. The special feature of this algorithm is that in conjunction with a special processor, it may be used for parallel processing.

3.3.2 Spatial Fringe Scanning Method

Spatial fringe scanning phase measurement was proposed by Toyooka *et al* [86] to measure a 3-D diffuse object. In this method, a known periodic grating is projected on the surface to be measured, and the image of the deformed grating is analysed by computer to determine the profile. During the processing, a deformed grating pattern projected on the object is regarded as a constant spatial carrier frequency.

The intensity distribution of the interference pattern with a tilted wavefront of the object is given by

$$I(x,y) = a(x,y) + b(x,y)\cos[2\pi f_0 x + \Phi(x,y)] \tag{3.2}$$

where $\Phi(x,y)$ is the phase modulation. The coefficients $a(x,y)$ and $b(x,y)$ are slowly varying functions of x and y compared with the spatial carrier frequency

f_0. If the carrier fringes are dense enough to greatly dominate the fringe pattern, $a(x,y)$ and $b(x,y)$ can be considered constant in each of the carrier fringe intervals, and can be represented by a_i and b_i in the i_{th} interval of the carrier. However, the carrier frequency should be less than the Nyquist sampling frequency and approximately N (10-15) times greater than the spatial frequency pattern of the disturbance under test (see Chapter 6 Section 5). The effects of carrier frequency on the interference phase measurement will be discussed in Chapter 6.

Suppose that the phase is linear in the x interval, therefore, the phase may be expressed as

$$\Phi(x_i) = \alpha_i + \beta_i x \tag{3.3}$$

then the intensity over the i_{th} interval of the carrier is given by

$$I(x_i) = a_i + b_i \cos[\alpha_i + (2\pi f_0 + \beta_i)x] \tag{3.4}$$

by applying the orthogonality of trigonometric functions to Eq. 3.4, we get

$$S[I_i] = \int_{i/f_0}^{(i+1)/f_0} I_i(x)\sin(2\pi f_0 x)dx = b_i\left(\frac{1}{4\pi f_0 + \beta_i} - \frac{1}{\beta_i}\right)\sin\bar{\Phi}_i \sin\left(\frac{\beta_i}{2f_0}\right) \tag{3.5}$$

and

$$C[I_i] = \int_{i/f_0}^{(i+1)/f_0} I_i(x)\cos(2\pi f_0 x)dx = b_i\left(\frac{1}{4\pi f_0 + \beta_i} + \frac{1}{\beta_i}\right)\cos\bar{\Phi}_i \sin\left(\frac{\beta_i}{2f_0}\right) \tag{3.6}$$

where

$$\bar{\Phi}_i = \alpha_i + \frac{2i+1}{2f_0}\beta_i \tag{3.7}$$

is the phase value at the middle point of the carrier. The ratio of Eq. 3.5 and Eq. 3.6 is

$$\frac{S[I_i]}{C[I_i]} = -\frac{2\pi f_0}{2\pi f_0 + \beta_i} \tan \bar{\Phi}_i \tag{3.8}$$

If the carrier frequency f_0 is sufficiently large, the coefficient $-\frac{2\pi f_0}{2\pi f_0+\beta_i}$ in Eq. 3.8 becomes unity, therefore, the phase value Φ_i is calculated by

$$\bar{\Phi}_i = -\arctan\left(\frac{S[I_i]}{C[I_i]}\right) \tag{3.9}$$

According to Eq. 3.9, the phase over every interval of the carrier is successively determined along a scanning line. The feature of this method is that interference fringes with a tilted wavefront are spatially scanned at right angles to the carrier fringes and analysed not by FFT but by simple correlation integrals. This will reduce the computation time dramatically in getting phase information.

In the above analysis, the phase $\bar{\Phi}_i$ is assumed to be linear in the interval of the carriers and the carrier frequency f_0 could be fine enough. Once the spatial frequency f_0 is determined, the functions $\sin(2\pi f_0 x)$ and $\cos(2\pi f_0 x)$ in the integrals of Eqs. 3.5, 3.6 are generated by the computer. Then the phase value of each point is calculated through Eq. 3.9.

3.3.3 Synchronous Phase Detection

3.3.3.1 Temporal Synchronous Detection

The temporal synchronous detection [86] of interferometric phase measurement can be obtained by shifting the reference phase of an interferometer linearly in time. The sinusoidal signal occurring at each point in the interferogram due to the reference shift can be written as

$$I(x,y,t) = a(x,y) + b(x,y)\cos\{2\pi[\omega t - \Phi(x,y)]\} \tag{3.10}$$

where $\Phi(x,y) = \frac{2\pi}{\lambda}\Delta L$ is the phase value at each point in the interferogram. To detect the phase value $\Phi(x,y)$ synchronously, it is correlated with the sinusoidal and cosinoidal signal of the same frequency and averaged over many periods of

oscillation. Then, the phase $\Phi(x, y)$ can be measured using the following formula.

$$C = \int_{-\infty}^{\infty} I(t)\cos(\omega t)dt = (b/2)\cos(\Phi) \tag{3.11}$$

$$S = \int_{-\infty}^{\infty} I(t)\sin(\omega t)dt = (b/2)\sin(\Phi) \tag{3.12}$$

Using Eqs 3.11 and 3.12, the phase $\Phi(x, y)$ can be calculated by

$$\Phi(x, y) = \arctan\left(\frac{S}{C}\right) \tag{3.13}$$

when discrete steps are used, and N measurements are equally spaced over one modulation period. The phase shift is

$$\delta_i = \frac{i2\pi}{N} \qquad i = 1, 2, 3, ...N \tag{3.14}$$

In a similar way to the integrated case, the intensity of interference fringes can be expressed using Fourier series

$$C = \sum_{i=1}^{N} I_i \cos(2\pi\frac{i}{N}) = \frac{b}{2}\cos\Phi \tag{3.15}$$

$$S = \sum_{i=1}^{N} I_i \sin(2\pi\frac{i}{N}) = \frac{b}{2}\sin\Phi \tag{3.16}$$

$$\Phi = \arctan\left(\frac{S}{C}\right) = \arctan\left(\frac{\sum_{i=1}^{N} I_i(x, y)\sin\delta_i}{\sum_{i=1}^{N} I_i(x, y)\cos\delta_i}\right) \tag{3.17}$$

3.3.3.2 Spatial Synchronous Detection

The spatial synchronous detection [86] of interferometric phase measurement provides a facility for the introduction of 'tilt' to the interferogram. In practice, this is usually accomplished by angular displacement of a mirror in the interferometer, which in turn, causes the fringe pattern to rotate/translate in the field of view.

Womack [87] has described a family of three spatial synchronous detection

phase measurement techniques. These are the quadrature multiplicative moire, sinusoidal window and complex exponential window algorithms. All three phase measurement methods involves summing over products of the form

(fringe pattern) × (reference term) × (filter envelope)

with the major difference in the argument of the reference term and filter transfer function.

(1) The quadrature multiplicative moire technique involves multiplying the fringe pattern $I(x,y)$ by a sinusoidal reference $R(x,y)$ then convolving the product $I(x,y)R(x,y)$ with a window function $H(x,y)$. The resulting low frequency moire can be written

$$\begin{aligned} M_1(x,y) &= [I(x,y)R(x,y)] \otimes H(x,y) \\ &= \frac{B}{2}\cos 2\pi[w(x,y) - w_r(x,y)] \end{aligned} \tag{3.18}$$

where $w(x,y)$ and $w_r(x,y)$ indicate unknown and known surface deviations. $\otimes$ denotes convolution. $I(x,y)$ and $R(x,y)$ are as follows:

$$I(x,y) = A(x,y) + B(x,y)\cos 2\pi w(x,y) \tag{3.19}$$

$$R(x,y) = \cos 2\pi w_r(x,y) \tag{3.20}$$

A second low spatial frequency moire is obtained by shifting a reference beam in phase by 90^0 with respect to the first.

$$\begin{aligned} M_2(x,y) &= [I(x,y)R(x,y)] \otimes H(x,y) \\ &= \frac{B}{2}\sin 2\pi[w(x,y) - w_r(x,y)] \end{aligned} \tag{3.21}$$

Phase is computed by taking the ratio of Eqs. 3.19 and 3.22

$$w(x,y) = w_r(x,y) + \frac{1}{2\pi}\arctan\frac{M_2(x,y)}{M_1(x,y)} \tag{3.22}$$

(2) In the sinusoidal window technique, the reference fringe is made a part of the window envelope. The double sideband high pass filter is used to obtain the side lobes. Phase is computed directly from the two filter outputs.

(3) The complex exponential window involves using a single sideband high pass filter in the spatial domain. The desired phase information is present in a single side lobe. The phase is computed from the ratio of real and imaginary parts of the filter output.

The advantages of the spatial synchronous detection are as follows:

- Only a single frame of interference data is needed.
- Phase shift elements such as a piezoelectric translator or an electron-optic crystal are not required.
- Since the method does not use the Fourier transform, computation times can be reduced by as much as a factor of 5 [88].

However, spatial synchronous detection requires a high fringe density and therefore a high resolution detector. The error of the measurement in this method is influenced by the performance of the digital filter to extract phase information.

3.3.4 Heterodyne Holographic Method

In interferometry, a single interferogram does not always contain sufficient information to allow automatic evaluation of the parameter to be measured. Ambiguities may arise due to a lack of explicit phase information. Therefore, heterodyne interferometers are specially designed to extract phase information and have been used in a wide variety of interferometric techniques [19, 20, 89, 90].

The heterodyne method applied to holographic interferometry has greatly improved the ability of holography for high resolution interference fringe interpolation. Levels of accuracy of 1/1000 of a fringe are achieved by this method [19].

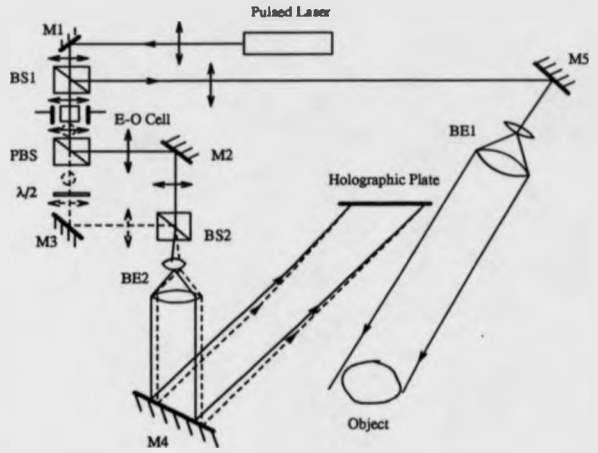


Figure 3.2: Heterodyne dual reference holographic interferometry using electro-optic cell for high speed recording.

However, this method requires sophisticated electronic equipment and mechanical scanning of the image by photodetectors. In addition to optical components used in conventional holography, heterodyne holography requires methods to generate the desired frequency offset, to detect the modulated signals, and to measure their relative phase change.

Wagner [20] proposed an arrangement for heterodyne holographic analysis (see Fig. 3.2). In this arrangement, two holographic exposures are recorded each with its own angularly separate reference beam. The laser output is a linearly polarised beam. The beam is divided into reference and object beams by means of a beam splitter (BS1). An electro-optic (E-O) cell is used to rotate the polarity of the light from the laser source. A polarising beam splitter (PBS) is used to route the reference beam along separate paths according to its polarity as indicated by solid and dashed double arrows. Between exposures, the reference beam can be switched from one path to another by switching the E-O cell. During the first exposure, the object beam reflected from BS1, directed by mirror M5 via beam expander BE1, illuminates the object being studied. The reference beam

passes through the E-O cell while the E-O cell is switched off. The reference beam reflected by the polarising beam splitter PBS, directed by mirror M2 via beam expander BE2, illuminates the holographic plate. By switching on the E-O cell, the polarity of the reference beam is rotated as indicated by the dashed circle. The reference beam is transmitted by PBS. Before the beam reaches the mirror M3, a half-wave plate is used to correct the polarity of the light rotated by the E-O cell after it has passed through the PBS. The reference beam reflected by beam splitter BS2, via beam expander BE2, illuminates the holographic plate. Therefore, dual reference double-exposure hologram can be recorded for heterodyne analysis.

3.3.5 Quasi-Heterodyne (Phase-Stepping) Interferometry Method

The quasi-heterodyne or phase-stepping method is well established in the field of holographic and speckle interferometry [19, 21, 22, 23, 24, 90]. The technique is similar to the phase heterodyne technique, but allows electronic scanning of the image by CCD or TV cameras and uses microcomputer controlled digital phase evaluation. This makes the instrumentation much simpler and allow direct interfacing to a computer image processing system. The technique allows one to measure the interference phase with an accuracy of 1/100 of a fringe at any point in the TV image [19]. The following shows the different algorithms for phase-stepping method of phase measurements.

3.3.5.1 General Phase Measurement

Since there are three unknowns in the interference equation, a minimum of three measurements is necessary. The fringe intensity at any point is

$$I = I_0[1 + \gamma_0 \cos(\Phi)] \tag{3.23}$$

where I_0 is the d.c. intensity, γ_0 the modulation of the interference fringes, and

Φ the phase value. For the general technique, the phase shift is linearly ramped during the detector integration time. The detector array will integrate the fringe intensity data over some change in relative phase of Δ.

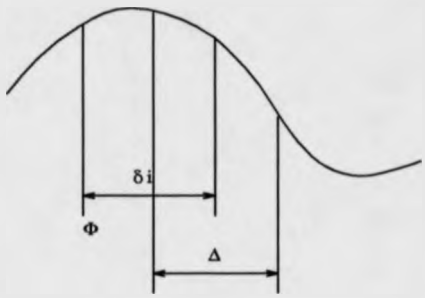


Figure 3.3: Averaged signal measurements with the integrating phase shifting method

As shown in Fig. 3.3, the average measurement is taken from $\delta = \delta_i - \frac{\Delta}{2}$ to $\delta = \delta_i + \frac{\Delta}{2}$, with centre value δ_i. Corresponding to reference phase δ_i, the average value of the irradiance is given by [21, 24, 90]

$$I_i(x,y) = \frac{1}{\Delta}\int_{\delta_i-\frac{\Delta}{2}}^{\delta_i+\frac{\Delta}{2}} I_0(x,y)\left\{1+\gamma_0\cos[\Phi(x,y)+\delta(t)]\right\}d\delta(t) \tag{3.24}$$

After integrating this expression, the recorded intensity is

$$I_i(x,y) = I_0(x,y)\left\{1+\gamma_0 \mathrm{sinc}(\Delta/2)\cos[\Phi(x,y)+\delta_i]\right\} \tag{3.25}$$

where

$$\mathrm{sinc}(\Delta/2) = \frac{\sin(\Delta/2)}{\Delta/2} \tag{3.26}$$

where δ is the average value of the relative phase shift for the i_{th} exposure. In the case of stepped phase shift, the sinc function would have a value of unity ($\Delta = 0$). Therefore, phase-stepping is a simplification of the integrating-bucket

method. Equation 3.25 can be rewritten as

$$I_i(x,y) = a(x,y) + b(x,y)\cos[\Phi(x,y)]\cos\delta_i - b(x,y)\sin[\Phi(x,y)]\sin\delta_i \quad (3.27)$$

where

$$\begin{aligned} a(x,y) &= \quad I_0(x,y) \\ b(x,y) &= I_0(x,y)\gamma_0 \mathrm{sinc}(\Delta/2) \end{aligned}$$

By using a least-squares technique, the unknown phase can be calculated for a total of N recorded intensity measurements $\hat{I}_i(x,y)$. The criterion for shifting N times is given by

$$e = \sum_{i=1}^{N} (I_i - \hat{I}_i)^2 = \sum_{i=1}^{N} (a + b\cos\Phi\cos\delta_i - b\sin\Phi\sin\delta_i - \hat{I}_i)^2 \quad (3.28)$$

To find three unknowns a, $b\cos\Phi$, and $b\sin\Phi$, that minimise the criterion e, the linear equations in matrix form are given by

$$\mathbf{A}(\delta_i)\mathbf{a} = \mathbf{B}(\delta) \quad (3.29)$$

where $\mathbf{A}, \mathbf{a}$ and $\mathbf{B}$ can be expressed by the following matrix.

$$\mathbf{A} = \begin{pmatrix} N & \sum_{i=1}^{N}\cos\delta_i & \sum_{i=1}^{N}\sin\delta_i \\ \sum_{i=1}^{N}\cos\delta_i & \sum_{i=1}^{N}\cos^2\delta_i & \sum_{i=1}^{N}\cos\delta_i\sin\delta_i \\ \sum_{i=1}^{N}\sin\delta_i & \sum_{i=1}^{N}\cos\delta_i\sin\delta_i & \sum_{i=1}^{N}\sin^2\delta_i \end{pmatrix} \quad (3.30)$$

$$\mathbf{B} = \begin{pmatrix} \sum_{i=1}^{N}\hat{I}_i \\ \sum_{i=1}^{N}\cos\Phi \\ \sum_{i=1}^{N}\sin\Phi \end{pmatrix} \quad (3.31)$$

$$\mathbf{a} = \begin{pmatrix} \sum_{i=1}^{N} a \\ \sum_{i=1}^{N} b\cos\Phi \\ \sum_{i=1}^{N} b\sin\Phi \end{pmatrix} \tag{3.32}$$

If the matrix **A** is well conditioned, then a matrix equation for the three unknowns results

$$\mathbf{a} = \mathbf{A}^{-1}(\delta_i)\mathbf{B}(\delta_i) \tag{3.33}$$

Because the matrix **A** depends only on the phase shift, it can be calculated and inverted just once. By evaluating the value of **B** at each point of intensity measurement and solving for the coefficients $b\cos\Phi$ and $b\sin\Phi$, the phase at each point in the interferogram is determined by

$$\Phi(x,y) = \arctan\left(\frac{b\sin[\Phi(x,y)]}{b\cos[\Phi(x,y)]}\right) \tag{3.34}$$

In practice, for most phase shifters, such as PZT, there exists a repeatability problem from hysteresis, nonlinearity, and temperature linear drift. The effective reference phase in the measurements is also affected by the surrounding disturbance. Lai and Yatagai [25] proposed a method which evaluates the actual phase difference between the interfering beams at the same time as the data are read.

3.3.5.2 Phase-Stepping Measurement

(1) Single step method

Kerr *et al* [91] have described a new approach utilising a single-phase-step technique to extract data from static and dynamic fringe patterns. This new technique significantly improved the environmental stability and reduced computational effort. It has been shown that there is no significant difference between single-phase-step and three, four or five steps, and the new technique offers important practical advantages.

The fundamental equation describing the intensity distribution in an interfer-

ogram is given by [91]

$$I(x,y) = I_1(x,y) + I_2(x,y) + 2\sqrt{I_1(x,y)I_2(x,y)}\cos\theta(x,y) \tag{3.35}$$

where $I_1(x,y)$ and $I_2(x,y)$ are the object and reference beam intensity, and $\theta(x,y)$ is the phase angle depending on the object speckle and the reference beam geometry. After the object has been displaced, the corresponding intensity equation is given by

$$\bar{I}(x,y) = I_1(x,y) + I_2(x,y) + 2\sqrt{I_1(x,y)I_2(x,y)}\cos(\theta(x,y) + \Phi(x,y)) \tag{3.36}$$

where the phase $\Phi(x,y)$ describes the information about object displacement. Assuming that during the displacement of the object, the angle $\theta(x,y)$ remains unchanged.

Subtracting Eq. 3.35 from Eq. 3.36 gives

$$I_{sub} = I - \bar{I} = 2\sqrt{I_1 I_2}[\cos\theta - \cos(\theta + \Phi)] \tag{3.37}$$

Since Eqs. 3.37 has only two unknowns, the term $\sqrt{I_1 I_2}$ and the phase Φ, only two equations are needed to solve the phase value $\Phi(x,y)$. In convenience, the values of the irradiance are measured using two difference values of the phase $0, -\pi/2$. In this case, the following two values of the irradiance are obtained, $I_{(0)}$ and $I_{-\pi/2}$.

$$I_{(0)} = C\cos(\theta) - C\cos(\theta + \Phi) \tag{3.38}$$

$$I_{-\pi/2} = C\cos(\theta) - C\cos(\theta + \Phi - \frac{\pi}{2}) \tag{3.39}$$

where C is the value $2\sqrt{I_1 I_2}$. From Eqs. 3.38, 3.39, the phase Φ can be calculated by

$$\Phi = \arctan \frac{I_{-\pi/2}}{I_{(0)}} \tag{3.40}$$

Figure 3.4 shows the intensity distribution of two interferograms corresponding to I_0 and $I_{-\pi/2}$. Since the argument in Eq. 3.40 is modulo 2π, the phase values have to be unwrapped in order to generate a continuous phase. Figure 3.5(a) is the wrapped phase value obtained by using Eq. 3.40. After phase-unwrapping, the continuous phase value is shown in Fig. 3.5(b).

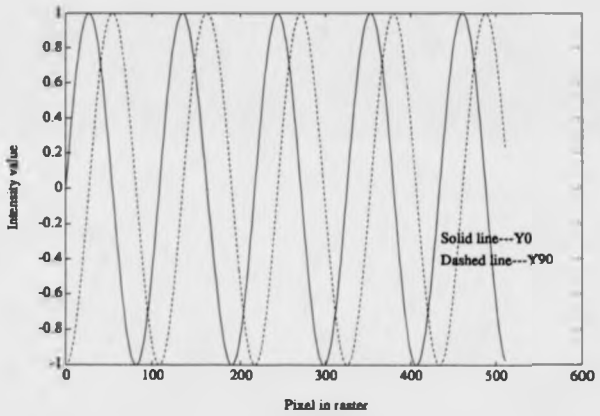


Figure 3.4: Intensity distribution of two interferograms

(2) Three step method

As there are three unknowns in Eq. 3.23, three interferograms of different reference beam phase are required. With three interferograms recorded at three different reference beam phases, the phase value can be calculated for each point on the interferogram from the three equations as

$$\Phi(x,y) = \arctan\left(\frac{(I_3 - I_2)\cos\delta_1 + (I_1 - I_3)\cos\delta_2 + (I_2 - I_1)\cos\delta_3}{(I_3 - I_2)\sin\delta_1 + (I_1 - I_3)\sin\delta_2 + (I_2 - I_1)\sin\delta_3}\right) \tag{3.41}$$

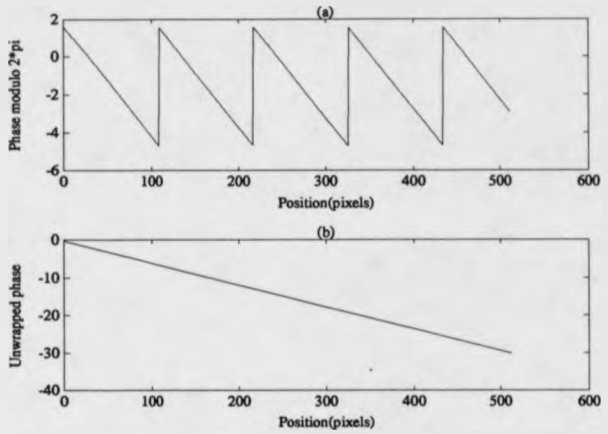


Figure 3.5: (a) Phase modulo 2π; (b) unwrapped phase value.

where δ_1, δ_2 and δ_3 refer to the average value of the relative phase shift for the each interferogram.

In the case of phase shift with $\delta_i = \frac{\pi}{4}, \frac{3\pi}{4}$ and $\frac{5\pi}{4}$. The three intensity measurements can be expressed as

$$\begin{aligned} I_1(x,y) &= I_0(x,y)\left\{1+\gamma\cos[\Phi(x,y)+\tfrac{\pi}{4}]\right\} \\ I_2(x,y) &= I_0(x,y)\left\{1+\gamma\cos[\Phi(x,y)+\tfrac{3\pi}{4}]\right\} \\ I_3(x,y) &= I_0(x,y)\left\{1+\gamma\cos[\Phi(x,y)+\tfrac{5\pi}{4}]\right\} \end{aligned} \tag{3.42}$$

The phase at each point in the interferogram is then simplified as

$$\Phi(x,y) = \arctan\left(\frac{I_3(x,y)-I_2(x,y)}{I_1(x,y)-I_2(x,y)}\right) \tag{3.43}$$

If the corresponding phase shifts are $\frac{2\pi}{3}$, 0, and -$\frac{2\pi}{3}$, the three intensity measurements become

$$\begin{aligned} I_1(x,y) &= I_0(x,y)\left\{1+\gamma\cos[\Phi(x,y)+\tfrac{2\pi}{3}]\right\} \\ I_2(x,y) &= I_0(x,y)\left\{1+\gamma\cos[\Phi(x,y)]\right\} \\ I_3(x,y) &= I_0(x,y)\left\{1+\gamma\cos[\Phi(x,y)-\tfrac{2\pi}{3}]\right\} \end{aligned} \tag{3.44}$$

The three interferograms (I_1, I_2, I_3) may be obtained by shifting the reference beam by $2\pi/3, 0, -2\pi/3$ as shown in Fig. 3.6.

The phase value $\Phi(x,y)$ is computed using the formula

$$\Phi(x,y) = \arctan\left(\frac{\sqrt{3}(I_3 - I_1)}{2I_2 - I_1 - I_3}\right) \tag{3.45}$$

Figure 3.7(a) nad (b) are the modulo 2π and unwrapped phase value for the three step method.

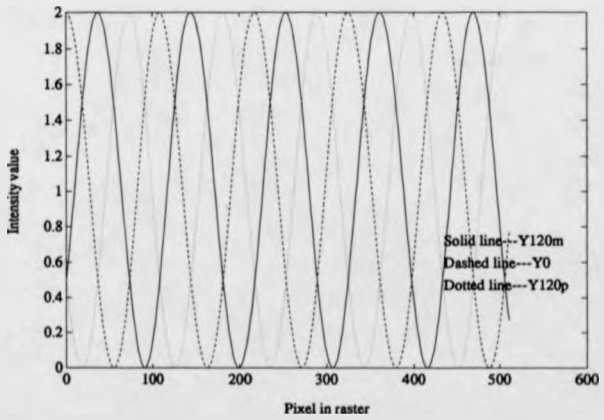


Figure 3.6: Intensity distribution of three interferograms

(3) Four step method

The four step method utilises four successive intensity data frames with the reference phase shifts between each frame. The phase steps of multiples of $\pi/2$ are convenient to achieve. When the phase shifts are introduced as $0, \pi/2, \pi, 3\pi/2$ within one period of the fringe, the phase $\Phi(x,y)$ at each detecting point (x,y)

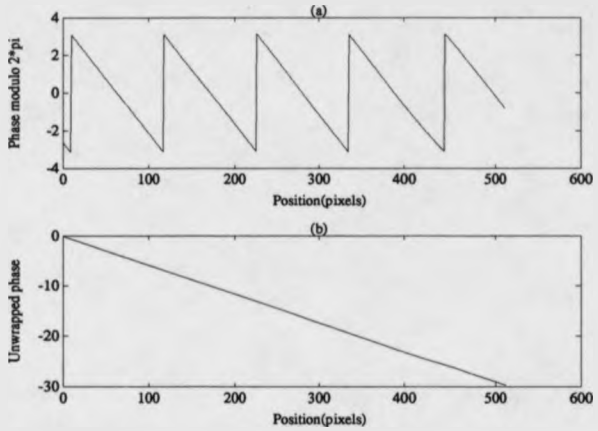


Figure 3.7: (a) Phase modulo 2π; (b) unwrapped phase value.

in the interferogram can be obtained by the relation

$$\Phi(x,y) = \arctan\left\{\frac{I_4(x,y) - I_2(x,y)}{I_1(x,y) - I_3(x,y)}\right\} \tag{3.46}$$

where I_1, I_2, I_3 and I_4 denote the intensities with additional phase shifts as expressed below:

$$\begin{aligned}
I_1(x,y) &= I_0(x,y)\{1 + \gamma\cos[\Phi(x,y)]\} \\
I_2(x,y) &= I_0(x,y)\{1 + \gamma\cos[\Phi(x,y) + \pi/2]\} = I_0(x,y)\{1 - \gamma\sin[\Phi(x,y)]\} \\
I_3(x,y) &= I_0(x,y)\{1 + \gamma\cos[\Phi(x,y) + \pi]\} = I_0(x,y)\{1 - \gamma\cos[\Phi(x,y)]\} \\
I_4(x,y) &= I_0(x,y)\{1 + \gamma\cos[\Phi(x,y) + 3/2\pi]\} = I_0(x,y)\{1 + \gamma\sin[\Phi(x,y)]\}
\end{aligned} \tag{3.47}$$

Figure 3.8 shows the intensity distribution of four interferograms obtained by shifting the reference beam by $0, \pi/2, \pi$, and $3\pi/2$ respectively. The phase value modulo 2π obtained by using Eq. 3.46 is shown in Fig. 3.9(a). The unwrapped phase value is shown in Fig. 3.9(b).

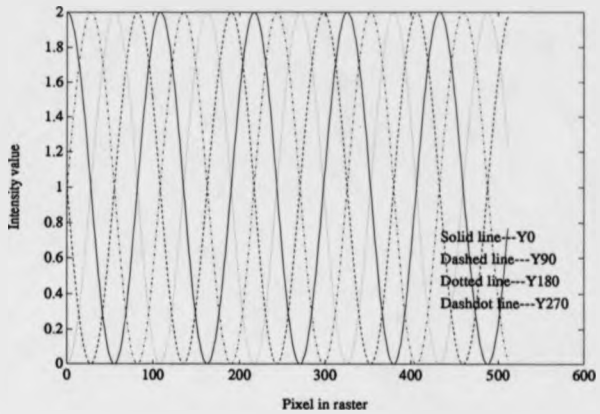


Figure 3.8: Intensity distribution of four interferograms

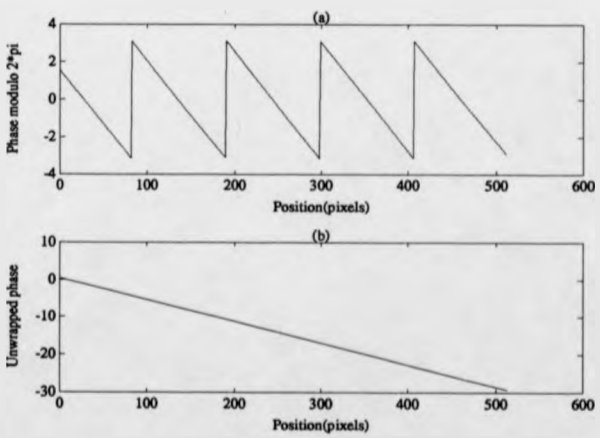


Figure 3.9: (a) Phase modulo 2 π; (b) unwrapped phase value.

(4) Five step method [92]

The five step approach was intended to accomplish an autocalibration of the piezoelectric motor used for phase-stepping in conventional interferometry. In this method, five interferograms are used for calculating the phase value $\Phi(x, y)$. Figure 3.10 shows the intensity distribution of five interferograms obtained by shifting the reference beam by $\pi/3$ for each interferogram.

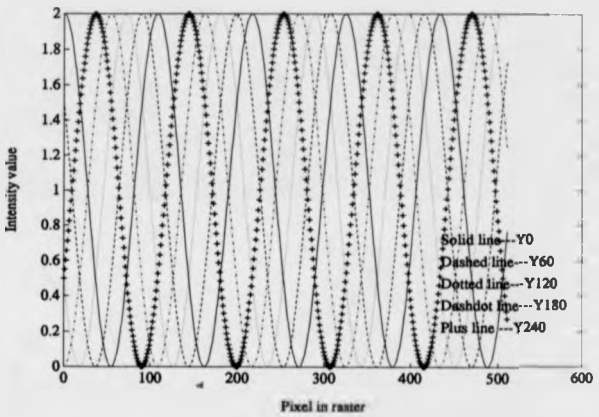


Figure 3.10: Intensity distribution of five interferograms

The expression for phase value $\Phi(x, y)$ in this case corresponding to the five intensities is given by

$$\Phi(x, y) = \frac{1}{2\pi} \arctan \frac{2(I_2 - I_4)}{2I_3 - I_1 - I_5} \tag{3.48}$$

Figure 3.11(a) and (b) show the results for the modulo 2π and unwrapped phase value respectively.

Figure 3.12 shows the unwrapped phase value computed by four different phase-stepping methods. It has been shown that from the above examples four different phase-stepping algorithms are quite suitable for the phase measurement. However, to reduce small fluctuations in the computed phase value, it is necessary

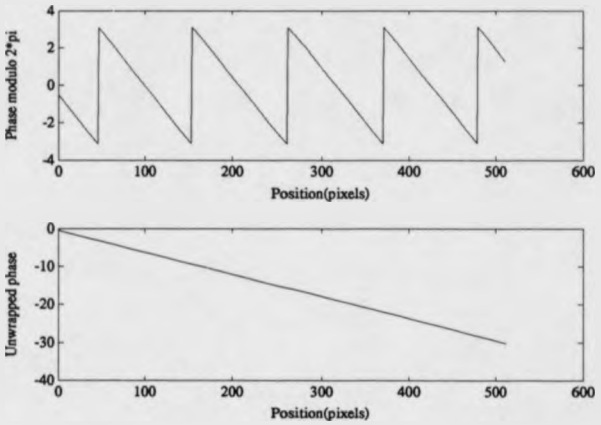


Figure 3.11: (a) Phase modulo 2 π; (b) unwrapped phase value.

to choose the algorithm of three step and four step.

(5) Any phase shift methods (Carre methods) [22, 93]

If the phase shift α is constant over all (x, y), the four stepped interferograms I_1, I_2, I_3 and I_4 are recorded. These may be written in the following way

$$\begin{aligned} I_1(x,y) &= I_0(x,y)\left\{1+\gamma\cos[\Phi(x,y)-\tfrac{3\alpha}{2}]\right\} \\ I_2(x,y) &= I_0(x,y)\left\{1+\gamma\cos[\Phi(x,y)-\tfrac{\alpha}{2}]\right\} \\ I_3(x,y) &= I_0(x,y)\left\{1+\gamma\cos[\Phi(x,y)+\tfrac{\alpha}{2}]\right\} \\ I_4(x,y) &= I_0(x,y)\left\{1+\gamma\cos[\Phi(x,y)+\tfrac{3\alpha}{2}]\right\} \end{aligned} \tag{3.49}$$

Suppose the phase shift is linear. From these equations, the measuring phase distribution $\Phi(x, y)$ at each point can be calculated by

$$\Phi(x,y) = \arctan\left\{\frac{\sqrt{[(I_1-I_4)+(I_2-I_3)][3(I_2-I_3)-(I_1-I_4)]}}{[(I_2-I_3)-(I_1+I_4)]}\right\} \tag{3.50}$$

The main advantage of this algorithm is that the phase calculated using

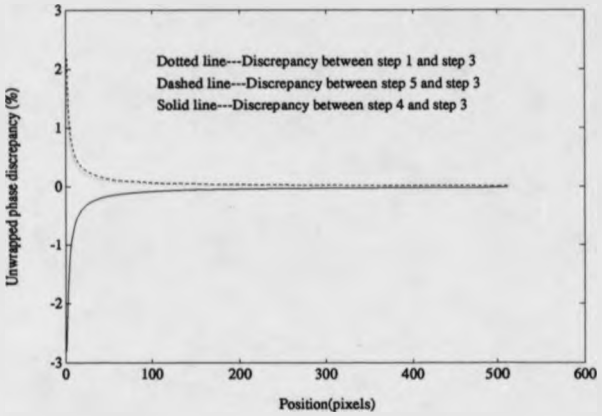


Figure 3.12: Comparison of four different phase step methods for unwrapped phase value calculation.

Eq. 3.49 is independent of the phase shift of the reference beam. Therefore, the need for PZT calibration can be eliminated. Also, it is easier to take data using a phase shifting method since the phase of the reference wave varies linearly with time rather than in discrete steps.

3.3.6 Fourier Transform Method

The Fourier transform analysis based on the FFT algorithm has been demonstrated to be a more suitable method for automated fringe pattern analysis [26, 27, 28, 62]. This technique has the advantage that only a single frame of irradiance data is required and phase-shift elements such as a PZT or an electro-optic crystal are not needed. It also has the advantage that background intensity variations and speckle noise can be reduced.

3.3.6.1 The Principle of the Fourier Transform

The Fourier transform pair is defined as [94]

$$G(u) = \int_{-\infty}^{\infty} g(x)e^{-j2\pi ux} \tag{3.51}$$

$$g(x) = \int_{-\infty}^{\infty} G(u)e^{j2\pi ux} \tag{3.52}$$

Equations 3.51 and 3.52 can be shown to exist if $g(x)$ is continuous and integrable and $G(u)$ is integrable. These conditions are almost satisfied in practice. Similarly, the Discrete Fourier Transform (DFT) is defined as

$$G(n) = \frac{1}{N}\sum_{k=0}^{N-1} g(k)e^{(-j2\pi nk)/N} \tag{3.53}$$

for $n = 0, 1, ...N-1$

$$g(k) = \sum_{n=0}^{N-1} G(n)e^{(j2\pi nk)/N} \tag{3.54}$$

for $n = 0, 1, ...N-1$

Notice that both Eq. 3.53 representing the transform and Eq. 3.54 representing the inverse transform are quite similar mathematically.

A two dimensional function $g(x, y)$ has a two dimensional Fourier transform $G(u, v)$ by

$$G(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(x,y)e^{-j2\pi(ux+vy)}dxdy \tag{3.55}$$

The two dimensional inverse Fourier transform is given by

$$g(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} G(u,v)e^{j2\pi(ux+vy)}dudv \tag{3.56}$$

Equations 3.55 and 3.56 are analogous to the one dimensional case. The two dimensional Fourier transform $G(u, v)$ can be expressed as successive one dimensional transforms.

Similar to the one dimensional case, the two dimensional discrete Fourier transform is defined as

$$G(u,v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} g(x,y) e^{-j2\pi(\frac{ux}{M}+\frac{vy}{N})} \tag{3.57}$$

for $u = 0, 1, 2.....M-1$ and $v = 0, 1, 2.....N-1$

$$g(x,y) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} G(u,v) e^{j2\pi(\frac{ux}{M}+\frac{vy}{N})} \tag{3.58}$$

for $u = 0, 1, 2.....M-1$ and $v = 0, 1, 2.....N-1$

As mentioned above, the two dimensional discrete Fourier transform can be implemented straightforwardly by computing the one dimensional discrete Fourier transforms.

In practice, the Fourier transform procedure is often performed by the FFT algorithm. This increases the computation efficiency dramatically. The FFT algorithm is simply a procedure for factoring an $N \times N$ matrix into γ matrices, such that each of the factored matrices has the special property of minimising the number of complex multiplications and additions. Figure 3.13 illustrates the relationship between the number of multiplications required using the direct method compared with the number of multiplications using the FFT algorithm. As shown, it is faster to employ the FFT for computation if N exceeds 64. The computing time is reduced by an additional factor of two.

Table 3.1 [94] gives a comparison of the computing times required for direct method and FFT algorithm.

Figure 3.14 [94] shows the two dimensional Fourier transform of the aperture function using the FFT algorithm. Figure 3.15 is the two dimensional Fourier transform of 30 pixels dot.

3.3.6.2 The Problems with the FFT

(1) Aliasing in FFT

The sampled wave form $h(t)$ with sample interval T can be expressed as

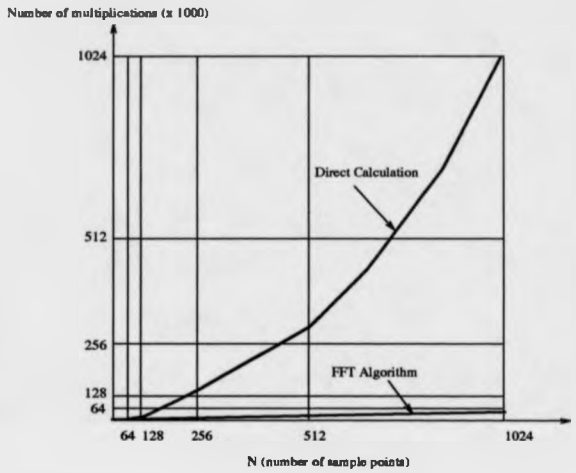


Figure 3.13: Comparison of the multiplications required by direct calculation and FFT algorithm.

Table 3.1: Comparison of computing times(seconds)

N	*Direct Method*	*FFT Method*	*Speed Factor*
16	0.0008	0.003	0.27
32	0.003	0.007	0.43
64	0.012	0.015	0.8
128	0.047	0.033	1.4
256	0.19	0.073	2.6
512	0.76	0.16	4.7
1024	2.7	0.36	7.5
2048	11.0	0.78	14.1
4096	43.7	1.68	26.0

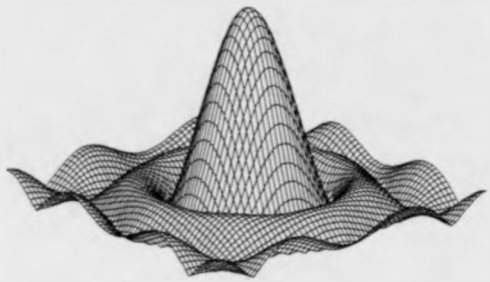


Figure 3.14: The mesh plot of the FFT of the aperture function(5 pixels).

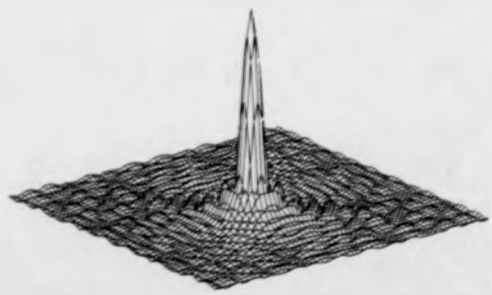


Figure 3.15: The mesh plot of the FFT of point size 30 pixels.

$$\hat{h}(t) = \sum_{n=-\infty}^{\infty} h(nT)\delta(t - nT) \tag{3.59}$$

Equation 3.59 shows an infinite sequence of equidistant impulses. One of the problems with the Fourier transform is aliasing. If the sample interval T is too large, aliasing occurs. In order to avoid the aliasing problem, a sample interval T has to be chosen equal to one-half the reciprocal of the highest frequency component, i.e. $1/T = 2f_c$, where f_c is the highest frequency component. The Fourier transform of a function $h(t)$ is band-limited at the frequency f_c and the sample spacing must be chosen as $T = 1/2f_c$, where the frequency $1/T = 2f_c$ is known as the Nyquist sampling rate. Only if the function meet the requirements of sampling theorems, can it be reconstructed from a knowledge of the impulse.

(2) Ripple effect in FFT

In the Discrete Fourier Transform, it is necessary to truncate the sampled $\hat{h}(t)$ so that only a finite number of points are considered. If the truncation interval is not wide enough, a ripple effect will be produced. This will cause corruption of the high frequency information associated with the phase function. In the case of Sinusoidal fringes, the spectral side-lobes of $C(f-f_0)$ and $C^*(-f-f_0)$ will overlap with the central lobe and with each other. Therefore, not only is high frequency information lost, but also high frequency information is corrupted because the tail of the side lobe of $C(f - f_0)$ extends beyond $f = 0$ and modifies the high frequency associated with $C^*(-f - f_0)$. The ripple effect can be reduced by increasing the truncation interval, then the Fourier transform of the truncation (rectangular) function $\sin(f)/f$ approaches an impulses, the more closely the $\sin(f)/f$ approaches an impulse, the less ripple or error is introduced by the convolutions that results from truncation. Hence, it is desirable to choose the length of the truncation function as long as possible.

(3) The leakage problem in FFT

In performing the Fourier transform fringe analysis, a leakage problem arises

due to the digitisation of data and truncation. There is always a finite sampling interval used in the digitisation of the interferogram, and therefore, a finite sampling interval in frequency domain. In the case of a rectangular truncation window, if the truncation interval is not equal to a period or integer multiple of a period, the convolutions of the time function does not produce the original periodic function. This convolutions introduces additional frequency domain components, because of the side-lobe characteristics of the sinc function. These additional components are known as the leakage. In order to reduce leakage, an appropriate time-domain truncation or weighting function has to be chosen. This weighting function has frequency domain side-lobe characteristics. The smaller the side lobes, the less leakage affects the results of the FFT. Before the FFT is computed, a weighting function is always applied to the N point sampled function. This is a way of improving the FFT approximation of the Fourier transform.

3.3.6.3 Application to Fringe Analysis

The Fourier transform method of fringe pattern analysis is a powerful way of evaluating conventional interferograms [26, 95]. A high spatial carrier frequency is introduced by tilting the wavefront. Due to the tilt given to the object beam at the second exposure in double-exposure holographic interferometry, the reconstructed image has a spatial carrier fringe pattern that is phase modulated by this tilt. This phase function must be a slowly varying function compared to the variation introduced by the carrier frequency.

In interferometry, the interference fringes of tilted wavefronts can be described by

$$g(x,y) = a(x,y) + b(x,y)\cos[2\pi f_0 x + \Phi(x,y)] \tag{3.60}$$

where $a(x,y)$ and $b(x,y)$ are unwanted background and modulation terms respectively, f_0 is a spatial carrier frequency introduced by tilting the wavefront, and the phase $\Phi(x,y)$ contains the desired information.

Figure 3.16 (a) shows the intensity distribution along raster 256 of the 512 in the image. The unwanted irradiance variations which are expressed by $a(x,y)$ and $b(x,y)$ in Eq. 3.60 can be seen in Fig. 3.16 (a). The end points of each raster are brought to zero by applying a Papoulis window across each raster in the spatial domain, to simulate a periodic function, as shown in Fig. 3.16 (b). The Papoulis window is defined as the square of the Hanning window [96].

For the purpose of Fourier fringe analysis, the input fringe pattern can be written in the following more convenient form:

$$g(x,y) = a(x,y) + c(x,y)e^{2\pi j f_0 x} + c^*(x,y)e^{-2\pi j f_0 x} \tag{3.61}$$

where $c(x,y) = \frac{1}{2}b(x,y)e^{j\Phi(x,y)}$ and * denotes a complex conjugate.

The Fourier transform $G(f,y)$ of recorded intensity distribution $g(x,y)$ is given by

$$G(f,y) = A(f,y) + C(f - f_0, y) + C^*(f + f_0, y) \tag{3.62}$$

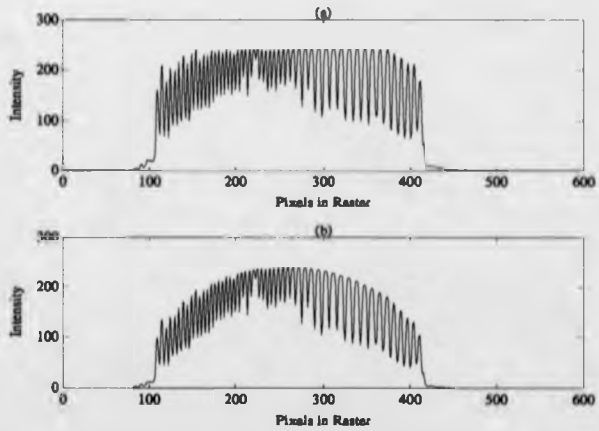


Figure 3.16: (a) Digitised intensity data of central raster in the interferogram; (b) intensity data weighted by Papoulis window

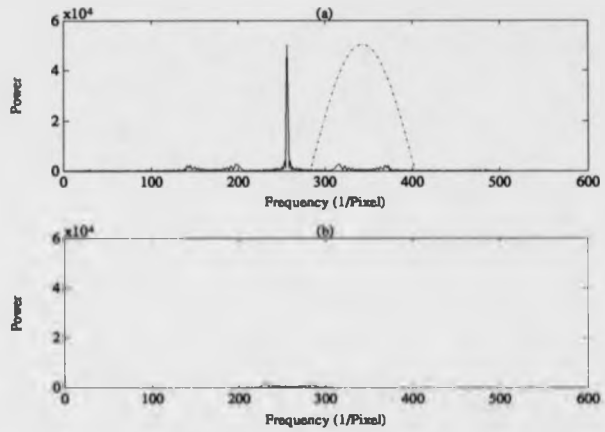


Figure 3.17: (a) Power spectrum of a raster from the interferogram with carrier and deformation (the window is indicated by the dashed dot line); (b) side lobe translated by the carrier frequency to the origin position

The Fourier transform of the data in Fig. 3.16 (b) is shown in Fig. 3.17 (a). Since in most cases $a(x,y), b(x,y)$ and $\Phi(x,y)$ vary very slowly compared to the carrier frequency f_0, all the spectra are separated from each other by the carrier frequency f_0, as shown in Fig. 3.17 (a). One of the side lobe is weighted by the Hanning window. Fig. 3.17 (b) shows the side lobe translated by f_0 in Fig. 3.17 (a) toward the origin to obtain $C(u,v)$. The central lobe and either of two spectral side lobes are filtered out by translating one of the side lobes to zero frequency. By applying the inverse Fourier transform, $c(x,y)$ is obtained. The phase distribution can be calculated pointwise by

$$\Phi(x,y) = \arctan\left(\frac{Im[c(x,y)]}{Re[c(x,y)]}\right) \tag{3.63}$$

where $Re[c(x,y)]$ and $Im[c(x,y)]$ denote the real and imaginary parts of $c(x,y)$, respectively. Since the phase calculation by computer gives principal values ranging from $-\pi$ to π, the phase distribution is wrapped into this range and consequently has discontinuities with 2π-phase jumps for variations more than 2π.

These discontinuities can be corrected easily by adding or subtracting 2π according to the phase jump ranging from $-\pi$ to π or vice versa.

As an example of the 2-D FFT for fringe analysis, Fig. 3.18 is the holographic interference pattern of a deformed silicon wafer with carrier fringes. The amplitude spectrum of the interferogram of Fig. 3.18 is shown in Fig. 3.19. The zero peak is in the centre of the image. After filtering by a band-pass filter in the frequency domain, one of the side-lobe is translated to the centre as illustrated in Fig. 3.20 and Fig. 3.21. The resulting interference phase distribution modulo 2π is given in Fig. 3.22. The unwrapped phase map of the hologram of a silicon wafer is shown in Fig. 3.23.

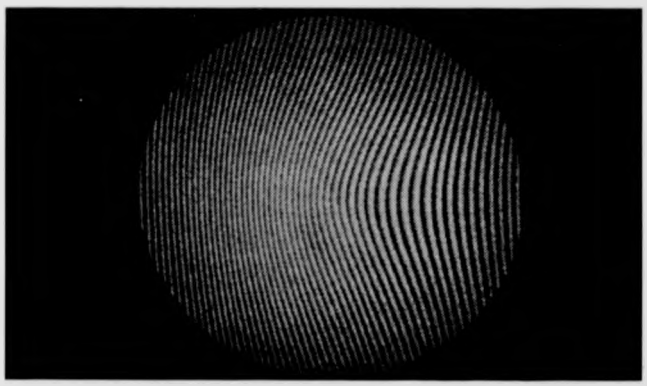


Figure 3.18: Holographic interference pattern showing deformed silicon wafer with carrier fringes.

3.4 The Methods of Phase-Unwrapping

In interferometry, the phase is obtained from an actangent function which gives values in the range $-\pi$ and $+\pi$. This produces "sawtooth" phase fringes, or wrapped phase. It is necessary to remove 2π phase steps to obtain a continuous map of wavefront coded in the interferogram. This can be achieved by adding

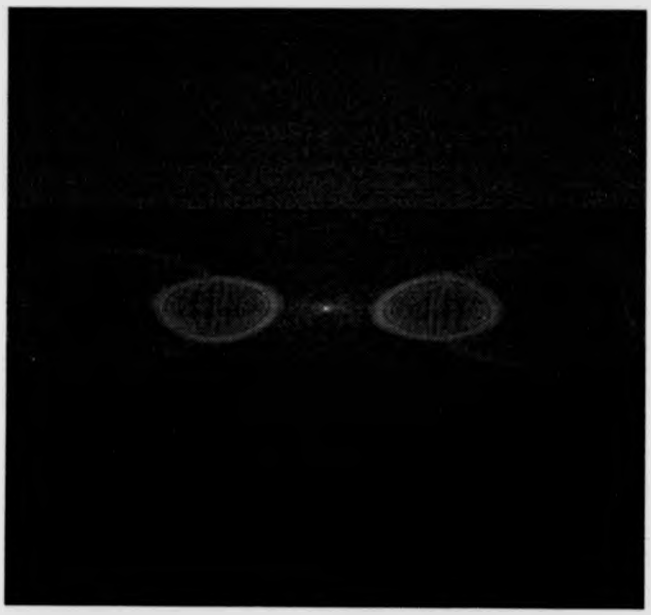


Figure 3.19: Spectrum of holographic interference pattern, the centre peak is the dc value.

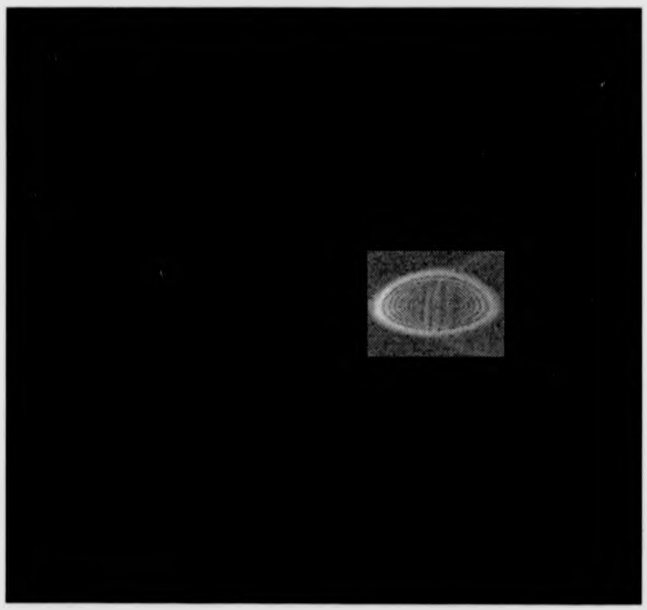


Figure 3.20: Filtered spectrum of holographic interference pattern by band-pass filter.

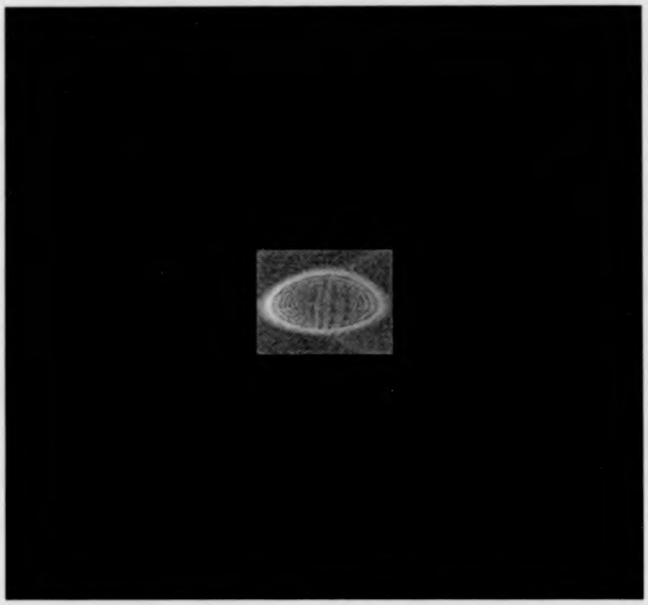


Figure 3.21: After band-pass filtering, the side-lobe is translated to the centre.

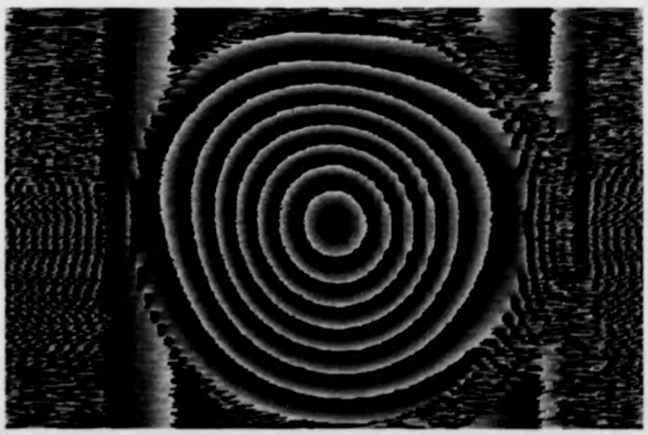


Figure 3.22: Interference phase modulo 2π.

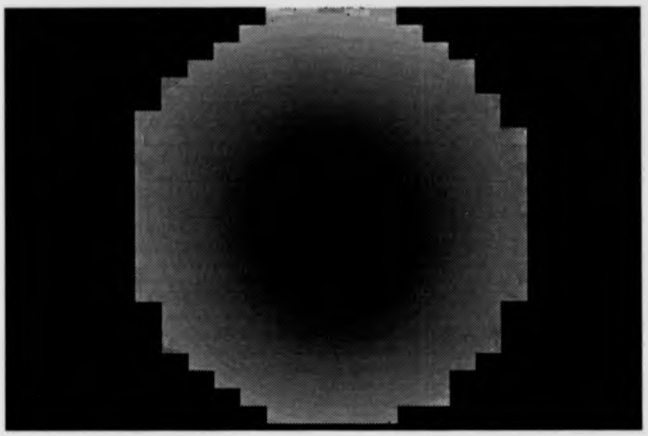


Figure 3.23: Unwrapped phase map of interference fringes.

or subtracting 2π step function to the wrapped phase, which removes discontinuities to get continuous phase map related to the measurement parameter. This is known as phase-unwrapping. In many holographic applications, automatic phase-unwrapping is a fundamental requirement for fringe analysis. This field has been the subject of considerable research activity in recent years. The phase-unwrapping technique involves calculating the fringe phase $\Phi(x, y)$ at each pixels, either by the phase-stepping method [24, 93, 97, 98], or by the Fourier transform method [26, 28, 62]. The phase value of $\Phi(x, y)$ obtained in this way is undetermined by some factor of 2π. Therefore, phase-unwrapping must be carried out in order to convert the discontinuous phase to a continuous one. This is normally achieved by adding or subtracting 2π to the wrapped phase. First the procedure is carried out along each row in turn, and then repeated along the columns. However, when random noise, object discontinuities, poorly modulated pixels are presented, the conventional phase-stepping algorithms would not unwrap the phase correctly.

Wrapping a phase function is equivalent to making the function modulo 2π. Takeda *et al* [26], Kreis [28] and Macy [95] reported the standard algorithm for

phase-unwrapping and computed phase by looking at the differences between adjacent pixels along a raster. The continuous phase in the 1D case $\Phi_c(x_i, y)$ is given by [28]

$$\Phi_c(x_i, y) = \Phi_d(x_i, y) + 2\pi n(x_i, y) \qquad i = 1, 2, 3..... \tag{3.64}$$

where $\Phi_d(x_i, y)$ is the computed wrapped phase at point $(x_i, y), n(x_i, y)$ is an integer. Since the sampling theorem demands at least two pixels per fringe, the phase difference between two adjacent pixels must be less than π. In this case, the integer value $n(x_i, y)$ may be expressed by [28]

$$n(x_i, y) = 0, \tag{3.65}$$

$$n(x_i, y) = \begin{cases} n(x_{i-1}, y) & \text{if } |\Phi(x_i, y) - \Phi(x_{i-1}, y)| < \pi \\ n(x_{i-1}, y) + 1 & \text{if } |\Phi(x_i, y) - \Phi(x_{i-1}, y)| \leq -\pi \\ n(x_{i-1}, y) - 1 & \text{if } |\Phi(x_i, y) - \Phi(x_{i-1}, y)| \geq \pi \end{cases}$$

$$i = 2, 3, 4..... \tag{3.66}$$

The above is a conventional phase-unwrapping algorithm that works correctly unless the sampling theorem requirement is satisfied. This approach is computationally efficient but has poor noise immunity.

Huntley [99] has described another method of noise immune phase- unwrapping, which is based on the requirement that the unwrapped map should be independent of route by which unwrapping takes place. This method relies on equality of phase change in paths between pixel pairs. Another method of phase-unwrapping is described by Ghiglia [100], using Celluar automata. The method as first outlined is time consuming, taking many iterations to converge.

The Celluar automata phase-unwrapping method is similar to that by Takeda [26], Kreis [28] and Macy [95]. The mathematical description of the method is as follows. In this method, a wrapping operator is needed which is defined as

$$W_l[\Phi(n)] = \Phi_p(n) \quad (n = 0, 1, 2.....N) \tag{3.67}$$

where $\Phi(n)$ is the unwrapped phase samples, $\Phi_p(n)$ is the sample of principle values, n is an integer, l is the label indicating different wrapping operators. The Eq. 3.67 may be written as

$$W_l[\Phi(n)] = \Phi_p(n) + 2\pi K_l \quad (n = 0, 1, 2.....N) \tag{3.68}$$

where K_l is a sequence of integers chosen so that

$$-\pi \leq W_l[\Phi(n)] \leq \pi \tag{3.69}$$

The difference operator can be also defined as

$$\Delta\Phi(n) = \Phi(n) - \Phi(n-1) \quad (n = 1, 2, 3.....N) \tag{3.70}$$

By using Eqs. 3.68 and 3.70, the difference between the principle values can be calculated as

$$\Delta W_1[\Phi(n)] = \Delta\Phi(n) + 2\pi\Delta K_l(n) \tag{3.71}$$

If applying the wrapping operator to Eq. 3.71, the wrapped difference of wrapped phases is obtained by

$$W_2\{\Delta W_1[\Phi(n)]\} = \Delta\Phi(n) + 2\pi[\Delta K_1(n) + K_2(n)] \tag{3.72}$$

According to the Nyquist sampling theorem, the wrapped phase produced by Eq. 3.72 must lie in the range between $-\pi$ and $+\pi$, therefore

$$-\pi \leq \Delta\Phi(n) \leq \pi \tag{3.73}$$

then the term $2\pi[\Delta K_1(n) + K_2(n)]$ in Eq. 3.72 becomes zero. Therefore,

$$\Delta\Phi(n) = W_2\{\Delta W_1[\Phi(n)]\} \tag{3.74}$$

which can be manipulated to produce

$$\Phi(m) = \Phi(0) + \sum_{n=1}^{m} W_2\{\Delta W_1[\Phi(n)]\} \tag{3.75}$$

Equation 3.75 indicates that the phase may be unwrapped by integrating the wrapped differences of wrapped phases providing that the Nyquist sampling theorem is satisfied. If the Nyquist sampling condition is violated, the true phase can not be recovered.

Ghiglia *et al* [100] first presented the original celluar automata algorithm which was applied to path independent phase-unwrapping. The routine to mask path inconsistences consists of test of every 2 × 2 group of pixels. If the sum of the wrapped phase difference along the path equals zero, then all four pixels are said to be consistent, otherwise all four pixels will be masked out. This is done for all 2 × 2 sample regions until the entire wrapped phase array is covered. This routine is applied only once before the phase-unwrapping, and this is not enough because the routine masks single point path dislocation appears on the image, only the ends of the discontinuity line are masked, not the whole path.

Spik [101] has made some modifications to Ghiglia's original algorithm which improved its performance when dealing with noise. Buckberry [102] has implemented a cellular automata method on a frame store with associated pixel processing functions, which has produced an efficient system.

Another new phase-unwrapping method was proposed by Judge *et al* [62, 107] This method produces results that are nearly path independent using the minimum spanning tree approach to phase-unwrapping. The minimum spanning tree is a mechanism for comparing the consistency of possible phase-unwrapping routes.

The minimum spanning tree is a product of graph theory [103, 104, 105]. Figure 3.24 shows an example of a weighted undirected graph, which consists of

vertices and edges. A tree is a connected graph without any circuits. Figure 3.25 is a tree. A spanning tree in a graph G is a tree connecting all the vertices of G. If graph G is a weighted graph (i.e., if there is a real number associated with each edge of G), then the weight of a spanning tree T of G is defined as the sum of the weights of all the branches in T. Among all of the spanning trees of G, those with the smallest weight are called minimum spanning trees. Figure 3.25 shows the minimum spanning tree of the graph in Fig. 3.24 [104].

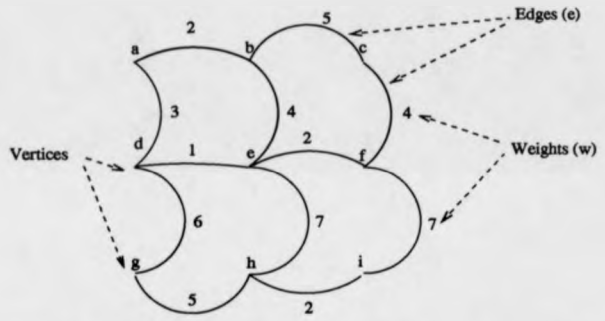


Figure 3.24: Example of a weighted undirected connected graph.

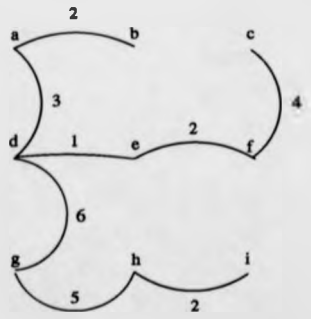


Figure 3.25: A minimum spanning tree of the weighted graph.

The minimum spanning tree may be constructed by various algorithms. Prim's [106] algorithm presents an efficient solution. The four edge weights are tabulated for

each pixel. Unwrapping begins at the pixel which holds the lowest edge weight.

On each iteration of Prim's algorithm a new edge e is added to the growing spanning tree T. The edge e is the edge of least weight connecting a vertex in the remainder of the graph with those in T.

The approach has developed from a fringe counting method. The initial procedures of both methods include the steps outlined below:

- computation of wrapped phase map
- fringe edge detection; a Sobel edge detector with hysteresis thresholding is employed here [107, 108]
- the unwrapping of the phase, row by row
- the unwrapping of the phase, column by column.

At this point the fringe counting and minimum spanning tree techniques divide. Figure 3.22 shows a wrapped phase map of a silicon wafer. Figure 3.23 is an unwrapped phase map of Fig. 3.22.

3.5 Image Capture Device and Digital Frame Board

3.5.1 CCD Camera

Charged Couple Device (CCD) camera is well suited for use as the electronic sensors in the interference plane [109]. It is a high resolution solid-state camera designed for scientific and industrial imaging applications, where high accuracy and image quality are important.

The key component of the CCD camera is the sensor. The Cohu solid state camera is used in all the experiments. The active area of the device (6.4×4.8mm) consists of 512 $\times$ 512 pixels. The camera can produce an accurate geometrical

grid of data points, and the sensor integration time can be readily controlled. The discrete video pixel output is also directly compatible with digital computer processing techniques.

The CCD is a concept of semiconductor electronics. It exhibits the characteristics in conversion of incident energy into electrical charge. The mechanism for moving and sensing this charge is an important feature of this device. The charge-coupling is easily visualised as a "bucket bridge" of transferred charge, where during the charge in one bucket is poured or moved into the adjacent bucket by the external manipulation of voltages [110]. The quantity of the stored charge on the applied voltages and on the capacitance of the storage element. The amount of electric charge in each bucket can represent information. Figure 3.26 shows a p-type semiconductor CCD cell [110].

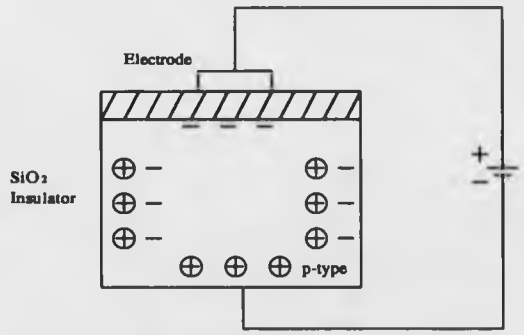


Figure 3.26: The basic CCD cell structure.

The basic CCD cell is an enhancement model MOS device that behaves like a capacitor. The device is formed by diffusing an impurity into pure silicon. This creates a semiconductor device upon which an insulator is formed. The electrodes are deposited on top of the insulator to facilitate charge transfer. The minority carriers are then moved by applying suitable voltages to the transfer conductors. The charge of interest is attracted to the electrodes when a positive voltage is applied to them. This creates charge buckets stored in potential wells under the

electrode.

Figure 3.27 shows the CCD imaging sensor consisting of an area array of the CCD cells [110]. This discrete sensing array involves a finite sensing area under each potential well that converts incident illumination in this area to minority carriers.

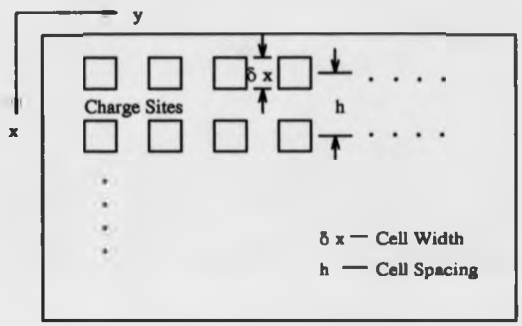


Figure 3.27: CCD sensor substrate with individual cells.

Figure 3.28 shows a 1-D horizontal slice of the array [110]. The light falling continuously on the photosensor sites produces electrons, which accumulate as charge buckets in the potential wells created by the photogate voltage. The size of charge bucket is dependent on the incident light intensity at this site. The amount of charge thus generated at each sensing site is also a function of the dimensions of site and time during which this accumulation is allowed. The time is determined by the frame rate of the system and typically 1/30 of a second.

The CCD camera has many characteristics, it can perform at full capability in the temperature range of -55^{0}c - +70^{0}c. The speed limitation of the CCD device is theoretically that of electron mobility in silicon. The reliability of the CCD device equals that of NMOS. The noise level of the CCD device is rather low, because the basic CCD register, heart of all CCD devices, is almost noiseless, since it does not have PN junctions as do MOS and bipolar devices. A typical structure of an area of a CCD solid-state camera is shown in Fig. 3.29, which

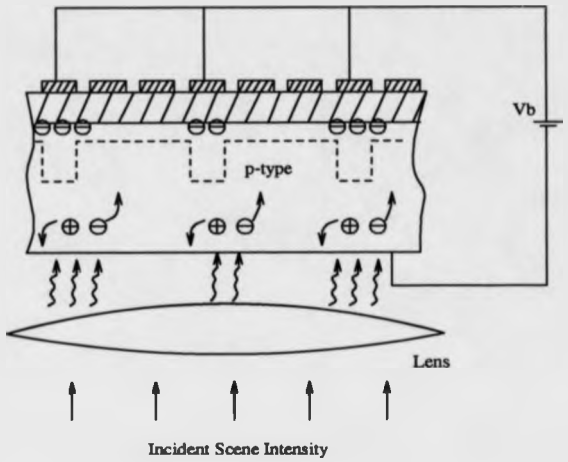


Figure 3.28: Generation of minority carrier charge bucket from incident illumination.

consists of a sense head and a camera control unit.

3.5.2 Digital Frame Store (PIP)

The PIP is a plug in card that allows an IBM PC microcomputer to perform frame grabbing operations on a video signal from an external source. It has a resolution of 512 × 512 pixels with eight bits per pixels. The PIP is capable of operating in continuous grab or snapshot mode, and also has a built-in video keying capabilities. Frames which have been stored in the on board frame buffer can be loaded into the PC's memory or into a disk. Conversely, video data, as well as lookup table data, can be written to the PIP from the PC. Pixels can be individually addressed by the PC using the X and Y address registers or they can be addressed sequentially using an auto-increment register. A block diagram of PIP operation is shown in Fig. 3.30.

The PIP has three inputs and an IBM compatible RGB output. There is also an internal feed back for diagnostic purposes. As shown in Fig. 3.30, input

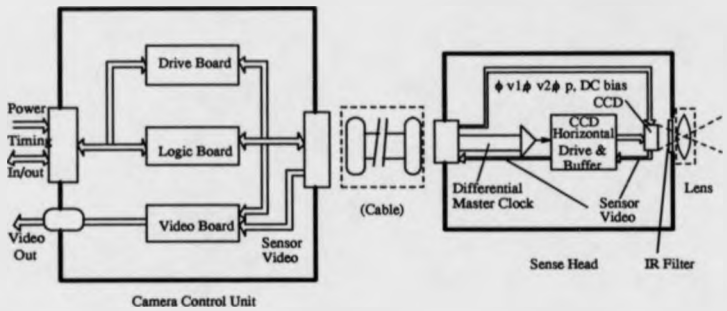


Figure 3.29: Typical structure of CCD camera.

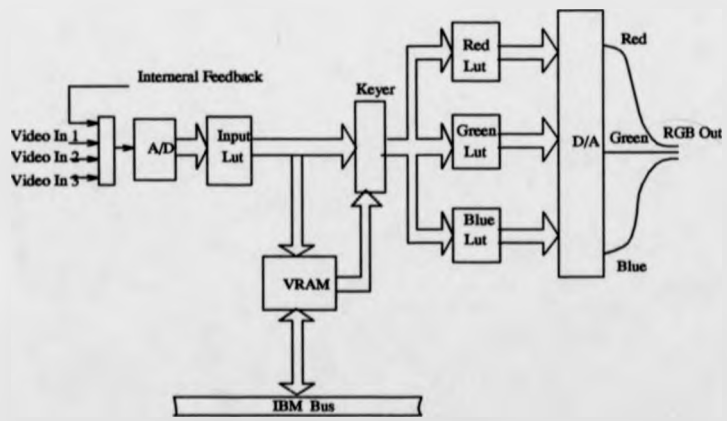


Figure 3.30: Block diagram of the PIP board.

signals are selected from one of three input ports on the PIP. The PIP will lock on to the input signals sync and use the sync signal to drive the video functions of the board. The board can also generate its own sync signal using a master clock. After the data passes the sync separator, it can be adjustable DC offset voltage along with the gain by the software. The video input is then digitised in real-time in the A/D converter, producing an eight bit number that is sent to the input lookup table. An input lookup table is used to allow the user to manipulate the video data as it is taken from the A/D converter. The PIP has a frame buffer RAM that is used to store frame grabbed data. When a frame is grabbed, it is taken from the selected input port and digitised. It then passes through an input lookup table and is stored in the frame buffer. The video output is controlled by the keyer. When video keying is selected, the keyer, on a pixel by pixel basis, tests the least significant bit of the frame buffer, when this bit is 0, the pixel from the input lookup table is sent to the output lookup tables, otherwise the frame buffer pixel is sent.

There are three output lookup tables, each of which has eight maps to choose from. Each lookup table receives all eight bits, and then using one of its eight maps, generates a new value. The output of each lookup table is sent to a D/A converter. These D/A converters produce three real-time analogue signals for RGB output. Finally the signals or images are displayed on the monitor.

3.6 Discussion

The methods for automated interference fringe analysis allow measurements with high accuracy [89]. A comparison of the methods for interference phase measurement is given in Table 3.2 [111, 112]. Accuracy of the order of 1/1000 of a fringe is obtainable using the heterodyne method [19]. However, this method requires sophisticated electric equipment and mechanical scanning of the image by photo-detectors. It is well suited for application in which very high resolution is needed. On the other hand, the phase-stepping method does not require

Table 3.2: Comparison of methods for interference phase measurement

Characteristics	**Fringe Tracking**	**Phase-Stepping**	**Heterodyne**	**Fourier Transform**
No. of Interferogram Required	1	≥ 2	1	1
Accuracy (Order of interference fringe)	1/10	1/100	1/1000	1/50
Real Time	Yes	Yes	No	Yes
Environment Requirements (Vibration)	Low	Medium	High	Low
Image Quality	Good	Good	Medium	Good
Electronics Requirements	Low	Medium	High	Low
Optics Requirements	Low	Medium	Medium	Low
Inherent Image Enhencement	No	Yes	No	Yes
Automatic Sign Detection	No	Yes	Yes	Yes
Sensitivity to Experimental Errors	Low	Medium	Very High	Low
Experimental Requirements	Low	High	Very High	Low

any special instrumentation apart from a CCD camera or other detectors. This method is well suited for real-time phase measurement and leads to fringe measurements to within 1/70 of a fringe [113]. The major sources of errors in the calculated phase values are phase shifter miscalibration, phase-shifter nonlinearity, detector nonlinearity, quantisation and vibration [93, 111]. The spatial fringe scanning phase measurement method results in determination of fringes to within 1/10 of a fringe [114]. The Fourier-transform method offers accuracy within 1/20 of a fringe [62, 26, 27, 28]. This method of computing phases can produce phase maps automatically without the need for locating fringe centres or assigning fringe orders. Compared with the phase-stepping methods, where three or four reconstruction with mutual phase shifts are recorded and evaluated, only one interference fringe pattern is needed provided there is knowledge of the direction of the added carrier fringes. The disadvantage of the FFT method is that it requires the sufficient carrier fringes to give a monotonic change in fringe field. In practice, this can not be guaranteed for an unknown fringe field. The errors in the FFT method are the incorrect filtering in the Fourier domain, the influence of random noise and spurious fringes in the fringe field, and also making incorrect choice for the carrier fringe frequency [115]. Kreis [28] reported a FFT method which no additional carrier fringe has to be introduced and compensated for by a frequency shift in the spatial frequency domain. However, this method needs two reconstructions of the holographic interference pattern with a mutual phase shift in order to remove the ambiguity and get a reliable detection of phase.

Chapter 4

Fibre Optics and Its Application to Holography

4.1 Introduction

As mentioned in the previous chapters, holographic interferometry is a versatile measurement tools, which can be used to measure the surface of rough objects, and in many applications, cannot be replaced by other techniques. In recent years, there is a gradually increasing employment of fibre optics for holographic interferometry. The use of optical fibres in connection with continuous wave (CW) and pulsed lasers has found many holographic applications [29, 30, 31, 32, 116, 117]. Investigation of obscured or remote objects or studies in difficult environment using holographic interferometry are now possible with optical fibres. This chapter discusses the properties of optical fibres and fibre optics application to holography and holographic interferometry. A holographic system based on the fibre optics was developed for many holographic applications, such as holographic contouring, holographic deformation measurement. This fibre optic holographic system is particularly important for carrier fringe technique and automated analysis of holographic interferograms.

Optical fibres have used in optical communication systems [118, 119]. They

are also used in sensor applications. Optical fibre sensors can be designed to detect many parameters, such as pressure, temperature, gas detection, strain and electrical and magnetic fields [120, 121, 122]. Another application of optical fibres is beam delivery for medical uses in surgery and diagnostics.

Research on optical fibre holography has now been carried out by many researchers [29, 30, 32]. The use of optical fibres offers several advantages in the field of optical holography [5, 32]:

- It requires much less space than the conventional system and allows the user almost unlimited freedom in selecting object and reference beam angles.
- The fibre holographic system is more flexible than conventional system and less optical equipment is needed. In the case of fibre holography, the basic equipment may be contained in a closed portable box as shown in Figure 4.1. In this system, two fibre-cable heads provide the object and reference beams for holographic recording. The system is dust free, portable, and easily setting-up. The beam-splitter can be of the varying ratio type, in order to have an optimum power distribution. During the exposure, only the fibre-cable ends need to be kept in their stable position.
- The fibre holographic system can be used to investigate an obscured or remote system, the phase-stepping approach can be easily applied by moving the fibre optic beams. The use of optical fibres in connection with CW lasers has already found interferometry applications with metrology [116]. Applications of pulsed fibre holography have also been reported [117].
- The carrier fringe pattern can be easily generated by translating fibre optic beams between exposures in holographic interferometry. This is very useful for implementing fast Fourier transform of fringe patterns [62, 39]. The developed holographic system based on the fibre optics can also be applied to holographic contouring [5].

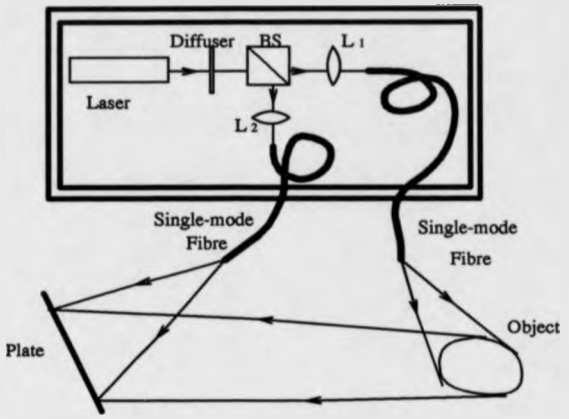


Figure 4.1: Portable fibre optical holographic system (Leite 1979).

4.2 Light in Optical Fibres

4.2.1 Introduction to Optical Fibres

Light can travel through a thin-glass fibre whose cross section and optical properties are designed on the phenomenon of total internal reflection. The cross section of a step-index optical fibre is shown in Figure 4.2.

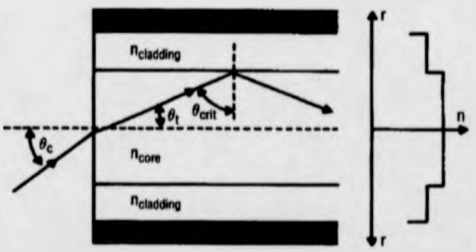


Figure 4.2: The cross section of step-index fibre.

A central part called the core of refractive index n_1 is surrounded by a cylindrical region, called the cladding of lower refractive index n_2. The cladding is

then covered with a protective plastic jacket. Therefore, electromagnetic waves can be confined in the core region and transmitted by total internal reflections at the core/cladding boundary. Two kinds of the most common fibres commercially available are step-index and graded-index as shown in Figure 4.3 [123].

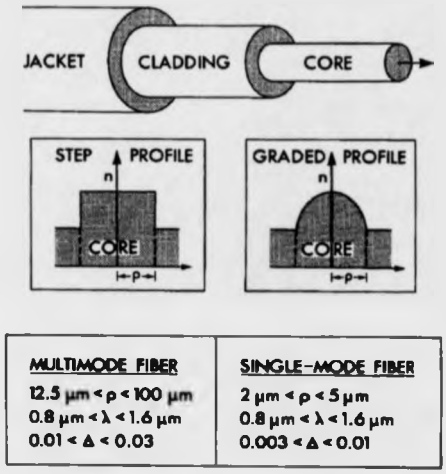


Figure 4.3: Profiles for typical optical fibres (Snyder *et al* 1983).

Furthermore, optical fibres can be conveniently divided into two subclasses called multimode fibres (with comparatively large cores) and single-mode fibres (with comparatively small cores). The range of dimensions for typical optical fibres is shown in Fig. 4.3. Where ρ is the core radius, λ is the free-space wavelength of light and Δ is the fractional refractive index difference.

4.2.2 Several Important Parameters

(1) Critical propagation angle

As shown in Figure 4.4, two rays are incident on an interface of fibre at different angles. One (the solid line) is incident at a lower angle than is the other (the dashed line). The dashed ray refracted at point A, the ray then gets to

the core-cladding interface at point B. Refaction takes place at point B. The ray bends and continues in the cladding. Finally, the rays bends and exits the fibre at the cladding-air interface at point C. The solid ray refracted at point A. At point B', the core-cladding interface, total reflection occurs. It should be noted that total reflection can occur only when light travels from high index to low index media. This ray is confined to the fibre optics. The angle of incidence at the core-cladding interface is called critical angle (α_c). An incident ray with an angle larger than α_c will propagate in the fibre. The critical ray makes angle θ_c with the fibre center. The angle θ_c is called the critical propagation angle [124], which is given by

$$\theta_c = \cos^{-1}(n_2/n_1) \tag{4.1}$$

All rays with angles less than θ_c will propagate in the fibre. These rays are referred to as modes of propagation. The total number of modes propagating in the fibre increases as θ_c increases.

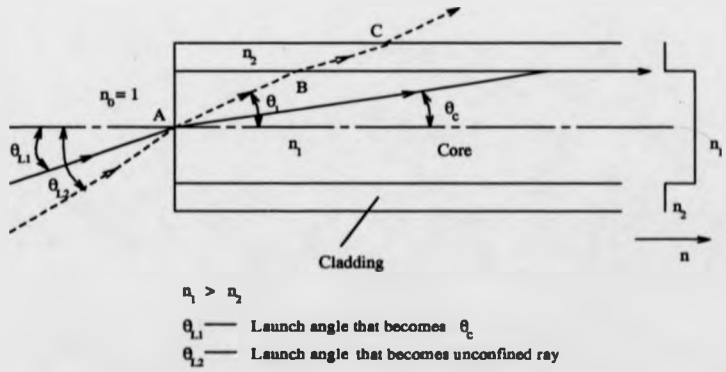


Figure 4.4: Light propagation in a step-index fibre (Zanger *et al* 1991).

(2) Acceptance angle

The light in an optical fibre are transmitted through total internal reflection at the core-cladding interface. The condition is that the propagation angle must

be equal to or less than the critical propagation angle. Figure 4.5 shows two rays incident at the critical α_c within the fibre at core-cladding interface. Both rays (solid and dahed rays) enter the fibre core at an angle θ_a to the fibre axis and are refracted at the aie core interface before transmission to the core-cladding interface at the critical angle. Therefore, any rays which are incident into the fibre core at angle greater than θ_a will be transmitted to the core-cladding interface at an angle less than α_c, and will not be accepted by the fibre. For rays to be transmitted by total internal reflection within the fibre core, they must be incident on the fibre core within an acceptance cone defined by the conical half angle θ_a. Hence θ_a is the maximum angle to the axis that light may enter the fibre in order to be propagated. The angle $2\theta_a$ is the acceptance angle [124], which is given by

$$2\theta_a = 2\sin^{-1}(n_1 \sin\theta_c) \doteq 2\sin^{-1}\sqrt{n_1^2 - n_2^2} \tag{4.2}$$

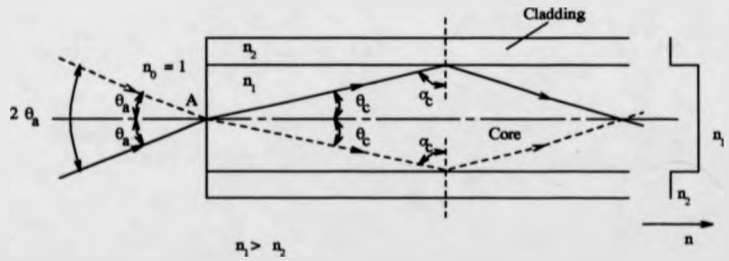


Figure 4.5: The acceptance angle θ_a when launching light into an optical fibre (Zanger *et al* 1991)

(3) Numerical aperture NA

The numerical aperture, NA, is a quantity which is frequently used to specify the light gathering power of an optical system. In the case of the optical fibre, it is a measure of the amount of light which traverses through the fibre. This is defined as the sine of the maximum ray angle in object space multiplied by the

refractive index of the object space.

$$\mathrm{NA} = n \sin(\theta_{max}) \tag{4.3}$$

If the light is incident from the air ($n = 1$), the numerical aperture of a step-index fibre is

$$\mathrm{NA} = \sqrt{n_1^2 - n_2^2} \tag{4.4}$$

The fractional refractive index difference is given by

$$\Delta = \frac{n_1 - n_2}{n_1} \tag{4.5}$$

When $\Delta \ll 1$, Eq. 4.4 can be approximated by

$$\mathrm{NA} = n_1\sqrt{2\Delta} \tag{4.6}$$

(4) Normalised frequency ν

An important quantity in a fibre is a parameter called the normalised frequency, which is defined as

$$\nu = k_0 a\sqrt{n_1^2 - n_2^2} \quad \text{or} \quad \nu = k_0 a \mathrm{NA} \tag{4.7}$$

where k_0 is the free space wave number, $\frac{2\pi}{\lambda_0}$, a is the radius of the core.

4.3 Polarisation in Fibre Optics

The direction of the electric field determines the polarisation of the waves. In many light sources, the polarisation of the light varies in random manner. Other source, such as the output of many lasers, are linearly polarised. When light is linearly polarised, the electric field vector maintains a constant orientation in space. In the case of optical fibres the polarisation of light transmitted through them may be preserved or it may be scrambled to yield randomly polarised light,

depending on the fibre that is used. For example, the output light of the He-Ne laser is linearly polarised, when launching this linearly polarised light into the fibre, the output will in general be elliptically polarised and dependence upon external factors (such as temperature changes, bending and twisting of the fibres).

In a perfectly symmetric circular fibre, the two polarised components HE_{11} mode (the LP_{01} modes with orthogonal polarisations) travel at the same velocity, since they have identical propagation constants. If the fibre is not perfectly symmetric, then the fibre will be birefringent. Therefore, the polarisation state of the optical signal will change on propagation through the fibre. If the changes are unequal, the fibre visibility is reduced. The solution to the problem of obtaining an output state of polarisation is to use specially manufactured fibre which maintains the polarisation state, such as Newport FSP-10 fibre used in the experiments.

When linearly polarised light is launched with the polarisation vector parallel to either the fast and slow axis of a polarisation-preserving fibre, the output polarisation will still be linearly polarised despite fibre bending. This polarisation-preserving fibre provides reduced sensitivity to environment effects. However, for other launch conditions, the output polarisation state will changed. In this case, the fibre will act to change the polarisation of the light. The actual output polarisation state will be determined by the launch conditions, the beat length, and fibre length [125]. Polarisation-preserving fibres have applications wherever the polarisation of the transmitted light must be stable and well defined. These applications include fibre interferometric sensors, fibre holography and heterodyne detection system.

4.4 Wave Optics and Modes in Optical Fibres

Although the ray picture of light propagation through a fibre is easy to depict, it does not reveal some of the intersting properties of light in optical fibres, particularly in those fibres where the core size is on the order of the wavelength

of the light. Therefore, the idea of representing electromagnetic waves as rays propagating in straight lines is valid only when the wavelength is small relative to the structure carrying the waves. When the core size is in the order of wavelength, for example, single-mode fibres, electromagnetic wave propagation is usually used to describe the transformation of light through the fibre.

The propagation of light in optical fibres is governed by Maxwell's equations. If the refractive indices and the boundary conditions for the cylindrical geometry of the core and cladding are known, a wave equation can be solved for the electromagnetic field distribution propagating through the fibre. The allowed distributions of the electromagnetic field across the fibre are referred to as the modes of the fibre. The description of the modes that propagate in a fibre can be given by solving the wave equation in a cylindrical waveguide as shown in Figure 4.6.

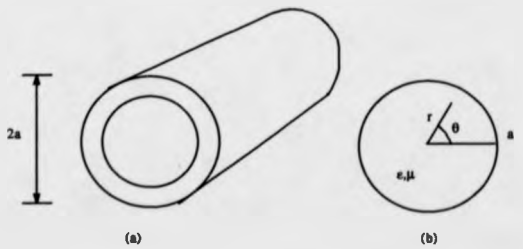


Figure 4.6: Cylindrical waveguide geometry.

The solution consists of cosine functions as below [125]

$$E(r,\theta,z) = f(r)\cos(\omega t - \beta z + \gamma)\cos(q\theta) \tag{4.8}$$

where ω is the frequency of light in radians, β is the propagation constant, γ is the phase constant, q is an integer.

In the analysis of the cylindrical optical fibre, the key parameter ν determines which modes can propagate and how tightly they are bound to the core. The number of modes that can propagate in a particular fibre, when the propagation

constants of the fibre modes are plotted as a function of ν number. Figure 4.7 shows a plot of the normalised propagation constant for a few of the low-order modes [126]. Each ν-number represents a different fibre configuration or a different wavelength of light in a given fibre configuration. According to Figure 4.7 only the HE_{11} mode can propagate in the fibre for fibres with $\nu < 2.405$. This kind of fibre is said to be single-mode fibre. The difference between HE and EH mode depends on whether H_z and E_z make the larger contribution to the transverse field. TE and TM modes are defined by depending on whether the components of E_z or H_z is zero [126].

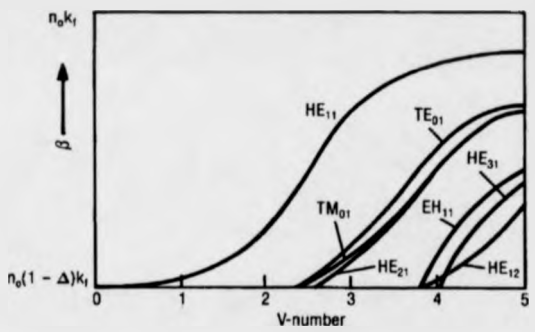


Figure 4.7: Low order modes of an optical fibre (Barnoski 1981).

When ν exceeds 2.405, more than one mode can propagate and share the available power. This fibre is the multimode fibre.

In the case of step-index fibre, a uniform core region of diameter $2a$ and index, n_1 is surrounded by a cladding of index, $n_1\ (1-\Delta)$. If $n_1 \doteq n_2$, the entire space is almost homogeneous, with a single index of refraction $n \doteq n_1 \doteq n_2$. There is a great similarity between the eigenvalues for the $HE_{\nu+1,\mu}$ and the $EH_{\nu-1,\mu}$. This is the case of weakly guiding which suggests a possible linear combination of solutions to achieve a simplicity. Gloge [127] arrived at approximate modes of propagation that are nearly polarised, called LP modes. Figure 4.8 shows the

propagation constants of these modes as a function of the ν-number.

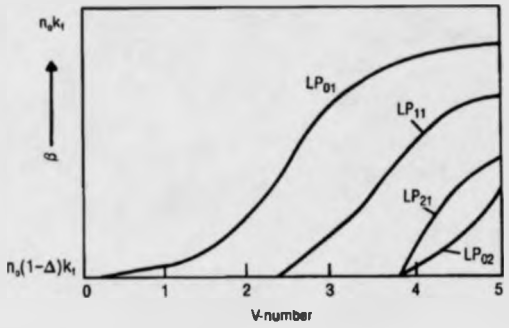


Figure 4.8: Low order linearly polarised modes of an optical fibre (Barnoski 1981).

The irradiance patterns for some low order linearly polarised modes can be shown in Figure 4.9

Figure 4.9: The irradiance pattern in the output of the fibre.

4.5 Applications of Fibre Optics to Holography

Both single-mode and multimode fibres can be used in holographic recording. Single-mode fibre optics have one important advantage over multimode fibres:

single-mode fibre optics can maintain good temporal and spatial coherence of laser light. Multimode fibres can be used for an object beam within an imaging and display holography. The output of such fibres is covered by speckles. In holographic interferometry, investigations using a CW laser for double-exposure holograms of deformation and vibration studies of objects were conducted by several researchers [116, 117]. Double-pulsed holograms of dynamic events were also studied [32]. Further developments in medical endoscopy as well as industrial measurements have been undertaken using optical fibres [128].

4.5.1 Fibre Optic Holography Using CW Lasers

Figure 4.10 shows a typical conventional arrangement for optical holography. In this case, a CW laser is used as a source of highly coherent illumination. In order to record a hologram on the plate, the mirrors, spatial filters and beam splitter are used.

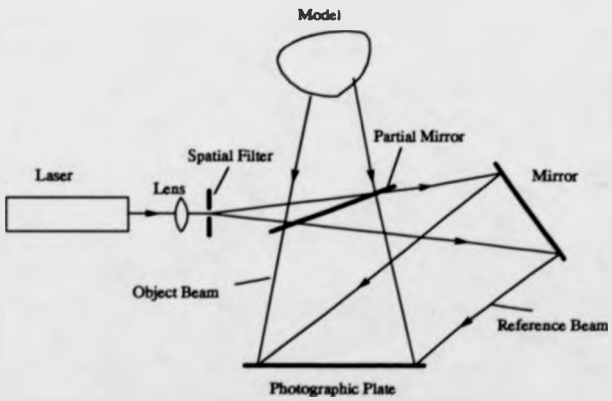


Figure 4.10: Conventional arrangement for reflection holography.

Figure 4.11 is a schematic diagram of a fibre optic system for reflection holography. In contrast, there is no need for mirrors, spatial filters and a beam splitter. The laser output is launched into a single-mode optical fibre. The directional

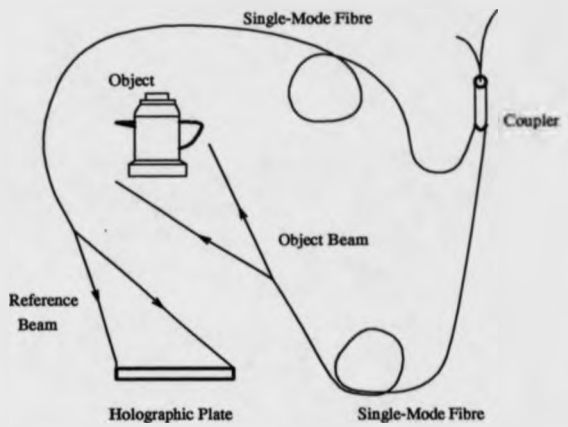


Figure 4.11: Schematic arrangement for optical fibre reflection holography.

coupler is used to divide the beam into object and reference beams. A setup for holographic investigation using such a system is rather simple. It also gives a high stability system for CW holography. However, the output ends of the fibres must be held in a stable holder. Here, the same demand for stability exists for conventional holography. Because the core of a single-mode fibre is very small (5μm), the output beams from such fibres are spatially filtered. The fibre ends act as a spatial filter. Figure 4.12 shows the reconstruction of the processed double-exposure hologram recorded with fibre optic holographic system.

4.5.2 Fibre Optic Holography Using Pulsed Lasers

Optical fibres have already been combined with holography using CW lasers. The use of pulsed lasers in combination with fibres is a more difficult problem to handle, mainly because of the very high peak power output from Q-switched ruby lasers [129]. The first use of a single-mode fibre for transmitting a reference beam from a Q-switched ruby laser was presented in 1982 by von Bally [128] in an investigation of an artificial tympana membrane. Dudderar and Gillbert [117] showed that a pulsed ruby laser could be used for recording both static and dy-

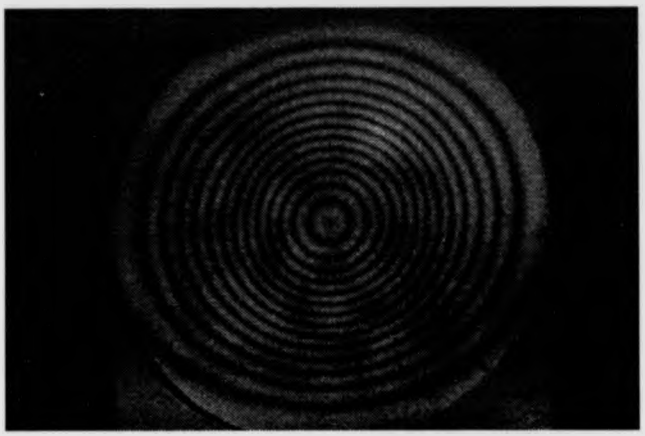


Figure 4.12: A hologram of displacement pattern of a centrally loaded clamped disk using single-mode fibres for reference and object beams (image size 512 × 512).

namic events. Both the object and reference light were guided through multimode fibres and the image was transmitted through an multimode fibre as shown in Fig. 4.13.

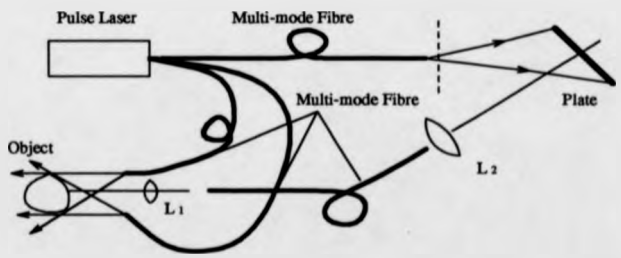


Figure 4.13: Schematic arrangement used to record remote image plane using pulsed ruby laser and multimode fibres (Albe *et al* 1984).

The advantages of this technique are that both the conventional image plane and the remote image plane holograms can be recorded by this system, and even at operating levels in the order of joules. The system is quite capable of handling

the light pulses without destroying the fibres and has the necessary coherence needed to generate holograms.

Bjelkhagen [130] has used single-mode fibres combined with a pulsed laser for high speed holographic recordings. The use of fibre optic pulsed holography has many advantages, in particular for investigations of non-contact vibration and deformation measurements, as well as obscured or remote high speed events, such as a wind tunnel, turbine blade, or combustion. However, in the case of a pulsed laser more problems may occur [129]:

- The very high powers may destroy the fibre. For example, a ruby pulsed laser which delivers 200mJ in 20ns gives a peak power of 10MW. For a single-mode fibre with a 5μm core diameter, the power density is $5.1\times10^{13}\mathrm{W}/cm^2$. An optically smooth silica end face which is perpendicular to the laser light pulse is reported to be able to withstand up to $5\times10^{10}\mathrm{W}/cm^2$. These conditions are rarely met so a safer limit for pulsed laser energy is about 1.0-1.5$\times10^{10}\mathrm{W}/cm^2$. The example above is far beyond this limit. Therefore, the limit for a Q-switched pulsed laser is approximately 0.1mJ for the single-mode fibres.

- The repetition rate of a pulsed laser source is also important in the area of interface between the fibre core and laser beam. If the peak power is high the repetition rate must be slow enough to permit dissipation of the lost energy between pulses, otherwise these will be a heat buildup and associated damage to the cladding and core.

- Air may be ionized and a breakdown may occur on the entrance face of the fibre or in its immediate vicinity.

In spite of those drawbacks, holograms of small objects have been recorded through multimode and single-mode fibres with the ruby laser by Albe *et al* [129]. Such an holographic systems is shown in Fig. 4.14.

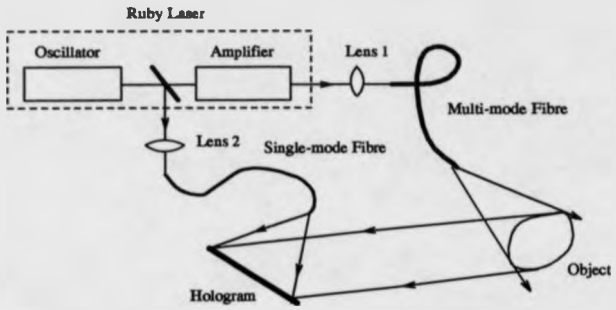


Figure 4.14: Holographic recording system through multimode and single-mode fibres using ruby laser (Albe *et al* 1984).

The advantage of the system described in Fig. 4.14 is that the energy of the reference beam is 100 times lower than that of the object beam before arriving in the fibre, and therefore the energy available of the reference beam on the exit face of the fibre did not exceed 0.1 mJ if the output of the ruby laser is 200 mJ.

As mentioned above, the use of fibre optic pulsed holography will open a complete new field of industrial applications such as holograms made from the inside of machine parts (e.g. engines, fuel injection particle distribution), PIV (Particle Image Velocimetry) as well as many new holographic non-destructive testing. Another important new field will be found within medicine and biology [128].

4.5.3 Endoscopic Holography Using Fibre Optics

The formation of fringes in endoscopic holographic interferometry using fibre optics has been studied in connection with medical diagnosis of living body cavities, non-destructive testing of the human ear canal and tympana membrane by von Bally [128]. He used a pulsed ruby laser and fibres for both the reference and object beam paths. This technique has been called endoscopic holographic interferometry or simply holographic endoscopy. A multimode fibre bundles was used as the object illuminating beam, while a single-mode fibre carried out the reference beam. The exit aperture of the multimode fibre was connected to a

rigid speculum which was inserted into a canal or inner part of a machine. The object illumination was transmitted along the periphery of speculum, while the light reflected from the object was imaged down the central core.

Motoki *et al* [131] investigated endoscopic holography using fibre bundles instead of single-mode fibres, to transmit high power laser light for object illumination to prevent the light path break caused by the high power density of concentrated laser light. They pointed out that the effect of the extension of the source is equivalent to that of the aperture, and the fringe visibility is given by the product of both effects. If the source size is smaller than the aperture size, the visibility reduction may be tolerable. The endoscopy holographic system is shown in Fig. 4.15.

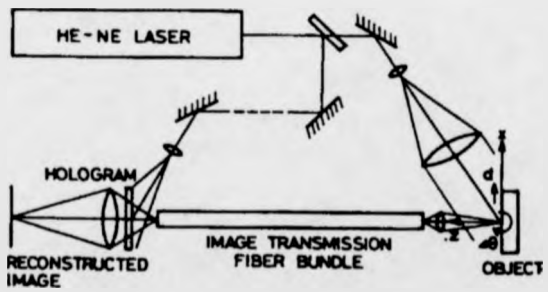


Figure 4.15: Experimental configuration for endoscopy holography (Yonemura *et al* 1981).

4.5.4 ESPI Using Fibre Optics

The Electronic Speckle Pattern Interferometry or ESPI was first demonstrated by Butters and Leendertz [132]. The major feature of ESPI is that it enables real-time correlation fringes to be displayed directly on a television monitor without photographic processing. The introduction of fibre optics in ESPI plays a key role in the miniaturisation of certain ESPI system and also their uses in environments

normally inaccessible to conventional optics. Tatam [34, 133] *et al* reported the ESPI surface contouring using optical fibres and wavelength modulation of laser diodes. The principle of ESPI has been reviewed in chapter 2, and several different contouring techniques will be discussed in the next chapter.

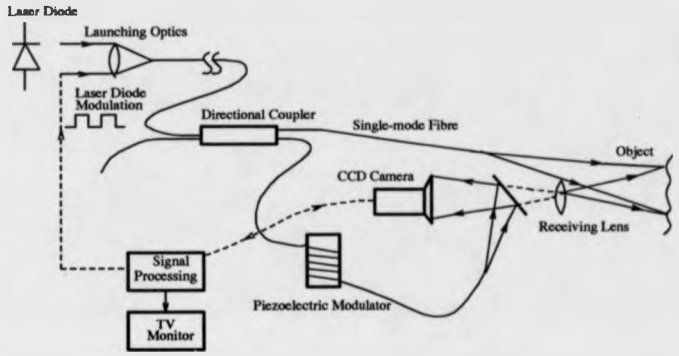


Figure 4.16: Experimental arrangement for ESPI surface contouring using fibre optics (Tatam *et al* 1990).

In Tatam's work, the optical-electronic arrangement of experimental system is shown in Fig. 4.16. The laser diode (50mW) is used as a light source. The advantage of using laser diode is that the wavelength modulation can be achieved by changing the injection current of the laser diode. Therefore, the two wavelength contouring technique can be applied to the surface contouring. The use of fibre optics in the ESPI system greatly simplifies the optical arrangement of experiment. The object and reference beams are both delivered by flexible single-mode fibres. This allows the ESPI system to access the difficult environment. Such ESPI system based on the fibre optics can be used as a real-time measurement in industrial measurement. By applying piezoelectric modulator in the reference beam, phase-stepping fringe analysis can be readily achieved. In addition, single-mode fibre also provide excellent spatial filtering of object and reference beam providing high quality fringes. The use of optical fibres in ESPI system have also demonstrated by Vrooman *et al* [134] and Kujawinska *et al* [135].

4.5.5 Application of Fibre Optics in Sensor Technology

There has been a great deal of work completed using fibre optics for practical sensing. Low cost fibres open new applications in interferometric temperature and pressure sensing [120].

In the case of temperature sensing, a fibre optic sensor relies on the change of the coefficient of thermal expansion at the core/cladding interface. To sense temperature, the first step is to convert the sensing temperature into the relative linear or axial change in the fibre core and cladding. This dimensional change is then converted into the corresponding fringe patterns. The second step is to calculate the fringe changes separately for the core and cladding. The third step is to calculate the overall system fringe changes. Wohlstein [120] reported that resolution of 1 fringe change per 0.5^0 C is achieved. Much higher resolution can be obtained, depending on the experimental setup, conditions, mounting and other factors.

Similar to the temperature sensing, pressure sensing involves three steps. The first step is to convert sensed pressure into relative volume change. Second, convert the volume information into linear or axial change. Third step is to convert this dimensional change into corresponding fringe patterns. Resolution of 1 fringe change per $0.38 kg/cm^2$ is achieved by Wohlstein [120].

Jackson and Jones [121, 122] reviewed the extrinsic fibre-optic sensors for remote measurement. Fibre optics provide a practical and economic solution to making traversed measurements. Particularly, in restricted access locations, fibre optics may give the only feasible means of measurement. The optical quality of the guided beam by single-mode fibres is much better than when formed directly from the laser.

4.5.6 Experimental Demonstration

Experiments were conducted on different objects using single-mode fibres for CW laser. It has been shown that it is feasible to use single-mode optical fibres in the

recording and reconstruction of holograms of ordinary scenes. The system based on the fibre optics is very flexible. It can be applied to holographic contouring, holographic deformation measurements using carrier fringe and FFT techniques. There are no problems with coherence, wavefront shape and resolution present. Multimode fibres can also be used, especially for pulsed holography.

Figure 4.17 shows the schematic diagram of the experimental arrangement for recording and reconstruction of holograms.

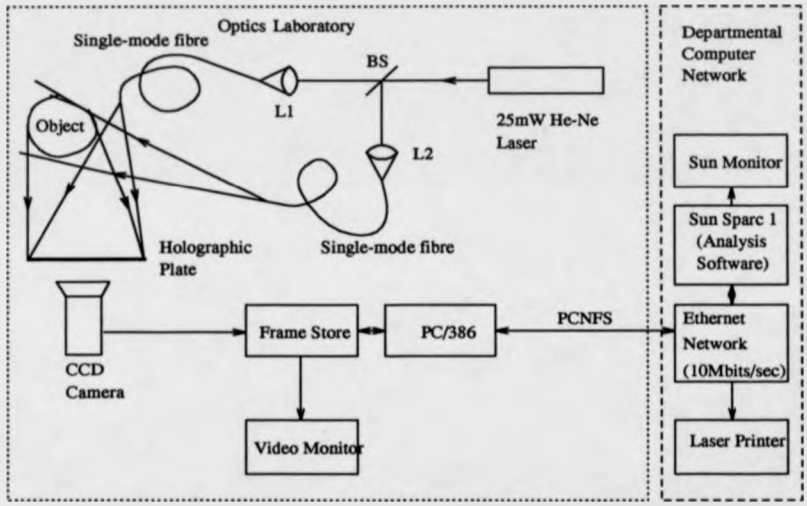


Figure 4.17: Schematic of the experimental arrangement for recording and reconstruction holograms.

An He-Ne laser operated at 0.6328μm with maximum power of about 25mW was employed. The optical fibre used was the Newport FSV-10 as single-mode for wavelength longer than 600nm. The light from a laser is a linearly polarised and divided into two beams by a beam splitter. Two 20X microscope objectives were used to couple the light into the fibre ends. The launching efficiency about 60% were obtained. One beam was arranged to illuminate an object and the other arranged to illuminate the hologram itself. The reference fibre end was at a large distance (1540mm) from the hologram to provide a reasonably uniform

illumination over entire holographic plate. The holograms were recorded in AGFA 8E75 plates, which were bleached in order to increase the diffraction efficiency. The exposure time was in the order of 4-5 second.

To maximise coupling into a single-mode fibre, the incident field distribution must be matched to that of the fibre mode. In this experiment, the fibre mode field diameter is given by [136]

$$w = 2a(0.65 + \frac{1.619}{V^{1.5}} + \frac{2.879}{V^6}) \tag{4.9}$$

For the Newport Corp. F-SV fibre, $V = 2$, the Gaussian width is approximately 28% larger than the core diameter, so the light should be focused to a spot size 1.28 times the core diameter at the fibre surface.

A microscope objective should be matched to laser and single-mode fibre by[15]

$$f = D(\frac{\pi}{4\lambda}) \tag{4.10}$$

where f is the objective focal length, D is the laser beam diameter at the coupler, λ is the laser wavelength. In this experiment, two M-20X microscope objectives were employed.

The Fresnel type hologram may then be reconstructed to produce the original image of the object as it was illuminated by the reference beam only. The reconstructed image was picked up by a CCD camera, digitised and stored in PIP digital frame store. An IBM compatible PC/386 was used to control the image capture, store and transmit. A software tool called *photo* was developed to make the snap and continuous frame grab. The images can be displayed on the monitor. An Ethernet card was used to connect the PC and Sun Sparc Workstation. In this case, the images can be sent to the Sun Sparc Workstation and displayed on the Sun monitor. The hardcopy of the holographic images was printed out by the laser printer.

4.6 Discussion

In this chapter, it was demonstrated that holograms can be recorded using fibre optics. Compared with conventional holography, fibre holography can be used to record remote objects, and makes the holographic recording task much easier. The reconstructed holographic fringe patterns can be digitised and with the use of a carrier fringe technique, may be interpreted automatically and unambiguously. This will be dealt with in next chapter.

In holographic interferometry, a fibre bundle can also be used to image the object to the holographic plate. In results presented, all of the holograms were recorded using the single-mode fibre optics. Single-mode fibres offer distinct advantage over the multimode fibres. For a single-mode fibre, surface speckle is substantially more stable, as compared to a multimode fibre.

Fibre optic holography has already found many applications in interferometric investigations using CW and pulsed lasers. The use of fibre optics in holography can deal with many interesting fields in which investigations have been difficult or impossible to perform. The important new fields will be found in medicine and biology. In general, using fibre optics in holographic interferometry makes it possible to vary the direction of object and reference beams, in turn, this makes more accurate deformation and displacement measurement. The pulsed laser combined with the fibre optics will be found in many industrial applications, such as taking holograms inside machine parts, for example, fuel injection, particle distribution etc. The new holographic equipment in which a pulsed laser and fibre optics are integrated in one unit will emerge in the future.

Chapter 5

Double-Source Holographic Contouring Using Fibre Optics

5.1 Introduction

Surface contouring by interferometry is widely used to determine the shape of a surface [36, 37, 38]. In many industrial metrology applications, this non-contact non-destructive automated surface shape measurement is a desirable tool for strain analysis [137], vibration analysis [138], and contour mapping [139]. Recently, this technique was applied to automatic measurements of surface defects in artwork [6]. In this work, the contouring technique is used to determine the position, the depth and the volume of a surface or subsurface defects in ancient wooden monuments. Contour generation is the formation of an image of an object, on which the object's surface intersect with a set of equally spaced planes parallel to the reference plane. The contour lines or fringes characterise the relief of the surface.

There are many ways to contour an object using interferometry. Some existing methods for surface contouring include the shadow moire method [140], the projection moire method, the ESPI method [133] and holographic method [5, 141]. The problem of contouring objects has been discussed from the viewpoint of

coherent optics by Hildebrand and Haines [37]. They described a holographic contouring technique using a two-frequency source. This technique generates contours on the object with a resolution that depends on the wavelengths of the source. Another method presented by Hildebrand and Haines, uses two-source holography for generation of surface contouring. In contrast to the first method. the two source technique provides a contour map whose resolution can be varied from the order of a wavelength up to the centimeter range. Later, Abramson [142] reported the sandwich holographic contouring technique. The contour maps were obtained by first tilting a plane-parallel glass plate in the illumination arm of the interferometer during recording, and then rotating the sandwich hologram during reconstruction. An automatic sandwich holographic contouring technique has recently been reported by Carelli *et al* [6] as applied to the automatic measurements of surface defects in artwork.

A different surface contouring approach, using incoherent light, is provided by moire interferometry [38, 139, 140]. The moire contouring technique is used currently in some measurement applications as well as in stress analysis [143]. An automated surface measurement technique using moire contouring has been proposed by Idesawa *et al* [38, 144].

Thalman *et al* [141] reported holographic contouring using electronic phase measurement. This technique is based on the two illumination source method and the use of a microcomputer for automated data reduction. Recently, Tatam *et al* [133] presented a paper on the development of an Electronic Speckle Pattern Interferometry (ESPI) surface contouring system, based on laser diodes and monomode fibre optics.

The aim of this chapter is to review different interferometric surface contouring techniques, and to present a novel holographic surface contouring technique based on the two-source method and fibre optics system. The reconstructed holographic contouring interferograms are captured by a CCD camera and stored in a digital frame buffer. The contouring fringes can be displayed on a monitor or transferred to a Sun Workstation for automatic fringe processing. With the aid

of the Fourier transform, a 3-D mesh plot can be obtained for different object shape measurements.

5.2 Moire Surface Contouring Techniques

Moire fringe patterns are formed when two gratings are superimposed. The resultant moire fringes correspond to a contour lines system of the object under certain conditions. This technique is widely used in experimental research in mechanics and it is also used for measuring a 3-D shape.

5.2.1 Moire Shadow Contouring Technique

Moire shadow contouring was proposed by Takasaki and Meadows *et al* [38, 139] in 1970. The technique was developed by use of moire to observe contour lines of an object. The shadow of two equidistantly placed plane gratings is projected onto an object by a point source and observed through the grating, as shown in Fig. 5.1. The resulting moire fringe is a contour line system showing equal depth from the plane of grating if the light source and observing point lie on a plane parallel to the grating. This technique has the property that constant height contours of equal separation are produced on the object. The position of these contours are then used to determine the shape of the surface in much the same ways as a topographic map is used. It has the advantage of non-contact contouring. Moreover, unlike holographic contouring, stability is not required. However, the depth of field of the moire contouring system is limited by diffraction, while the depth of the holographic contouring system is limited by the depth field of the hologram. Therefore, the resolution of the holographic contour method is of the order of a wavelength, while the resolution of the moire method is obtained down to about 25μm[140].

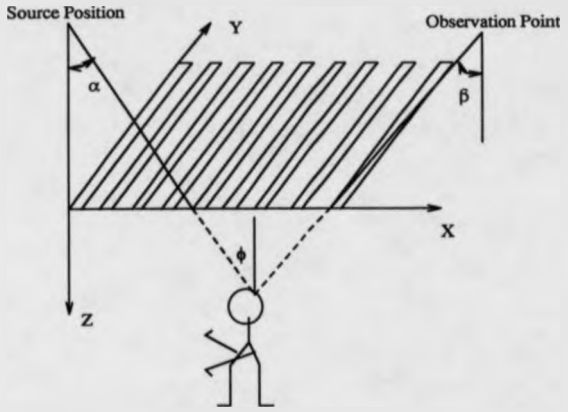


Figure 5.1: Shadow moire contouring system with the source and observer at infinity (Meadows *et al* 1970).

5.2.2 Projection Moire Contouring Technique

An alternative to the shadow moire contouring technique is the projection-type moire contouring technique which has been introduced by Miles and Speight [145]. Later, this technique was developed by Benoit *et al* [146]. The projection system forms the image of a linear grating onto the object surface under test. The object is viewed through another grating. If the object is not plain, lines of the grating visualised by the detection system in the plain of the detection grating create moire fringes which carry information about the shape of the object. The optical arrangement of projection moire contouring is shown in Fig. 5.2. In this method, two systems of fringes are observed, the fringes of the object due to the structured illumination, the fringes formed between projection grating and viewing grating. These two fringe system are multiplied, leading to a new type of moire fringe. In practice, this technique is very efficient and is not influenced by the surface roughness of the objects. The visibility of the contour lines is good.

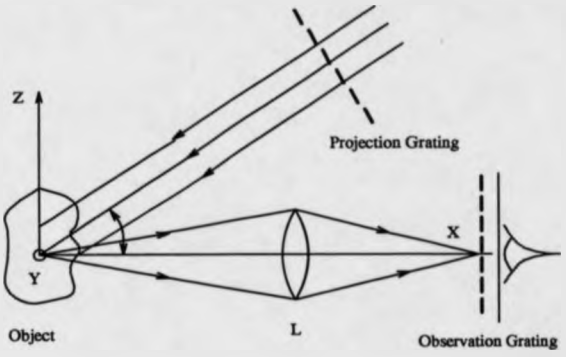


Figure 5.2: Projection-type moire contouring system (Benoit *et al* 1975).

5.2.3 Automatic Measurement of 3-D Shape by Moire Interferometry

Two-dimensional moire fringe patterns can be used for extracting qualitative information, but are sometimes extremely tedious and time-consuming to evaluate quantitative data from the fringe pattern. Yatagai *et al* [144] developed automatic measurement of 3-D shapes using moire techniques. Moire fringes are generated by an observing device as shown in Fig. 5.3. A moire fringe is observed on a TV monitor. The accuracy of the measurement is considerably influenced by misalignment of the directions θ and ϕ of the projection system. The technique is applicable to real-time measurement by means of appropriate scanning devices. This system can be used to measure area of a surface, or the vibration of a surface.

Dai and Chiang [143] recently reported double-exposure moire interferometry technique. In this technique, a curved surface coated with light sensitive material is exposed twice in a volume of virtual gratings formed by the interference of two coherent light beams. The advantage of this method is that it offers both good fringe quality and easily adjustable high sensitivity.

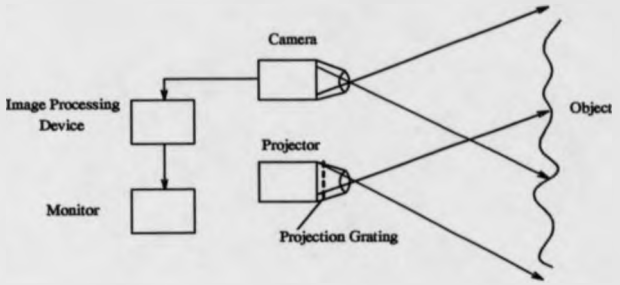


Figure 5.3: Schematic diagram of scanning moire contouring (Yatagai & Idesawa 1982).

5.3 ESPI Surface Contouring Technique

Electronic Speckle Pattern Interferometry (ESPI) is widely used in industrial measurement. This technique was first demonstrated by Butters and Leendertz [132]. The major advantage of ESPI is that it enables real-time fringes to be displayed on a television monitor without any form of photographic processing. Bergguist *et al* [147] described a technique for surface contouring by ESPI. They demonstrated that the correlation fringe contours obtained were equivalent to those of a projected two-beam interference fringe contour. In their work, the double-exposure technique is used. Between exposures, the direction of the object illumination is slightly altered, and the first speckle pattern is subtracted from the second. The change of phase is due only to the rotation of wavefront incident on the object.

Recently, Tatam *et al* [133] has reported an ESPI surface contouring system using laser diodes and single-mode fibre optics. This technique is based on the two wavelength method. Wavelength modulation is achieved by modulating the injection current of a laser diode. The system is compact and flexible, containing fibre optics for illumination, a CCD camera, and a laser diode. The contour fringe interval is in the range of 0.5 to 5mm. However, the drawback of this technique is that the visibility of the fringes is poor, and fringe quality is lower than that

of holographic contouring.

5.4 Holographic Contouring Technique

Holographic contouring is a well-known method of determining the shape of an object. The contour fringe pattern is generated by the interference of two reconstructed images of a double-exposure hologram. There are various types of holographic contouring, including sandwich holography [142], the two-index method [35, 148, 149], the two-wavelength method [33, 37, 150, 151, 152] and the two-source method [4, 5, 153]. The applications of holographic contouring include inspection of very flat surfaces with high resolution, and recording the shape of the surface of a 3-D object.

5.4.1 Sandwich Holographic Contouring Technique

Sandwich holography was first proposed by Gates [154] to study air flows, and subsequently developed by Abramson for holographic contouring as well as other problems in mechanics [142]. In this contouring technique, after one holographic plate is exposed, the point source of light illuminating the object is displaced laterally to produce a sheared illumination beam. A second exposure is made on another plate. After their exposures and photographic processing, two holograms are obtained and bonded together with their emulation outward. When illuminating them with a reference beam, contouring fringes are formed that represent the intersection of the object by a set of interference surfaces. By using sandwich holography, the contouring interference fringe surfaces can be rotated through an angle close to 180^0. By tilting the sandwich hologram through quite a small angle during reconstruction, the fringe patterns can be rotated in any direction of 3-D reconstructed object surface. Abramson [142] derives the relationship between the rotation angle of the hologram and the rotation angle of the interference surfaces, but recommends that calibration techniques should be

used for measuring purposes using reference objects with known dimensions and angles. The advantage of this technique is that the sign of the slant of the object (uphill or downhill) is easily found, and it also combines the illumination and observation shearing. Fringes with good contrast are obtained by this technique. The drawback of the method is that it requires a special kinematic fixture for a precise return of the hologram back to the position at which it was recorded.

5.4.2 Two-Index Method (Immersion Method)

The immersion holographic contouring technique was first reported by Tsuruta *et al* [35, 148, 149]. Zelenka *et al* described a multiple-index holographic contouring technique which is an improved two-index method. The object under study is placed into an immersion cell as shown in Fig. 5.4.

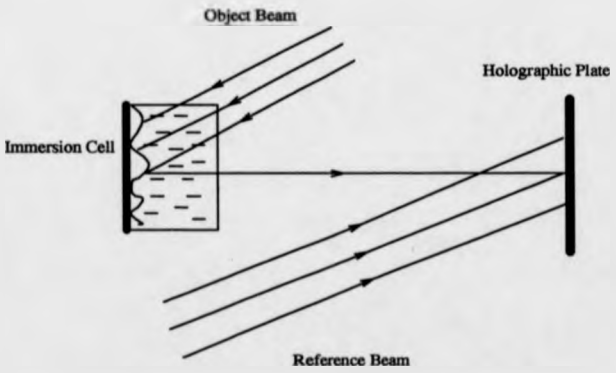


Figure 5.4: Schematic arrangement of the immersion method (Ostrovsky *et al* 1991).

The double-exposure technique is employed in the immersion method. Before the first exposure, the immersion cell is filled with a transparent liquid or gas of refractive index n_1, and before the second exposure it is filled with a medium of refractive index n_2. The resulting change in the index of refraction introduces an optical path difference. Therefore, a pair of slightly displaced images form

during the reconstruction process. A schematic diagram indicating that how the calculation of the optical path difference in the immersion method is made is shown in Fig. 5.5.

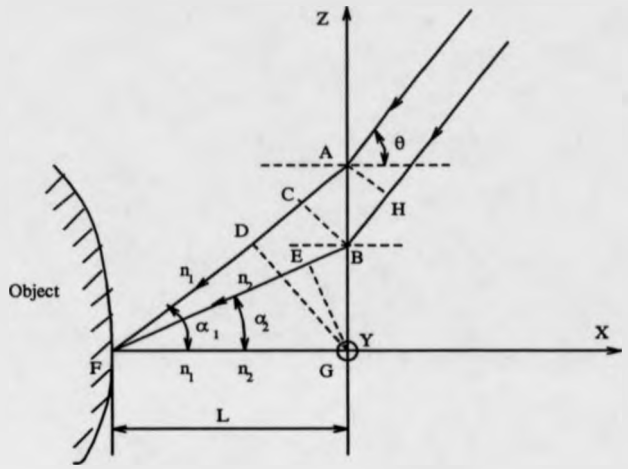


Figure 5.5: Geometry of calculating the path difference in the immersion method.

The two beams arriving at the point F on the surface of an object are reflected from it in a direction perpendicular to the window of the immersion cell. The optical path difference of two beams with two different refractive indexes n_1 and n_2 is as follows

$$\Delta l = (AF + L)n_1 - (BF + L)n_2 - HB \tag{5.1}$$

From Fig. 5.5, $AF = L/\cos\alpha_1$, $BF = L/\cos\alpha_2$, $HB = L(\tan\alpha_1 - \tan\alpha_2)\sin\theta$, Eq. 5.1 can be written as

$$\Delta l = (\frac{L}{\cos\alpha_1} + L)n_1 - (\frac{L}{\cos\alpha_2} + L)n_2 - L(\tan\alpha_1 - \tan\alpha_2)\sin\theta \tag{5.2}$$

According to Snell's law, $\sin\theta = n_1\sin\alpha_1, \sin\theta = n_2\sin\alpha_2$, the Eq. 5.2 be-

Table 5.1: Fringe spacing in immersion method

Materials	**Refractive Index**	**Fringe Spacing**
Air	1.00028	1 – several mm
Water & Ethylene Glycol	1.333, 1.427	3.4μm
Water & Ethyl Alcohol	1.333, 1.360	11.7μm

comes

$$\Delta \mathrm{l} = \mathrm{L}(n_1 - n_2 + \sqrt{n_1^2 - \sin^2\theta} - \sqrt{n_2^2 - \sin^2\theta}) \tag{5.3}$$

The condition for the fringe maximum has the form

$$\Delta \mathrm{l} = (2K + 1)\frac{\lambda}{2} \tag{5.4}$$

where $K = 0, 1, 2, 3.....$ Substituting Eq. 5.4 into Eq. 5.3, we get

$$\mathrm{L} = \frac{(2K + 1)\lambda}{2(n_1 - n_2 + \sqrt{n_1^2 - \sin^2\theta} - \sqrt{n_2^2 - \sin^2\theta})} \tag{5.5}$$

The value of a fringe corresponds to a change of Δl in one wavelength, therefore,

$$\Delta \mathrm{l} = \mathrm{L}_{K+1} - \mathrm{L}_K = \frac{\lambda}{n_1 - n_2 + \sqrt{n_1^2 - \sin^2\theta} - \sqrt{n_2^2 - \sin^2\theta}} \tag{5.6}$$

When illuminating the object from the side of the hologram ($\theta \doteq 0$), Eq. 5.6 becomes simplying to

$$\Delta \mathrm{l} = \frac{\lambda}{2(n_1 - n_2)} \tag{5.7}$$

By using materials with different refractive indexes, the fringe interval can range from a few micrometer to several millimeters. Table 5.1 shows the fringe interval of different materials in the immersion method [148, 155, 156, 157].

5.4.3 Two-Wavelength Method

The two-wavelength holographic contouring technique was first presented by Hildebrand and Haines [37]. The hologram is generated by using two closely spaced wavelengths λ_1 and λ_2. Such a two-wavelength hologram can be considered as an incoherent superposition of two holograms. When reconstructed, each of these holograms gives an image of the surface in the light of the same wavelength of the reconstructing source λ_3. In this case, the contour spacing is given by $\frac{\lambda_1\lambda_2}{2\Delta\lambda}$, when λ_1 and λ_2 are the two wavelengths and $\Delta\lambda = |\lambda_1 - \lambda_2|$. Contours as small as 9.25μm using 632.8nm and 611.8nm light from a He-Ne laser have been reported by Zelenka *et al* [150]. Holographic contour maps were made using two frequencies from a pulsed ruby laser by Heflinger *et al* [33]. Multifrequency emissions arise from the use of a resonant output reflector in the laser cavity, producing contour intervals equal to the optical thickness of the resonator. The feasibility of extending contour mapping with dye lasers to very large objects was reported by Schmidt *et al* [152].

Figure 5.6 shows the calculation of phase difference in the two-wavelength contouring method [72].

In Fig. 5.6, the object is illuminated with a plane wave containing light of two wavelengths. The phase is measured from the plane A (dashed line). The phases of the illuminating wave for the two-wavelengths at point B are

$$\phi_{B1} = \frac{2\ \pi\ y_0}{\lambda_1} \tag{5.8}$$

$$\phi_{B2} = \frac{2\ \pi\ y_0}{\lambda_2} \tag{5.9}$$

where y_0 is a point of the object in the coordinate system. The phases of the spherical waves incident from point B are

$$\phi_1 = \frac{2\ \pi\ y_0}{\lambda_1} + \frac{2\ \pi\ r}{\lambda_1} \tag{5.10}$$

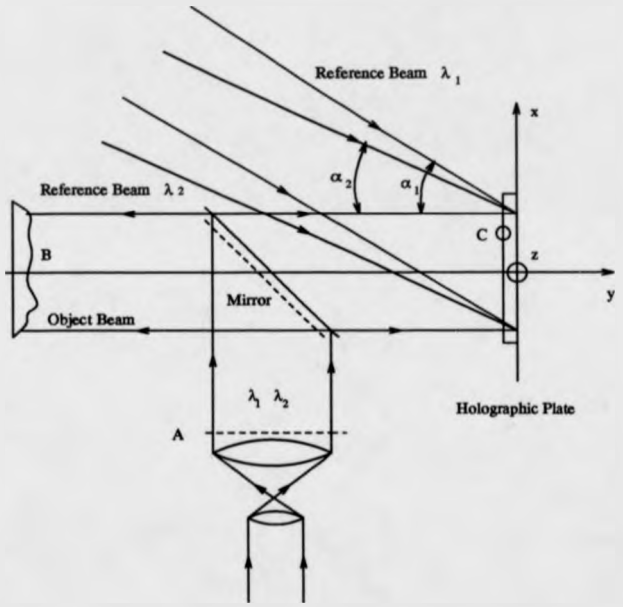


Figure 5.6: Phase difference calculation in the two-wavelength method of surface contour mapping (Ostrovsky *et al* 1991).

$$\phi_2 = \frac{2\,\pi\,y_0}{\lambda_2} + \frac{2\,\pi\,r}{\lambda_2} \tag{5.11}$$

where r is the distance from point B to an arbitrary point C on the holographic plate with the coordinate x measured from the centre of the hologram.

$$r = [y_0^2 + (x - x_0)^2]^{1/2} \doteq y_0[1 + \frac{1}{2}(\frac{x - x_0}{y_0})^2] \tag{5.12}$$

In the small angle approximation, $r = y_0$, therefore,

$$\phi_1 = \frac{4\,\pi\,y_0}{\lambda_1} \tag{5.13}$$

$$\phi_2 = \frac{4\,\pi\,y_0}{\lambda_2} \tag{5.14}$$

As shown in Fig. 5.6, the two reference waves arrive at the hologram under the angles α_1 and α_2. The phases of these waves are

$$\phi_{r1} = \frac{2\,\pi\,x\,\alpha_1}{\lambda_1} \tag{5.15}$$

$$\phi_{r2} = \frac{2\,\pi\,x\,\alpha_2}{\lambda_2} \tag{5.16}$$

By considering Eqs. 5.13, 5.14, 5.15 and 5.16, the phase difference between the object and reference beams are given by

$$\phi_1 - \phi_{r1} = \frac{4\,\pi\,y_0}{\lambda_1} - \frac{2\,\pi\,x\,\alpha_1}{\lambda_1} \tag{5.17}$$

$$\phi_2 - \phi_{r2} = \frac{4\,\pi\,y_0}{\lambda_2} - \frac{2\,\pi\,x\,\alpha_2}{\lambda_2} \tag{5.18}$$

By illuminating the hologram with a plane wave incident under an angle α_3 with the wavelength λ_3, the phases of reconstructed hologram are

$$\phi_{1c} = \phi_1 - \phi_{r1} + \frac{2\,\pi\,x\,\alpha_3}{\lambda_3} = \frac{2\,\pi}{\lambda_1}\,(2\,y_0 - x\,\alpha_1) + \frac{2\,\pi\,x\,\alpha_3}{\lambda_3} \tag{5.19}$$

$$\phi_{2c} = \phi_2 - \phi_{r2} + \frac{2\,\pi\,x\,\alpha_3}{\lambda_3} = \frac{2\,\pi}{\lambda_1}\,(2\,y_0 - x\,\alpha_2) + \frac{2\,\pi\,x\,\alpha_3}{\lambda_3} \tag{5.20}$$

The phase difference of these waves in the plane of the hologram is

$$\Delta\phi = \phi_{1c} - \phi_{2c} = 4\,\pi\,y_0\,(\frac{1}{\lambda_1} - \frac{1}{\lambda_2}) - 2\,\pi\,x\,(\frac{\alpha_1}{\lambda_1} - \frac{\alpha_2}{\lambda_2}) \tag{5.21}$$

In order to avoid the displacement of the reconstructed images corresponding to different wavelengths, the following condition is derived [72]

$$\frac{\alpha_3}{\lambda_3} = \frac{\alpha_2}{\lambda_2} = \frac{\alpha_1}{\lambda_1} \tag{5.22}$$

By taking into account Eq. 5.22, the phase difference in the hologram is given by

$$\Delta\phi = \frac{4\,\pi\,y_0\,\Delta\lambda}{\lambda_1\,\lambda_2} \tag{5.23}$$

Because a change of the phase difference of $2\,\pi$ corresponding to the distance between the maximum of adjacent interference fringes, therefore,

$$\Delta y = \frac{\lambda_1\,\lambda_2}{2\,\Delta\lambda} = \frac{\lambda^2}{2\,\Delta\lambda} \tag{5.24}$$

The advantage of the method is that the depth distance between these contour fringes can be freely varied by varying the wavelength difference between the laser lines.

Table 5.2 shows the different type of lasers used in the two frequency method.

Figure 5.7 [155] shows the contour spacing formed when using the different type of lasers given in Table 5.2.

Table 5.2: Type of lasers used in two-wavelength contouring

Laser	**Wavelength (nm)**					
He-Ne Laser	632.8	611.8				
Diode Laser	635 – 830					
Xe^+ Laser	539.5	535.3	526.1	515.0	500.9	495.4
Ar^+ Laser	514.5	501.7	526.1	515.0	500.9	495.4
Ruby Laser	694.3					
Dye Laser	400.0–700.0					

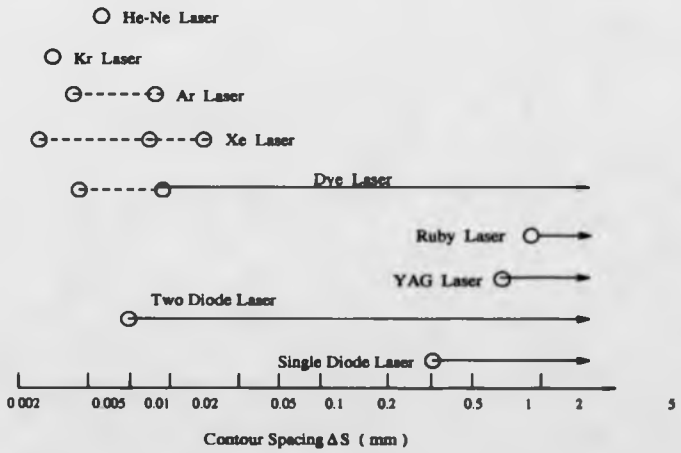


Figure 5.7: The contour spacing of two-wavelength method.

5.4.4 Two-Source Method

The contour fringe pattern on the object can be generated from the interference of two different illumination sources. This technique was first proposed by Hildebrand and Haines [37]. Figure 5.8 shows a schematic diagram of the two-source method. S_1 and S_2 are two coherent illumination sources. The interference phase at the object point D can be calculated from the path difference between S_1 and S_2. The hologram is recorded by the double-exposure technique. During the first exposure, the hologram is illuminated with S_1 at an angle θ_1, and during the second exposure with S_2 at an angle θ_2. Assuming the holographic plate is perpendicular to the viewing direction when the hologram is reconstructed, the object D is viewed through the point C on the hologram.

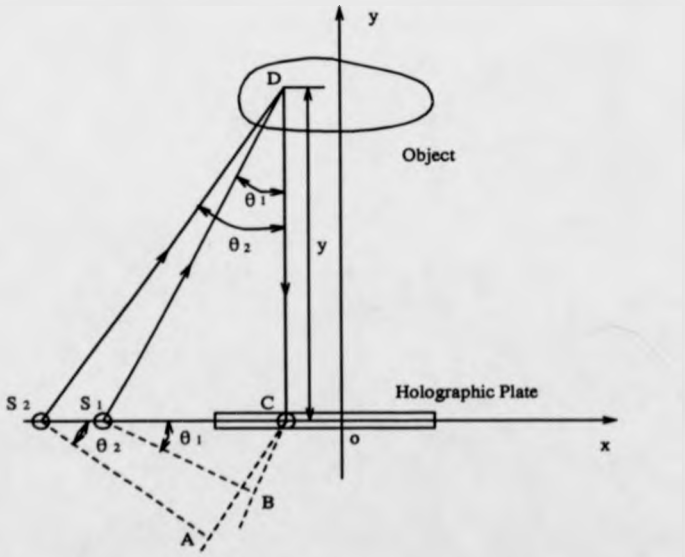


Figure 5.8: Contour map generation with the two-source method.

The interference phase $\phi(x, y)$ at the point D is given by

$$\phi(x, y) = k(\Delta S_1 - \Delta S_2) \tag{5.25}$$

where $k = \frac{2\pi}{\lambda}$ and ΔS_1 and ΔS_2 are the path difference at the point D between the object and reference beams in the first and second exposures. From Fig. 5.8, the ΔS_1 and ΔS_2 are given by

$$\Delta S_1 = S_1D + DC - CB \tag{5.26}$$

$$\Delta S_2 = S_2D + DC - CA \tag{5.27}$$

where $S_1D = y/\cos\theta_1, S_2D = y/\cos\theta_2, DC = y, CB = y\tan\theta_1\sin\theta_1, CA = y\tan\theta_2\sin\theta_2$. Therefore, the phase difference ϕ is

$$\phi = k(\frac{y}{\cos\theta_1} - \frac{y}{\cos\theta_2} - y\tan\theta_1\sin\theta_1 + y\tan\theta_2\sin\theta_2) \tag{5.28}$$

Equation 5.28 shows that the phase difference is proportional to the distance between the source and the points on the object surface. Bright fringes are observed for phase values $\phi = 2\,\pi n, (n = 0, 1, 2...)$. The phase change corresponding to one fringe is

$$\Delta\phi = \phi_{n+1} - \phi_n = \frac{2\pi}{\lambda}\Delta y(\frac{1}{\cos\theta_1} - \frac{1}{\cos\theta_2} - \tan\theta_1\sin\theta_1 + \tan\theta_2\sin\theta_2) \tag{5.29}$$

By manipulating Eq. 5.29, the value of one fringe Δy can be found

$$\Delta y = \frac{\lambda}{2\sin[(\theta_2 + \theta_1)/2]\sin[(\theta_2 - \theta_1)/2]} \tag{5.30}$$

Equation 5.30 can be simplified by setting $\theta = (\theta_2 + \theta_1)/2$, and $\Delta\theta = (\theta_2 - \theta_1)/2$, thus

$$\Delta y = \frac{\lambda}{2\sin(\theta)\sin(\Delta\theta)} \tag{5.31}$$

The sensitivity of the measurement described here is determined by the wavelength of the light source, and the angle between the illuminating and viewing

directions. If $\theta_1 = 180^0$, $\theta_2 = 0$, then the sensitivity is at its greatest. When $\theta_1 = \theta_2$, the contour interval is infinite and the sensitivity of its measurement is zero. The obvious drawback of the method is that portions of the surface of the object are in shadow. These are the areas of the object that can not be illuminated with such an arrangement of light sources. The solution to this problem is to illuminate the object from many directions. Another approach to solving this problem is to illuminate the object at an oblique angle to the line of observation.

Figure 5.9 shows a contour interferogram of a ball on top of a cylinder. The contour interval is 3.1mm. The shadow effect is clearly seen in Fig. 5.9.

Figure 5.9: Holographic contouring fringes on a ball and a cylinder (image size 512 × 512).

5.4.5 Discussion

All of the techniques in holographic contouring described above can produce fringes which correspond to contours of equal height on the object. Compared with moire fringe contouring, the holographic contouring method uses a coherent light source.

The sensitivity of holographic contouring fringes can be in the range of 1μm up

to several millimeter. Figure 5.10 shows the contour sensitivity of the two-source method for different angles of separation of the two sources.

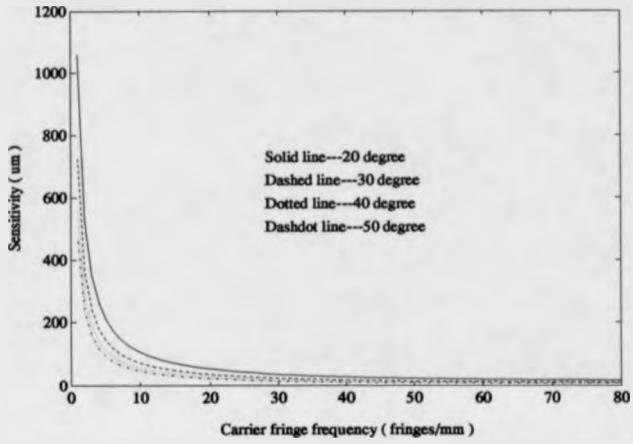


Figure 5.10: Contour sensitivity under different fringe frequencies for the two-source method for different values of θ.

In all techniques, the same contour interval is determined either by fringe spacing or the angle between the illumination and viewing directions. The FFT and phase-stepping techniques can be applied to any of the techniques described to produce quantitative height information, as described in the next section.

The FFT and phase-stepping techniques determine the surface height according to the relevant fringe patterns. In the case of the FFT approach, if the contour fringes are straight and equally spaced in object space, the carrier fringe technique can be applied to these fringe patterns. The straight and spaced fringes are the carriers. If the spatial carrier frequency is properly selected, the surface profile can be produced using the inverse Fourier transform and phase-unwrapping techniques. The contouring fringes are viewed directly by a CCD camera, the output is digitised into an array of 512 × 512 pixels and stored in 512 × 512 × 8 memory. The image is converted into a TIF (Tagged Image Format) file. The

surface height can be determined by computing the phase information. It has been shown that it is an advantage to use as many fringes to provide as high a spatial carrier frequency as possible, providing it is within the resolution of the detection device.

The sandwich holographic contouring method makes it possible to rotate a fringe pattern in any direction of 3-D reconstructed object space. This is achieved simply by tilting the sandwich hologram through a much smaller angle during reconstruction. This method has the disadvantage of requiring complicated repositioning device.

The two-index method has a higher contouring sensitivity, as small as 0.75λ [148] using liquid and air. However, this method has the disadvantage of requiring two exposures to produce contours, therefore, eliminating transient or moving objects.

The two-wavelength method has the advantage that the depth distance between these contour fringes may be freely varied by varying the wavelength difference between the laser lines. In the case of the two-source method, the interference fringes may be produced between the reference wavefront and the tilted wavefront. This method is easy to implement. However, the method has a low sensitivity of measurement. Rastogi [158] described a fresh approach to enhance the sensitivity: with this method the sensitivity can be as high as 14μm. The tilted wavefront can be generated by different methods. Table 5.3 shows a direct comparison of the different contouring methods described above.

5.5 Double-Source Holographic Contouring Using Fibre Optics

5.5.1 Double-Exposure Holography

In double-exposure holography, when the object beam is displaced between the exposures, a fringe pattern is produced. As shown in Figure 5.11 a hologram

Table 5.3: Comparison of different contouring methods

Charateristics	**Moire**	**ESPI**	**Two Source**	**Two Index**	**Two Frequency**
Maximum Resolution (μm)	25	10	14	3.4	3.5
Fringe Visibility	Good	Poor	Good	Good	Good
Light Source	Incoherent	Coherent	Coherent	Coherent	Coherent
Real-Time	Yes	Yes	Yes	No	Yes

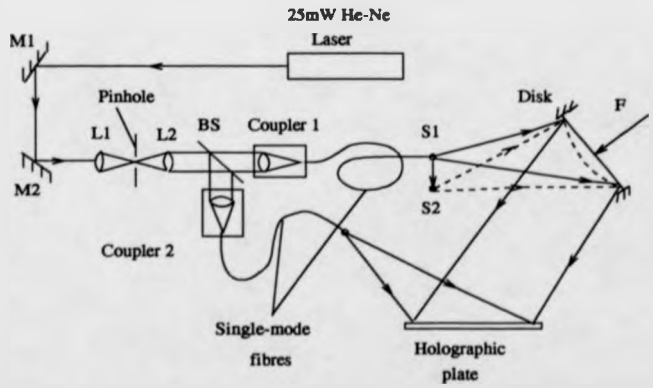


Figure 5.11: Experimental arrangement for recording contour holograms by the double-source method using fibre optics.

is exposed twice and the illuminating object source is shifted from position S_1 to S_2 between the exposures. Owing to the shift of the object beam, the phases of the scattered waves are different.

If both exposures are made with identical conditions of illumination and have the same duration, then the equation describing a family of hyperboloids of revolution whose axes pass through the two source points is given by (see Chapter 2 Section 2.4.3)

$$\frac{2\pi}{\lambda}(r_{o1} - r_{o2}) = 2n\pi \tag{5.32}$$

or

$$r_{o1} - r_{o2} = n\lambda \tag{5.33}$$

where n is an integer.

A cross-sectional view of these hyperboloids is shown in Figure 5.12. A family of hyperboloids also corresponds to the dark fringes, namely,

$$r_{o1} - r_{o2} = \frac{(2n+1)\lambda}{2} \tag{5.34}$$

The two exposures represent two coherent simultaneous or sequential points of illumination that produce a fringe system (Young's fringes) in space in the form of a set of rotational hyperboloids. Their common foci are the two points of illumination separated by the distance ΔS. The distance d between two adjacent hyperboloid shells is [159]

$$d = \frac{\lambda}{2\sin\frac{\theta}{2}} \tag{5.35}$$

where λ is the wavelength of light and θ is the angle between the two points of illumination as seen from the object, see Fig. 5.11.

The accuracy of the object contour is mainly limited by the errors in the interference phase measurement [141]. The linear shift ΔS determines the fringe

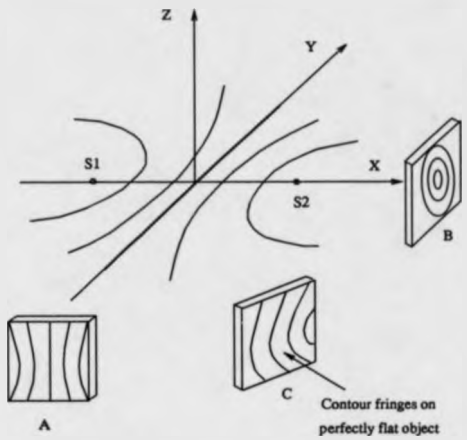


Figure 5.12: Schematic diagram of loci interference for the double-source hologram. A is the position of the object used in experimentation, B is collinear with the displacement, C is the intermediate case

density on the object, as shown in Fig. 5.13 and Fig. 5.14. The maximum fringe density will be determined by the detector resolution.

5.5.2 Experimental Set-Up

Figure 5.11 shows the experimental set-up for fibre optics double-exposure holography. The object beam was moved between the exposures by a micro-translator. The light from 25mW He-Ne laser ($\lambda = 0.6328\mu$m) was launched into the expander and divided into two by a (60/40) beam splitter (BS). Two single-mode optical fibres about 2m long were used, one arranged to illuminate an object and the other arranged to illuminate the hologram plate (H). Losses through the single-mode fibres are around 50-60%, most of which occurs at the launch. Because the fibre core diameter is very small (5μm), the irradiance distribution of the laser light from the fibre ends can be considered homogeneous, as for light emerging from a spatial filter. Three-axis directional micropositioners were used to couple light into fibres. Good coupling efficiency requires precise positioning

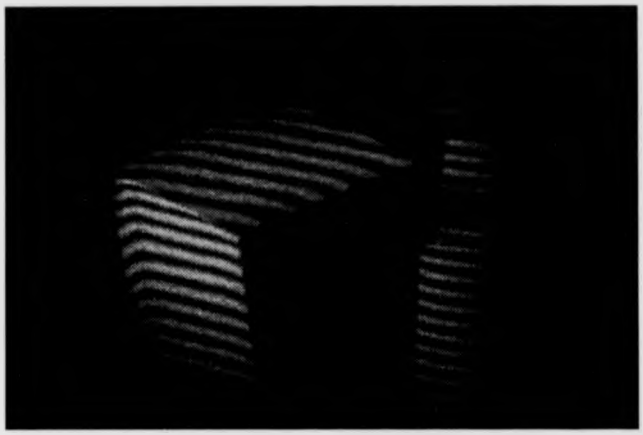


Figure 5.13: Holographic contouring on a cube, a ball and a cylinder (image size 512 × 512).

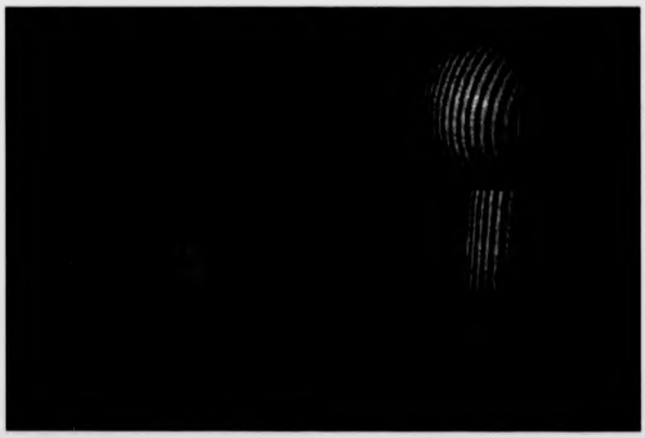


Figure 5.14: Holographic contouring on the same objects as in Figure 5.13, but with the translation of the object beam horizontally (image size 512 × 512).

of fibre to centre the core in the focused laser beam.

Two exposures were used, one for each source position. An exposure was taken with the illuminating object beam at position S1 in Fig. 5.11. Then the object beam was translated sideways to position S2 to form the contouring fringes, after which a second exposure was made on the same plate. The reconstructed holographic image reveals the object intersected by surfaces of a set of rotational symmetric hyperboloids.

5.5.3 Practical Problems

All experiments were performed in a temperature controlled basement laboratory. This room has been adapted for use as a dark room and also contains an optical bench. This bench consists of rectangular cast iron block of dimensions 4.3m × 1.7m × 0.185m and weighs approximately 6 metric tones. This is, in turn, mounted on pneumatic, vibration isolation supports. In addition all measurements were performed after working hours to reduce influences from traffic, machinery and other human activity. However, other perturbations would be extant during experimentation due to physical adjustments and thermal radiation from operators. To reduce these effects a period of 3 minutes was left to elapse before each exposure of the photographic plate.

Translation of the optical fibres has been achieved using a standard slideway with positional feedback via an LVDT. The gauging had a resolution of 10nm and appeared to be stable to better than 1μm throughout a single experiment. However, the large measurement loop precludes confidence on accuracy to better than 3%.

Additional errors may result from thermal fluctuations of fibre optics resulting in variation of refractive index, inaccuracies in timing of the two exposures due to manual control and also non-optimal developing procedures. All of these will result in less contrast of the developed hologram. A technique for the compensation of the thermally induced fluctuations of fibre optics has been reported by Jones

et al [31]. In the normal laboratory environment thermally induced phase fluctuations with instantaneous velocities of up to 5 rad/s, and total phase excursions of up to 20 rad in one minute were observed [31]. Consequently, although satisfactory holograms were obtained with short exposures longer exposures could not be used. Such effects may be readily compensated for using the phase modulators described by Jones *et al* [31], Jackson *et al* [121, 122].

Finally, a CCD camera was used to capture the virtual image. The video signal from the CCD camera is received by a digital frame-store card and can be displayed on a monitor. During reconstruction, speckle noise from the laser contributes to the background of the image.

Figure 5.15 shows the system for reconstructing holograms. A reconstructed image is picked up by a CCD camera and is stored in digital frame memory in the form of 512 × 512 × 8 bit data. The image received by CCD camera can be displayed on a monitor. The Olivetti M28 microcomputer is used to make digital file on a disk which is then transformed through a communication line to a Sun Workstation and processed. The final results are printed out by a laser printer.

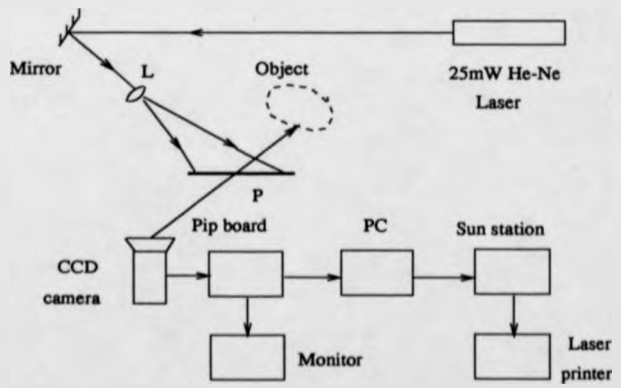


Figure 5.15: Schematic diagram of hologram reconstruction.

5.5.4 Results and Evaluation

With this set-up, several holograms were made on different objects. These objects were a ball on a top of the cylinder, cube, telephone, cup, aluminium can, model footballer and balls. In order to get a much brighter hologram, an optimum ratio of intensities between reference and object beam of 3:1 is required, and an angle of $30^0 - 60^0$ between the two beams is essential to ensure high contrast and resolution.

In this contouring technique, different types of translation of the fibre optics beam give rise to different fringes. For example, translations tend to give fringes which are located at infinity. And the fringe pattern also depends on the direction of the translation.

In Figure 5.16 a model footballer hologram can be seen with curvature fringes on the football surface. Figure 5.17 shows the same model footballer hologram but with an object beam displacement of 300μm. In this case, the fringe spacing is nearly twice smaller than that of previous one. Figure 5.18 shows the hologram contours on a light bulb. Figure 5.19 shows the holographic contouring fringes on a cube. In all of these holograms, sharp fringes can be seen on the object surface. The fringe quality is much better than that of ESPI images.

The objects shown in the Fig. 5.13 have three different shapes and the contour fringes can be seen on the surface of the objects. The hologram was made by vertically translating the object beam between two exposures. Figure 5.14 is a hologram of the same objects as in the Fig. 5.13. The hologram was made by translating the object beam 200μm horizontally. The fringe pattern in Fig. 5.14 is different from that in the Fig. 5.13. Figure 5.20 is a hologram of the same objects used in the Fig. 5.14, but with a different amount (150μm) of translation of the object beam between two exposures, and also the direction of translation is opposite to the previous one. In both cases, the linear fringes can be seen on the surface of the cube and cylinder. The curvature of the fringes on the ball corresponding to the height of the surface.

Figure 5.16: Holographic contouring fringes on a model footballer (image size 512 × 512).

Figure 5.17: Holographic contouring fringes on a model footballer (image size 512 × 512).

Figure 5.18: Holographic contouring fringes on a light bulb (image size 512 × 512).

Figure 5.19: Holographic contouring fringes on a cube (image size 512 × 512).

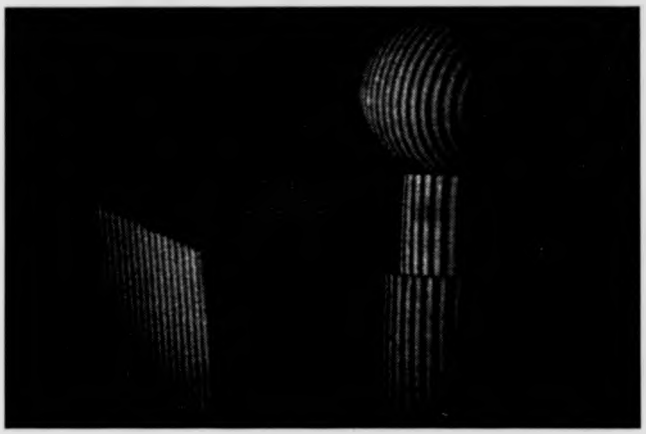


Figure 5.20: Holographic contouring on the same objects as in Figure 5.14, but with opposite translation direction of the object beam horizontally (image size 512 × 512).

Figure 5.21: Holographic contouring fringes on an aluminium can (image size 512 × 512).

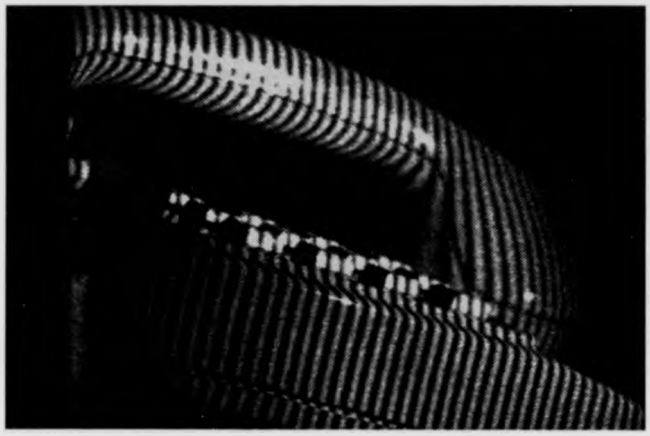


Figure 5.22: Holographic contouring on a telephone (image size 512 × 512).

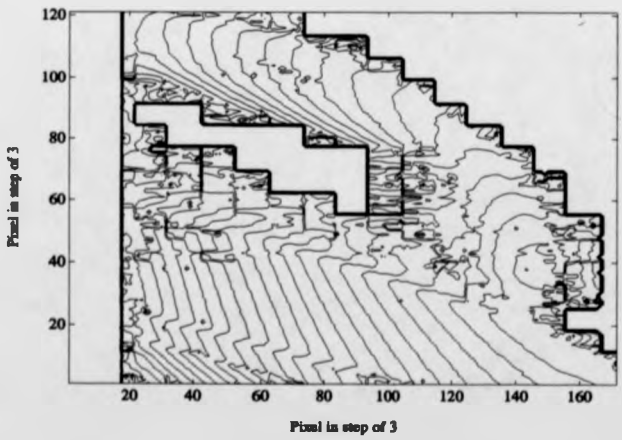


Figure 5.23: Contouring map of a telephone.

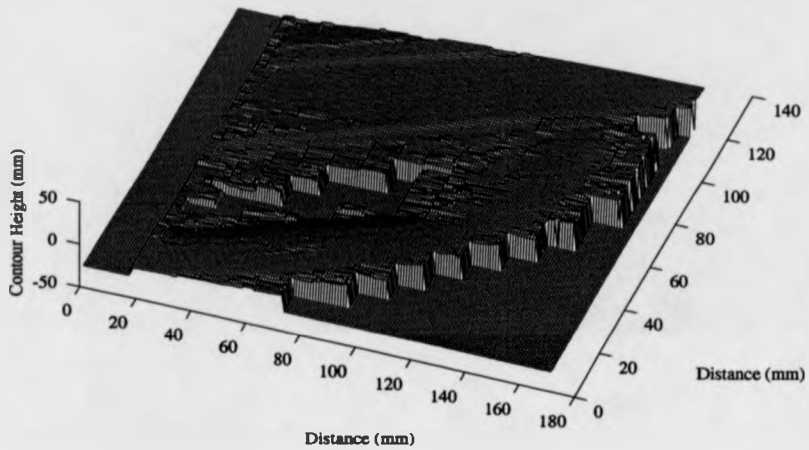


Figure 5.24: 3-D perspective plot of a telephone.

Figure 5.25: Holographic contouring fringes on a tray and a cup (image size 512 × 512).

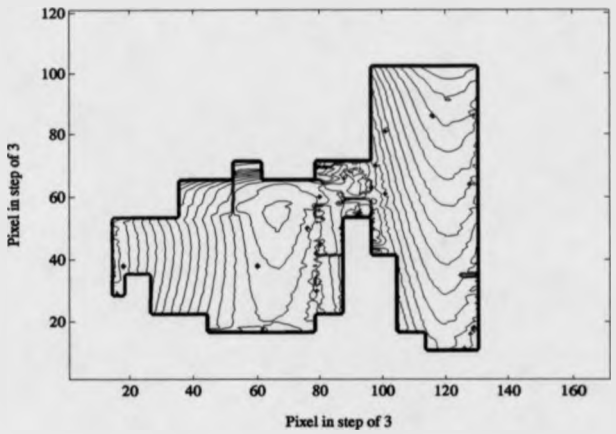


Figure 5.26: Contouring map of a tray and a cup.

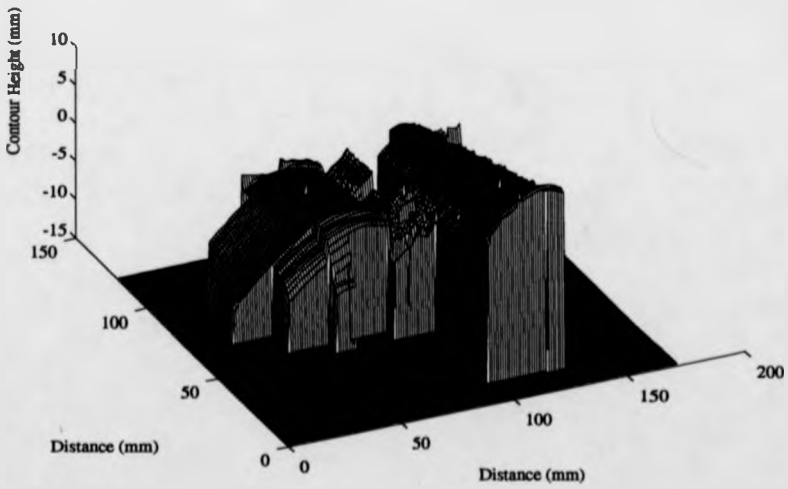


Figure 5.27: 3-D perspective plot of a tray and a cup.

Figure 5.28: Holographic contouring fringes on the balls (image size 512 × 512).

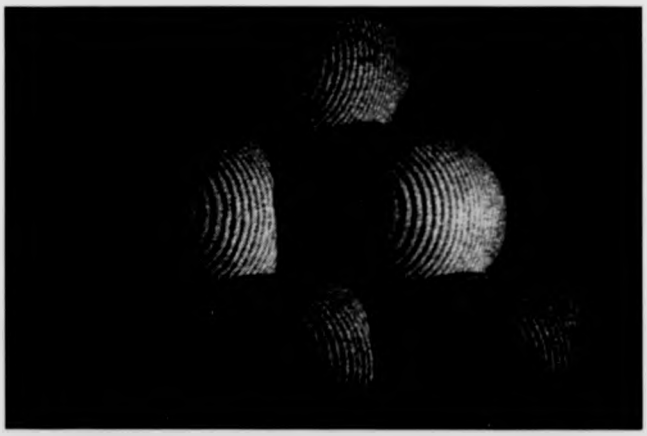


Figure 5.29: Holographic contouring fringes on the same balls as in Figure 5.28, but with different viewing direction (image size 512 × 512).

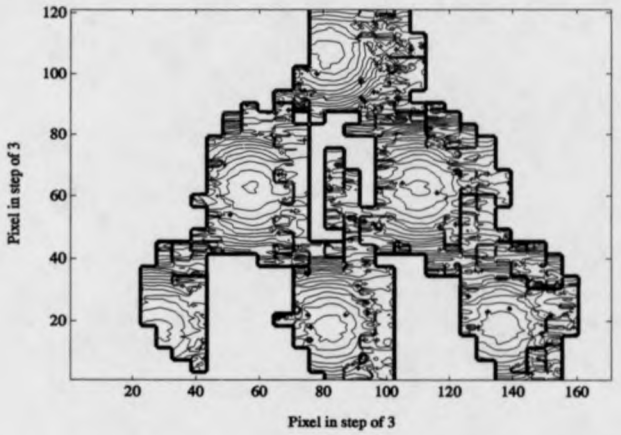


Figure 5.30: Contouring map of balls in Figure 5.28.

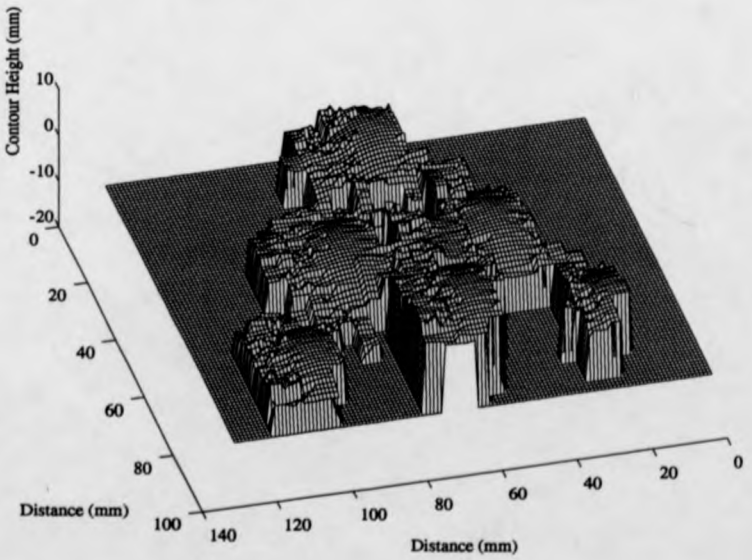


Figure 5.31: 3-D perspective plot of balls.

In Figure 5.21 we see the aluminium can hologram with 300μm linear movement of the object beam between two exposures. Figure 5.22 is a hologram of the telephone as produced using an object beam displacement of 300μm. The contour map of the telephone produced by the Matlab package is shown in Fig. 5.23. A 3-D perspective plot of the telephone is shown in the Fig. 5.24. This was generated by using Fourier transform of the holographic image and phase-unwrapping techniques. In Fig. 5.25 a tray and a cup hologram can be seen with curvature fringes on the objects' surface. A contour map and a mesh plot of the objects are shown in Fig. 5.26 and Fig. 5.27 respectively.

Finally, the contouring method has been applied to balls that are glued to the object. Figure 5.28 shows the holographic fringe pattern on the balls. The different contour fringes of balls is shown in Fig.5.29. This hologram was recorded after the balls were rotated. By studying the fringe patterns on the balls, it is possible to determine that a complicated fringe pattern can be caused not only by object deformation but also by the contouring of a non-deformed translated or rotated object having a complicated shape. A contour map and 3-D perspective plot of balls are shown in Fig.5.30 and Fig.5.31 respectively.

The holograms demonstrated above show that contouring fringes can be made by a well planed translation of the object beam between two exposures. The use of fibre optics greatly simplifies the holographic contouring system. The contouring fringes may be conveniently formed without the use of excessive auxiliary components or sophisticated experimental devices. The contouring fringe spacing can be varied during the recording by precise controlling the translation of the fibre optic beams. Holographic contouring using fibre optics can be competitive with other standard optical methods used in surface contouring. Volume and depth of a surface can be automatically obtained by using the Fourier transform technique. Also, the advantage of the method used here is simplicity in experimental realisation and the possibility of working in real-time (without the hologram).

The disadvantage of using this method for contouring purposes is easily seen from Figs. 5.21, 5.25, 5.28, 5.29. It is a fact that if the object has convolutions,

there are shadowed areas where no light strikes the surface and hence no reconstruction occurs.

5.6 Conclusion

A method for obtaining three-dimensional surface contouring has been described. The fibre optics holographic system makes it easier to achieve object beam linear movement of angular displacement. This method may be applied to other types of fringe measurement technique. The flexibility of using fibre optics for holographic interferometry introduces the possibility of easily varying the direction of object illumination. This will enable more accurate deformation displacement measurements. So, combined with a pulsed laser, vibration and surface deformation analysis will be much easier than using conventional holographic methods. For the future, one can see the possibility of holographic equipment in which a pulsed laser, fibre coupler, and CCD camera are integrated in one unit, to make real-time and remote holographic measurement.

Chapter 6

Theoretical Analysis of the Carrier Fringe and FFT Techniques

6.1 Introduction

This chapter will present a theoretical analysis of the carrier fringe technique. This technique for interference fringe pattern analysis is well known and has been widely examined during the last few years in several papers [40, 41, 42, 43, 44]. The technique is based on generating a high density linear phase carrier and is therefore very suitable for automated complex fringe processing. Introduction of the carrier fringes results in the formation of a fringe pattern that has monotonically increasing fringe orders. Using this technique only one interferogram is needed to obtain the object phase distribution of interest, and there is no need for the special phase-shifting actuators used commonly in phase-shifting-interferograms.

The carrier fringe technique has been developed to solve problems of phase measurement in many fringe field. Reynolds *et al* [42] reported holographic fringe linearisation interferometry for defect detection. A linear fringe pattern is gener-

ated by using the object beam between two exposures. An improved fringe carrier technique for unambiguous determination of holographically recorded displacements was proposed by Plotkowski *et al* [41]. Mattys *et al* [43] used carrier fringe method for automated analysis of holo-interferograms for the determination of surface displacement. Also, Long *et al* [45] presented a cylindrical fringe carrier technique in holographic interferometry. Quan, Judge and Bryanston-Cross [39] reported the holographic measurement of deformation using the carrier fringe and FFT techniques. The novel feature in this work is the use of fibre optics generating carrier fringes and fringe analysis by FFT.

Han *et al* [44], Takezaki *et al* [160] applied the carrier fringe technique to moire interferometry for strain and stress measurements. They illustrated how moire carrier fringes were used to obtain the desired data in complex specimens. Measurement of in-plane displacements using the carrier fringe technique will be presented in Chapter 8. This new technique of strain and stress measurement has two advantages, firstly, the stress fringe patterns can be modulated by linear carriers and secondly, the results can be automatically processed by the FFT fringe analysis method.

In interferometry, a spatial carrier fringe technique has been applied to precision surface metrology by Tischer [161] and Frankowski *et al* [162].

6.2 Objective of the Holographic Carrier Fringe Technique

The deformation fringe pattern obtained from double-exposure holographic interferometry can be complex. This complexity can be reduced to yield a simple fringe pattern using carrier fringe linearisation. This is the first objective of holographic carrier fringe technique. The second objective is to yield a single interferogram for analysis, which encodes direction as well as displacement. In this carrier fringe technique, the fringes are introduced in the formation step

of double-exposure holographic interferometry by shifting the fibre optic beam, illuminating the object, between the two exposures. This principle of shifting the object wavefront between exposures to create linear cosine fringes has been illustrated in the literature [42]. The fringes have a period ρ in object space, which is given by [5]

$$\rho = \frac{\lambda}{2 \sin \frac{\gamma}{2}} \tag{6.1}$$

where λ is the wavelength of the laser light and γ is the amount of angular shift between the object beams.

Figure 6.1 shows a double-exposure hologram of an aluminium can stressed with an elastic band obtained from a fibre optic holographic system. The can is stressed during one exposure but unstressed during the other. Between two exposures, the rubber band was delicately cut off, thus removing the cause of movement in the can's surface. The interference fringes on the can surface reveal the flexing caused by the rubber band between the first and second exposures. The area near the rubber band flexed the most, the bottom of the can hardly at all.

Figure 6.2 shows a double-exposure hologram of an aluminium can with added linear carrier fringes. The linear carrier fringes is produced by shifting the fibre optic beam illuminating the objects between exposures. In Fig. 6.2, a complicated fringe pattern of irregularly curved stripes and rings is replaced by simplified linear fringes. Because of the curvature of the object, the fringes are not quite linear, but rather quadratic. Figures 6.1, and 6.2 demonstrate the idea of simplification of interference fringes. This forms the basis of carrier fringe concepts. With the aid of automatic readout by the FFT technique, the carrier fringe technique is of great importance in the field of holographic interferometry.

Figure 6.1: Double-exposure hologram of a stressed aluminium can (image size 512 × 512).

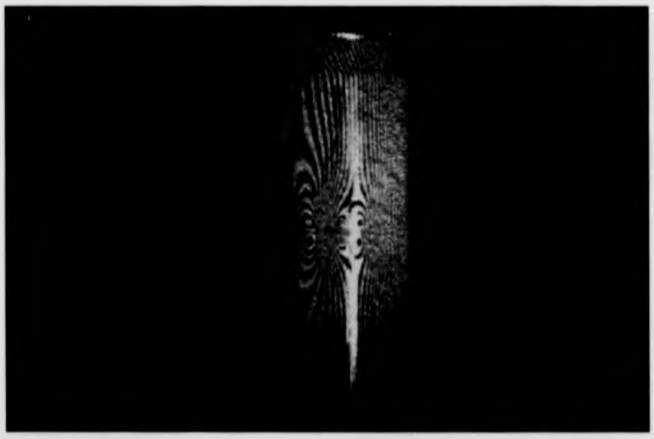


Figure 6.2: Double-exposure hologram of a deformed aluminium can with carrier fringes (image size 512 × 512).

6.3 Methods of Generating Carrier Fringes

The carrier fringe pattern is an important ingredient in holographic, speckle, and moire interferometry. In the case of a fringe field, a very large number of carrier fringes is sometimes introduced in order to dominate the fringe field. In such cases, the load induced fringes are modulated by a high frequency pattern of uniformly spaced carrier fringes. This is analogous to carrier frequencies in communications technology. There are several ways of generating carrier fringes.

(1) By applying a very small tilt to the wavefront illuminating the object field between the exposures. This results in a background pattern of fringes which is nearly equidistant straight lines [163]. The optical system for generating such linear fringe patterns is shown in Fig. 6.3.

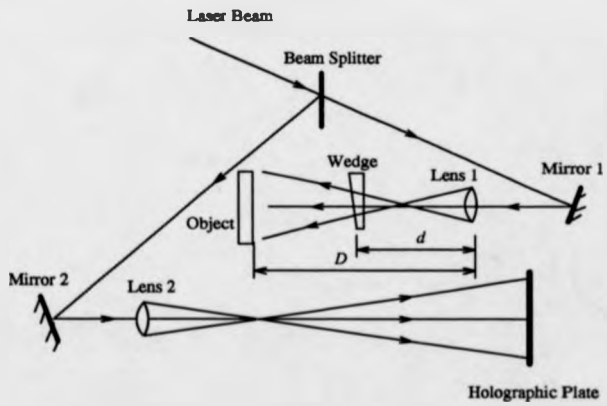


Figure 6.3: Schematic diagram showing generation of carrier fringes in double-exposure hologram (Hariharan *et al* 1973).

In this system, between two exposures, a precisely controlled tilt in the object wavefront is obtained by rotating a thin mounted glass wedge. In the absence of any changes in the object between the two exposures, this would give rise to straight, vertical fringes in the reconstructed image with spacing $\lambda/2(n-1)(d/D)\Delta\phi \sin\theta$, where d and D are indicated in Fig. 6.3, $\Delta\phi$ is the wedge angle,

θ is the wedge rotation angle, n is the index of the glass wedge.

(2) By rotating a mirror so that the point source appears to have moved [41]. The method of generating high density fringes is shown in Fig. 6.4. This is the typical double-exposure hologram system. However, mirror 2 is mounted in a rotation stage. Between two exposures, mirror 2 is rotated slightly. This translation produces linear parallel fringes added to the phase changes that are to be measured. In order to determine the fringe orders without ambiguity, the carrier fringes should be of high density to dominate greatly the fringe pattern. Given the knowledge of the direction of rotation of the mirror, the sign of the fringes can be determined.

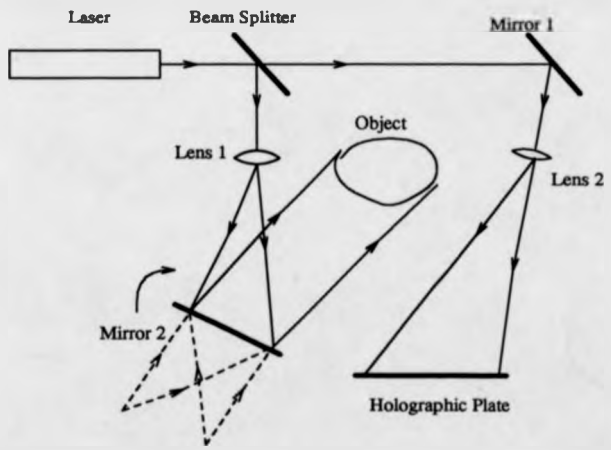


Figure 6.4: Schematic diagram of generating carrier fringes (Plotkowski *et al* 1985).

(3) By tilting the object between two exposures [43]. In this method, the object is tilted about a horizontal axis through the center of the object surface, with the bottom of the object being moved away or toward the observer between the exposures. Since the direction of the tilt is known, the sign of the fringe orders can be assigned without ambiguity. The zero fringe order is located in the

centre of the object.

(4) By translating the cylindrical lens in double-exposure holography [45]. In this cylindrical carrier fringe technique, a linear carrier fringe of unequal spacing is created by using a cylindrical wave as shown in Fig. 6.5. In this system, a cylindrical lens is used to illuminate the object. Between two exposures, the cylindrical lens is moved along the optical axis a small distance **d**. The advantages of the technique are: 1) the object beam need not be very precisely adjusted to provide the required fringe spacing. 2) the spacing of the modulated carrier fringe can be controlled so as to be fairly constant, and 3) it is to count fringe orders and measure the fringe positions along a line perpendicular to the carrier fringes.

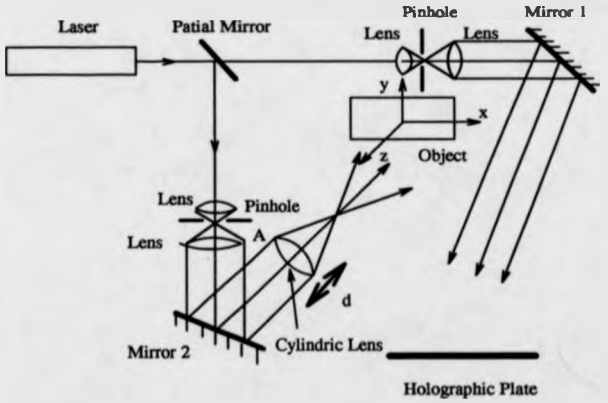


Figure 6.5: Schematic diagram for generating carrier fringes by a cylindrical wave (Long *et al* 1988).

(5) The linear carrier fringes are generated by translating the illuminating lens (in shearography strains measurement) [160]

As shown in Fig. 6.6, a point light source is located at the focal length of the lens illuminating the object. This is the same as a spherical wave diverging from virtual point S. After translating the lens, the virtual point S is also moved to S_d, therefore, a set of linear fringes are generated by translating lens and the carrier

fringes are orthogonal to the shearing direction.

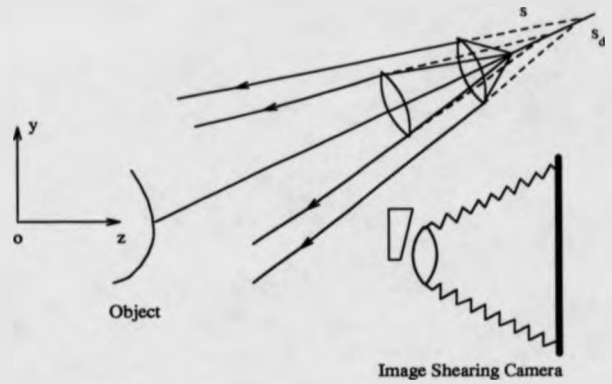


Figure 6.6: Schematic diagram showing generation of carrier fringes in shearography strain analysis (Takezaki *et al* 1986).

(6) The tilted plate method for introducing carrier fringes in moire interferometry [44]. The idea of generating carrier fringes in this method is simple as shown in Fig. 6.7. Carrier fringes are introduced by inserting a glass plate of uniform thickness in the diverging beam between the light source and the collimator. The carrier frequency can be controlled by tilting the plate. Carrier fringes are very useful in the case of moire interferometry, for example, in distinguishing the sign of the displacements by introducing a carrier pattern of known sign and to determine gradients when they are not adequately represented in the load induced fringe pattern.

(7) Carrier fringes are produced by translating the fibre optic beam illuminating the object, as shown in Fig. 6.8 [39]. The use of fibre optics offers advantages in holographic interferometry. If fibre optics are used the holographic system is greatly simplified and very flexible. The spacing of the carrier fringe can be adjusted precisely by controlling the amount of translating fibre optic beam. The carrier fringe is very sensitive to the direction of deformation.

Figure 6.9 shows a double-exposure hologram of a centrally loaded silicon

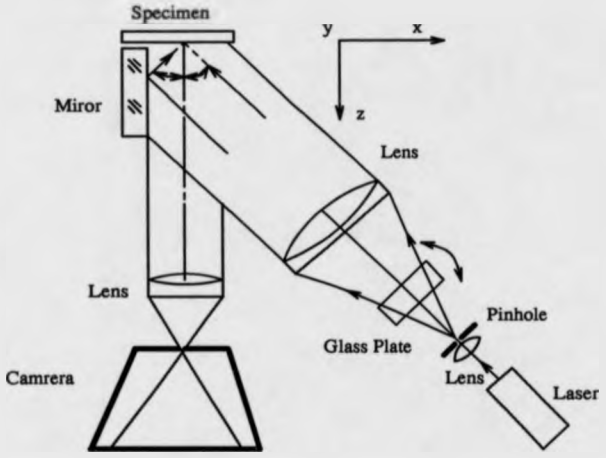


Figure 6.7: Schematic diagram showing generation of carrier fringes in moire interferometry (Han *et al* 1989).

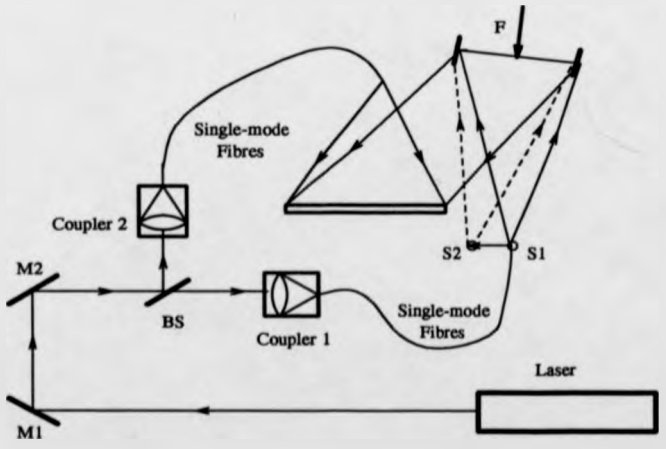


Figure 6.8: Schematic diagram of generating carrier fringes by translating fibre optic beams (Quan *et al* 1991).

wafer obtained in a fibre optic holographic system. Figure 6.10 shows the effect of adding the linear fringe pattern to the deformed silicon wafer. The first exposure was recorded with a central loading. Before the second exposure, the fibre optic illuminating the object was translated 350 μm along the line perpendicular to the illuminating direction. The second exposure was recorded with a deformed silicon wafer and a large number of carrier fringes. It can be seen that the complicated fringe pattern in Fig. 6.9 has been simplified to the linear fringe pattern of Fig. 6.10.

Since the direction of the translating fibre optics beam is known, counting and measuring the fringe order and sign is obtained easily. This is illustrated in the Figs. 6.10, 6.11, 6.12 and 6.13. Figure 6.11 shows the same double-exposure hologram of a centrally loaded silicon wafer with the carrier fringes in the opposite direction to Fig. 6.10. After processing the holographic images shown in Fig. 6.10 and Fig. 6.11, 3-D deformation perspective plots can be generated. Figure 6.12 and Figure 6.13 show 3-D perspective plots of the deformed silicon wafer corresponding to Figs. 6.10 and 6.11 respectively. Note that the deformation in Fig. 6.12 and Fig. 6.13 are in opposite directions. In other words, if the direction of the translating fibre optic beam is the same in the case of Fig. 6.10 and Fig. 6.11, Fig. 6.10 indicates that the direction of deformation is towards the observer, while Fig. 6.11 indicates the direction of deformation is away from the observer.

6.4 Theoretical Analysis of Carrier Fringes

The coordinate system for hologram recording and reconstruction is shown in Fig. 6.14. The distance from the object plane to the hologram plane is z_1, the distance from the hologram plane to the reconstruction plane is z_2, the recording wavelength is λ_1, the reconstruction wavelength is λ_2. Object plane is denoted by (x_1, y_1), the hologram plane by (x_2, y_2), the reconstruction plane by (x_3, y_3).

The geometry and notation for calculation of the side-band Fresnel hologram

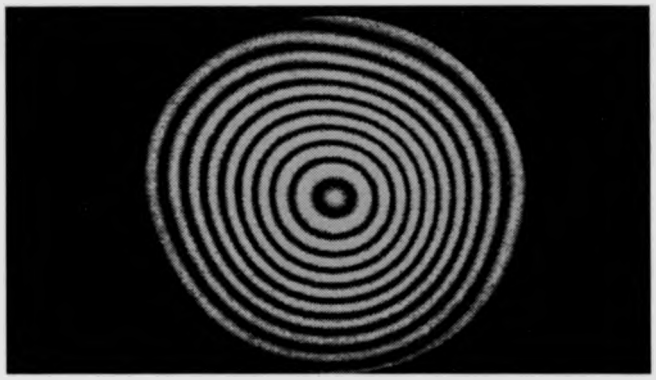


Figure 6.9: Double-exposure hologram of a centrally loaded silicon wafer (image size 512 × 512).

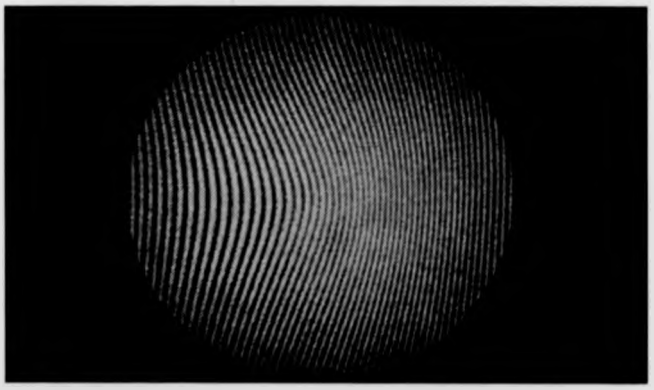


Figure 6.10: Double-exposure hologram of a centrally loaded silicon wafer with carrier fringes (image size 512 × 512).

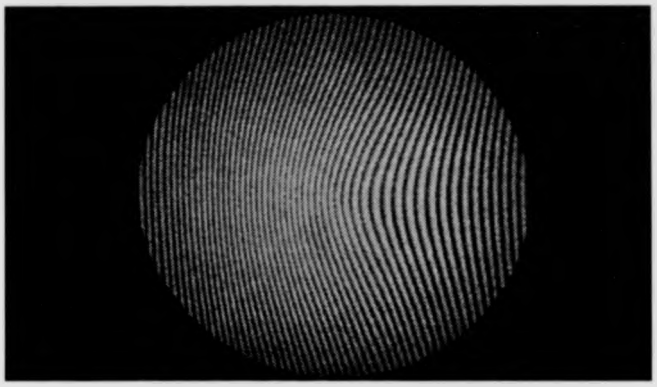


Figure 6.11: Double-exposure hologram of a centrally loaded silicon wafer with opposite direction of carrier fringes (image size 512 × 512).

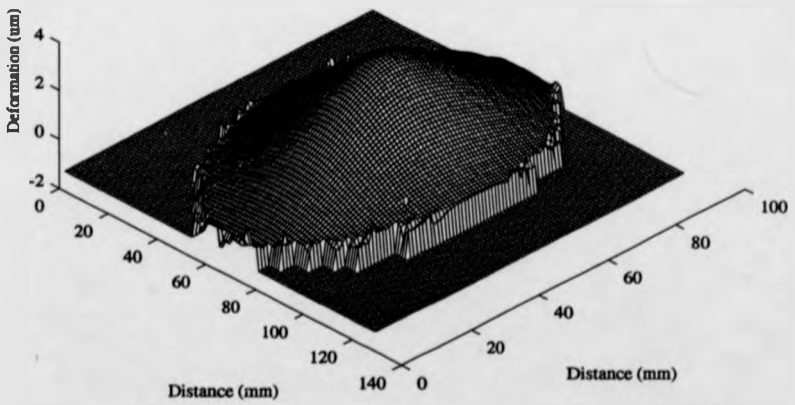


Figure 6.12: 3-D perspective plot of deformed silicon wafer.

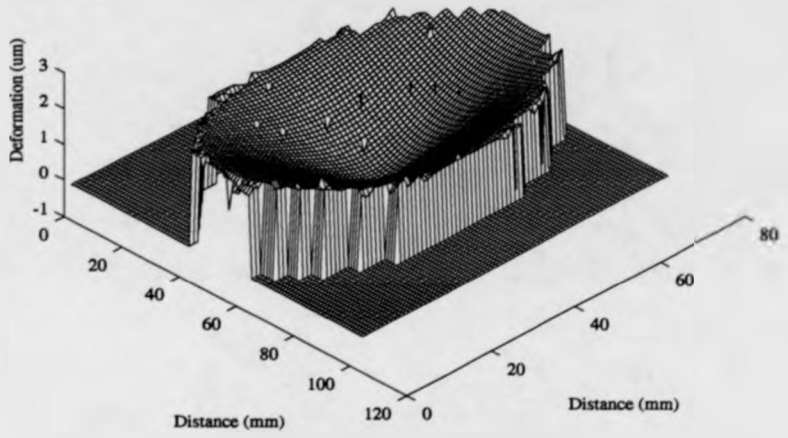


Figure 6.13: 3-D perspective plot of deformed silicon wafer with opposite carrier direction.

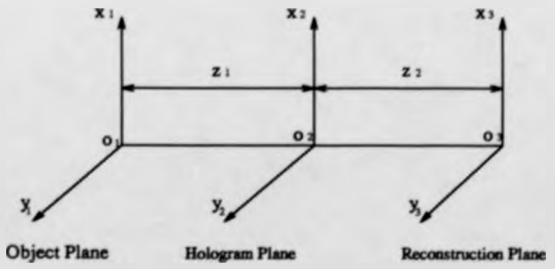


Figure 6.14: Coordinates and notation for hologram process.

is illustrated in Fig. 6.15. The light is reflected from the object and made to interfere with the reference beam. In the case of generating carrier fringes, the object beam is shifted by an angle of θ. For an opaque object, the field $\Psi(\vec{x}_2)$ in the hologram plane can be expressed in the form [57]

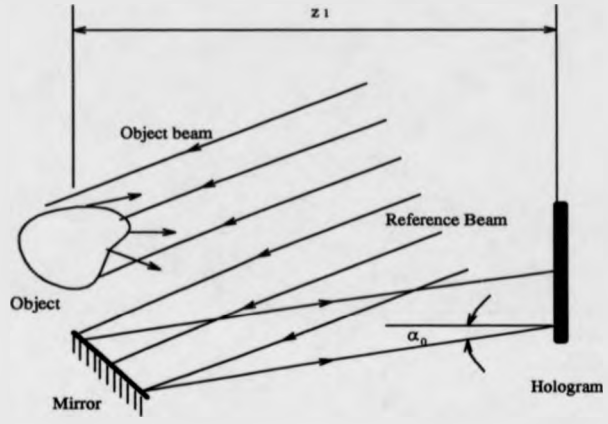


Figure 6.15: Side band Fresnel hologram recording with tilted object beam (Develis *et al* 1971).

$$\Psi(\vec{x}_2) = \int\int CK[1 - D(\vec{x}_1)]exp[ik_1 r(\vec{x}_1, \vec{x}_2)]d\vec{x}_1 \tag{6.2}$$

where

$$C = \frac{(1 - ik_1 r)\cos\beta}{2\pi r^2} \doteq \frac{-ik_1 \cos\beta}{2\pi r} \tag{6.3}$$

where K is the amplitude of the incidence field, $D(\vec{x}_1)$ is the object transmission. β is the angle between the propagation vector k_1 and the optical axis. The term $r(\vec{x}_1, \vec{x}_2)$ is the distance from a typical point $\vec{x}_1$ in the object plane to a point $\vec{x}_2$ in the hologram plane, and is given by

$$r = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + z_1^2} \tag{6.4}$$

For simplicity, $\cos\beta \doteq \frac{z_1}{r} \doteq 1$, therefore, r may be expressed by a series expansion,

$$r \doteq z_1 + \frac{x_1^2+y_1^2}{2z_1} + \frac{x_2^2+y_2^2}{2z_1} - \frac{x_1x_2}{z_1} - \frac{y_1y_2}{z_1} - \frac{x_1^4-x_2^4}{8z_1^3} - \frac{y_1^4-y_2^4}{8z_1^3} + ... \quad (6.5)$$

Omitting the terms of order $\frac{1}{z_1^3}$ and higher, Eq. 6.2 becomes

$$\Psi(\vec{x}_2) = CKe^{ik_1z_1}\int\int exp\{ik_1[\frac{(x_1-x_2)^2+(y_1-y_2)^2}{2z_1}]\}dx_1dy_1 - CKe^{ik_1z_1}$$
$$e^{\frac{ik_1(x_2^2+y_2^2)}{2z_1}}\int\int D(\vec{x}_1)exp[ik_1(\frac{x_1^2+y_1^2}{2z_1} - \frac{x_1x_2+y_1y_2}{z_1})]dx_1dy_1 \quad (6.6)$$

Suppose the reference field is a plane wave during the recording, the diffracted object field $R(\vec{x}_2)$ is combined with this plane wave to form a hologram, the intensity distribution is

$$I(\vec{x}_2) = |\bar{K}exp(i\vec{k}_1\vec{r}') + R(\vec{x}_2)|^2 \quad (6.7)$$

where

$$\vec{k}_1\vec{r}' = |k_1|(x_2\sin\alpha_0 + y_2\cos\alpha_0\cos\Phi + z_1\cos\alpha_0\sin\Phi)$$
$$\vec{r}' = \vec{i}x_2 + \vec{j}y_2 + \vec{k}z_1$$
$$\vec{k}_1 = \vec{i}k_x + \vec{j}k_y + \vec{k}k_z$$

In the equations above, $k_1 = |k_1| = \frac{2\pi}{\lambda_1}$, α_0 is the angle between the reference beam and normal of holographic plate, Φ is the angle in the plane measured from y-axis. In order to simplify the analysis, the one dimensional case is considered, and scalar form is used. In Eq. 6.6, only the second term is related to the diffracted object wave, therefore, the diffracted object wave is simply given by

the second term in Eq. 6.6, which is [57]

$$R(x_2) = -CKe^{ik_1z_1}e^{\frac{ik_1x_2^2}{2z_1}}\int D(x_1)exp[ik_1(\frac{x_1^2}{2z_1} - \frac{x_1x_2}{z_1})]exp(ik_1x_1\sin\theta)dx_1 \quad (6.8)$$

where the term $exp(ik_1x_1\sin\theta)$ denotes the effect of adding the linear fringe pattern by shifting the object beam an angle θ. Substituting Eq. 6.8 into Eq. 6.7 gives the intensity distribution

$$\begin{aligned} I(x_2) = {} & |\tilde{K}|^2 + |C|^2|K|^2|f(x_2)|^2 - \tilde{K}^*K\ Cexp(ik_1z_1) \\ & exp(\frac{ik_1x_2^2}{2z_1})f(x_2)exp(-ik_1x_2\sin\alpha_0) \\ & -\tilde{K}K^*C^*exp(-ik_1z_1)exp(-\frac{ik_1x_2^2}{2z_1})f^*(x_2)exp(ik_1x_2\sin\alpha_0) \end{aligned} \quad (6.9)$$

where

$$f(x_2) = \int D(x_1)exp[ik_1(\frac{x_1^2}{2z_1} - \frac{x_1x_2}{z_1})]exp(ik_1x_1\sin\theta)dx_1 \quad (6.10)$$

Upon reconstruction, the hologram is illuminated with a plane reference wave, the amplitude transmittance is given by

$$\begin{aligned} T(x_2) = {} & |\tilde{K}|^2 + \frac{\gamma}{2}|C|^2|K|^2|f(x_2)|^2 - \frac{\gamma}{2}\tilde{K}^*K\ Cexp(ik_1z_1) \\ & exp(\frac{ik_1x_2^2}{2z_1})f(x_2)exp(-ik_1x_2\sin\alpha_0) \\ & -\frac{\gamma}{2}\tilde{K}K^*C^*exp(-ik_1z_1)exp(-\frac{ik_1x_2^2}{2z_1})f^*(x_2)exp(ik_1x_2\sin\alpha_0) \end{aligned} \quad (6.11)$$

where γ is the slope of the H-D curve of the photographic process. In Eq. 6.11, when $\tilde{K} \gg K$, the second term may be omitted. The first term is a diffraction pattern characteristic of the hologram size centred on the optical axis in the image plane, the third and fourth terms represent virtual and real image respectively.

These two terms have different focusing conditions. The focusing condition for focused real image is $k_1/z_1 = k_2/z_2$. In convenience, the real image is considered (a similar analysis holds for the focused virtual image). If the real image is reconstructed with a beam of wavelength λ_2, the amplitude transmittance in the reconstructed image is given by

$$\Psi_{rel}(x_3) = A_1 \int exp(\frac{ik_1x_2^2}{2z_1}) f^*(x_2) exp(ik_1x_2 \sin\alpha_0) \\ exp[ik_2\vec{r}(x_2,x_3)]dx_2 \tag{6.12}$$

where

$$|\vec{r}(x_2,x_3)| = \sqrt{(x_2-x_3)^2 + (y_2-y_3)^2 + z_2^2} \tag{6.13}$$

$$A_1 = -\frac{\gamma}{2}\tilde{K}K^*C^* exp(-ik_1z_1) \tag{6.14}$$

$$f^*(x_2) = \int D^*(x_1) exp[-ik_1(\frac{x_1^2}{2z_1} - \frac{x_1x_2}{z_1})] exp(-ik_1x_1 \sin\theta)dx_1 \tag{6.15}$$

Consistent with paraxial approximation, $|r(x_2,x_3)|$ may be expressed as

$$r \doteq z_2 + \frac{x_2^2+y_2^2}{2z_2} + \frac{x_3^2+y_3^2}{2z_2} - \frac{x_2x_3}{z_2} - \frac{y_2y_3}{z_2} - \frac{x_2^4-x_3^4}{8z_2^3} - \frac{y_2^4-y_3^4}{8z_2^3} + ... \tag{6.16}$$

Substituting Eqs. 6.13, 6.14, 6.15, and 6.16 into Eq. 6.12, the reconstructed amplitude transmittance becomes

$$\Psi_{rel}(x_3) = A_2 \int [\int D^*(x_1) exp(-ik_1\frac{x_1^2}{2z_1}) exp(\frac{ik_1x_1x_2}{z_1}) \\ exp(-ik_1x_1 \sin\theta)dx_1] exp[\frac{ix_2^2}{2}(\frac{k_2}{z_2} - \frac{k_1}{z_1})] \\ exp(ik_1x_2 \sin\alpha_0) exp(\frac{-ik_2x_2x_3}{z_2})dx_2 \tag{6.17}$$

where

$$A_2 = A_1 exp(ik_2 z_2) exp(\frac{ik_2 x_3^2}{2z_2}) \tag{6.18}$$

Using the focusing condition $k_1/z_1 = k_2/z_2$ to remove the quadratic phase factor in Eq. 6.17, Eq. 6.17 becomes

$$\Psi_{rel}(x_3) = A_2 \int \{ \int F(x_1) exp[i2\pi(\frac{x_2}{z_1\lambda_1} - \frac{\sin\theta}{\lambda_1})x_1] dx_1 \}$$
$$exp[ix_2(k_1 \sin\alpha_0 - \frac{k_2 x_3}{z_2})] dx_2 \tag{6.19}$$

where

$$F(x_1) = D^*(x_1) exp(-\frac{ik_1 x_1^2}{2z_1}) \tag{6.20}$$

Suppose infinite limits are considered in integrating x_1, the contents in curly bracket is just a Fourier transform of a Fresnel wavefront, therefore,

$$\Psi_{rel}(x_3) = A_2 \int \tilde{F}(\frac{x_2}{\lambda_1 z_1} - \frac{\sin\theta}{\lambda_1}) exp[ix_2(k_1 \sin\alpha_0 - \frac{k_2 x_3}{z_2})] dx_2$$
$$= A_2 \int \tilde{D^*}(\frac{x_2}{\lambda_1 z_1} - \frac{\sin\theta}{\lambda_1}) exp[-\frac{ik_1}{2z_1}(\frac{x_2}{\lambda_1 z_1} - \frac{\sin\theta}{\lambda_1})^2]$$
$$exp[ix_2(k_1 \sin\alpha_0 - \frac{k_2 x_3}{z_2})] dx_2 \tag{6.21}$$

where $\tilde{F}$ denotes the Fourier transform of F, let $q = \frac{x_2}{\lambda_1 z_1} - \frac{\sin\theta}{\lambda_1}$, then $x_2 = \lambda_1 z_1 (q + \frac{\sin\theta}{\lambda_1})$ and $dx_2 = \lambda_1 z_1 dq$. Substituting above formulas into Eq. 6.21, Eq. 6.21 becomes

$$\Psi_{rel}(x_3) = A_3 exp[i2\pi z_1 \sin\theta(\frac{\sin\alpha_0}{\lambda_1} - \frac{x_3}{\lambda_2 z_2})]$$
$$\int G(q) exp[i2\pi(z_1 \sin\alpha_0 - x_3)q] dq \tag{6.22}$$

where

$$A_3 = \lambda_1 z_1 A_2$$
$$G(q) = \bar{D}^*(q) exp[-\frac{ik_1}{2z_1}q^2]$$

The integral of q is also a Fourier transform pattern, hence Eq. 6.22 becomes

$$\begin{aligned}\Psi_{rel}(x_3) &= A_3\ exp[i2\ \pi z_1 \sin\theta(\frac{\sin\alpha_0}{\lambda_1} - \frac{x_3}{\lambda_2 z_2})]\bar{G}(z_1\sin\alpha_0 - x_3) \\ &= A_3\ exp[i2\ \pi z_1 \sin\theta(\frac{\sin\alpha_0}{\lambda_1} - \frac{x_3}{\lambda_2 z_2})]\bar{D}^*(z_1\sin\alpha_0 - x_3) \\ &\quad exp[-\frac{ik_1}{2z_1}(z_1\sin\alpha_0 - x_3)^2] \qquad (6.23)\end{aligned}$$

If $\sin\theta \doteq \theta$, $\sin\alpha_0 \doteq \alpha_0$, the amplitude transmittance is given by

$$\begin{aligned}\Psi_{rel}(x_3) &= A_3\ exp[i2\ \pi z_1\theta(\frac{\alpha_0}{\lambda_1} - \frac{x_3}{\lambda_2 z_2})] \\ &\quad \bar{D}^*(z_1\alpha_0 - x_3) exp[-\frac{ik_1}{2z_1}(z_1\alpha_0 - x_3)^2] \qquad (6.24)\end{aligned}$$

In Eq. 6.24, the term $exp[i2\ \pi z_1\theta(\frac{\alpha_0}{\lambda_1} - \frac{x_3}{\lambda_2 z_2})]$ indicates a set of cosine fringes, which are linear carrier fringes. The phase factor ϕ of carrier fringes is

$$\phi = z_1\theta(\frac{\alpha_0}{\lambda_1} - \frac{x_3}{\lambda_2 z_2}) \qquad (6.25)$$

Equation 6.25 shows that the phase of the carrier fringes is proportional to the shift angle θ, recording distance z_1 and wavelength λ_1, reconstruction distance z_2 and wavelength λ_2, angle between normal of hologram plate and the reference beam α_0. If the recording distance z_1 is equal to the reconstruction distance z_2 and the reconstruction wavelength λ_2 is the same as the recording wavelength λ_1, the Eq. 6.25 can be simplified as

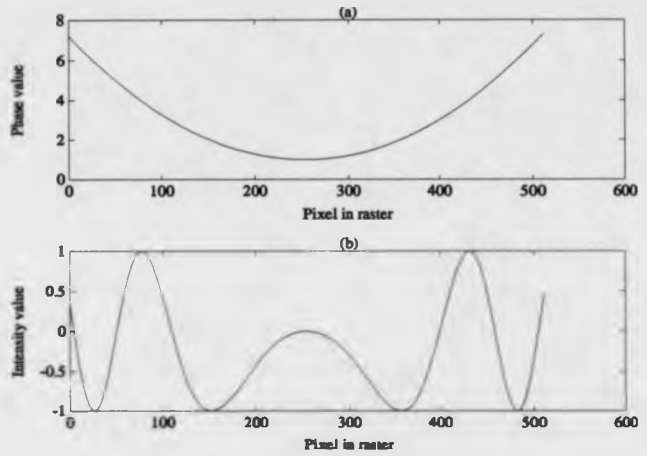


Figure 6.16: (a) Object phase variation along the pixel positions; (b) intensity distribution of interference fringes of the object under study.

$$\phi = \frac{\theta\ (z_1\ \alpha_0 - x_3)}{\lambda_1} \tag{6.26}$$

6.5 The Effects of Carrier Fringes on the Interference Phase Measurement

The irradiance distribution in a general interferogram can be written as

$$g(x,y) = g_0(x,y)[1 + v(x,y)\cos\Phi(x,y)] + n(x,y) \tag{6.27}$$

where $g(x,y)$ is the measured intensity distribution, $g_0(x,y), v(x,y)$ and $n(x,y)$ are the background illumination, fringe contrast and noise distribution. $\Phi(x,y)$ is the phase of the object wavefront. Figure 6.16(a) is the object phase variation $\Phi(x,y)$ along the pixel positions.

The intensity distribution of the object under study is shown in Fig.6.16(b). Figure 6.17 displays the power spectrum of the interference fringes of the object

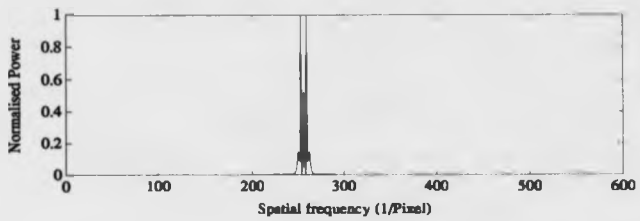


Figure 6.17: Spectrum of the interference fringes of the object under study.

under study.

By tilting the wave front of the interferometer, the carrier fringes are generated. In this case, the phase value of carriers is linear as shown in Fig. 6.18(a). Figure 6.18(b) is the intensity distribution of the carrier fringes. Figure 6.19 shows the spectrum of the linear carrier fringes.

After adding the linear carrier fringes to the fringe field under study, Eq. 6.27 can be written in the form

$$g(x,y) = g_0(x,y)[1 + v(x,y)\cos(\Phi(x,y) + \Phi_0(x,y))] + n(x,y) \tag{6.28}$$

where $\Phi_0(x,y)$ is the phase value due to the carrier fringes, which is given by

$$\Phi_0(x,y) = 2\pi f_0 x \tag{6.29}$$

where f_0 is the spatial carrier frequency. This spatial carrier frequency is modulated by the object phase $\Phi(x,y)$, so that the interference fringes become closely spaced and nearly linear. Figure 6.20(a) is the phase value of the carrier fringes modulated by the object. Figure 6.20(b) shows the intensity variation of the object under study with the carriers. A spectral plot of the intensity distribution is shown in Fig. 6.21.

According to the Nyquist sampling theorem, the phase value between the

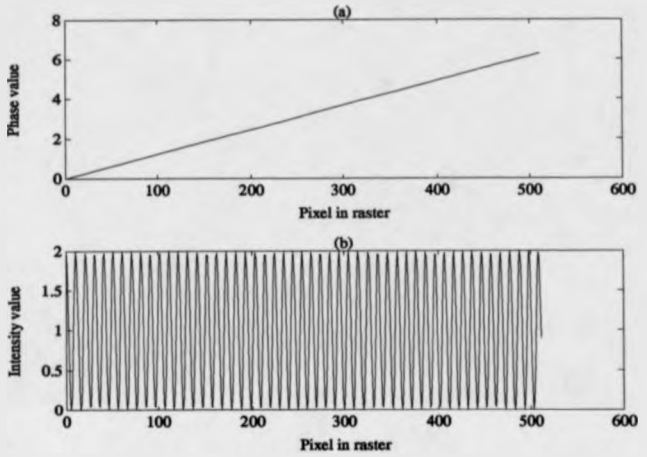


Figure 6.18: (a) Linear phase value of the carrier fringes; (b) intensity distribution of the linear carrier fringes.

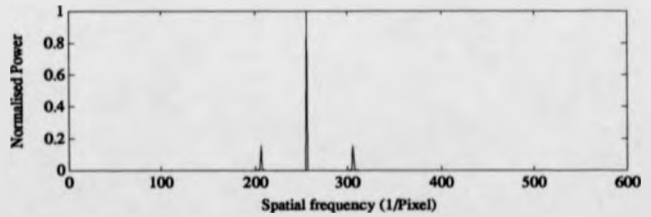


Figure 6.19: Spectrum of the linear carrier fringes.

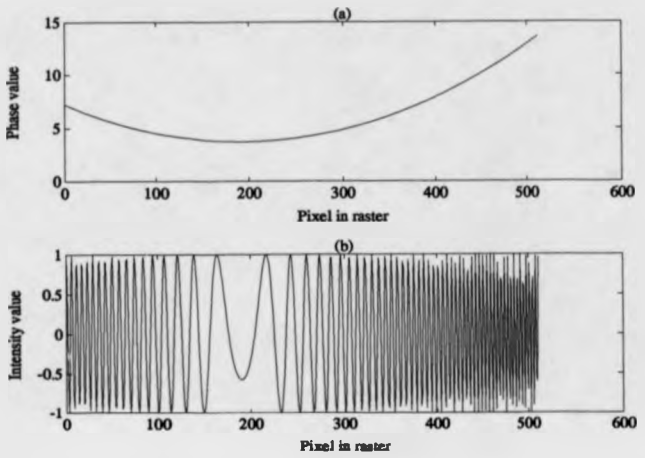


Figure 6.20: (a) Phase value of the carrier fringes plus object; (b) intensity distribution of the carrier fringes modulated by the object.

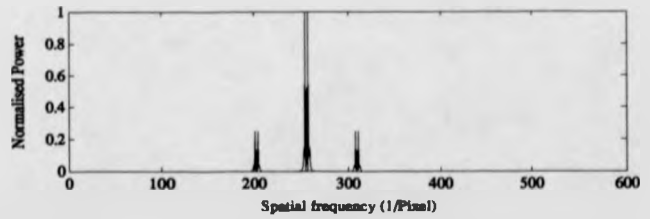


Figure 6.21: Spectrum plot of the carrier fringe plus the object fringes.

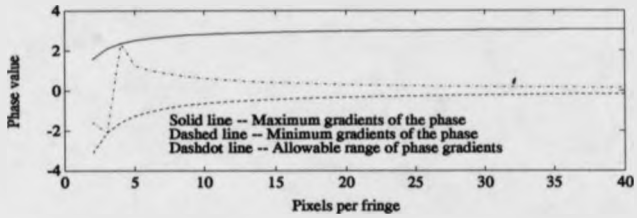


Figure 6.22: Allowable gradients for different spatial carrier frequencies.

adjacent pixels must be less than π, provided that the interference pattern is sampled using a sampling rate f, the condition for phase difference is [162]

$$max|grad\Delta(\Phi + \Phi_0)| < \pi f \tag{6.30}$$

where Φ_0 is the carrier fringe phase value ($\Phi_0 = 2\pi f_0 x$). From Eq. 6.30, the conditions for limiting the range of the allowable gradients of the object phase using the spatial carrier frequency can be given by

$$\Delta\Phi_{max} < \pi(1 - 2\frac{f_0}{f}) \tag{6.31}$$

$$\Delta\Phi_{min} < -2\pi\frac{f_0}{f} \tag{6.32}$$

Figure 6.22 shows the maximum and minimum gradients of the phase. They depend on the ratio of the spatial carrier frequency and the sampling rate Eqs. 6.31 and 6.32.

Note that the difference between the maximum and minimum gradients is always π. The minimum of the absolute value of the gradients is given by

$$\Delta\Phi_x = min(\Delta\Phi_{max}, |\Delta\Phi_{min}|) \tag{6.33}$$

The phase value using Eq. 6.33 is plotted in Fig. 6.22 as dashdot line. It

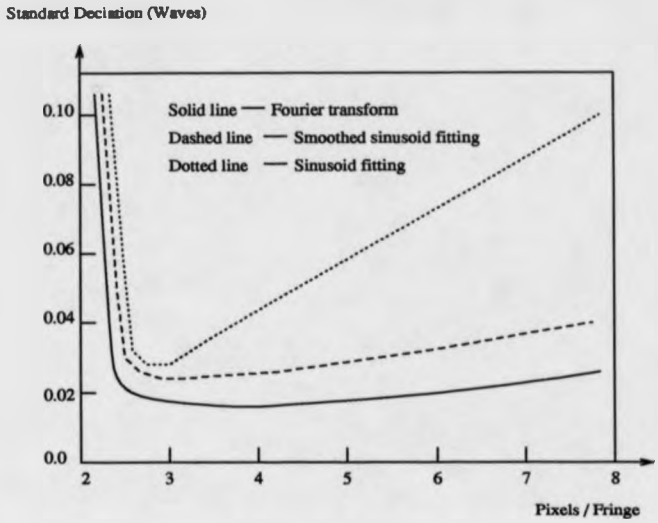


Figure 6.23: Dependence of the accuracy of each method on the carrier frequency.

has been found that the optimum ratio of f_0/f is 1/4, (four pixels per fringe), because the allowable range is divided into equal parts of positive and negative gradients.

Macy [95] also reported results indicating the dependence of the accuracy of each method on the frequency of the carriers. Figure 6.23 shows the relationship between the standard deviation of the phase computed from the test interferograms with the known phase and carrier frequency [95]. It is shown that the accuracy decreases sharply as the fringe spacing approaches 2, which corresponds to the Nyquist frequency, and the accuracy also decreases as the carrier fringe spacing increases. For the Fourier transform method, the optimum fringe spacing is 4 pixels per fringe, but for the sinusoid-fitting and smoothly sinusoid-fitting methods, the optimum fringe spacing is 3 pixels per fringe.

The effects in the inference fringe pattern due to the spatial carrier frequency have been investigated by using computer generated holograms. In this simulation, instead of sampling by a CCD camera, a hologram of a vibration disk

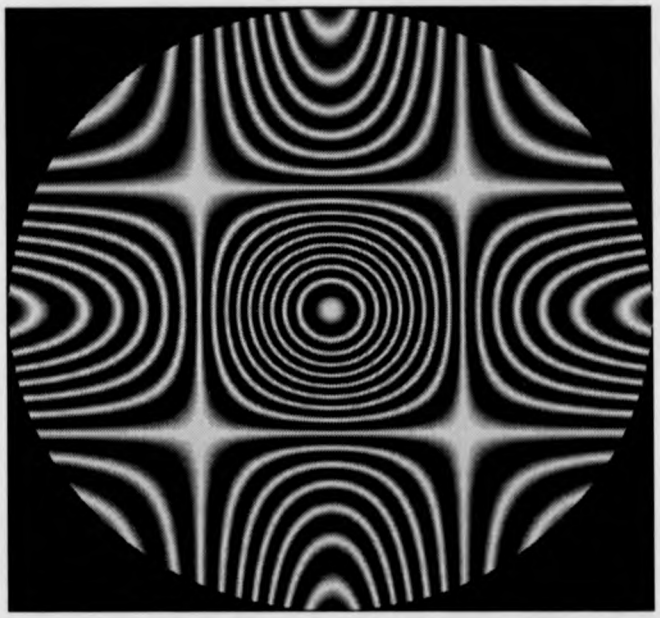


Figure 6.24: Computer generated holograms of vibration disk (image size 512 × 512).

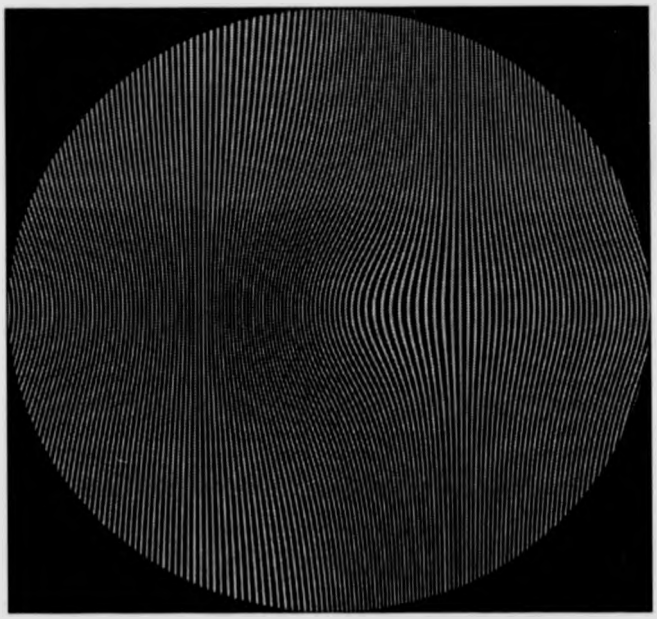


Figure 6.25: Computer generated hologram of vibration disk plus carrier fringes (4 pixels/fringe, image size 512 × 512).

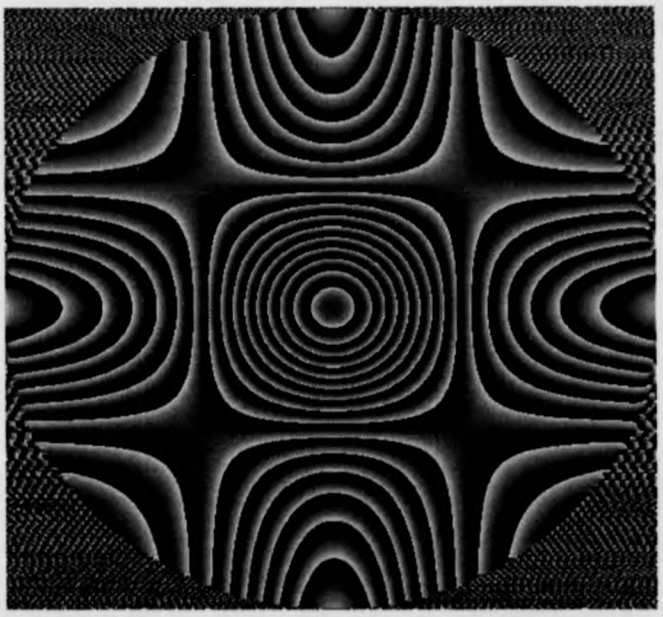


Figure 6.26: Wrapped phase map of vibration disk (image size 512 × 512).

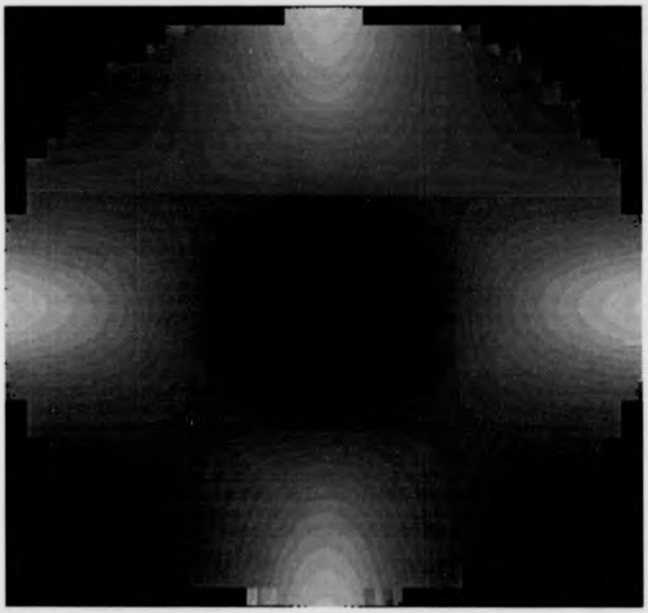


Figure 6.27: Unwrapped phase map of a vibration disk (image size 512 × 512).

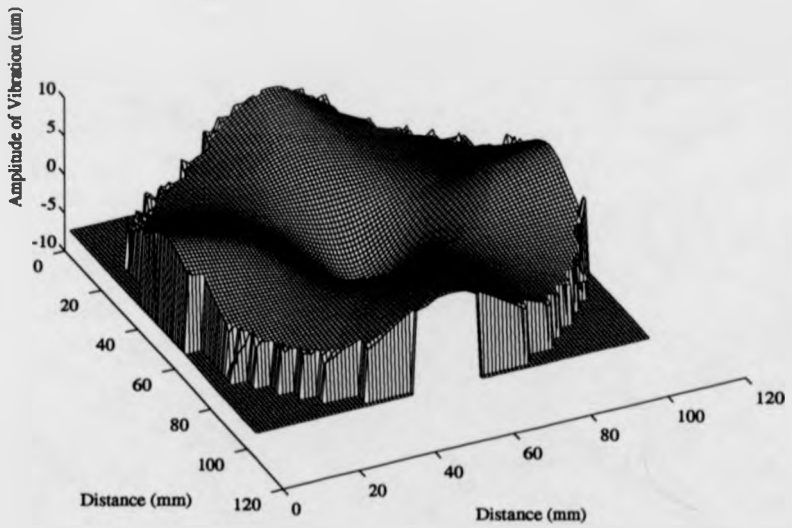


Figure 6.28: 3-D mesh plot of the vibration disk.

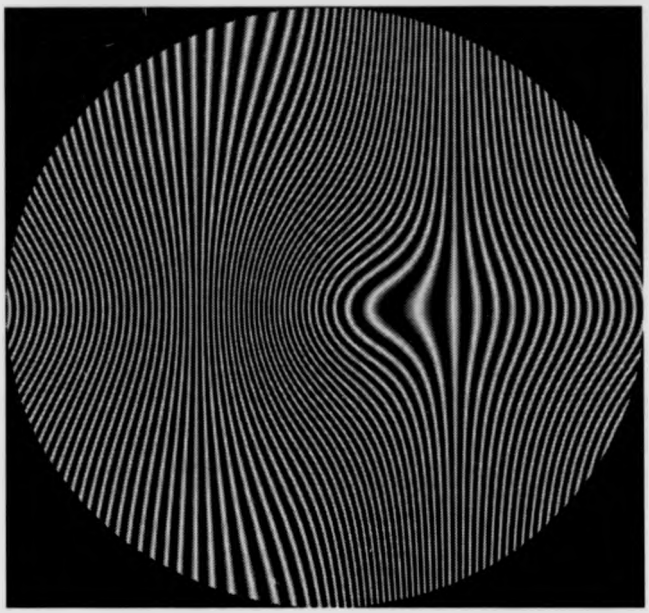


Figure 6.29: Computer generated hologram of a vibration disk plus carrier fringes (9 pixels/fringe, image size 512 × 512).

generated by computer is shown in Fig. 6.24. The carrier fringes (4 pixels per fringe) are also generated by the computer. They are shown superimposed on Fig. 6.24 in Fig. 6.25.

By taking the Fourier transform, the wrapped phase map is computed, as shown in Fig. 6.26. The unwrapped phase map is obtained by using the phase unwrapping procedure, as shown in Fig. 6.27. Figure 6.28 shows the 3-D mesh plot of the vibration disk.

If the carrier frequency is over a certain limit, the unwrapped phase map can not be recovered from the Fourier transform. On the other hand, if the carrier frequency is under a certain limit, for example, approaching the Nyquist sample frequency, the wrapped phase map will have large errors.

Figure 6.29 is a computer generated hologram of a vibration disk with carrier

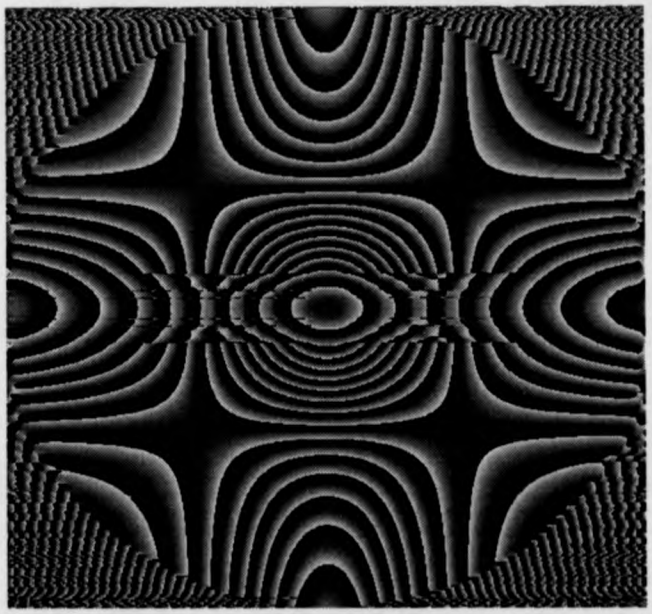


Figure 6.30: Wrapped phase map of a vibration disk with some errors (image size 512 × 512).

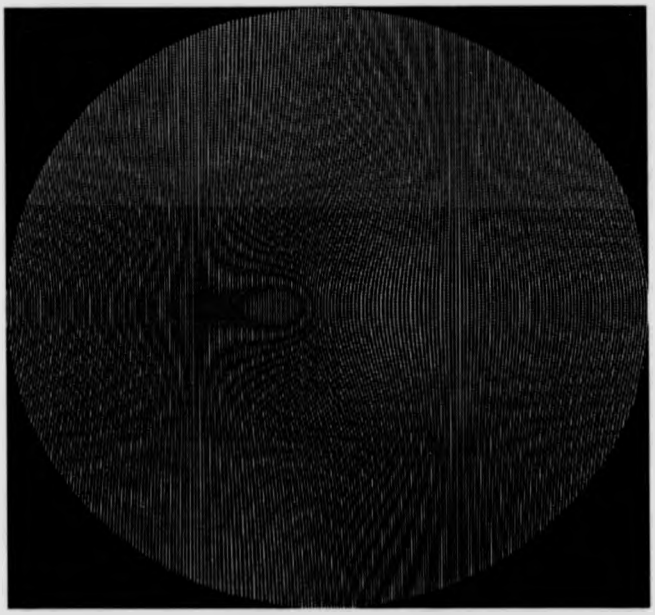


Figure 6.31: Computer generated hologram of a vibration disk plus carrier fringes (2.2 pixels/fringe, image size 512 × 512).

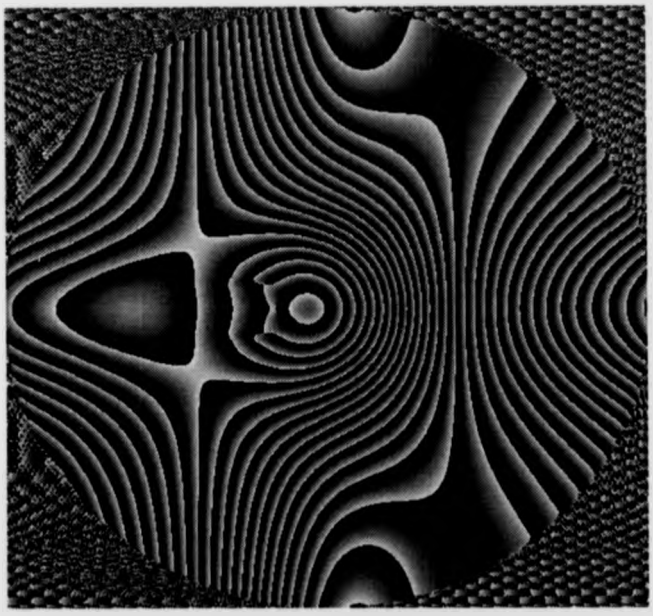


Figure 6.32: Wrapped phase map of a vibration disk with some errors (image size 512 × 512).

fringes, a carrier frequency corresponding to 9 pixels/fringe. Figure 6.30 shows the wrapped phase map of the computer generated hologram of the vibration disk. Note some strips in the centre part of the wrapped phase map.

Figure 6.31 is a computer generated hologram of a vibration disk with the carriers, a carrier frequency corresponding to 2.2 pixels/fringe. Figure 6.32 shows the wrapped phase map of a vibration disk. It is shown that the true wrapped phase can not be recovered if the carrier frequency is approaching the Nyquist frequency limit.

6.6 Conclusion

In this chapter, the feasibility of the carrier fringe and FFT technique concepts has been demonstrated. A theoretical analysis of the technique is given. The effects of carrier fringes on the interference phase measurement are investigated with computer generated holograms. Several different methods for generating carrier fringes are discussed. Carrier patterns are easily introduced and controlled by adjustments of the fibre optic holographic system. The procedure for extracting the data is facilitated when carrier fringes are introduced. Once carrier fringes have been generated automatic data reduction using image processing software and digital Fourier transform methods is possible. The major advantage of the technique is that only one image is needed to extract the interference phase information. The carrier fringe and FFT technique can be applied to various fringes obtained by photoelasticity, holographic interferometry, speckle interferometry, moire interferometry, and other interferometry.

Chapter 7

Holographic Deformation Measurements by Carrier Fringe and FFT Techniques

The work done in this chapter was done in collaboration with Dr. T. R. Judge. I performed all the experimental work including test rig build-up, hologram recording and reconstruction as well as parts of the data analysis. Dr. Judge made contributions on fringe analysis aspects especially on the tile-level, minimum spanning tree and phase unwrapping techniques. This chapter describes a carrier fringe technique for measuring surface deformation. In contrast to conventional holography and fringe analysis, this holographic system is based on the fibre optics and automatic spatial carrier fringe pattern analysis techniques. Carrier fringes are generated by simply translating the object beam between two exposures. Single-mode optical fibres are used to transfer both the object and reference beams. The use of optical fibres offers advantages in the field of optical holography. The fast Fourier transform method is used to process the interferograms. This method of fringe pattern analysis is a successful technique for extracting phase information from fringe patterns which result from the interference of tilted wavefronts. The chapter also introduces the tile-level, minimum spanning tree

and phase unwrapping techniques. The method is illustrated by measuring the centrally loaded disk and side loaded cantilever beam. A correction is made for the non uniform spacing of the carrier fringes, which was introduced by a slight inclination of object. The results are given for an aluminium disk and a cantilever beam, including the perspective plot of the out of plane deformation field, the maps of wrapped and unwrapped phase, and a contour map of the unwrapped phase.

7.1 Introduction

A major limitation in applying holographic interferometry to deformation studies is the difficulty in determining the sign of the displacement vector; i.e. whether the surface has moved toward or away from the observer [164]. Moreover, the interference fringe patterns obtained in practical cases are often complicated that it is difficult to determine fringe order. Such problems have been obstacles to the widespread application of holographic interferometry for automatic data acquisition. Recently, several methods have been suggested for solving these problems by the addition of a carrier fringe [40, 41, 43, 160, 163]. The idea behind of the technique is that by introducing a known linear carrier fringes, the deformation fringes are modulated by superimposed carriers. Within defined limits this makes the fringe orders of the resulting fringe pattern monotonically increasing. The fringe orders in the interferogram can then be determined in a straightforward way, therefore, providing the automatic data reduction.

The Fourier transform method of fringe pattern analysis is a successful technique for extracting phase information from fringe patterns that result from the interference of tilted wavefronts [27, 28, 87, 114]. Tilting the wavefront generates a linearised fringe pattern which facilitates the use of computerised automatic analysis. The Fourier transform methods for computing phases can produce phase maps automatically without the need for locating fringe centres or assigning fringe orders. Compared with the phase-stepping method, where three or

four reconstructions with mutual phase shifts are recorded and evaluated, only one interference fringe pattern is needed provided the 'direction' of the added carrier fringes is known. If this information is not available, then a solution might be produced that is a negative of the true deformation, like a mold is to a cast object.

Having applied the FFT technique, the phase distribution is wrapped into the range between $-\pi$ and π and consequently has discontinuities with 2π-phase jumps for variations more than 2π. This is known as a "wrapped" phase map. Phase-unwrapping schemes are required to produce a continuous phase distribution. The "unwrapped" full-field phase can now be related to the measurement parameter encoded with the intensity distribution. A variety of phase-unwrapping approaches have been proposed over the last few years. The following are useful references. For fringe scanning, Refs. [165] and [166] present an previous method which is prone to corruption due to noise. For cellular automata, Refs. [100] and [101] present a capable method requiring a large number of iterations, but which does not address large scale discontinuities. Reference [167] presents a regional method, with some similarities to tiling found in Refs. [98] and [113]. Reference [99] presents a method that places walls or cut lines in the phase map to prevent unwrapping through erroneous regions. Finally, a method that inspects the frequency bandwidth of the fringe pattern to eliminate the steps in phase at fringe boundaries is found in Ref. [168].

The method presented here is immune to spike noise, and unlike the algorithms listed above also presents a good degree of immunity to larger discontinuities (detected via tiling) that cannot be detected by pixel comparisons. Such discontinuity types might be introduced by aliasing, for example.

7.2 Interpretation of Deformation Fringe Patterns in Holographic Interferometry

When the hologram is recorded, the main source of information on the changes that occur in an object is the interference pattern. This interference fringe pattern can be interpreted by the vectors analysis method.

In the holographic recording system, the light source is located at the point O. The object surface being studied is displaced between exposures from point P_1 to P_2. The interference phase value at the point P_3 can be calculated [48, 72]

$$\phi_1 = \vec{k}_1\vec{r}_1 + \vec{k}_2(\vec{R} - \vec{r}_1) + \phi_0 \tag{7.1}$$

$$\phi_2 = \vec{k}_3\vec{r}_3 + \vec{k}_4(\vec{R} - \vec{r}_3) + \phi_0 \tag{7.2}$$

where ϕ_0 is the constant phase shift of light wave of λ.

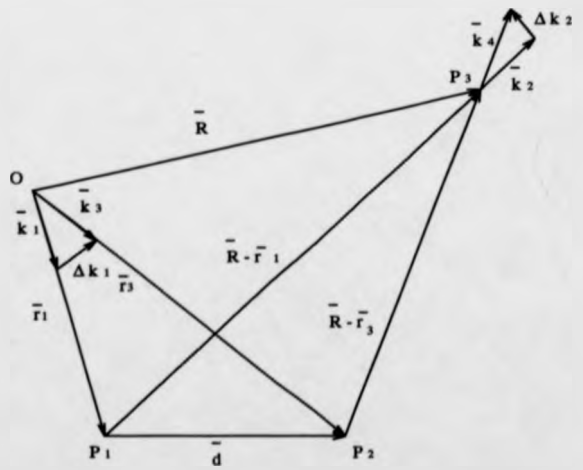


Figure 7.1: Schematic diagram of vector analysis for hologram recording (Ostrovsky *et al* 1981).

The phase difference at point P_3 is

$$\phi = \phi_2 - \phi_1 = (\vec{k}_1 - \vec{k}_2)(\vec{r}_1 - \vec{r}_3) - \Delta\vec{k}_1\vec{r}_3 - \Delta\vec{k}_2(\vec{R} - \vec{r}_3) \tag{7.3}$$

If the distance $|\vec{r}_1| \doteq |\vec{r}_3| \gg |\vec{d}| = |\vec{r}_3 - \vec{r}_1|$, then $\Delta\vec{k}_1$ and $\Delta\vec{k}_2$ are perpendicular to $\vec{r}_3$ and $(\vec{R} - \vec{r}_3)$, respectively

$$\phi = (\vec{k}_1 - \vec{k}_2)(\vec{r}_1 - \vec{r}_3) \tag{7.4}$$

Equation 7.4 is the basic relationship for analysis of static displacement in any holographic interferogram of diffusely reflecting surface.

In the case of two dimensions, all of the vectors in Eq. 7.4 are in one plane, as shown in Figure 7.2

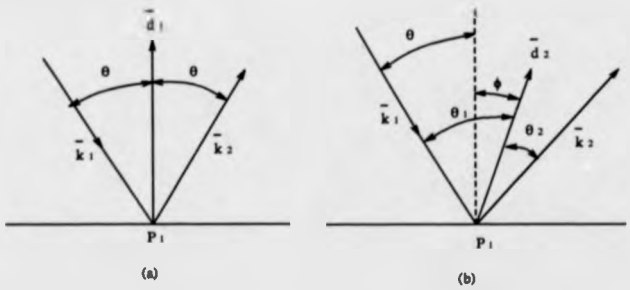


Figure 7.2: Vector analysis of hologram formation in two dimensional case (Ostrovsky *et al* 1981).

The displacement vector $\vec{d}$ is coplanar with the incident and scattered light as shown in Figure 7.2(a) and can be expressed

$$\vec{d} = \vec{r}_3 - \vec{r}_1 \tag{7.5}$$

Therefore, the Equation 7.4 becomes

$$\phi = -(\vec{k}_1 - \vec{k}_2)\vec{d} \tag{7.6}$$

Case 1: $\theta_1 = \theta_2$, In this case, the displacement vector $\vec{d_1}$ is normal to the object surface. Let $\theta_1 = \theta_2 = \theta$, since $|\vec{k_1}| = |\vec{k_2}| = \frac{2\pi}{\lambda}$ and $\cos(\pi - \theta) = -\cos\theta$, we have

$$\phi = \frac{4\pi}{\lambda} d_1 \cos\theta \tag{7.7}$$

If the number of interference fringes is m, we get

$$2m\pi = \frac{4\pi}{\lambda} d_1 \cos\theta$$

$$d_1 = \frac{m\lambda}{2\cos\theta} \tag{7.8}$$

Case 2: $\theta_1 \neq \theta_2$, in this case, the displacement vector $\vec{d_2}$ is not perpendicular to the surface of the object, as shown in Figure 7.2(b). The angle θ and Φ are introduced in Figure 7.2(b), i.e. $\theta = \frac{\theta_1 + \theta_2}{2}, \Phi = \frac{\theta_1 - \theta_2}{2}$. Therefore,

$$\phi = \frac{4\pi}{\lambda} d_2 \cos\theta \cos\Phi \tag{7.9}$$

Similarly, we have

$$d_2 = \frac{m\lambda}{2\cos\theta\cos\Phi} \tag{7.10}$$

where m is the fringe number.

From Equations 7.8, 7.10, the difference of displacement is given by

$$\Delta d = d_2 - d_1 = \frac{m\lambda}{2\cos\theta}[\frac{1}{\cos\Phi} - 1] \tag{7.11}$$

When $\Phi = 0, \theta_1 = \theta_2, \Delta d = 0$. This is the measurement for symmetrical case. When $\Phi \neq 0, \theta_1 \neq \theta_2, \Delta d \neq 0$, which is the measurement for non-symmetrical case.

7.3 Image Processing Technique

7.3.1 Windowing to Separate out a Sidelobe

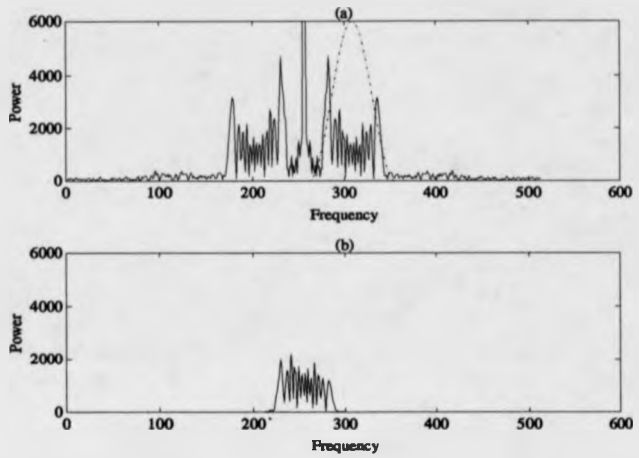


Figure 7.3: Various window functions for FFT fringe analysis.

Each raster of the interferogram produces a power spectrum similar to that shown in Fig. 7.3. Each of the spectral sidelobes represent modulated fringes contouring the measurement parameter. The centre lobe represents unwanted background variations in the intensity of the interferogram, which are low in frequency.

The FFT technique eliminates background variations by considering just one of the sidelobes. The frequency tails of the lobes may overlap if the carrier frequency is not high enough, so an appropriate division point between the lobes must be found. An upper frequency limit for the sidelobe must also be found. Once a division point has been determined, a window may be used to isolate the sidelobe using the cut points to determine the size of the window.

A variety of strategies may be employed to decide upon these cut points. Here, the average power in a band of frequencies at either side of the carrier has

been measured at successive offsets from the carrier. The criterion for these cut points is that the average power in the check-bands must fall below a certain threshold on both sides of the carrier. This threshold is initially set very low. If no bands are found with an average power less than the threshold, then the threshold is incremented slightly and the process repeated. The window function is computed over the range that this test suggests; the window may differ from scan line to scan line.

In the FFT fringe analysis, several different weighting functions are used as shown in Fig. 7.4

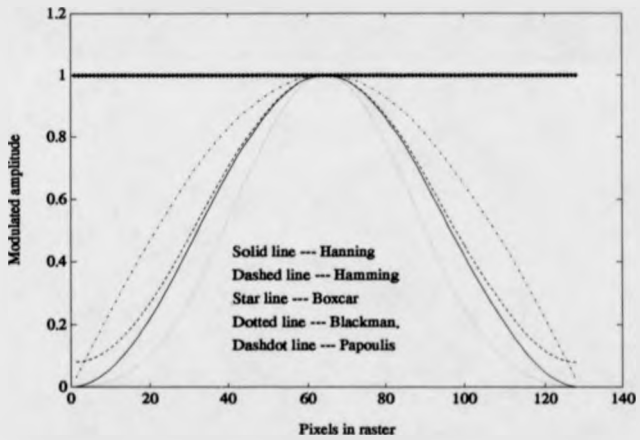


Figure 7.4: Various window functions for FFT fringe analysis.

The corresponding frequency functions are illustrated in Fig. 7.5.

As shown in Fig. 7.5, all weighting functions have sidelobe in the frequency domain of lower amplitude than that of the rectangular functions. Therefore, these weighting functions produce less leakage than that of the rectangular function. However, these weighting function also have a broader central lobe. Because the spatial domain weighting is a frequency domain convolution, the broader the central lobe of the weighting function, the less the capability of the FFT to dis-

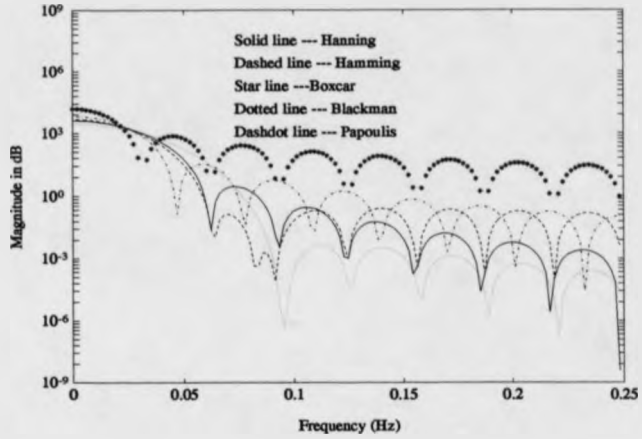


Figure 7.5: The frequency response of various window functions.

tinguish frequencies. There is a relationship between leakage and the central lobe bandwidth. The smaller the sidelobes are, the more smeared the results of the FFT appear.

In the experimental work described here, a Papoulis window has been employed in the spatial domain to bring the spatial data to zero at the edges of the images, as shown in Fig. 7.6.

This window has been selected for its sharp cutoff. It is obtained by taking the square root of the Hanning window. A Hanning window has been used in the frequency domain to isolate the sidelobe, as shown in Fig. 7.3. The windowing procedures can have dramatic effects on the quality of the results obtained. This can be illustrated by Figs. 7.7, 7.8, 7.9, 7.10. Figure 7.7(a) shows the digitised intensity data weighted by a boxcar window, and 7.7(b) shows the sidelobe of the power spectrum also weighted by a boxcar window. After calculating the phase from the arctangent of the ratio of the imaginary and real parts of the inverted FFT, the wrapped phase map can be obtained as shown in Fig. 7.8. Note that a phase error is introduced. The right hand side of the phase map is smeared

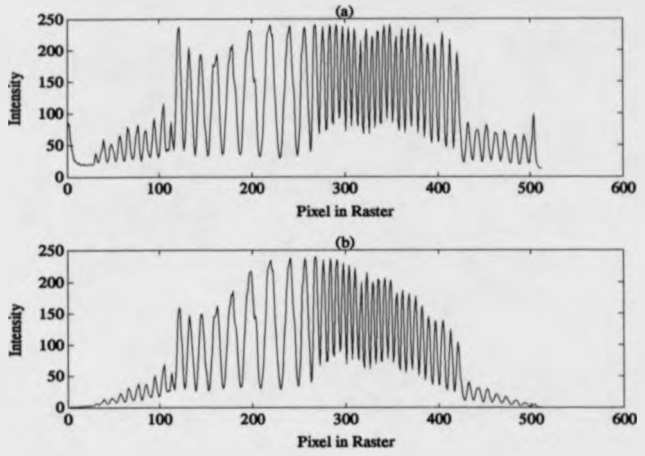


Figure 7.6: (a) Digitised intensity data of central raster in the interferogram; (b) intensity data weighted by a Papoulis window.

by using the boxcar window in the spatial domain. In comparison, Fig. 7.9(a) shows the same intensity data weighted by a Papoulis window, and with the same condition in frequency domain, i.e. the sidelobe of the power spectrum weighted by a boxcar window as shown in Fig. 7.9(b). The wrapped phase map is shown in Fig. 7.10. In this case, there is no phase error introduced by Papoulis window.

It should be addressed that the weighting function whose characteristics are best suited to the problems being encountered.

Kreis [28] proposed a two dimensional transform for the evaluation of an interferogram of a thermally loaded panel. The procedure described by Kreis enables the sidelobe to be isolated without a windowing operation in the Fourier domain. The sidelobe is isolated from the centre lobe without having to use a window. However, in his method, two images are required. A background image is needed in addition to the encoded fringe pattern.

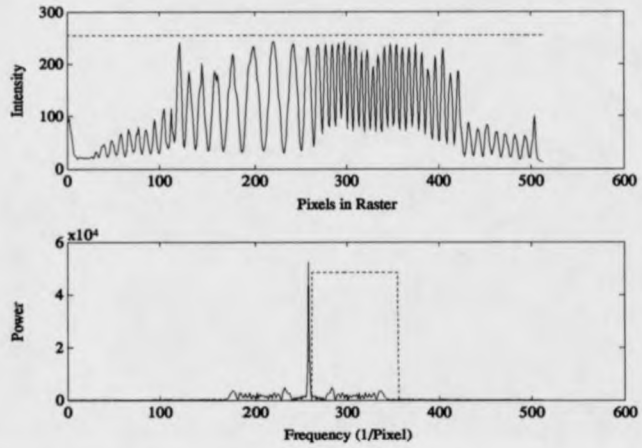


Figure 7.7: (a) Digitised intensity data weighted by a boxcar window (indicated by the dashed line) in the interferogram; (b) power spectrum of central raster from the interferogram (sidelobe weighted by a boxcar window).

Figure 7.8: Wrapped phase map with some errors (image size 512 × 512).

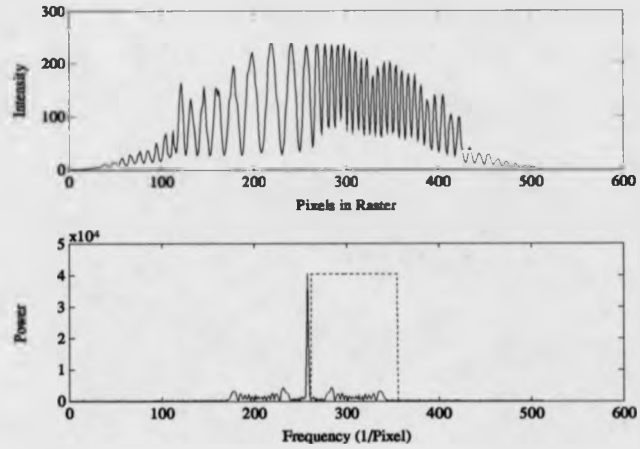


Figure 7.9: (a) Digitised intensity data weighted by the Papoulis window; (b) power spectrum of central raster from the interferogram (sidelobe weighted by a boxcar window).

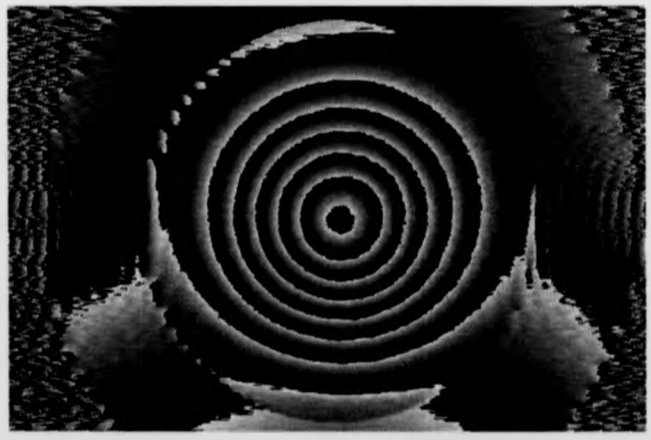


Figure 7.10: Wrapped phase map without errors (image size 512 × 512.

7.3.2 Accounting for Areas of the Interferogram Where Fringes are Absent

The FFT technique, in common with the quasi-heterodyne method produces a wrapped phase map. However, the use of a carrier frequency introduces an added complexity that is not present in quasi-heterodyne maps. That is, those parts of the field lacking the carrier may be aliased so that rogue fringes appear in the wrapped phase map.

In a phase-stepping system with three fringe fields at a phase step of α, the expression below may be employed to compute the phase ϕ at a given pixel [169]

$$\phi = \arctan\left[\frac{(I_3 - I_2)\cos\alpha + (I_1 - I_3)\cos 2\alpha + (I_2 - I_1)\cos 3\alpha}{(I_3 - I_2)\sin\alpha + (I_1 - I_3)\sin 2\alpha + (I_2 - I_1)\sin 3\alpha}\right] \tag{7.12}$$

where I_1, I_2, and I_3 are the intensities of the interferograms at the three phase positions α, 2α, and 3α, respectively.

A low-modulation point may be identified when both the numerator and denominator are small for the quantity under the arctan. The probability of error in ϕ is then high.

This is a very successful method of detecting bad data points in phase-stepping systems (The critical size for the numerator and denominator is experimentally determined). It was seen above that in a system employing the Fourier transform technique, the phase is computed from Eq. 3.63.

The FFT analysis technique assumes that carrier fringes are present in all areas of the interferogram. However, in practice this may not be the case. For example, an object with fringes projected on to it may be set on a neutral background, or experimental apparatus may cast shadows across the field.

The low-modulation test may be applied to the FFT method to detect bad data points, but it is not as effective as when applied to the phase-stepping method. Figure 7.11 shows in white the low-modulation points that were detected

during the FFT process for the disk. While a percentage of bad (aliased) data points have been detected around the edge of the image by the low-modulation test, many aliased points have remained undetected. This is because some of the bad data points have been left well modulated by the FFT process, and are therefore not detected by the low-modulation test.

Figure 7.11: Bad data points detected by the low modulation test as shown in white (image size 512 × 512).

An alternative strategy for detecting areas lacking carrier fringes is to examine the digitised cosinoidal interferogram prior to the FFT process to locate areas where fringes are absent. A thresholding strategy may then be employed to detect the presence or absence of fringes. This strategy has been implemented along with the low-modulation test; the areas examined correspond to those of the tiles used in phase unwrapping. Any tiles covering an area lacking the carrier are not processed.

Donovan *et al* [170] introduce the carrier into such areas of the original cosinoidal interferogram that have lacked them. These areas are again detected by a thresholding strategy.

7.3.3 The Minimum Spanning Tree Approach to Phase Unwrapping, Applied to the FFT Method

The tree is employed in a hierarchical unwrapping strategy [62, 107]. At the lowest level paths between pixels are compared to circumvent spike noise. At the higher level tiles of pixels are compared to circumvent aliasing induced inconsistencies and natural inconsistencies. The latter might be formed by objects overlapping, for examples.

7.3.3.1 Pixel Level Noise Immune Phase Unwrapping

Each pixel in the wrapped phase map is considered to be a vertex in a graph of confidence. The problem is to construct a path for unwrapping which maximises confidence. The weights of graph edges will be used to signal the confidence of a particular route, pixel to pixel.

Each pixel has four neighbours, north, south, east and west and a corresponding edge in the graph for each pixel neighbour. The weight of each edge is calculated from the rate at which phase would change if unwrapping took place between each pixel pair. The minimum spanning tree would therefore minimise pixel to pixel phase changes in the unwrap path.

Spike noise may be characterised by a rapid change in phase. If phase is unwrapped by a path tree which seeks to minimise the change in phase at each step, then noise points are prevented from entering the solution until the tips of the tree branches are reached, where they do not threaten the solution. The minimum spanning tree approach therefore presents a noise immune phase unwrapping strategy.

7.3.3.2 Calculation of Edge Weights at the Pixel Level

The edge weights may be calculated from the horizontal and vertical scans. These contain partially unwrapped data, localised to either rows or columns. The local nature of the data means that the absolute offset of the data relative to the rest

of the field is unknown. However, for the purpose of weight computation, only relative phase changes are necessary.

There is an implicit assumption that the fringe field represents a continuous function that is changing relatively slowly. That is, the signal does not, in the main, contain high frequency components that could be confused with noise. Each pixel effectively represents a crossroads at which the best direction to move, to minimise phase unwrapping errors must be determined. The vertical scan set weighs the merits of the north/south options, while the horizontal scan set weighs the merits of the east/west options.

The steps in the unwrapping procedure include:

1. computation of east/west pixel edge weights
2. computation of north/south pixel edge weights
3. phase unwrap during calculation of minimum spanning tree.

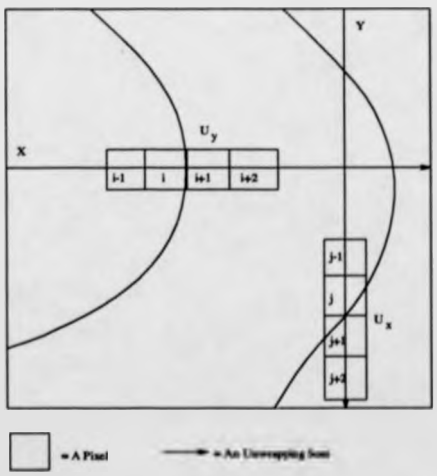


Figure 7.12: Pixel level, computation of edge weights. Weights are computed over four pixels for each pair.

The edge weights are actually computed as an average over a two pixel area(see Fig. 7.12). The effect of an *averaging* filter placed over a spike noise point is to spread the spike so that it raises the values of surrounding pixels. Using the average in the graph-weighting strategy serves to increase the weight of graph edges, which connect to pixels surrounding the noise point. These are embedded in the unwrapping path before the noise spike is actually reached, thereby causing the algorithm to delay further its approach to the noise spike.

In reference to Fig. 7.12, X and Y represent a horizontal and vertical unwrap scan, respectively. Let the unwrapped phase of row scan y be denoted by $U_y(i)$, where i is an index to the pixels of the scan, and similarly let $U_x(j)$ represent the unwrapped phase of column x at pixel j of the scan. For a row pixel pair i and $i+1$, the weight of the edge connecting the pixel vertices of row y is denoted by w_y and calculated via Eq. 7.13 from the unwrapped rows,

$$w_y = |(U_y(i) + U_y(i-1)) - (U_y(i+1) + U_y(i+2))| \tag{7.13}$$

and similarly the north/south edge weights are calculated from the unwrapped columns in Eq. 7.14;

$$w_x = |(U_x(j) + U_x(j-1)) - (U_x(j+1) + U_x(j+2))| \tag{7.14}$$

The change in phase between the unwrapped pixel pairs δ is also recorded so that phase may be unwrapped during formation of the minimum spanning tree;

$$\delta = U(i+1) - U(i) \tag{7.15}$$

7.3.3.3 Tile Level Phase Unwrapping

This level of phase unwrapping is aimed at dealing with local area inconsistencies, such as aliasing induced inconsistencies, which are not distinguishable at the pixel level.

A similar strategy applies to that adopted for coping with spike noise in that

confidence is weighed by the minimum spanning tree. The goal is to keep areas of inconsistency contained within tiles. By comparing the unwrapped edges of neighbouring tiles, these contained inconsistencies may be detected and avoided in the unwrap path. The tiles are made to overlap slightly to facilitate this comparison.

To further aid in the avoidance of inconsistencies, the graph weights also take account of the number of fringes across tiles so that areas where fringes are sparse are unwrapped first, and those where fringes are dense last.

In advance of pixel-level unwrapping, an edge detection is performed on the tile. If no fringe edges are found, then clearly the tile does not need to be unwrapped, therefore the pixel level phase unwrap is bypassed. This significantly improves the speed of processing.

Tiles may be completely omitted from the solution if for example they contain a high percentage of low modulation points, or cover an area where carrier fringes are not present. This tiling approach has been described in Refs. [39, 62, 98, 107, 113, 171].

7.4 Experimental Description and Results

7.4.1 Direct Measurement of the Test Objects

To demonstrate the fringe analysis method for measuring deformation, an aluminium disk and a cantilever beam have been chosen as experimental objects. A direct deformation measurement for an ideal clamped circular disk may be calculated from theory. The deflection of the disk at a distance r from the centre is given by [172]

$$w = \frac{Fr^2}{8\pi D} ln \frac{r}{R} + \frac{F(R^2 - r^2)}{16\pi D} \tag{7.16}$$

where

$$D = \frac{Eh^3}{12(1-v^2)} \tag{7.17}$$

Equations 7.16 and 7.17 are for a disk of radius R, thickness h, Young's modulus E and Poisson's ratio v. The load, F, is determined from the change in deflection of the centre of the disk imposed between exposures. In the case of the aluminium disk, the parameters for Eqs. 7.16 and 7.17 are as follows: $R = 50$mm, $r = 20$mm, $E = 70$kN/mm^2, $v = 0.33$, $h = 2$mm. By using a precision gauge the deformation w_1 measured at the radial distance r was 1.3μm $\pm$ 20nm on a precision gauge. Therefore, the force F can be calculated from the equations above:

$$F = \frac{w_1}{\frac{r^2}{8\pi D} ln\frac{r}{R} + \frac{R^2-r^2}{16\pi D}} = 2.5\text{N}$$

From Eqs. 7.16 and 7.17, the maximum deflection at the centre of the clamped disk, when loaded at the centre, is given by

$$w_{max} = \frac{FR^2}{16\pi D} = 2.38\mu\text{m}$$

The loading scheme and the aluminium disk used for the experiments is shown in Fig. 7.13.

Measurements were also carried out with a test object consists of a vertical cantilever beam 125mm long, 28.5mm wide, and 3mm thick, which was clamped at the lower end and free at the upper end. A concentrated load was applied to the distance 25mm from the free end of the cantilever beam. The precision gauge was used to measure the deflection at the distance 45mm from the free end of the cantilever. The deflection at the distance 45mm from the free end was measured as 3.0 μm.

The test beam and loading scheme is shown in Fig. 7.14. When the load in the position shown in Fig. 7.14 cantilever is effectively only 100mm long, the remaining 25mm being unloaded and therefore not bending. Thus the standard

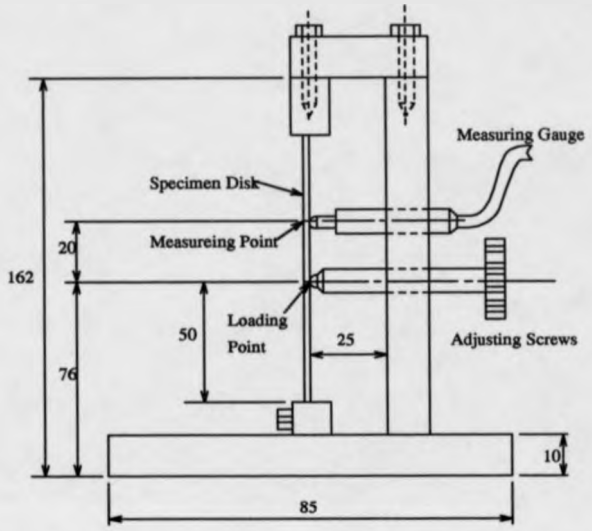


Figure 7.13: The test specimen of clamped aluminium disk.

equations for deflections and slope apply between A and B only. The deflection of the cantilever is given by the following equation [173].

$$w = \frac{F}{6EI}(2l^3 - 3l^2x + x^3) \tag{7.18}$$

where l is the length of the cantilever, x the loading point from the free end. The parameters for effective cantilever beam during the experiment are: length l = 100mm, width h = 28.5mm, thickness t = 3mm, E = 70kN/mm^2, I = 64.1mm^4, therefore, the force F can be calculated from Eq. 7.18,

$$F = \frac{6EIw}{1.408l^3} = 5.74 \times 10^{-2}\text{N}$$

Horizontal deflection (δ_1) and slope (θ_h) of the cantilever beam at point B are given by the following respectively

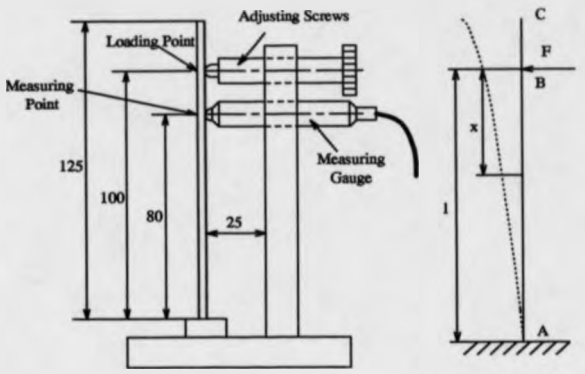


Figure 7.14: The test specimen of a cantilever beam.

$$\delta_1 = \frac{F\ l^3}{2\ E\ I} = 4.26\mu\text{m}$$
$$\theta_h = \frac{F\ l^2}{2\ E\ I} = 6.39 \times 10^{-5}\text{rad}$$

The part BC of the cantilever beam remains straight since it is not subject to bending.

$$\delta_2 = \theta_h\ l = 1.6\mu\text{m}$$

Hence, the horizontal deflection of cantilever beam at point C is

$$\delta = \delta_1 + \delta_2 = 5.86\mu\text{m}$$

7.4.2 Experimental Procedures

A schematic diagram of the experimental system is shown in Fig. 7.15. The first object is a centrally loaded aluminium disk that has a 100-mm diameter. The light from a 25-mW He-Ne laser ($\lambda = 0.6328\mu$m) was launched into the

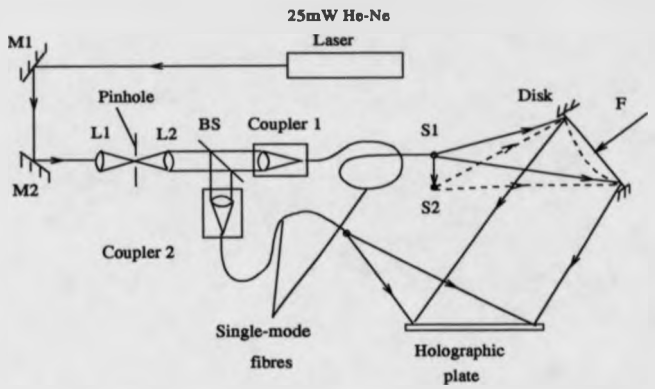


Figure 7.15: Experimental arrangement for recording holograms of the disk by carrier fringe technique.

expander and divided in two by a (60/40) beam splitter (BS). Two single-mode optical fibres about 2 m long were used, one arranged to illuminate the object and the other arranged to illuminate the holographic plate (H). The object is recorded before and after the deformation by the double-exposure technique. As mentioned earlier, the deformation was measured by a precision gauge, which has an accuracy of ± 20 nm. To generate carrier fringes for quantitative analysis, the object beam was moved between the exposures by a microtranslator. The number of carrier fringes is determined by the amount of movement of the object beam. The carrier fringes should be of high enough density to clearly distinguish the sidelobe from the centre lobe. In this case, the frequencies of the deformed fringes are small compared to the spatial carrier fringes. In this experiment the carrier frequency was slightly lower than the ideal. That is, the sidelobe is not clearly separated from the central lobe. The effect of this can be seen in the contour plot of Fig. 7.16.

There is a slight ripple in the contours to the left and right of the central peak. If these areas are inspected in the original interferogram of Fig. 7.17, it shows that these areas have, respectively, low and high frequencies when compared to the carrier. These frequencies, therefore, lie in the frequency tails to left and right of

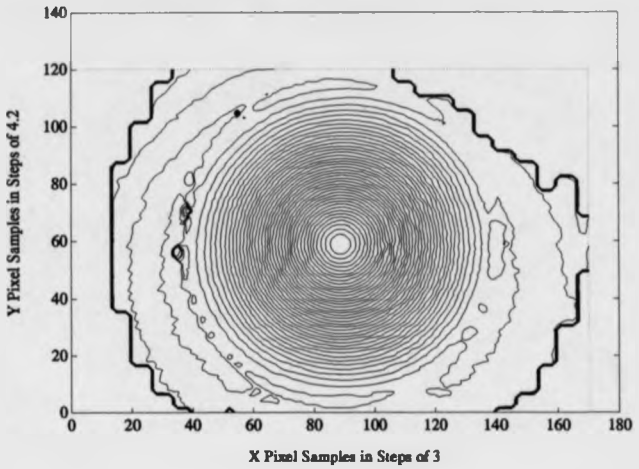


Figure 7.16: Contour map of unwrapped and corrected phase data.

the sidelobe in the power spectrum, see Fig. 7.3. They are therefore particularly sensitive to defects in the windowing procedure that extracts the sidelobe.

An exposure was taken with the illuminating object beam at position S1 in Fig 7.15. Then the object beam was translated 200 μm sideways to position S2 to form the carrier fringes, after which a second exposure was made on the same plate. Between the exposures, the disk was deformed by central loading. The deflection at the centre was estimated from the calculated force as 2.38 μm.

Figure 7.18 shows the system for reconstructing holograms. A reconstructed image is picked up by a CCD camera and is stored in digital frame memory in the form of 512 $\times$ 512 $\times$ 8 bit data. An IBM compatible microcomputer has served as host for the frame capture board. This is connected via Ethernet and the PC Net Filing System (PCNFS) to the departmental Sun Network (data transfer is transparent to the PC, which simply saves an image file to a disk). Image processing takes place on a Sun Sparc Station 1, the result may then be read by the PC and displayed (again transparently by reading a file from a disk). Hardcopy of the results is produced by a laser printer.

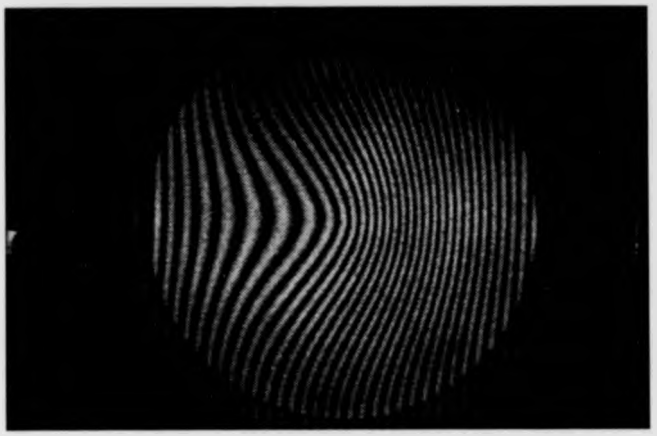


Figure 7.17: Holographic carrier fringe interferogram of the deformed disk.

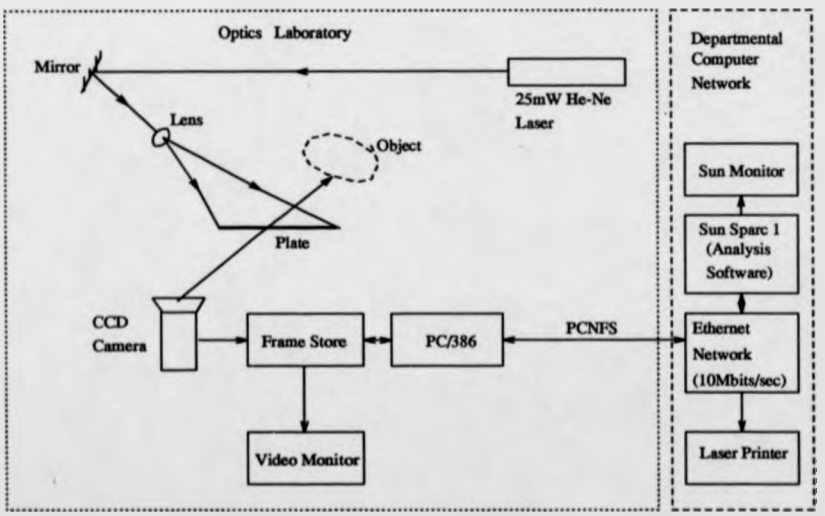


Figure 7.18: Optical and electronic arrangement for reconstruction of the holograms.

7.4.3 Correction for Nonlinearity of Carrier Fringes

The carrier fringe maxima, as seen around the edge of the deformed disk, can be shown to follow Eq. 7.19;

$$x = \frac{m\lambda a}{b + m\lambda \tan\theta} \tag{7.19}$$

where $(m = 0, 1, 2, 3, ...)$ is the carrier fringe order number, x is the distance of the intensity maxima in the carrier fringes from an origin position on the flat disk, b is the displacement between source position S1 and S2, and a is the distance from source to object, as shown in Fig. 7.19. The effect of this nonuniform carrier may be observed in the unwrapped data, see Fig. 7.20.

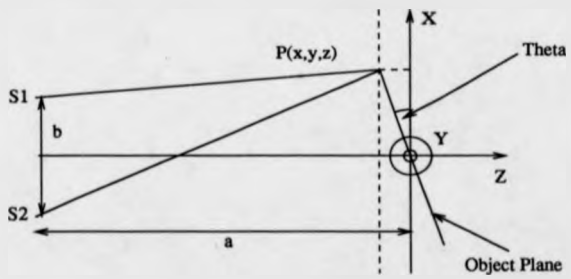


Figure 7.19: Schematic diagram for generation of carrier fringes at slight inclination.

The difference between the function represented by Eq. 7.19 and the ramp expected has been superposed on the unwrapped data. The carrier wavelength of the ramp expected was selected by the carrier fringe spacing at the centre of the interferogram. At the centre the unwrapped data is correct, however the carrier wavelength following Eq. 7.19 becomes stretched, nonlinearly, to the left of the interferogram and compressed to the right. This can be seen as the bowl shape in the unwrapped data of Fig. 7.20. The bowl shape s is proportional to Eq. 7.20.

$$s \propto \frac{m\lambda a}{b + m\lambda \tan\theta} - m\kappa \tag{7.20}$$

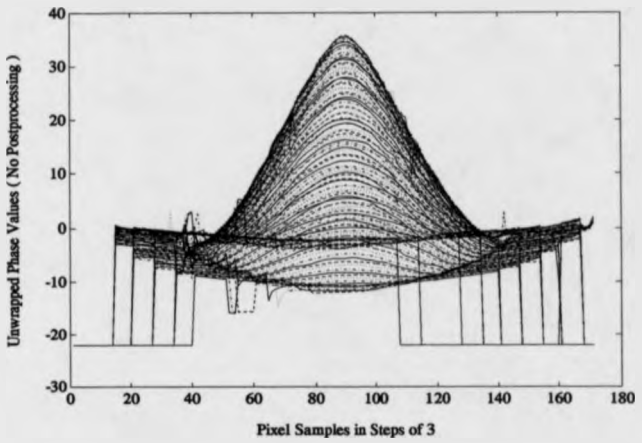


Figure 7.20: Side view of unwrapped numerical phase data.

where κ is the carrier fringe spacing at the centre of the object. Figure 7.21(a) shows the theoretical carrier fringe ramp and that of measured. The difference between these carrier fringe ramp is shown in Fig. 7.21(b) (like a bowl shape).

This superposed deformation may be eliminated from the unwrapped phase map by fitting a polynomial to the data representing the flat surround at the edge of the interferogram and subtracting this fitted curve across the unwrapped map.

Since the carrier fringes were not exactly vertical, a slight ramp running down the image, was also imposed. This has been corrected for in a similar manner to the above.

7.4.4 Results and Evaluation

Examples of deformation measurement for a centrally loaded disk and for an end loaded cantilever beam are given to illustrate the aspects of the complete measurement system discussed above. In the case of a centrally loaded disk, the interferogram obtained by making a double-exposure hologram of the disk before

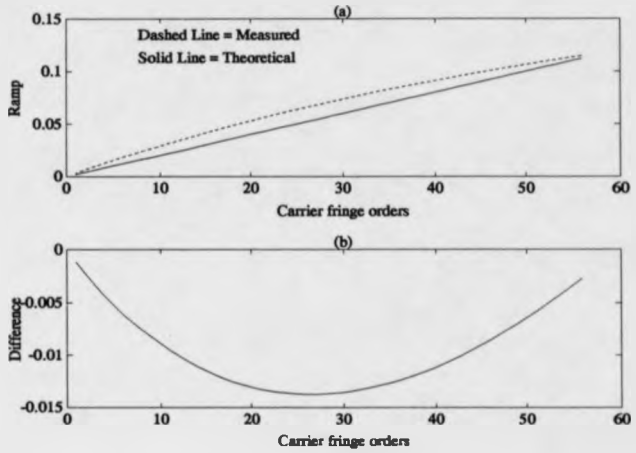


Figure 7.21: (a) The theoretical and measured carrier fringe ramp; (b) the difference between the theoretical and measured carrier fringe ramp.

and after applying the central loading is shown in Fig. 7.22. The interference fringes are basically a set of concentric rings.

Figure 7.17 shows the fringe pattern produced by both carrier and deformation present during the recording. The carrier has been created so that the fringe orders increase monotonically from left to right. The location of fringes in the undeformed carrier can be seen around the edge of the disk.

Figure 7.23 shows the wrapped phase map produced by the FFT process. Figure 7.24 shows the edge detection of the wrapped phase map, using a Sobel with 3×3 kernel and hysteresis thresholding. That is, the Sobel technique is employed with two threshold levels. Points above the lower threshold level are connected into the edge image only if they, or their neighbours, are connected to point(s) above the high threshold for edge detection. In the past this approach has, for example, been applied to the Canny [108] edge detector. Figure 7.25 shows a normalised grey scale plot of the unwrapped phase before it has been corrected for the nonlinearity of the carrier fringes.

Figure 7.22: Double-exposure holographic interferogram of a centrally loaded disk showing surface deformation (image size 512 × 512).

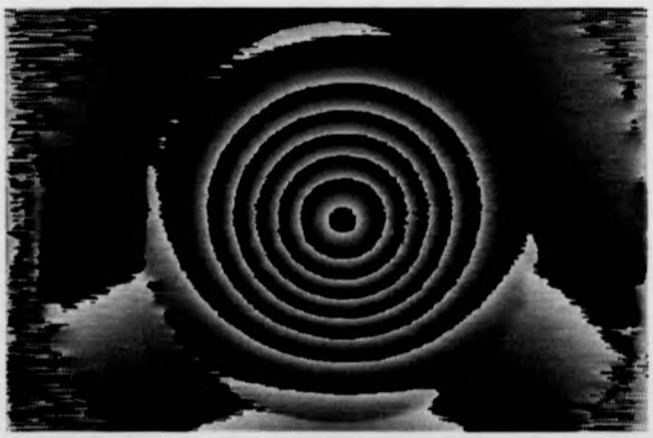


Figure 7.23: Wrapped phase map generated by FFT technique (image size 512 × 512).

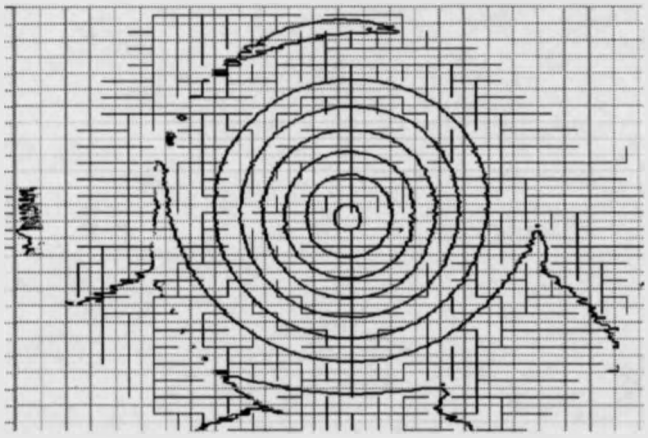


Figure 7.24: Sobel edge detection (with hysteresis thresholding) of wrapped phase map and tile level connection tree (image size 512 × 512).

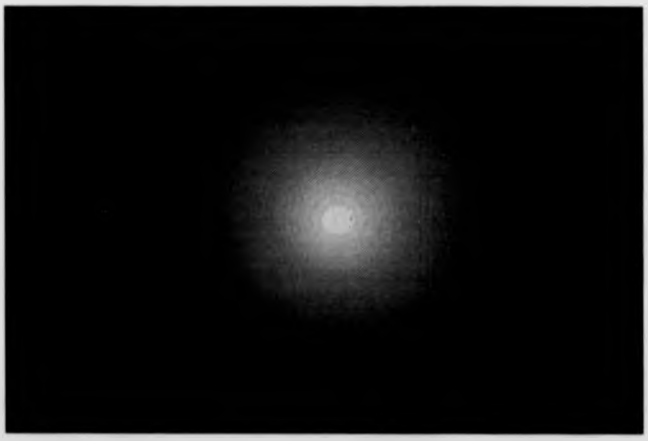


Figure 7.25: Normalised grey scale plot of deformation produced by unwrapping procedure, before correction for non linearity of carrier fringes (image size 512 × 512).

The input data for computer processing consists solely of the digitised image of the hologram. By applying the inverse FFT of $C(u,y)$ with respect to u, the term $c(x,y)$ in Eq. 3.61 is obtained. The phase $\phi(x,y)$ can be computed pointwise by Eq. 3.63. Figure 7.6 (a) shows the intensity distribution along raster 256 of the 512 in the image. The unwanted irradiance variations that are expressed by $a(x,y)$ and $b(x,y)$ in Eq. 3.60 can be seen in Fig. 7.6 (a), for example, between x-pixel indices 100 and 200. The end points of each raster are brought to zero by applying a Papoulis window across each raster in the spatial domain to simulate a periodic function, as shown in Fig. 7.6 (b). The Fourier transform of the data in Fig. 7.6 (b) is shown in Fig. 7.3 (a). Figure 7.3 (b) shows the sidelobe translated by u_0 in Fig. 7.3 (a) toward the origin to obtain $C(u,y)$.

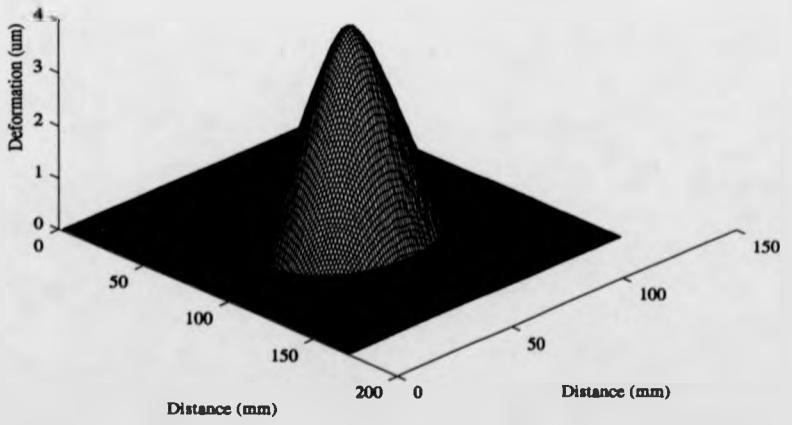


Figure 7.26: 3-D perspective plot of the out of plane displacement of the centrally loaded disk after phase unwrapping and correction.

The phase unwrapping algorithm counts up the agreements at tile boundaries

across the interferogram to give an estimate of the field area solved. In the case of the disk with a tile size of 44 pixels including an overlap of 4 pixels:

1. Total number of tile sides in complete frame = 624
2. Number of sides which computed between tiles = 480
3. Number of sides which did not appear to match = 20
4. Estimate of frame area solved = 76.6%

The 3-D perspective plot shown in Fig. 7.26 shows the shape of the deformed disk. A side view of the unwrapped phase is shown in Fig. 7.20, using the carrier frequency appropriate for the centre of each raster as stated above. The same view is shown in Fig. 7.27 after the phase values have been corrected for the non uniformity of the carrier. A contour map of the deformed disk is shown in Fig. 7.16, obtained from the again unwrapped and corrected numerical data.

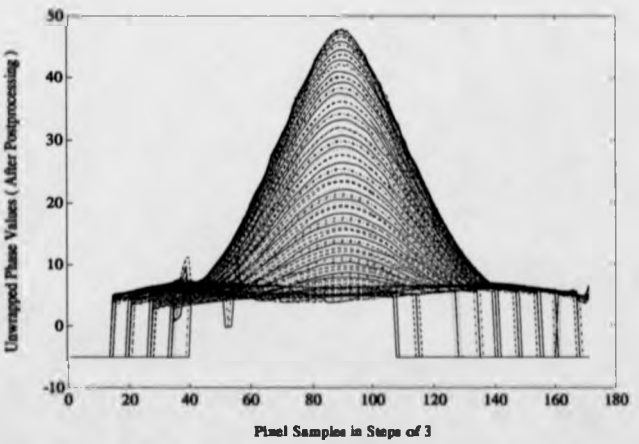


Figure 7.27: Side view of the unwrapped and corrected numerical phase data.

The experiment was arranged so that there existed a large separation between the disk and the fibre optic setup. In this case the theoretical out-of-plane

displacement component w_{th} is given by [43]

$$w_{th} = \frac{m\lambda}{2\cos\alpha} \tag{7.21}$$

where m is the fringe number, λ the wavelength and α the angle between the propagation vectors in the direction of illumination and normal of the object surface.

In fact, Eq. 7.21 is only valid for symmetrical cases where the illumination angle and the viewing angle are equally valued on either side of the surface normal. If this condition were not followed an error would be introduced. The rate of change of the out-of-plane displacement with respect to deviation of the illumination angle from its symmetrical position is given by

$$\frac{dw_{th}}{d\alpha} = \frac{m\ \lambda\ \tan\alpha}{2\ \cos\alpha} \tag{7.22}$$

which is assuming that the observation angle remains unchanged.

By moving a cursor over the unwrapped map, the displacement between two selected points could be calculated. Single precision floating point has been employed in all computations following data capture. As most processors now have associated floating point co-processors, this does not normally introduce a significant overhead on processing speed, and in some cases (notably with 286 PCs) is actually faster than using integer arithmetic. In this way coding is simplified (only one numerical type need be considered) and accuracy guaranteed. The intensity resolution of data capture devices is increasing, with the use of cooling to reduce dark current for example, and so it is wise to design analysis systems with a latent capacity in terms of intensity resolution.

A point on the disk edge and a second at the maximum were sampled. By applying Eq. 7.21 to the unwrapped phase values of these points, the relative deformation was found. The angle α was measured as 25 degree. The maximum displacement at the centre was calculated as 2.35μm in this way.

A cross section through the point of maximum deformation, obtained from

the phase unwrapping procedure, is shown in Fig. 7.28, along with a plot of the theoretical deformation curve. Figure 7.29 shows the plot of the discrepancy between the theoretical and that measured. The maximum discrepancy between the theoretical deformation and that measured is therefore 0.074μm. From this comparison it is seen that the larger deviation in the comparison occurs at the edges of the disk. This result may be partially explained by the imperfect clamping of the disk at its edges. This means that a larger than expected deformation might appear at the disk edges.

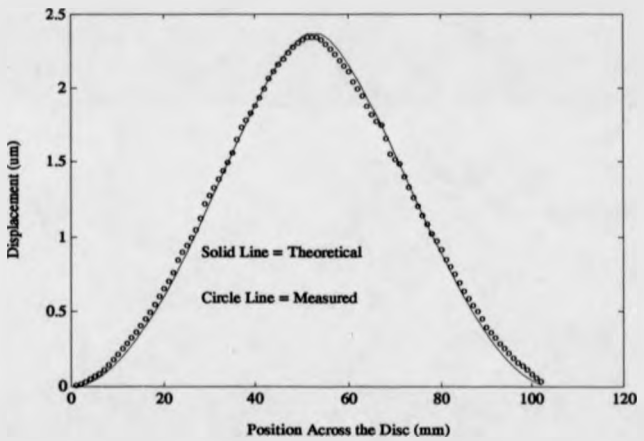


Figure 7.28: Comparison between the theoretical and measured deformation for cross section.

In the case of deformation measurement of cantilever beam, the arrangement of the apparatus and the experimental procedures was the same as in the case of aluminium disk. The experiments were carried out with a first exposure of the cantilever beam with an initial state without loading. Followed by a second exposure of the same duration as the first. In order to get better quality of interferogram and good fringe contrast, the experimental conditions were finally set at exposure times of 10 second, a ratio between object wave intensity and

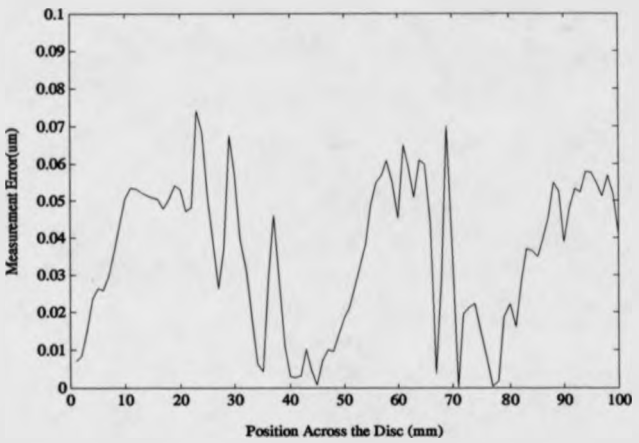


Figure 7.29: Discrepancy between the theoretical and measured deformation for cross section.

reference wave intensity from 1:2 to 1:4 and a fibre optics object illumination angle of 30 degree to the holographic plate perpendicular.

The holographic fringe pattern depicting the beam deflection is shown in Fig. 7.30. Figure 7.31 displays a double-exposure interferogram of the deflection of a cantilever beam with carrier fringes.

The carrier fringe technique not only benefits the automatic fringe analysis using FFT approach, but also removes the sign ambiguity of the displacement. The carrier fringes can be generated by translating the fibre optics illuminating objects between two exposures. The fringe orders increase monotonically from left to right. The location of fringes in the undeformed carrier can be seen on the frame of the cantilever beam in Fig. 7.31. Since the direction of translating the fibre optic beam is known, the sign of the displacement of cantilever beam can be determined by examining the fringe pattern obtained by a superposition of the fringe field under study on a known linear fringe field. In Fig. 7.31 the deformed fringe field plus carriers are bending to the right opposite to the direction of

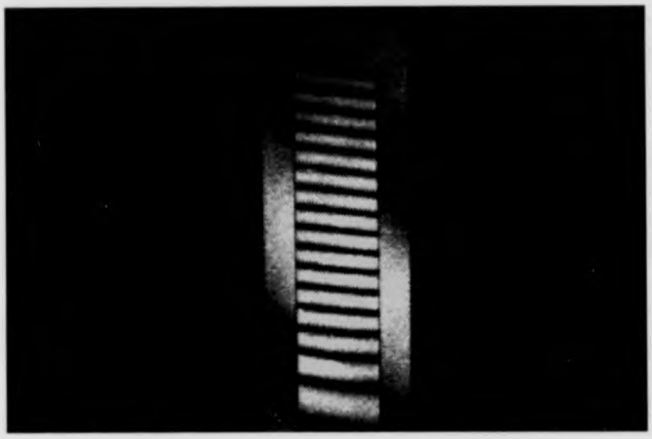


Figure 7.30: Reconstructed image of a deflected cantilever beam (image size 512 × 512).

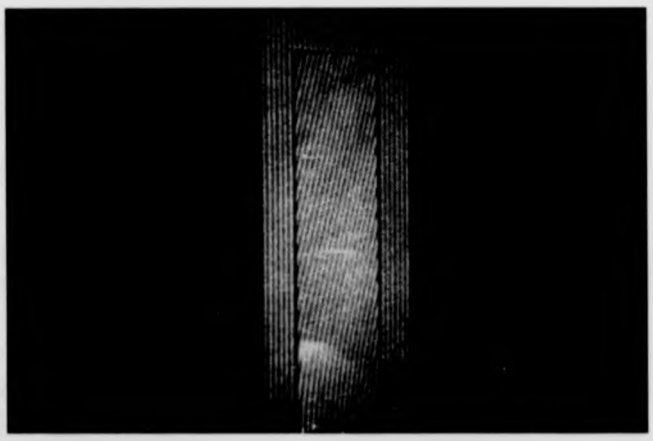


Figure 7.31: Hologram of deflection of a cantilever beam with carriers (image size 512 × 512).

translation of the fibre optic beam, therefore, the deflection of the beam is towards the observer.

Other methods for determining the sign of displacement have been reported in double-exposure holographic interferometry. One of them is based on shifting the strained object in a known direction between two exposures [174]. The additional shift is caused by turning either the object or the hologram. Plotkowski [41] proposed a carrier technique for unambiguous determination of sign of recorded displacements. The carrier fringes are generated by rotating one of the mirrors illuminating the objects between two exposures.

By using the FFT fringe analysis, the wrapped phase map can be generated. The unwrapped phase map can be produced with the aid of automatic phase unwrapping procedure. Figures 7.32 and 7.33 show the wrapped phase map and unwrapped phase map of deformed cantilever beam respectively.

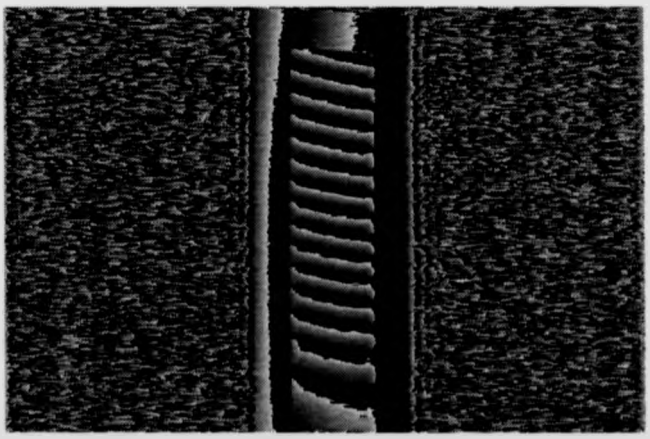


Figure 7.32: Wrapped phase map of deformed cantilever beam (image size 512 × 512).

Figure 7.34 is a 3-D mesh plot of cantilever beam. Figure 7.35 shows the side view of numerical phase data for deformed cantilever beam. Contour map of deformation of cantilever beam is shown in Fig. 7.36.

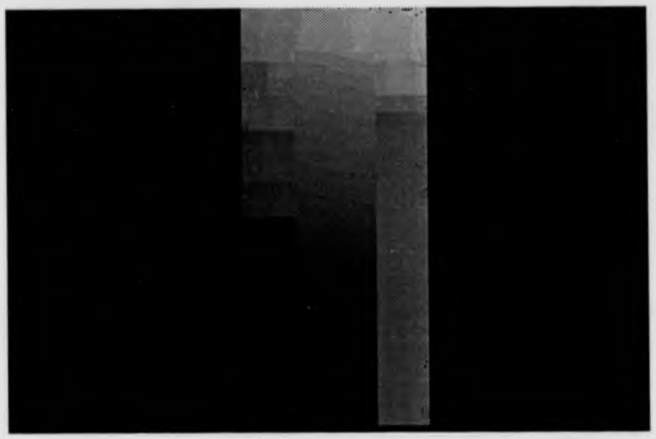


Figure 7.33: Unwrapped phase map of the deformed cantilever beam (image size 512 × 512).

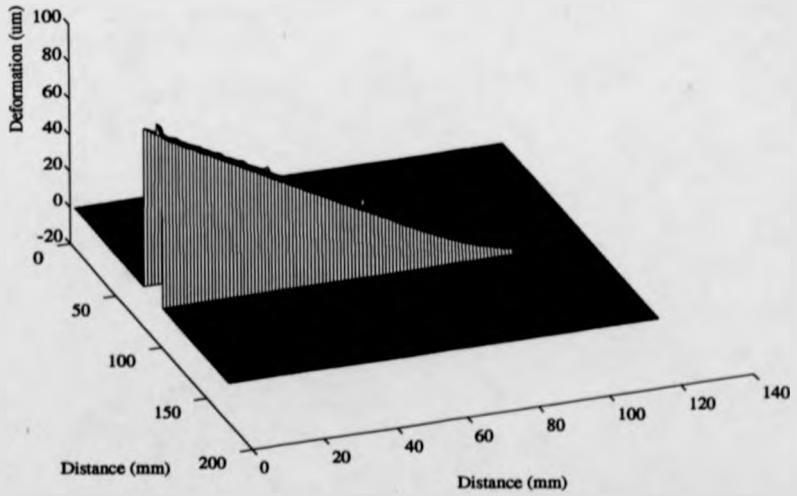


Figure 7.34: 3-D perspective plot of deformation of a cantilever beam.

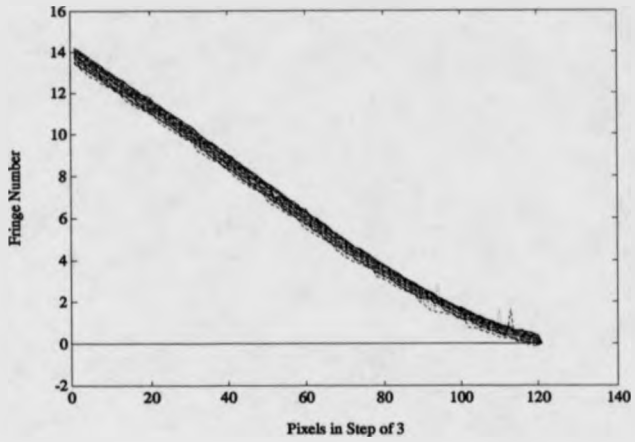


Figure 7.35: Side view of unwrapped phase data of deformed cantilever beam.

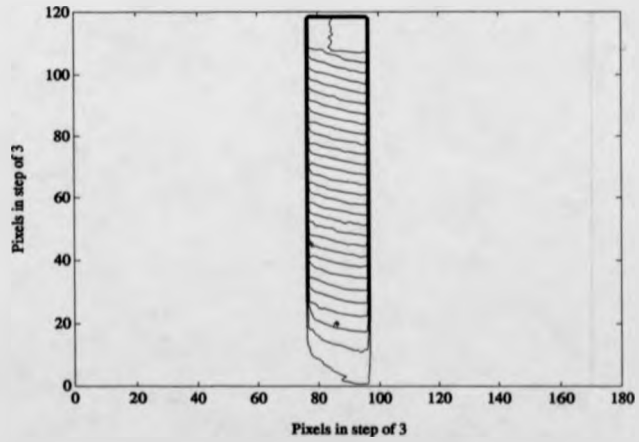


Figure 7.36: Contour map of deformation of a cantilever beam.

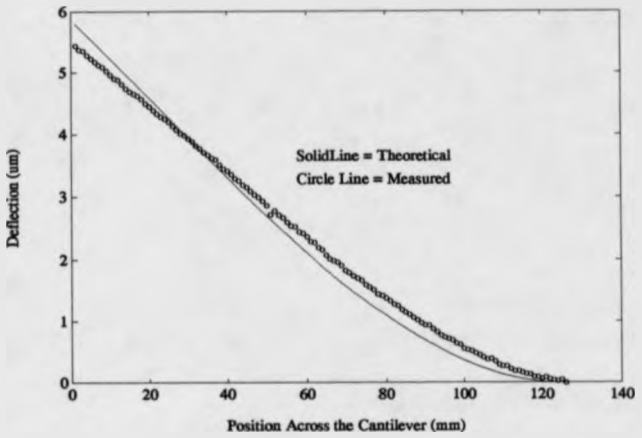


Figure 7.37: Comparison between the theoretical and measured deformation for cross section of a cantilever beam.

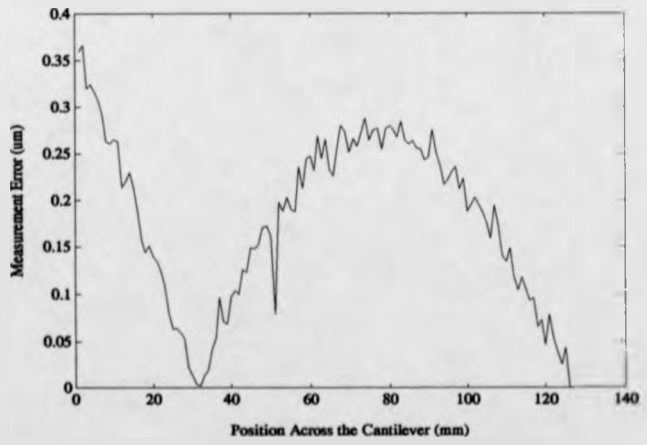


Figure 7.38: Measurement error between the theoretical and measured deformation for cross section of a cantilever beam.

A cross section through the point of maximum deflection of the beam is shown in Fig. 7.37, along with a plot of the theoretical deflection curve. It has been shown that the maximum deviation between the value of theoretical and that of measurement is 0.35 μm (see Fig. 7.38). The maximum discrepancy is at the free end of the cantilever. There are two major sources of error in this measurement. One is the error associated with the measuring gauge. Another is due to the tortion of the cantilever beam. This can be seen in Fig. 7.30. The fringe pattern is slightly inclined to the horizontal. This angle is caused by deflection and torsion when loading the cantilever. If there was no torsion, the fringe pattern would be horizontal, not inclined by a small angle to the horizontal.

7.5 Conclusion

In this chapter, the spatial carrier fringe technique has been applied to holographic interferometry. The spatial carrier across the object can be easily generated by moving fibre optics which illuminate the object. The use of fibre optics to carry the object and reference beams in holographic interferometry greatly facilitates the carrier fringe technique and the FFT fringe analysis method. The carrier fringe frequency can be set accurately by moving the optic fibres. The carrier fringes allow the sign ambiguity of the fringes to be removed by modulating them so that the fringe patterns can be analysed by the Fourier transform method. The non uniformity of the carrier fringes has been dealt with by considering the non uniformity as a superposed deformation.

It has been shown that the Fourier transform method of fringe analysis combined with the carrier fringe technique can be used to extract deformation information automatically from complex interferograms.

A new noise immune pixel level phase unwrapping strategy has been introduced. This strategy compliments the tile level of phase unwrapping to produce a robust system that may attack any fringe field. The image processing phase, including the FFTs, automatic window selection, generation of wrapped phase

map, phase unwrapping, and saving grey scale image and numeric ascii data (on 3 pixel grid) were all executed in software and required 2 minute 15 seconds of processing on a Sun Sparc station 1.

Chapter 8

Photoelasticity Stress Analysis Using Carrier Fringe and FFT Techniques

8.1 Introduction

The photoelastic stress analysis method is widely used in evaluating engineering structures. The technique requires a scale model made of birefringent materials and tested in a polariscope. It is particularly useful for analysis at the product design stage.

Photoelastic stress analysis is conducted using whole field stress or strain optical data. This method is sometimes in combination with a numerical procedure, like a least squares technique [175] or a hybrid method [176]. Compared to mechanical, electrical, and optical strain gauges, photoelastic stress analysis methods have the advantage of giving an overall picture of stress distribution, rather than point-by-point information. This property allows a relatively easy determination of complete stress fields.

In recent years, microcomputer and image processing techniques have been applied to photoelastic stress analysis. James *et al* [46] presented a review of

computerised optical fringe pattern analysis in photomechanics. They described a set of image processing tools used to analyze many of the optical fringe patterns in photomechanics. Video technology has been used to store the complete fringe pattern in a digitised form. Once stored, a microcomputer can be used to directly access any pixel in the image allowing the possibility of an automatic real-time analysis of the whole stress field. Various attempts have been made to develop the computer aided photoelastic stress analysis [177, 178, 179, 180, 181, 182, 183]. One of these is 'half-fringe photoelasticity' method [179, 183]. This method is based on a combination of classical photoelasticity and modern digital image analysis. It permits the materials and load to be chosen so that no more than one half of a fringe order appears in an area of interest. In this case, a relatively low birefringent response can be analyzed. Low loads can therefore be used to avoid local yielding and large deformation. However, photoelastic fringe data can also be processed quickly and accurately by appropriate computer software. Brown and Sullivan [182] developed a computer-aided holophotoelastic method. In this approach, isopachic, isochromatic, and isoclinic fringes are processed automatically by computer. The isopachic data is obtained from a phase-stepping procedure applied to a real time holographic absolute retardation fringe system. In this case, three phase-stepped images are needed to produce a relevant phase value. The isochromatic results are acquired using half-fringe photoelasticity.

This chapter presents a computer aided photoelastic stress analysis method using an added carrier fringe and FFT analysis techniques. The FFT approach is simple and direct. Data can be collected within a single stored video frame. Compared with the phase-stepping method, where three or four images with mutual phase shifts are recorded and evaluated, only one fringe pattern is needed to extract information from the fringe data. Optically stored experimental stress data obtained using a diametrically loaded disk and ring has been used for the validation of the method. The principle of the carrier fringe and FFT techniques are discussed. The experimental set-up and data acquisition software design are described.

The system used consists of a computer controlled polariscope, a CCD camera and a monitor. The photoelastic fringes are captured by a CCD camera and analyzed by a Sun Work-Station. The analysis algorithms are based on the carrier fringe and FFT techniques. In the analysis, the carrier fringes are generated using a birefringent wedge. The carriers are superimposed on a stress fringe field. This modulated fringe field is processed and the results of the three dimensional distribution of principal stress difference are given. It has been found that the system developed is of importance to automatic photoelastic stress analysis.

8.2 Basic Concept of Photoelastic Stress Analysis

8.2.1 Birefringence and Photoelasticity

Photoelastic models are constructed from a special class of transparent materials which exhibit a property known as birefringence, i.e. they have the ability to split an incident plane polarised ray into two component rays. This property is only exhibited when the material is under stress, and the direction of the component rays coincides with the direction of the principal stresses. The speeds of the rays are proportional to the magnitudes of the respective stresses in each direction, so that the subsequent interference of two polarised rays produces fringe patterns when combined. These fringes are ordered according to the darkness and brightness cycles that take place at a point in the model. The fringe patterns provide immediate qualitative information about the general distribution of stress, positions of stress concentrations and areas of low stress. By means of the stress-optic law, the stress and strain of the model under certain loading condition can be determined. On the basis of these results, designs may be modified to reduce or disperse concentrations of stress or to remove excess material from areas of low stress.

The most commonly used materials in photoelastic stress analysis are given

Table 8.1: Table of birefringent materials used in photoelastic stress analysis

Material	**Tensile Strength** (MN/m^2)	**Modulus of Elasticity** (MN/m^2)	**Fringe Value** (at 0.5893 μm) (KN/m.fr)	**Poisson's Ratio**
Araldite CT-200/HT-901	70-90	2700-3400	9.6-10.5	0.36
Araldite CT-200/HT-907	70-90	2700-3400	14.7-15.8	0.34
Columbia Resin CR-39	48	1700-2400	14.9-17.5	0.44
Photoflex	-	0.7-4.1	0.14-0.17	0.48

in Table 8.1 [184]. The materials used in the experiments are Araldite CT-200 Epoxy Resin and Photoflex Polyurethane Rubber. CT-200 Epoxy Resin exhibits good mechanical properties and strength, with little creep or time-edge effect. Photoflex Rubber is used as photoelastic materials for certain dynamic applications. The material has a very high stress-optic sensitivity.

The stress analysis in the photoelastic technique is usually done by using information from isoclinic and isochromatic as well as isopachic in holo-photoelasticity. A major feature of the photoelastic technique is that it allows one to effectively "look into" the component and pin-point flaws or weakness in design. Stress concentrations are immediately visible, stress values around the edge or boundary of the model are easily obtained.

8.2.2 Isoclinic Fringes with Plane-Polariscope

Figure 8.1 shows the optical arrangement for the formation of isoclinic fringes. Light emerging from the polariser will be polarised along the optical axis of the polariser. When a model of an engineering component constructed from a birefringent materials is stressed, the incident plane polarised light will be split into two separate components, one of which vibrates in the direction of the algebraic maximum principal stress σ_1 and the other in the direction of algebraic

minimum principal stress σ_2. This is because the model became birefringent or double refracting under strain. Each component wave will pass through the thickness of the model at a velocity which will depend upon the magnitude of the particular principal stress along whose axis each component passes and upon the optical properties of the model itself. If σ_1 has a larger magnitude than σ_2 , then there will be a phase difference in the emerging component waves which can used as a measurement of the magnitude of $\sigma_1 - \sigma_2$. When these two component waves arrive at the analyser position, only their horizontal components will pass through and these will combine to produce interference fringes as shown in Fig. 8.2. The fringes are termed as isoclinic. The definition of an isoclinic is that it is the locus of points at which there is a constant inclination of principal stress directions. It should be noted that an isoclinic fringe is always black, whether viewed in monochromatic or white light. Isoclinic fringe occurs in a model when in plane polariscope, but not in a circular polariscope.

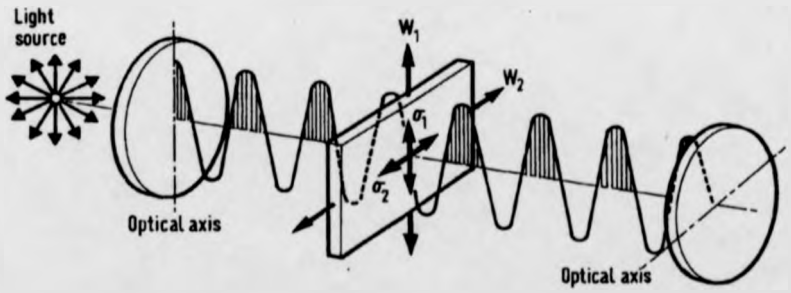


Figure 8.1: A typical arrangement for plane polariscope (Sharples 1981).

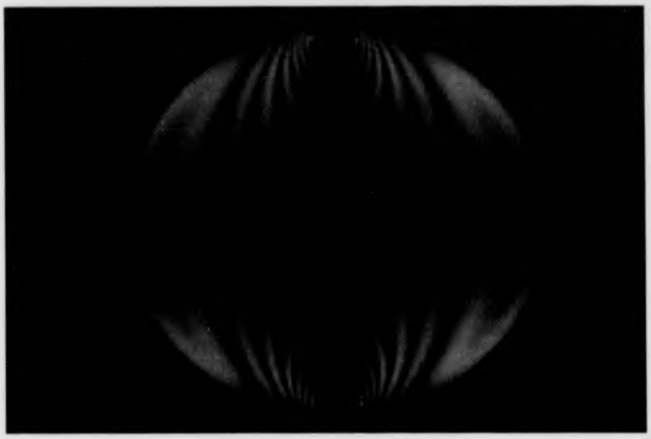


Figure 8.2: Isoclinic fringe pattern generated by a plane polariscope (image size 512 × 512).

8.2.3 Isochromatic Fringes with Circular Polariscope

When a stressed birefringent model is observed in a plane polariscope, both isoclinic and isochromatic fringes are present. If the elements of a plane polariscope are rotated relative to the birefringent model, whilst remaining crossed, the isoclinic fringes moves across the model and the isochromatics remain stationary. It would be possible to devise a mechanical system to rotate the polarising elements together at high speed, so that the isoclinic fringes moved rapidly and became blurred, and to the observer, virtually eliminated. The isoclinics would thus be suppressed as a result of the continual change in the plane of polarisation of the light passing through the model. The continual rotation of the polarising axes can be achieved by optical rather than mechanical means. The necessary optical arrangement is called a circular polariscope.

The circular polariscope differs from that in Fig. 8.1 by having two quarter-wave plates ($\lambda/4$ plates) placed between polariser and analyser. Each quater-wave plates has two polarising axes which at right angles and are called the fast and the slow axis. Light vibrating in a plane parallel to the slow axis is retarded by a

quater of a wave length relative to the light vibrating in the perpendicular plate. The fast and slow axis of these retarders may be crossed or parallel but at 45^0 with the polariser to convert the light from plane to circularly polarised and back to plane polarised as shown in Fig. 8.3. The relative retardation produced by the first plate is thus restored by the second. Since the model is immediately after the first quarter-wave plate, the light incident upon the model is resultant of the two perpendicular waves, 90^0 out of phase, emerging from the quarter-wave plate. The resultant polarised wave is of constant amplitudes nut varies in direction, the tip of the vector representing the vibration describing a circular path. The light entering the model is said to be circularly polarised, and is indicated in Fig. 8.3. The effect of this circularly polarised light upon the fringe pattern in the model is similar to that of the mechanical system, so that the isoclinic fringes are eliminated, but the isochromatic fringes are unchanged.

The isochromatic fringes are generally of greater importance, as they relate to stress magnitudes, and it is often convenient when analysing them, to eliminate the isoclinic fringes. Isochromatic fringe patterns contain information that is used to calculate the differences of principal stress. The fundamental formula for obtaining quantitative stress determinations from isochromatic fringe patterns is as follows

$$\sigma_1 - \sigma_2 = \frac{nf}{t} \tag{8.1}$$

where σ_1 is the algebraic maximum principal stress, σ_2 the algebraic minimum principal stress, n the fringe order, f the material fringe value, and t the model thickness.

Figures 8.4 and 8.5 show the isochromatic fringe pattern of dark (two $\lambda/4$ plates crossed) and light (two $\lambda/4$ plates paralleled) fields produced by a circular polariscope.

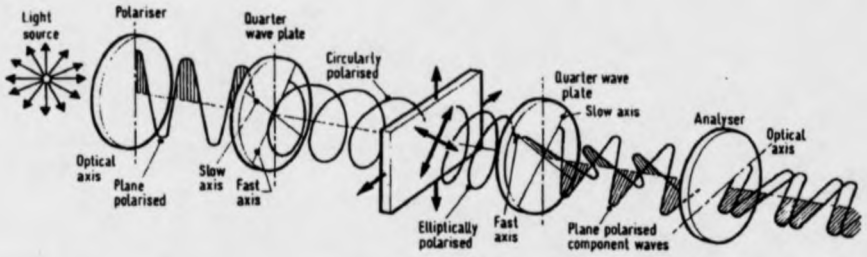


Figure 8.3: A typical arrangement for a circular polariscope (Sharples 1981).

Figure 8.4: Isochromatic fringe pattern of dark field obtained using circular polariscope (image size 512 × 512).

Figure 8.5: Isochromatic fringe pattern of light field obtained using circular polariscope (image size 512 × 512).

8.2.4 Isopachic Fringes with Holographic Photoelasticity

Holographic interferometry has been applied to photoelasticity by several authors [185, 186, 187, 188, 189]. Sanford [185] presented a survey of the role of holography in the field of photoelasticity. In holographic interferometry, a single-exposure provides all the same information that can be obtained with conventional photoelastic procedures, and double-exposure is often used to obtain isopachic fringe patterns. Ajovalasit *et al* [187] also proposed a real-time method of holographic interferometry to determine the absolute retardation fringes in photoelastic analysis of plane models. The proposed technique requires only a single hologram, which several holograms are required by double-exposure holographic interferometry. However, the disadvantage of this technique is the increased complexity and stability requirements. Holloway and Johnson [189] described dynamic holography experiments using a pulsed ruby laser. Recently, Qin *et al* [3] proposed a new method of resolving transient stresses in dynamic holophotoelasticity. In this method, a double pulsed ruby laser and a terbium glass Faraday rotator is employed to resolve the transient isochromatic and isopachic

patterns.

Figure 8.6 shows the geometry of the holographic polariscope and the orientation of the elements in the object and reference beams [187]. The laser produces plane polarised light in vertical direction **y**. The quarter-wave plates are arranged with their principal axes at $\pi/4$ with the horizontal **x** and vertical **y** axes. The light emerging from the quarter-wave plates is circularly polarised. The polarisers with their axes at an angle of θ to the **x** axes, transform the circular light into plane polarised light. It is necessary to insert the pinhole and positioning device for spatial filtering of both the reference and object beam. As a spatial filter, the pinhole offers an additional advantage: it serves as an optical isolator between the laser and the remainder of the optical system.

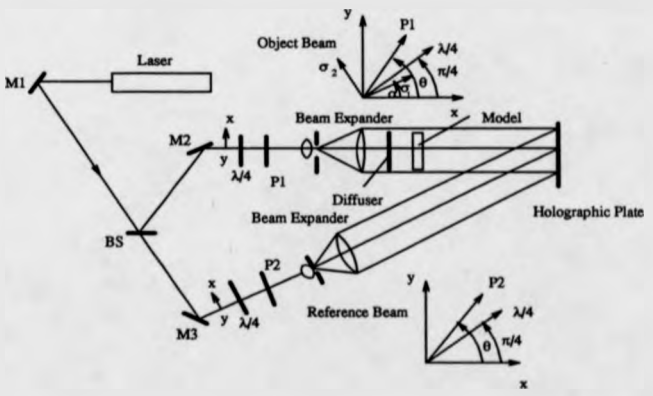


Figure 8.6: Schematic presentation of a holographic polariscope (Ajovlast 1970).

Compared with conventional photoelasticity, holographic photoelasticity provides isopachic fringe patterns related to the absolute retardation of the polarised light. It also produces isochromatic and isoclinic fringe patterns using single-exposure holograms. The holographic polariscope acts both a conventional polariscope and a photoelastic interferometer. In addition, the unique property of the reconstructed real image is to provide greatly enlarged images without the aid of an auxiliary lens [185]. The disadvantage is the high cost of establishing

a holography laboratory: the cost of equipment is greater than the cost of a conventionally available photoelastic polariscope.

8.3 Carrier Fringe Analysis Using Jones Matrix in Photoelasticity

8.3.1 Principle of Photocarrier Technique in Polariscope

The carrier fringe technique can be applied to photoelastic stress analysis. Carrier fringes are generated by inserting a crystalline quartz wedge or a temporary birefringent frozen wedge into a circular polariscope, as shown in Fig. 8.7.

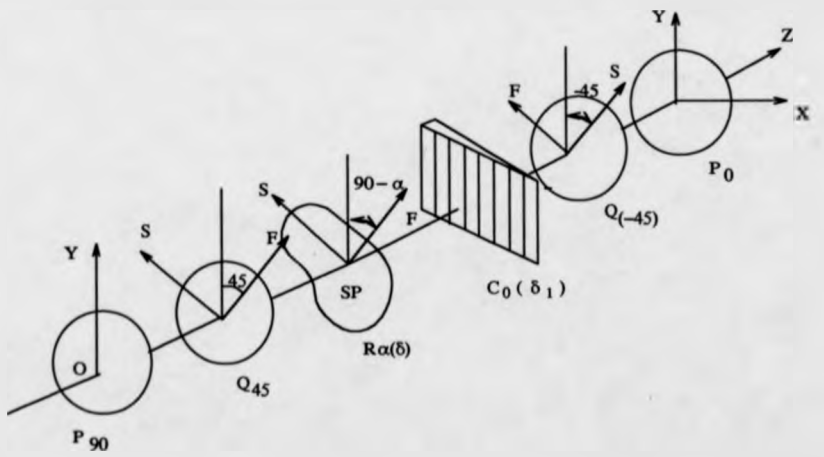


Figure 8.7: Experimental arrangement for a circular polariscope with birefringent wedge.

A circular polariscope with a carrier wedge and sample consists of the following optical elements.

1. A linear polariser $\vec{P}_{90}$, whose optical axis is vertical.

2. A quarter-wave plate $\vec{Q}_{45}$, whose fast axis is at an angle of 45^0 to the x axis.

3. A birefringent material with retardation δ when stressed, whose fast axis is at an angle of α to the x axis.

4. A photocarrier $\vec{C}_0$ made of quartz wedge which generates the equidistant straight fringes in the polariscope, whose fast axis is along the x axis.

5. A quarter-wave plate $\vec{Q}_{-45}$ whose fast axis is at an angle of -45^0 for a dark field circular polariscope, or $+45^0$ for the light field.

6. A linear polariser $\vec{P}_0$, whose optical axis is horizontal.

The incoming light after passing through the vertical polariser can be expressed by

$$\vec{P}_{90} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{8.2}$$

The Jones matrices [190] for the quarter-wave plates, whose fast axis is at an angle of $\pm 45^0$ to the x axis, is obtained by putting $\delta = 90^0$, and $\alpha = \pm 45^0$ in the general form of the expression of the retardation plate.

$$\vec{Q}_{45} = \frac{1}{2} \begin{bmatrix} i+1 & i-1 \\ i-1 & i+1 \end{bmatrix} = \frac{i+1}{2} \begin{bmatrix} 1 & i \\ i & 1 \end{bmatrix} \tag{8.3}$$

$$\vec{Q}_{-45} = \frac{1}{2} \begin{bmatrix} i+1 & -(i-1) \\ -(i-1) & i+1 \end{bmatrix} = \frac{i+1}{2} \begin{bmatrix} 1 & -i \\ -i & 1 \end{bmatrix} \tag{8.4}$$

The Jones matrix of the birefringent plate $R_\alpha(\delta)$ with retardation δ, whose fast axis is at an angle of α to x axis is given by

$$\vec{R}_\alpha(\delta) = \begin{bmatrix} e^{i\delta}\cos^2\alpha + \sin^2\alpha & (e^{i\delta}-1)\sin\alpha\cos\alpha \\ (e^{i\delta}-1)\sin\alpha\cos\alpha & e^{i\delta}\sin^2\alpha + \cos^2\alpha \end{bmatrix} \tag{8.5}$$

The Jones matrix of the carrier wedge $\vec{C}_\theta(\delta_1)$ with retardation δ_1, whose fast axis is along the **x** axis is given by

$$\vec{C}_0(\delta_1) = \begin{bmatrix} e^{i\delta_1} & 0 \\ 0 & 1 \end{bmatrix} \tag{8.6}$$

The Jones matrix of the analyzer, whose optical axis is horizontal, is given by

$$\vec{P}_0 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \tag{8.7}$$

According to the Jones calculus, the Jones vector $\vec{a}'$ of the light beam that emerges from the analyzer is

$$\vec{a}' = \vec{P}_0\ \vec{Q}_{-45}\ \vec{C}_0(\delta_1)\ \vec{R}_\alpha(\delta)\ \vec{Q}_{45}\ \vec{P}_{90} \tag{8.8}$$

By substituting Eqs. 8.2, 8.3, 8.4, 8.5, 8.6 and 8.7 into 8.8, for the dark-field polariscope, we have

$$\begin{aligned} \vec{a} = & \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \frac{i+1}{2} \begin{bmatrix} 1 & -i \\ -i & 1 \end{bmatrix} \begin{bmatrix} e^{i\delta_1} & 0 \\ 0 & 1 \end{bmatrix} \\ & \begin{bmatrix} e^{i\delta}\cos^2\alpha + \sin^2\alpha & (e^{i\delta}-1)\sin\alpha\cos\alpha \\ (e^{i\delta}-1)\sin\alpha\cos\alpha & e^{i\delta}\sin^2\alpha + \cos^2\alpha \end{bmatrix} \\ & \frac{i+1}{2} \begin{bmatrix} 1 & i \\ i & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \end{aligned} \tag{8.9}$$

The intensity of the fringe pattern in the observing plane is given by (see Appendix I)

$$I = \sin^2\left(\frac{\delta-\delta_1}{2}\right)\sin^2(\alpha) + \sin^2\left(\frac{\delta+\delta_1}{2}\right)\cos^2(\alpha) \tag{8.10}$$

When $\alpha = 0$, i.e. the fast axis of the model disk is along the **x** axis, and

Eq. 8.10 becomes,

$$I = \sin^2 (\frac{\delta + \delta_1}{2}) = 1 - \cos(\delta + \delta_1) \tag{8.11}$$

The Eq. 8.11 shows that the fringes are the type of cosinoidal, the phase of the stress fringes is modulated by the phase of the carriers. After taking FFT fringe analysis, the complicated phase of stress fringe field can then be demodulated. This will be seen in the following sections.

8.4 Hardware and Software Considerations

Over the years, several computer aided polariscope systems have been developed by the following authors, Muller & Saackel[177], Seguchi *et al* [178], Voloshin & Burger [179], Hecker & Morche [180], Gillies [181], Brown & Sullivan [182]. In these methods, direct exploration of digital computers was introduced to photoelastic fringe analysis. The techniques are based on the ability of an image analysis system to rapidly acquire information on light intensity over a whole field, and to store it digitally in a frame store. The information in the digital frame store can be accessed by computer and processed to produce the stress field.

Voloshin *et al* [179] and Wang *et al* [183] proposed "half-fringe photoelasticity (HFP)" which is based on classical photoelasticity and modern image analysis. This method operates effectively with less than one-half wavelength of relative retardation.

Brown *et al* [182] presented a computer-aided holo-photoelastic method. In this method, the isopachic data are obtained from a phase-stepping procedure and the isochromatic results are acquired using a half-fringe photoelasticity technique. The direction of principal stress is obtained from a series of stepped isoclinic images. Hecker *et al* [180] also reported computer-aided measurement of relative retardation in plane photoelasticity. The idea of phase-shifting was applied for digital measurement of traditional orders of isochromatic fringes.

This section presents a new experiment for whole field stress analysis based on the photocarrier fringe and FFT techniques. The isochromatic fringes produced by a circular polariscope are modulated by photocarrier fringes. The photocarrier fringes are generated by a crystalline quartz wedge. The fringe patterns are captured by a CCD camera, and automatically processed by the Fourier transform technique to generate the wrapped phase map of the stress fringe pattern. The continuous phase map of stress fringes is produced by the phase-unwrapping procedure used in the last chapter.

8.4.1 Description of the System

Figure 8.8 shows the automatic whole field stress analysis system. The system includes hardware, interfacing, software control and processing. A side view of the polariscope is shown in Fig. 8.9. A block of diagram of the whole system is shown in Fig. 8.10.

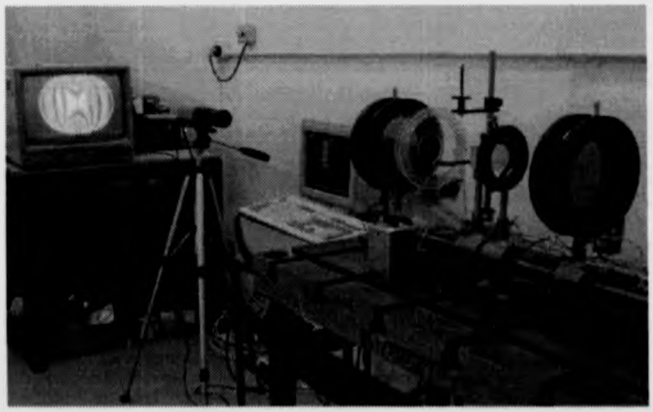


Figure 8.8: The automatic stress analysis system (image size 666 × 446).

The system consists of a traditional polariscope setup consisting of a light source (a sodium light), a polariser plus quarter-wave plate, a diffuser, a model, a quarter-wave plate plus analyser, and a means of recording the image. Instead

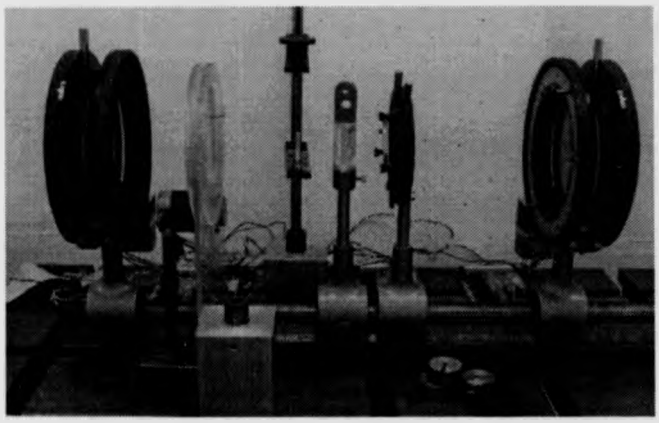


Figure 8.9: A side view of the polariscope (image size 643 × 437).

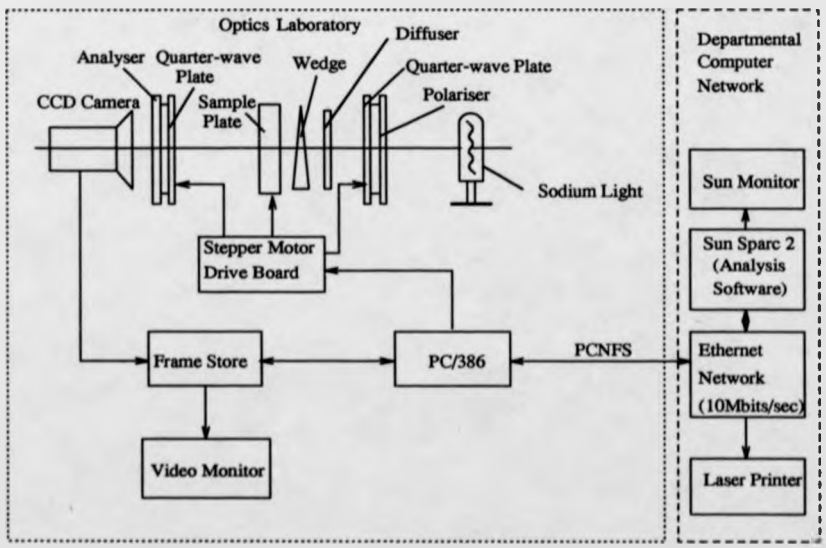


Figure 8.10: Diagram of the automatic stress analysis system.

of using the classical camera or photodiodes, a CCD camera is used to capture the whole field stress pattern. The image is captured by a CCD camera and digitised into 512 × 512 × 8 pixels by a digital frame board (PIP board). Every pixel has a resolution of 8 bits, i.e. 256 grey levels. Three stepper motors are employed to drive the two quarter-wave plates and a sample plate. A microcomputer (PC/386) is used to control the stepper motors, image capture and display as well as to process the image. A monitor is used to visualise the information. A PC Ethernet card is installed in the microcomputer so that the digital image can be transferred to the departmental computer network and analysed by a Sun Workstation in 2 minutes and a half.

8.4.2 Stepper Motor Control and Interface

Three RS 332-947 stepper motors are used to drive two quarter-wave plates and a loading plate respectively, as shown in Fig. 8.11.

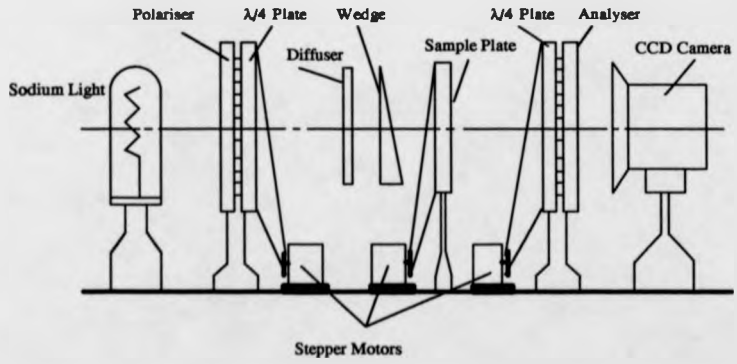


Figure 8.11: Schematic diagram of stepper controlled polariscope.

Each stepper motor has four 12v dc windings (coil) and a permanent magnet rotor construction. The stepper motor is designed for unipolar drive, therefore, the motors are easily interfaced for simple and relatively low power electronics, thus providing an economical means of motion and speed control.

Three unipolar stepper motor drive boards are employed to interface between the stepper motors and the computer. The features of the drive boards are: (1) directly compatible with any of the RS stepper motors; (2) eurocard system compatible; (3) full step and half step drive modes; (4) external control inputs are CMOS and open collector TTL logic compatible. Figure 8.12 shows the unipolar stepper motor drive board connections.

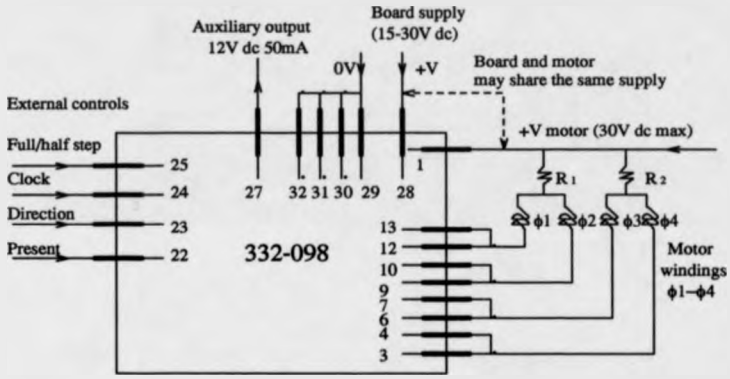


Figure 8.12: Unipolar stepper motor drive board connections.

There are four switching logic control inputs for the stepper motor drive board. (1) Full/half step input, level "1" is the full step and level "0" is half step, the voltage level is applied to pin 25 of the board. (2) Direction input, the direction of rotation of the motor is determined by the voltage level applied to pin 23 of the board, level "1" gives a clockwise motor direction and level "0" gives anti-clockwise motor rotation. (3) Trigger input, the stepping rate and ultimate angular position of the motor drive shaft are determined by the repetition rate and the number of the pulses applied to pin 22 of the board. (4) Clock step input, the clock frequency lie between 1Hz and 25kHz, 10μs minimum pulse with negative edge trigged.

The interface between the stepper motor drive board and the computer is performed via an 8255 PPI (Programmable Peripheral Interface), which is held on

an IBM microcomputer interface board. This board is connected to the IBM via the integral 62 way edge connector. The interface board also holds an 8253 PIT (Programmable Interval Timer). The 8255 is a general purpose programmable peripheral interface. The device has 24 I/O pins arranged as 3 ports A, B and C. Port A consists of one 8-bit data output latch/buffer and data input buffer, port B consists of one 8-bit data input/output latch/buffer and one data input buffer, port C consists of one 8-bit data output latch/buffer and one input buffer, this can split under mode control into two 4-bit ports.

The external interface is connected to the IBM internal interface using a 40 way ribbon cable and IDC connector. The connections between port A and stepper motors are as follows, PA0 and PA1 are for pulse generation and direction of steeper motor 1, PA2 and PA3 are for pulse generation and direction of stepper motor 2, PA4 and PA5 are for pulse generation and direction of stepper motor 3 respectively, PA6 and PA7 are not used in this case. The address of port A is 432. The mode control address is 435.

8.4.3 Birefringent Wedge Design

The carrier fringes are produced by a birefringent wedge. In the experiments, two birefringent wedge are designed for generating carrier fringes in the polariscope. The first one is the frozen stress wedge. The material used to design the wedge is Araldite CT-200 Epoxy Resin. This resin is the most widely used photoelastic material in applications. In the cast state, it exhibits good mechanical properties and strength, with little creep or time-edge effect. At freezing point, the amount of creep is negligible. The wedge is made in two stages. First, the sample plate is loaded in an oven, in order to produce straight fringes in the plate. The maximum fringe order is determined by the ultimate tensile strength of the material. The temperature inside the oven is reduced after several hours of loading, thus, the fringes are frozen in the sample plate. Second, a wedge is cut from a frozen stress plate, the fringe spacing is determined by the angle of the wedge.

The ultimate tensile strength of CT-200 epoxy at the stress freezing temperature is 2.07 $MN.m^{-2}$, material fringe value f' is 0.28 $KN.m^{-1}.fr^{-1}$. The fringe order which is possible to freeze into the material according to the flowing equation

$$\sigma = \frac{nf'}{h} \tag{8.12}$$

where σ is the principal stress difference, n the fringe order, h the thickness of material. The maximum fringe order can be calculated by

$$n_{max} = \frac{\sigma t}{f'} = 44.25 \tag{8.13}$$

The length of wedge (L) can be machined from a height (H) of 6 mm plate of epoxy material and the wedge will taper down to 0 thickness at one end, as shown in Fig. 8.13. The fringe spacing (s) is 0.5 mm, therefore, the angle of wedge θ is

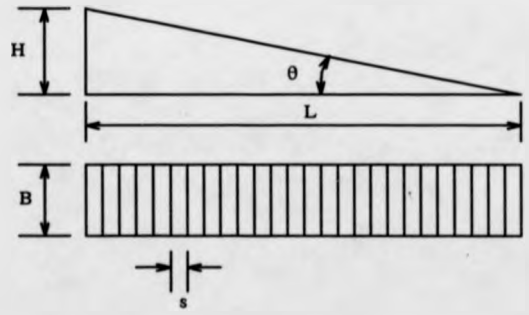


Figure 8.13: Schematic diagram of a frozen stress wedge.

$$\theta = \arctan(\frac{6}{n\ 0.5}) = 15^0 \tag{8.14}$$

It must be realised that the above calculations are based upon a nominal value at an ultimate tensile strength of 300 $lbf.in^{-2}$ and that no factor of safety has been included. Only by actual stress freezing can the practicality of using an

ultimate tensile strength of 300 $lbf.in^{-2}$ be proved. An alternative could be the use of two wedges with 20 fringes in each as illustrated in Fig. 8.14

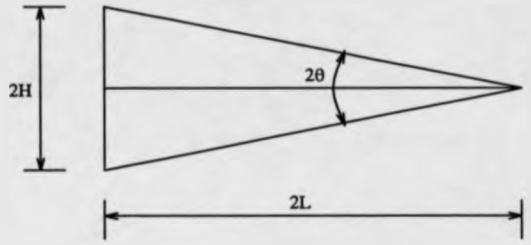


Figure 8.14: An alternative to frozen stress wedge made from two separate wedge.

The second wedge in the experiment is designed by using a birefringent crystalline quartz. Many crystalline substances are optically anisotropic, i.e. their optical properties are not the same in all direction. A material of this sort, which displays two different indices of refraction, is said to be birefringent. A sketch of the wedge is shown in Fig. 8.15.

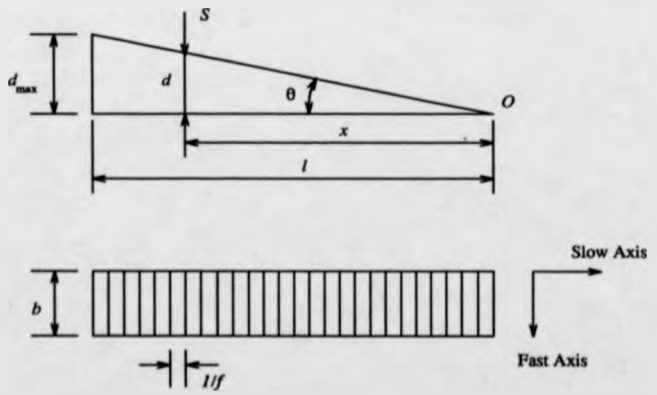


Figure 8.15: Sketch of crystalline quartz wedge.

Suppose that the fast axis of the crystal is normal to both the front and back surfaces. The light will be resolved into two rays travelling with different

velocities in the wedge. Consider a section S, parallel to the wedge edges and perpendicular to the boundary planes. If its thickness is d, and δ is the relevant phase difference introduced by the wedge at this section, then,

$$\delta = \frac{2\pi}{\lambda} d\,(n_e - n_o) \tag{8.15}$$

With reference to Fig. 8.15, $d = x \tan\theta$, where θ is the wedge angle, x the distance of S from a suitable origin O, the d can be calculated as

$$d = x\ \tan\theta = \frac{N\ \lambda}{n_e - n_o} \tag{8.16}$$

where

$$\tan\theta = \frac{N\ \lambda}{(n_e - n_o)x} = \frac{f\ \lambda}{n_e - n_o} \tag{8.17}$$

where f is the carrier frequency (line/mm), N the fringe values. For crystalline quartz, $n_e = 1.5534, n_o = 1.5443$. The required frequency (f) is 3 lines/mm, λ the wavelength (0.5893μm), the wedge angle θ is

$$\theta = \arctan\frac{f\ \lambda}{n_e - n_o} = 10^0\ 59' \tag{8.18}$$

When the wedge is placed between two quarter-plates in a dark field circular polariscope as shown in Fig. 8.16, a series of dark equi-spaced fringes are produced with illumination of monochromatic light. Figure 8.17 shows the carrier fringe patterns generated by crystalline quartz wedge in a dark field circular polariscope. The optical transforms that takes place in a circular polariscope can be described as follows:

- The polariser transforms the monochromatic light into plane polarised light.
- The first quarter-wave plate transforms the plane polarised light into circularly polarised light.
- The circularly polarised light is resolved into two components along the

principal axes of the birefringent wedge. The doubly refracting wedge will produce an angular phase difference as the two beams travel through the wedge. The phase difference introduced by the wedge, and hence the amount of interference produced is determined by the angle of the wedge and refractive index of the materials.

- The second quarter-wave plate restores the original plane polarised light or rotates it through 90^0, depending on whether it is crossed or parallel to the first quarter-wave plate.

- The analyser blocks or lets through the plane polarised light according to the desired background. The emerging ordinary and extraordinary rays are combined to produce the interference fringe patterns.

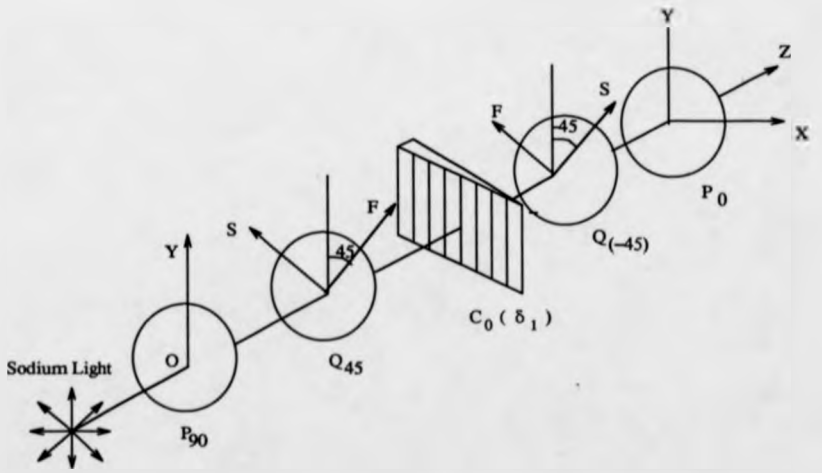


Figure 8.16: Schematic diagram of optical transformations with a birefringent wedge in a dark field circular polariscope.

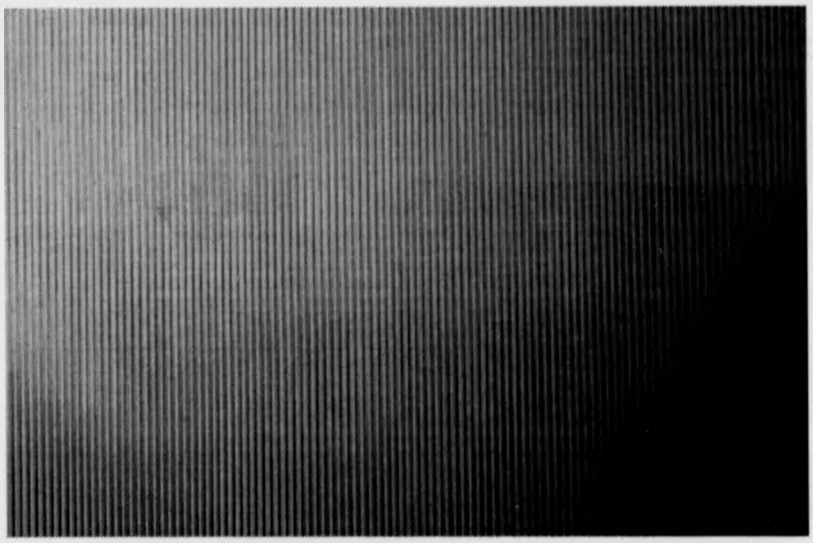


Figure 8.17: Carrier fringe pattern produced by crystalline quartz wedge in a dark field circular polariscope.

8.4.4 Software Development

An interactive software tool has developed for automatic photoelastic stress analysis. The software programme is written in the Turbo C programming language. The programme is menu driven to perform essential functions. There are six options on the menu,

1. **Motor**, this option allows the stepper motor to be controlled by computer. The stepper motors are used to drive two quarter-wave plates and a sample plate.

2. **Quadr**, this option allows the visible quadrant to be changed. There are 4 quadrants on the PIP board numbered 0 to 3.

3. **Video**, selecting this option causes the video camera to capture and preprocess the images.

4. **Fringes**, selecting this option causes the package to do fringe analysis by the phase-stepping and FFT approaches.

5. **Dos**, this is a system shell option, selecting this option allows execution of an operating system command such as "dir".

6. **Exit**, selecting this option causes the package to exit.

Figure 8.18 shows an overview of the system.

Several subroutines are developed to support these six options. There are four subroutines in the **Motor** option menu. All of the stepper motors are controlled via the IBM interface. The mode control of the 8255 is accomplished by a control word writing to the port control register (435). The statement **outport(control, 128)** configures the model control and port A as output. The direction and pulses are determined by writing an integer value to port A. A time delay of 5ms is required for each pulse. The stepper motor can be driven in a clockwise or an anti-clockwise direction by selecting the relevant bits of port A. A subroutine

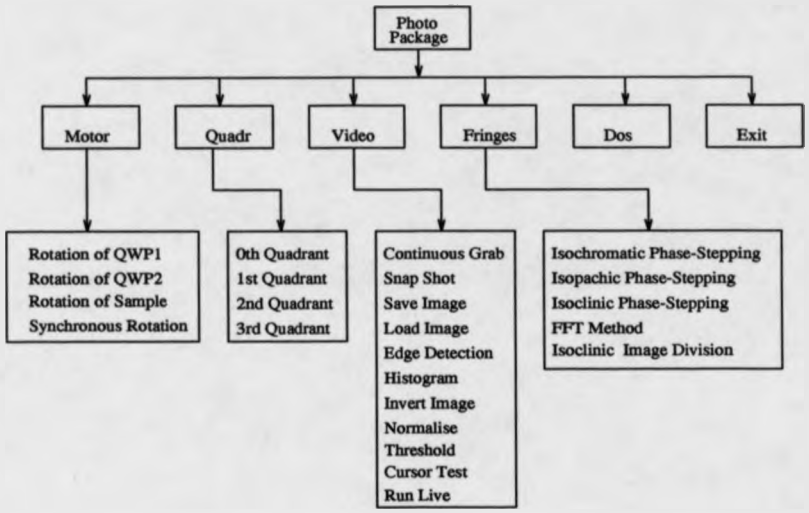


Figure 8.18: An overview of the interactive software tool for photoelastic stress analysis.

Synchronous-Rotation() is developed to rotate two quarter-wave plates at the same time, through the same angle but in opposite direction.

In the **Quadr** option, four 512×512 frame buffers or quadrants can be used to store four images. Each time one of the images can be displayed on the monitor through software control.

In the **Video** option, subroutines are written to enable the PIP to do a continuous grab and snapshot, and also to do some image processing, such as thresholding, normalising. The images can be loaded from a PC and saved into a PC under this option.

In the **Fringes** option, the images can be processed by either the phase-stepping or FFT methods. After computer processing, the phase information is extracted from the stress fringe pattern. Phase-unwrapping is performed to obtain a continuous phase, showing the stress distribution. Because of the large amount of computation, the processing takes 15 minutes. In order to speed up the computation, the PC is connected to the departmental computer network through

Ethernet and the PC Net Filing System (PCNFS). The image processing takes only 2 minutes and half to get the unwrapped phase map and also mesh data.

8.5 Results and Discussion

In order to illustrate the capability of the proposed photocarrier and FFT methods for stress fringe pattern processing, two specimens are investigated in a computer controlled polariscope. The first specimen is a circular disk 40mm in diameter, cut from a sheet of CT-200 Epoxy Resin, under a diametral load. The second specimen is a circular ring, also made of CT-200 Epoxy Resin, under a diametral load. With a sodium light, the intensities of the photoelastic patterns for the circular disk is recorded by a CCD camera. The dark field isochromatic fringe pattern is digitised and stored in the frame buffer of the PIP board of 512 × 512 pixels and 256 grey levels as shown in Fig. 8.19.

Figure 8.19: Isochromatic stress pattern of a circular disk under diametral compression (image size 512 × 512).

To facilitate FFT fringe analysis for extracting phase information, a photocarrier system of fringes is introduced in the computer controlled polariscope.

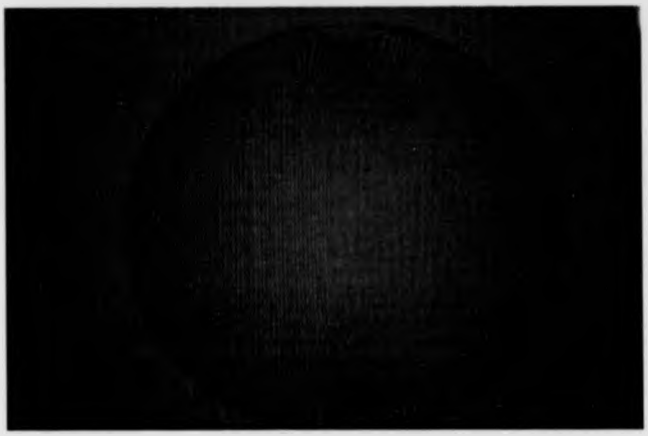


Figure 8.20: Circular disk under diametral load with carrier fringes (image size 512 × 512).

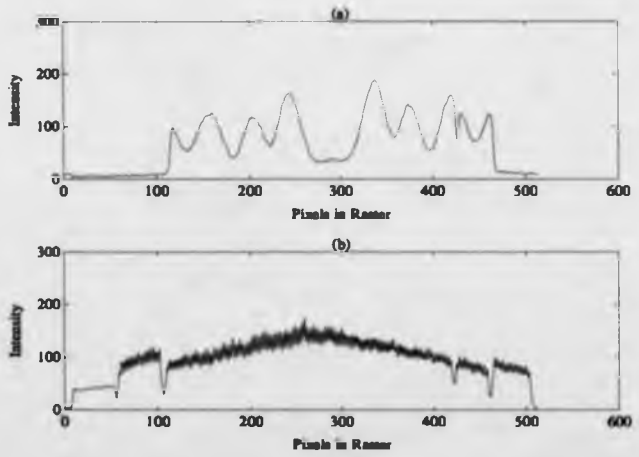


Figure 8.21: (a) Digitised intensity data of central raster in a circular disk; (b) digitised intensity data of central raster in a circular disk with carrier fringes.

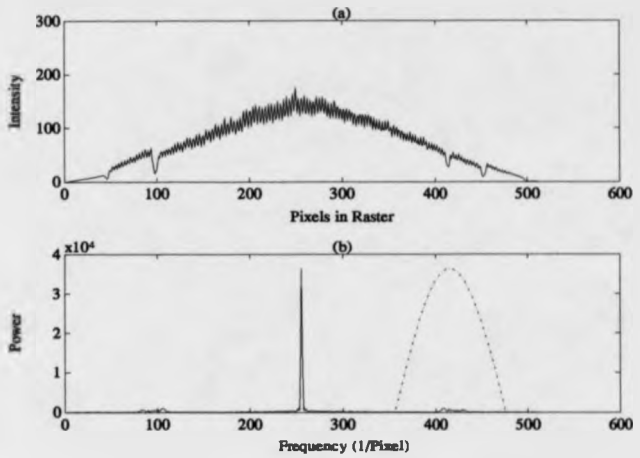


Figure 8.22: (a) Digitised intensity data of central raster in a circular disk with carrier fringes weighted by Papoulis window; (b) power spectrum of central raster from the circular disk with photocarrier (sidelobe weighted by Hanning window).

This is performed by putting a birefringent wedge in front of the specimen under study. Changes of phase smaller than 2π in the whole field of interest are difficult to detect with accuracy. However, by introducing a carrier, the problem can be solved.

In the case presented here, both carrier and stress patterns are multiplied and recorded with a CCD camera. Figure 8.20 shows the multiplied fringe patterns of the circular disk. The intensity data of the central raster for the circular disk with carrier and without carrier are shown in Fig. 8.21(b) and Fig. 8.21(a) respectively.

The phase information of the principal stress difference of the circular disk can be obtained from the FFT. The intensity data is weighted by a Papoulis window as shown in Fig. 8.22(a), before taking the FFT. Figure 8.22(b) is the power spectrum of the central raster of the intensity data in a circular disk. Figure 8.23 shows the wrapped phase map of the circular disk. Despite the time-edge effect and the scratched surface of the circular disk, the resultant phase profile is smooth as long as the phase modulation by the carrier is high. However

discontinuities may occur when the phase modulation by the carrier is small, as shown in Fig. 8.24 (top and bottom of the circular disk). Figure 8.25 shows a side view of the principal stress difference distribution of a disk. Figure 8.26 shows a 3-D plot of stress difference distribution of the circular disk. It is shown that the value of the stress difference is high near the loading point. Because of the free boundary of the circular disk, the value of the stress difference is zero around the edge of the disk. Figure 8.27 is a contour map of a disk under diametral compression.

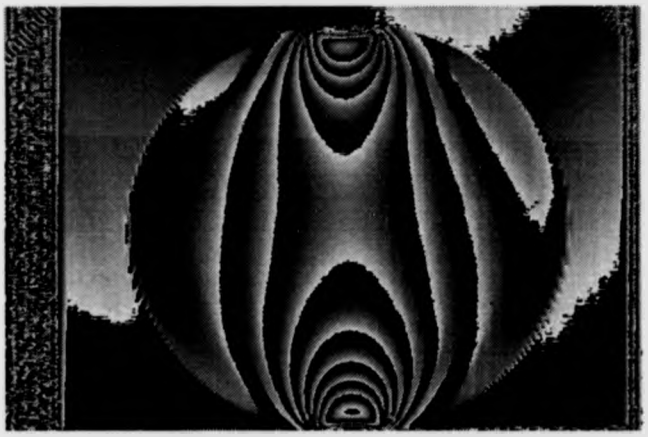


Figure 8.23: Wrapped phase map of circular disk under diametral load (image size 512 × 512).

For comparison, a frozen wedge which has a low carrier frequency (1 mm/line) is used as a photocarrier. Because of the low carrier frequency, the stress pattern of the disk is not well modulated, especially in the top and the bottom of the disk as shown in Fig. 8.28. Figure 8.29(b) and 8.29(a) show the digitised intensity of a central raster with and without carrier fringes respectively. The intensity data weighted by a Papoulis window, and the power spectrum of the central raster are shown in Figure 8.30. Note that in Fig. 8.30, the carrier frequency is not high enough to separate the sidelobe from the centre lobe. This will cause large errors

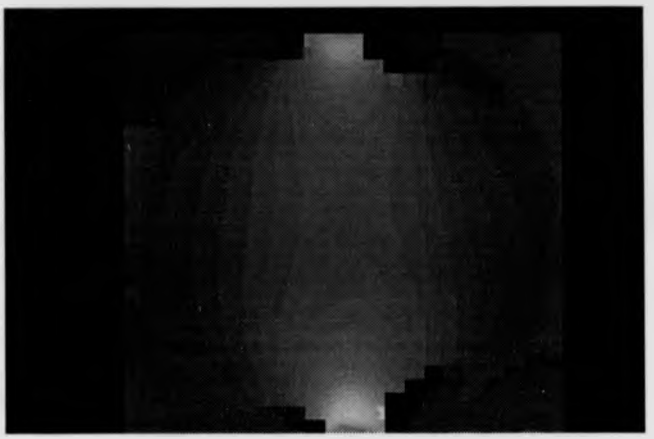


Figure 8.24: Unwrapped phase map of circular disk under diametral load (image size 512 × 512).

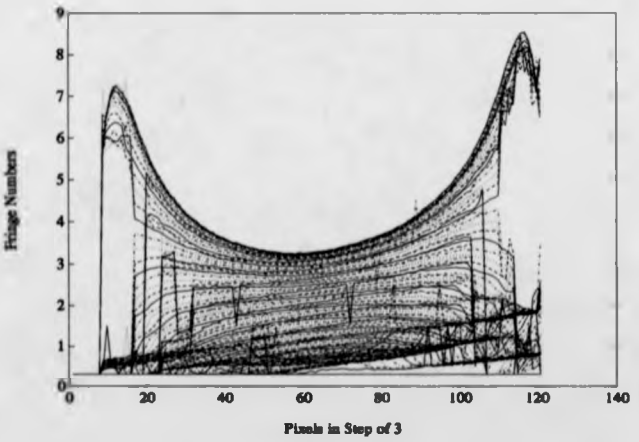


Figure 8.25: Side view of unwrapped phase map of a disk.

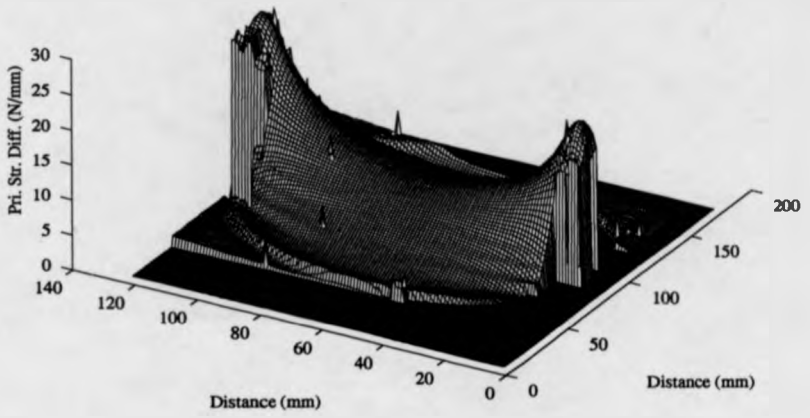


Figure 8.26: 3-D perspective plot of principal stress difference for a circular disk.

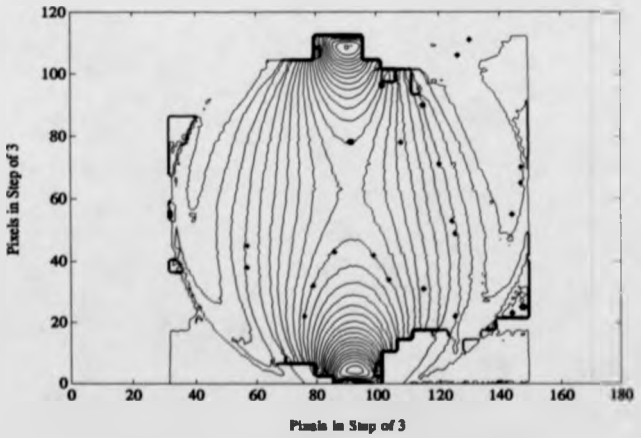


Figure 8.27: Contour map of a disk.

Figure 8.28: Isochromatic fringe pattern of a circular disk with low carrier frequency (image size 512 × 512).

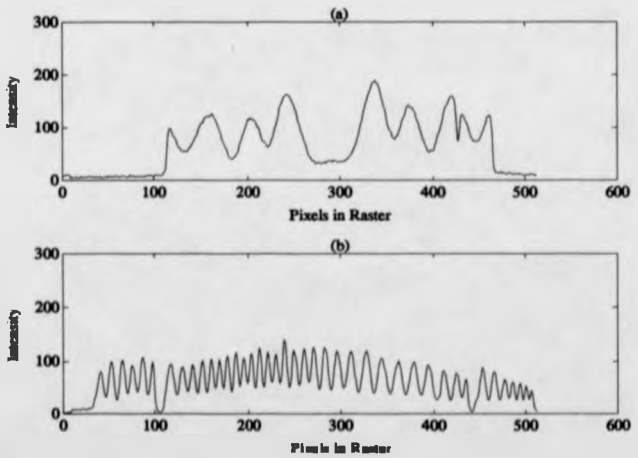


Figure 8.29: (a) Digitised intensity data of central raster in a disk; (b) digitised intensity data of central raster in a disk with less carrier fringes.

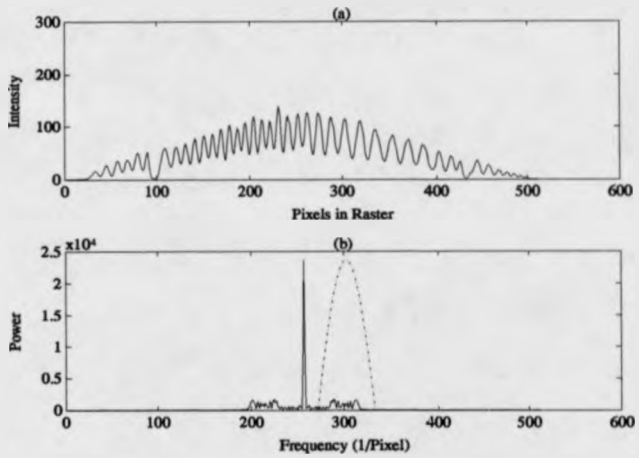


Figure 8.30: (a) Digitised intensity data of central raster in a disk with carrier fringes weighted by Papoulis window; (b) power spectrum of central raster from the disk with carrier fringes (sidelobe weighted by Hanning window).

Figure 8.31: Wrapped phase map of a circular disk with errors (image size 512 × 512).

in the wrapped phase map as shown in Fig. 8.31.

The stress in a circular disk can be found from the theory of elasticity [191]. Figure 8.32 shows a circular disk under two equal and opposite forces load. The stress along a diameter CD can be expressed by (see Appendix II)

$$\sigma_x = \frac{2P}{\pi d}\left[1 - \frac{16x^2d^2}{(d^2+4x^2)^2}\right] \tag{8.19}$$

$$\sigma_y = \frac{2P}{\pi d}\left[1 - \frac{4d^4}{(d^2+4x^2)^2}\right] \tag{8.20}$$

The maximum compressive stress along the diameter CD is calculated as $5.46N/mm^2$, which is located at the centre of the disk. The discrepancy between the theoretical stress and that measured is $0.17N/mm^2$. Figure 8.33 shows a cross section through the point of maximum stress along a diameter CD, obtained from FFT and a plot of the theoretical deformation curve. From this comparison it is seen that the maximum deviation in the comparison occurs at the edge of the disk. This is because of the time edge effect of the disk when making the sample models.

For a circular ring, the fringe pattern is much more complicated than that of a circular disk. Figure 8.34 shows the dark field isochromatic fringe pattern of a circular ring under a diametral load. Figure 8.35 shows the fringe pattern resulting from the superposition of a diametrally compressed circular ring and a carrier wedge.

The digitised intensity data of the central raster in a circular ring is shown in Fig. 8.36(a). Figure 8.36(b) is the intensity data of the central raster in a circular ring with carrier fringes. The intensity data is weighted using a Papoulis window (see Fig. 8.37(a)) before analysis with the FFT. The power spectrum of the intensity data for a circular ring is shown in Fig. 8.37(b). The wrapped phase map of a circular ring is shown in Fig. 8.38. Because of the low phase modulation by the carrier near the edge of the ring, discontinuities occur. An example can

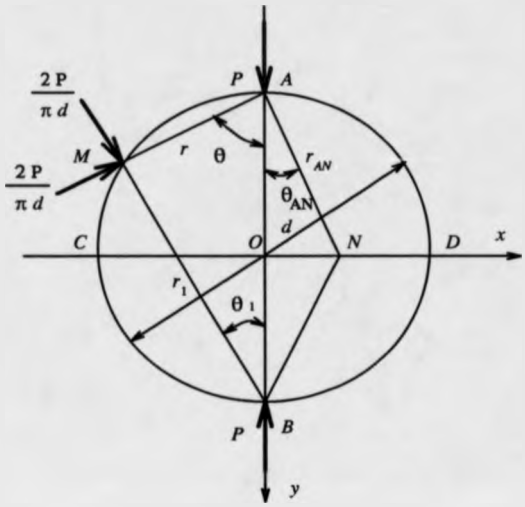


Figure 8.32: Circular disk under a diametral load.

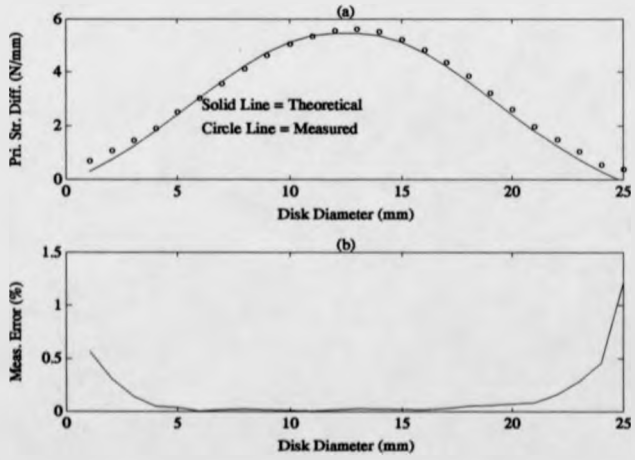


Figure 8.33: Comparison between the theoretical and measured stress for cross section. (a) Plots of principal stress difference for both theoretical and measured; (b) measurement errors.

be seen close to the edge of the circular ring in Fig. 8.38. The unwrapped phase map is shown in Fig. 8.39. Figure 8.40 shows a side view of the unwrapped phase map of a circular ring. A 3-D perspective plot of the principal stress difference is shown in Fig. 8.41. Figure 8.42 is a contour map of a circular ring under a diametral load.

Figure 8.34: Isochromatic stress pattern of circular ring under diametral load (image size 512 × 512).

8.6 Conclusion

The main contribution of this chapter is in applying the carrier fringe and FFT techniques to photoelastic stress measurement. The novel feature of the work is to generate linear carrier fringes in the stress fringe field. The carrier fringes are generated by using a birefringent wedge. The stress fringes are modulated by the linear carriers in the polariscope. A theoretical analysis of carrier fringe technique is carried out by using Jones matrix method. An interactive software tool for automatic analysis of photoelastic stress has also been developed. This full field, fully automated photoelastic stress measurement procedure is a new

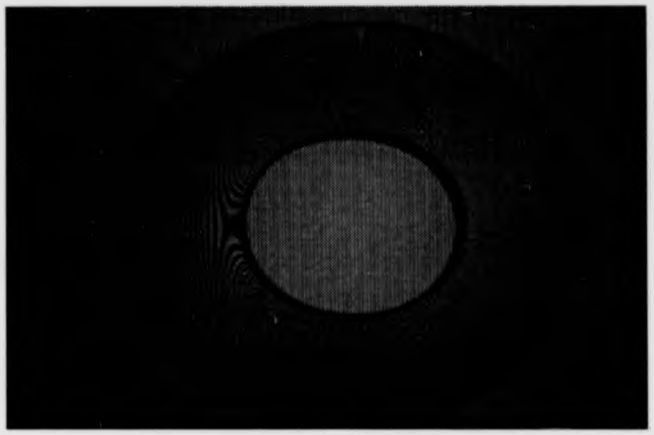


Figure 8.35: Circular ring under diametral load with photocarrier (image size 512 × 512).

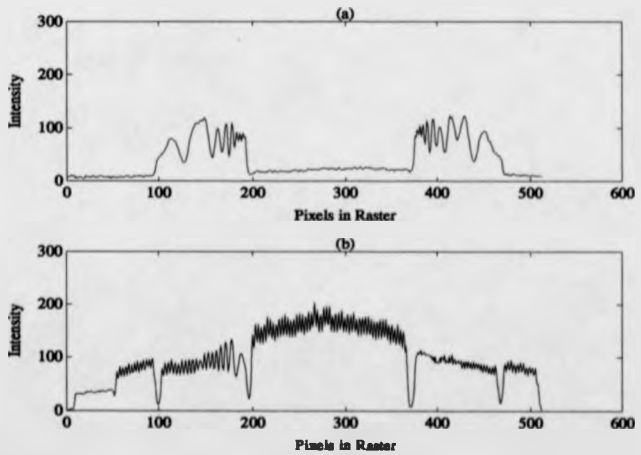


Figure 8.36: (a) Digitised intensity data of central raster in a circular ring; (b) digitised intensity data of central raster in a circular ring with carrier fringes.

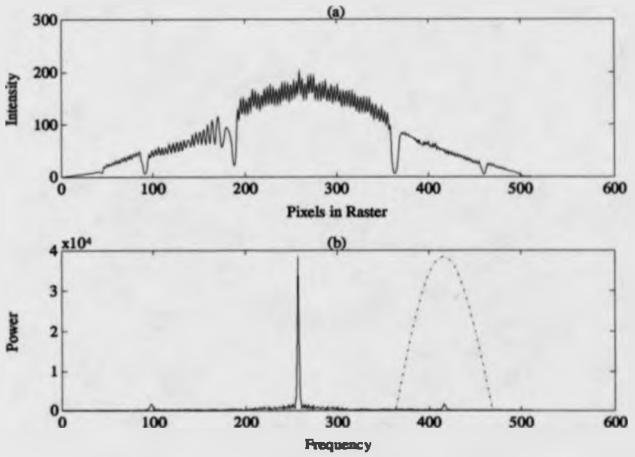


Figure 8.37: (a) Digitised intensity data of central raster in a circular ring with carrier fringes weighted by Papoulis window; (b) power spectrum of central raster from the circular ring with carrier fringes (sidelobe weighted by Hanning window).

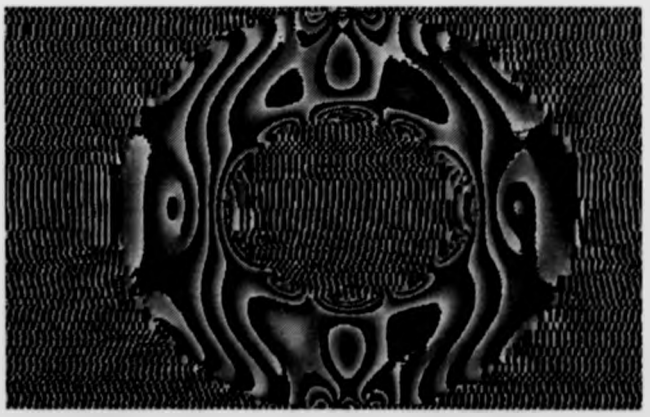


Figure 8.38: Wrapped phase map of circular ring (image size 512×512).

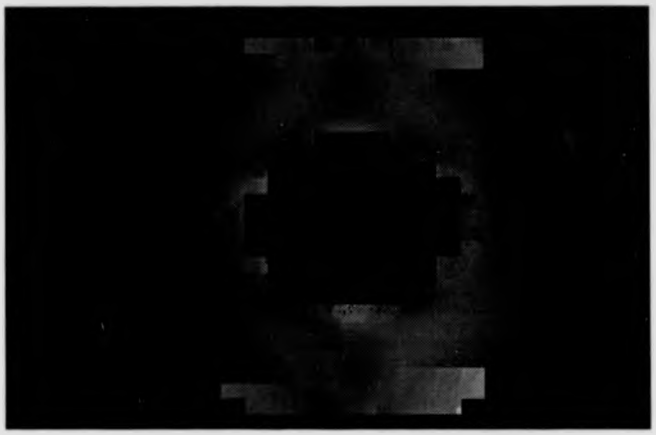


Figure 8.39: Unwrapped phase map of circular ring (image size 512 × 512).

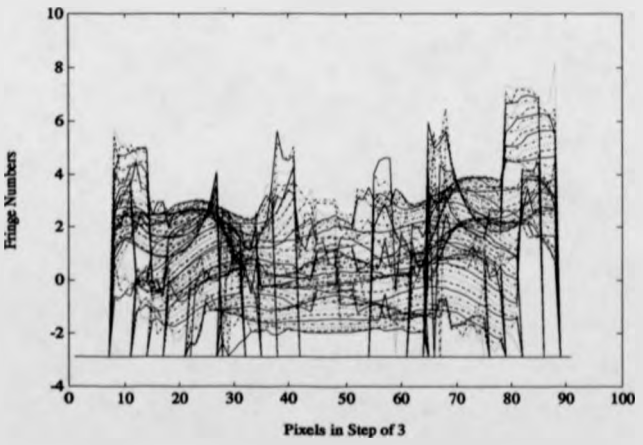


Figure 8.40: Side view of unwrapped phase map of circular ring.

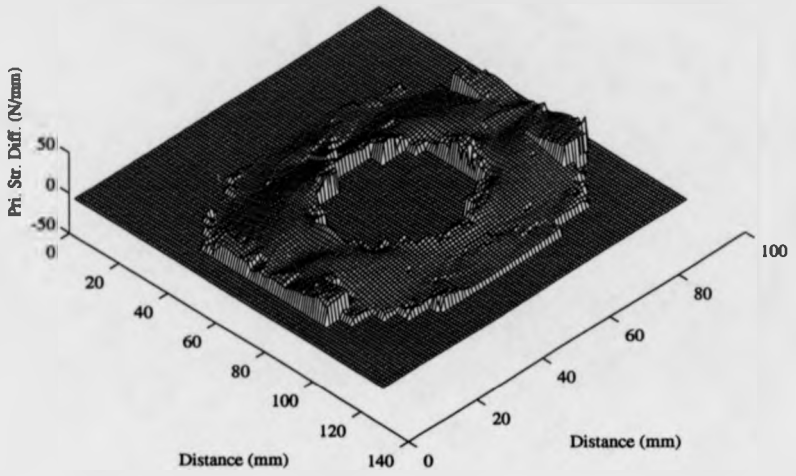


Figure 8.41: 3-D perspective plot of principal stress difference for a circular ring.

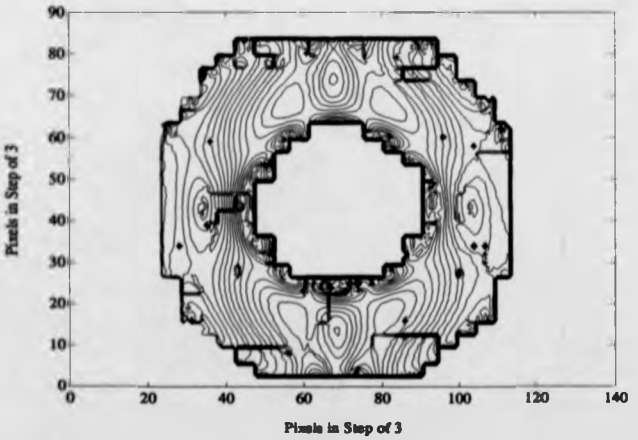


Figure 8.42: Contour map of a circular ring under diametral loading.

approach to photoelasticity. It has been found that high stress concentration areas can not be solved completely even using high density carrier fringes. The next stage of development is to tackle the problem of resolving the high stress concentrations as well as to develop the stress separation method. This will result in the development of a new efficient software system.

Chapter 9

Conclusion

This chapter summaries the results and discussions presented in the thesis. It also makes suggestions for possible areas for future research.

A survey of holographic techniques and the automation of the analysis of holographic interference patterns has been extensively discussed. The most important advantage of holographic interferometry is the fact that objects with a rough surface and complicated shape may be investigated with its help. There is a gradually increasing employment of fibre optics for coherent illumination. Data reduction is an important factor in holographic interferometry. The general requirement for optical fringe pattern interpretation is that the quantitative information relating to measurement parameter can be automatically and efficiently extracted. Several algorithms for fringe analysis with particular emphasis on the automated carrier fringe and FFT techniques have been described. All of the techniques described can significantly extract quantitative data from optical fringe patterns and provide a very large number of data points in a short period of time. The fringe tracking method automatically finds fringe maximas and minimas by algorithms based on line tracking. The disadvantage of the fringe tracking method is its low accuracy compared to other methods. There is no interference phase directly determined at the points between the fringe maxima and minima. The heterodyne method has the advantages of the high accuracy

and high resolution ($\lambda/1000$). However, it suffers from mechanical scanning of the reconstruction and high repositioning requirements. The phase-stepping method is widely used in the phase measuring of fringe patterns. The phase-stepping method has been applied to all types of holographic interferometric measurement and offers high accuracy ($\lambda/100$). The Fourier transform method combined with the carrier fringe technique can be used to extract the deformation information from complicated fringe patterns. This method has a number of advantages. It requires only one interferogram to extract phase information where as the phase-stepping method requires multiple interferograms taken at different times. There are no special devices required for generating carrier fringes where as the phase-stepping method requires a phase shifting device. Image enhancement and noise suppression are performed by filtering away distortions in the frequency domain. The achievable accuracy is almost as good as the phase-stepping method.

A novel fibre-optic system for the production of holograms has been constructed. Such system is used for holographic contouring, holographic deformation measurement. The introduction of fibre optics has resulted in easy handling of holographic interferometry. A new technique for surface contouring has been proposed and verified by experiments based on the fibre-optic system. The system is suitable for automatic measurement of a 3-D object shape. The fringe pattern is Fourier-transformed and processed in the spatial domain as well as in the frequency domain. The technique has a higher sensitivity than the conventional Moire technique and can be made fully automatic. The precision of measurement of the surface height depends on the number of fringes present. Therefore, there should be as many fringes as possible on the surface of object. The contour interval can be adjusted during the recording by varying the translation of the fibre optic beams. If a high resolution CCD camera is employed, the technique can be applied to real-time surface contouring using ESPI.

A theoretical analysis of the carrier fringe and FFT techniques has been presented and successfully applied to holographic interferometry. By using the carrier fringe technique, a complicated interference pattern resulting from double-

exposure holographic interferometry is replaced by a simple linear fringe pattern. The effects of carrier frequency on the interferogram is also investigated by computer generated holograms. It has been shown that the carrier frequency should be high enough to greatly dominate the fringe pattern. The maximum carrier frequency is limited by the Nyquist frequency. The optimum carrier frequency has been found to be 4 pixels/fringe. In the Fourier transform method, phase errors in the phase detection are caused by truncation, sampling, incorrect filtering in the frequency domain, random noise, and also by incorrect choosing the carrier frequency. The error can be reduced if the frequency of the carrier is chosen properly. The use of a Hanning window in the frequency domain, and the Papoulis window in the spatial domain can also reduce the errors in phase detection. The use of fibre optics in holographic interferometry greatly facilitate the carrier fringe technique and FFT fringe analysis method. The sign ambiguities of displacement to an object surface is removed by the carrier fringe technique. It has been shown that the Fourier transform method can produce phase maps automatically without the need for locating fringe centres or assigning fringe orders. The significant feature of the FFT method simplifies analysis of interferograms because it only needs one image to get phase information. To demonstrate the technique, several results from a centrally loaded disk and side-loaded cantilever beam have been presented illustrating the use of the technique on real image data and illustrated its advantages in studying the complicated fringe patterns. An excellent correlation between the theoretical deformation profile and that suggested by the technique is given. The accuracy of the measurement for a centrally loaded aluminum disk is 0.05μm.

The carrier fringe technique has also been applied to photoelastic stress measurement. A computer controlled photoelastic stress measurement rig has been constructed. The carrier fringes can be generated easily by inserting a birefringent wedge in the polariscope. The combined fringe pattern is captured by a CCD camera and analyzed by the Fourier transform method. The quantitative 3-D principal stress difference can be produced using this technique. There is

an excellent correlation between the results obtained using experimental method and the theoretical analysis. A theoretical analysis of the carrier fringe technique is carried out by using Jone's matrix method. An interactive software tool for automatic analysis of photoelastic stress has also been developed. This full field, fully automated photoelastic stress measurement procedure is a new approach to photoelasticity. Unlike phase-stepping method, only one frame is needed to form the fringe information using FFT method. It has been shown that the Fourier transform method of fringe analysis, combined with the carrier fringe technique can be used to extract principal stress difference information automatically from complex stress fringe patterns.

As a consequence of the research work presented in the thesis, a number of feasibility studies should be mentioned.

- With the refinement of computer aided optical analysis produced by this thesis, novel instrumentation for the whole surface displacement mapping in the industrial environment will be constructed and new understandings to present theoretical applications must be possible.

- It has been found that the high stress concentration areas can not be solved completely even using high density carrier fringes. Therefore, the next stage will be to solve the high stress concentrations and to develop a stress separation method. This will result in the development of a new efficient software system.

Bibliography

[1] H. M. Smith, *Principles of Holography*, Wiley-Interscience, New York, 35-48(1969).

[2] R. J. Parker and D. G. Jones, "Holography in an industrial environment", *Opt. Eng.* **27**(1), 55-66(1988).

[3] Y. W. Qin, C. T. Tou and Z. S. Ye, "A new method of resolving transient stresses dynamic holo-photoelasticity", *Exp. Mexh.* **30**(3), 1-3(1990).

[4] N. Abramson, "Holographic contouring by translation," *Appl. Opt.* **15**, 1018(1976).

[5] C. Quan and P. J. Bryanston-Cross, "Double-source holographic contouring using fibre optics," *Opt. Laser Tech.* **22**(4), 255-259(1990).

[6] P. Carelli, D. Paoletti, G. S. Spagnolo and A. D'Altorio, "Holographic contouring method: application to automatic measurements of surface defects in artwork", *Opt. Eng.* **30**(9), 1294-1298(1991).

[7] P. H. Malyak and B. J. Thompson, "Particle displacement and velocity measurement using holography", *Opt. Eng.*, **23**(5), 567-576(1984).

[8] P. J. Bryanston-Cross, M. Funes-Gallanzi, C. Quan, and T. R. Judge, "Holographic particle image velocimetry (HPIV)", *Opt. Laser Tech.*, Vol. 24, No. 5, 251-256(1992).

[9] A. V. Baez, "A study in diffraction microscopy with special reference to X-rays", *J. Opt. Soc. Am.* **42**, 756(1952).

[10] M. R. Howells, "X-ray holography", *Synchrotron Radiation News* **3**(4), 23-28(1990).

[11] D. Gabor, "A new microscopic principle," *Nature* **161**, 777-778(1948).

[12] D. Gabor, "Microscopy by reconstructed wave-fronts," *Proc. Roy. Soc.* **A197**, 454-487(1949).

[13] D. Gabor, "Microscopy by reconstructed wave-fronts," *Proc. Roy. Sco.* **B64**, 449-469(1951).

[14] E. N. Leith and J. Upatnieks, "Reconstructed wavefronts and communication theory," *J. Opt. Sco. Am.* **52**, 1123(1962).

[15] E. N. Leith and J. Upatnieks, "Wavefront reconstruction with continuous-tone objects," *J. Opt. Sco. Am.* **53**, 1377(1963).

[16] E. N. Leith and J. Upatnieks, "Wavefront reconstruction with diffused illumination and three-dimensional objects," *J. Opt. Sco. Am.* **54**, 1295(1964).

[17] T. Yatagai, M. Idesawa, Y. Yamaashi and M. Suzuki, "Interactive fringe analysis system: application to moire contourogram and interferogram," *Opt. Eng.* **21**(5), 901-906(1982).

[18] D. W. Robinson and D. C. Williams, "Automatic fringe analysis in double exposure and live fringe holographic interferometry", *Proc. SPIE* **599**, 134-140(1985).

[19] R. Dandliker and R. Thalman, "Heterodyne and quasi-heterodyne holographic interferometry," *Opt. Eng.* **24**(5), 824-831(1985).

[20] J. W. Wagner, "High speed applications of heterodyne hologram interferometry", *Proc. SPIE* **746**, 194-200(1987).

[21] D. Malacara, "Phase shifting interferometry", *Revisa Mexicana de Fisica* **36**(1), 6-22(1990).

[22] M. Chang, Ching-Piao Hu and J. C. Wyant, "Phase-shifting holographic interferometry(PHSI)," *Proc. SPIE* bf 599, 149-154(1985).

[23] T. A. W. M. Lanen, C. Nebbeling and J. L. van Ingen, "Digital phase-stepping holographic interferometry in measuring 2-D density fields", *Exp. in Fluids* **9**, 231-235(1990).

[24] K. Creath, "Phase-measurement interferometry techniques," *Prog. in Opt.* **26**, 349-393(1988).

[25] G. Lai and T. Yatagai, "Generalised phase-shifting interferometry", *J. Opt. Soc. Am.* bf 8(5), 822-827(1991).

[26] M. Takeda, H. Ina, and S. Kobayashi, "Fourier-transform method of fringe-pattern analysis for computer-based topography and interferometry," *J. Opt. Soc. Am.* **72**, 156-160(1982).

[27] R. J. Green, J. G. Walker, and D. W. Robinson, "Investigation of the Fourier-transform method of fringe pattern analysis," *Opt. Laser Eng.* **8**, 29-44(1988).

[28] T. Kreis, "Digital holographic interference-phase measurement using the Fourier-transform method," *J. Opt. Soc. Am.* **3**(6), 847-855(1986).

[29] A. M. P. P. Leite, "Optical fibre illuminators for holography", *Opt. Commun.* bf 28(3), 303-308(1979).

[30] J. A. Gilbert, T. D. Dudderar, M. E. Schultz and A. J. Boehnlein, "The monomode fibre - a new tool for holographic interferometry", *Exp. Mech.* bf 23, 190-195(1983).

[31] J. D. C. Jones, M. Corke, A. D. Kersey, and D. A. Jackson, "Single-mode fibre- optic holography", *J. Phys. E: Sci. Instrum.* bf 17, 271-273(1984).

[32] H. I. Bjelkhagen, "Pulsed fibre holography: a new technique for hologram interferometry", *Opt. Eng.* **24**(4), 645-649(1985).

[33] L. O. Heflinger and R. F. Wuerker, "Holographic contouring via multifrequency lasers", *Appl. Phys. Lett.* **15**(1), 28-30(1969).

[34] R. P. Tatam, T. C. Davies, C. H. Buckberry, J. D. C. Jones, "Holographic surface contouring using wavelength modulation of laser diodes", *Opt. Laser Tech.* **22**(5), 317-321(1990).

[35] P. Hariharan and B. F. Oreb, "Two-index holographic contouring: application of digital techniques", *Opt. Commun.* **51**(3), 142-144(1984).

[36] B. P. Hildebrand and K. A. Haines, "The generation of three-dimensional contour maps by wavefront reconstruction", *Phys. Lett.* **21**, 422(1966).

[37] B. P. Hildebrand and K. A. Haines, "Multiple-wavelength and multiple-source holography applied to contouring generation," *J. Opt. Soc. Am.* **57**, 155-156(1967).

[38] M. Idesawa, T. Yatagai and T. Soma, "Scanning moire method and automatic measurement of 3-D shapes", *Appl. Opt.* **16**(8), 2152-2162(1977).

[39] C. Quan, T. R. Judge and Bryanston-Cross, "Holographic measurement of deformation using carrier fringe and FFT techniques", *Proc. SPIE* **1507**, 463-475(1991).

[40] M. Takeda and Z. Tung, "Subfringe holographic interferometry by computer-based spatial-carrier fringe-pattern analysis," *J. Optics* **16**(3), 127-131(1985).

[41] P. D. Plotkowski, Y. Y. Hung, J. D. Hovanesian, and Grant Gerhart, "Improved fringe carrier technique for unambiguous determination of holographicly recorded displacements," *Opt. Eng.* **24**(5), 754-756(1985).

[42] G. O. Reynolds, D. A. Servaes, L. Ramos-lzquierdo, J. B. BeVelis, D. C. Peirce, P. D. Hilton, and R. A. Mayville, "Holographic fringe linearization interferometry for defect detection," *Opt. Eng.* bf 24(5), 757-768(1985).

[43] D. R. Matthys, J. A. Gilbert, T. D. Dudderar, and K. W. Koening, "A windowing technique for the automated analysis of holo-interferograms," *Opt Laser Eng.* **8**, 123-136(1988).

[44] B. Han and D. Post, "The tilted-plate method for introducing carrier fringes of extension in moire interferometry", *Exp. Tech.* **13**(7), 25-29(1989).

[45] P. Long, D. Hsu and B. Wang, "Cylindrical fringe carrier technique in holographic interferometry", *Opt. Eng.* **27**(10), 867-869(1988).

[46] S. S. James, Y. M. Chen, H. Singh and A. Y. Cheng, "Computerized optical fringe pattern analysis in photomechanics: a review", *Opt. Eng.*, **31**(2), 304-314(1992).

[47] A. Nussbaum and R. A. Phillips, *Contemporary Optics for Scientists and Engineers*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 149(1976).

[48] Yu. I. Ostrovsky, M. M. Butusov, G. V. Ostrovskaya, *Interferometry by Holography*, Spriger-Verlag, Berlin, 65-73(1980).

[49] K. Lizuka, *Engineering Optics*, Springer-Verlag, Berlin, 27-35(1987).

[50] B. Bon and E. Wolf, *Principles of Optics*, Pergamon Press Ltd., Headington Hill Hall, Oxford, 256-267(1965).

[51] K. D. Moller, *Optics*, University Science Books, Mill Valley, California, 91(1988).

[52] J. W. C. Gates, "The influence of holography on measurement technology", *J. Phys. E: Sci. Instrum.* **19**, 998-1007(1986).

[53] I. Weingartner, "Holography - techniques and applications", *J. Phys. E: Sci. Instrum.* **16**, 16-23(1983).

[54] G. Wyszecki and W. S. Stiles, *Color Science*, Section 3 and 4, Wiley, New York, 1967.

[55] C. B. Burckhart, "Information reduction in holograms for visual display," *J. Opt. Sco. Am.* **58**, 241(1968).

[56] H. Kogelnik, "Coupled wave theory for thick hologram gratings," *Bell. Syst. Tech. J.* **48**, 2909(1969).

[57] J. B. Develis, G. O. Regndds, *Theory and Applications of Holography*, Addison-Wesley, New York, 1971.

[58] J. C. MacBain, J. E. Horner, W. A. Stange and J. S. Ogg, "Vibration analysis of spinning disk using image-derotated holographic interferometry," *Exp. Mech.* **19**, 17-22(1979).

[59] P. J. Bryanston-Cross, J. W. Gardner, "Application of holographic interferometry to the vibrational analysis of the harpsichord," *Opt. Lasers Tech.* **20**(4), 199-204(1988).

[60] P. J. Bryanston-Cross, J. W. Gardner, "Holographic visualization of a combustion flame," *Opt. Lasers in Eng.* **9**, 85-100(1988).

[61] R. J. Collier, *Optical Holography*, Academic Press, New York and London, 68-78(1971).

[62] T. R. Judge, C. Quan and P. J. Bryanston-Cross, "Holographic deformation measurements by Fourier-transform technique with automatic phase unwrapping," *Opt. Eng.* **31**(3), 533-543(1992).

[63] P. J. Bryanston-Cross, "Holographic flow visualisation", *J. Photogra. Sci.* **37**, 339-345(1989).

[64] P. J. Bryanston-Cross, T. Lang, M. L. G. Oldfield, and R. J. G. Norton, "Interferometric measurements in a turbine cascade using image-plane holography," *J. Eng. Power* **103**(1), 124-130(1981).

[65] R. L. Powell and K. A. Stetson, "Interferometric vibration analysis of three-dimensional objects by wave-front reconstruction," *J. Opt. Soc. Am.* **55**, 612(1965).

[66] R. L. Powell and K. A. Stetson, "Interferometric vibration analysis by wave-front reconstruction," *J. Opt. Soc. Am.* **55**, 1593-1598(1965).

[67] M. A. Monahan and K. Bromley, "Vibration analysis by holographic interferometry," *J. Opt. Soc. Am.* **45**, 1166(1969).

[68] D. P. Tower, T. R. Judge and P. J. Bryanston-Cross, "Vibration measurements using dual reference beam holography," *Proc. SPIE* **1084**, 1989.

[69] R. M. Grant, and G. M. Brown, "Holographic nondestructive testing," *Materials Evaluation* **27**(4), 79-84(1969).

[70] E. Archbold, J. M. Burch, and A. E. Ennos, "The application of holography to the comparison of cylinder bares," *J. Sci. Instrum.* **44**, 489-494(1967).

[71] P. J. Bryanston-Cross, C. Quan, M. Funes-Gallanzi and T. R. Judge, "Quantitative analysis of holographic particle data", accepted for publication in *Opt. Laser Eng.*, June 1992.

[72] Yu. I. Ostrovsky, V. P. Shchepinov, V. V. Yakovlev, ***Holographic Interferometry in Experimental Mechanics***, Springer-Verlag, Berlin, 35-37(1991).

[73] L. O. Heflinger, R. F. Wuerker, and R. E. Brooks, "Holographic interferometry," *J. Appl. Phys.* **37**, 642(1966).

[74] J. N. Butters and J. A. Leendertz, "Holographic and video techniques applied to engineering measurement", *J. Meas. Control* **4**, 344(1971).

[75] D. W. Robinson and D. C. Williams, "Digital phase stepping speckle interferometry", *Opt. Commun.* **57**(1), 26-30(1986).

[76] K. A. Stetson and W. R. Brohinsky,"Electronic holography system fro vibration analysis and nondestructive testing", *Opt. Eng.* **26**(12), 1234-1239(1987).

[77] E. Vikhagen, "Nondestructive testing by use of TV holography and deformation phase gradient calculation", *Appl. Opt.* **29**(1), 137-144(1990).

[78] R. S. Sirohi, A. R. Ganesan and M. P. Kothiyal, "Some new techniques with digital speckle pattern interferometry (DSPI)", *Proc. SPIE* **952**, 218-224(1988).

[79] M. O. Petersen and P. D. Jensen, "Computer-aided electronic speckle pattern interferometry (ESPI), deformation analysis by fringe manipulation", *NDT International* **21**(6), 422-426(1988).

[80] R. Jones and C. Wykes, "Electronic speckle pattern interferometry", in *Holographic and Speckle Interferometry*, Chap. 4., Cambridge University Press, (1983).

[81] B. Sharp, "Electronic speckle pattern interferometry (ESPI)", *Opt. Laser Eng.* **11**, 241-255(1989).

[82] K. Creath and G. A. Slettemoen, "Vibration observation techniques for digital speckle pattern interferometry", *J. Opt. Soc. Am.* **A2**, 1629-1636(1985).

[83] N. Eichhorn and W. Osten, "An algorithm for the fast derivation of line structures from interferograms", *J. Moden Opt* **35**(10), 1717-1725(1988).

[84] F. Becker and H. Yung,"Digital fringe reduction techniques applied to the measurement of three-dimensional transonic flow", *Opt. Eng.* **24**(3), 429-434(1985).

[85] D. W. Robinson, "Automatic fringe analysis with a computer image-processing system", *Appl. Opt. 22*(14), 2169-2176(1983).

[86] S. Toyooka and M. Tominaga, "Spatial fringe scanning for optical phase measurement," *Opt. Commun.* **51**(2), 68-70(1984).

[87] K. H. Womack, "Interferometric phase measurement using spatial synchronous detection", *Opt. Eng.* **23**(4), 391-395(1984).

[88] J. S. Sirkis, Y. M. Chen, H. Singh and A. Y. Cheng, "Computerized optical fringe pattern analysis in photomechanics: a review", *Opt. Eng.* **31**(2), 304-314(1992).

[89] R. J. Pryputniewicz, "Review of methods for automatic analysis of fringes in hologram interferometry," *Proc. SPIE* **816**, 140-149(1987).

[90] J. E. Gerivenkamp, "Generalised data reduction for heterodyne interferometry", *Opt. Eng.* **23**(4), 350-352(1984).

[91] D. Kerr, F. M. Santoyo and J. R. Tyrer, "Extraction of phase data from electronic speckle pattern interferometric fringes using a single-phase-step method: a novel approach", *J. Opt. Soc. Am.* **A7**(5), 820-826(1990).

[92] P. Hariharan, F. B. Oreb, and T. Eiju, "Digital phase-shifting interferometry: a simple error-compensating phase calculation algorithm ", *Appl. Opt.* **26**, 2504-2506(1981).

[93] K. Creath, "Comparison of phase-measurement algorithms," *Proc. SPIE* **680**, 19-28(1986).

[94] E. O. Brigham, *The fast Fourier transform and its applications*, Englewood Cliffs, N. J., Prentice Hall, 1988.

[95] W. W. Macy, Jr., "Two-dimensional fringe-pattern analysis," *Appl. Opt.* **22**(23), 3898-3901(1983).

[96] F. Obene, *Phd Thesis*, Engineering Department, University of Warwick, January, 1991.

[97] D. P. Towers, *Phd Thesis*, Engineering Department, University of Warwick, April, 1991.

[98] D. P. Towers, T. R. Judge, P. J. Bryanston-Cross, "Analysis of holographic fringe data using the dual reference approach", *Opt. Eng.* **30**(4), 452-460(1991).

[99] J. M. Huntley, "Noise-immune phase unwrapping algorithm," *Appl. Opt.* **28**(15), 3268-3270(1989).

[100] D. C. Ghiglia, G. A. Mastin, and L. A. Romero, "Cellular-automata method for phase unwrapping," *J. Opt. Soc. Am.* **4A**, 267-280(1987).

[101] A. Spik, D. W. Robinson, "Investigation of the cellular automata method for phase unwrapping and its implementation on an array processor", *Opt. Lasers Eng.* **14**(1), 25-37(1991).

[102] C. Buckberry, J. Davies, "Digital phase-shifting interferometry and its application to automotive structures", *Appl. Opt. Digest* 17th-20th Sept. 275-276(1990).

[103] A. Gibbons, *Algorithmic Graph Theory*, Cambridge University Press, Cambridge, 40-41(1985).

[104] N. Deo, *Graph Theory with Applications to Engineering and Computer Science*, Prentice-Hall, London, 60-64(1974).

[105] E. Shimon, *Graph Algorithms*, Pitman Publishing Limited, London, (1979).

[106] R. C. Prim, "Shortest connection networks and some generalizations," *Bell System Tech. Journal* **36**, 1389-1401(1957).

[107] T. R. Judge, *Phd Thesis*, Engineering Department, Warwick University, December, 1991.

[108] J. Canny, "A Computational Approach to Edge Detection", *IEEE Transactions on Pattern Analysis and Machine Intelligence* **PAMI-8**(6), 679-698(1986).

[109] K. N. Prettyjohns, "Charged-couple device image acquisition for digital phase measurement interferometry", *Opt. Eng.* **23**(4), 371-378(1984).

[110] R. J. Schalkoff, *Digital Image Processing and Computer Vision*, John Wiley & Sons, Inc. New York, 399-402(1989).

[111] A. Ettemeyer, H. Rottenkolber and J. Schorner, "A comparison of the methods available for the evaluation of holographic images," *Proc. SPIE* **952**, 168-174(1988).

[112] T. Kries, "Quantitative evaluation of interference patterns", *Proc. SPIE* **863**, 68-77(1987).

[113] D. P. Towers, T. R. Judge and P. J. Bryanston-Cross, "A quasi heterodyne holographic technique and automatic algorithms for phase unwrapping," *Proc. SPIE* **1163**, 95-119(1989).

[114] S. Toyooka and M. Yominaga, "Spatial fringe scanning for optical phase measurement", *Opt. Commun.* **51**(2), 68-70(1984).

[115] M. Kujawinska and J. Wojciak, "High accuracy Fourier transform fringe pattern analysis", *Opt. Lasers Eng.* **14**, 325-339(1991).

[116] J. A. Gilbert, T. D. Dudderar and A. J. Boehnlein, "Ultra low frequency holographic interferometry using fibre optics", *Opt. Laser Eng.* **5**, 29-40(1984).

[117] T. D. Dudder and J. A. Gilbert, "Fibre optic pulsed laser holography", *Appl. Phys. Lett.* **43**, 730-732(1983).

[118] Y. Suematsu, Ken-Ichi IGA, H. Matsumura and W. A. Gambling, *Introduction to Optical Fibre Communications*, John Wiley & Sons, New York, 1982.

[119] J. E. Midwinter, *Optical Fibres for Transmission*, John Wiley & Sons, New York, 1979.

[120] S. D. Wohlstein, "Using fibre optics for practical sensing", *Lasers & Optronics* **8**, 73-76(1989).

[121] D. A. Jackson and J. D. C. Jones, "Extrinsic fibre-optic sensors for remote measurement: Part one," *Opt. Laser Tech.* **18**(5), 243-252(1986).

[122] D. A. Jackson and J. D. C. Jones, "Extrinsic fibre-optic sensors for remote measurement: Part two," *Opt. Laser Tech.* **18**(6), 299-307(1986).

[123] A. W. Snyder, J. D. Love, *Optical Waveguid Theory*, Chapman and Hall, London, 1983.

[124] H. Zanger and C. Zanger, *Fibre Optics Communication and Other Application*, Collier Publishing Company, New York, 42-47(1991).

[125] Newport Corporation, *Projects in Fibre Optics*, U.S.A. 3-11(1986).

[126] M. K. Barnoski, ed., *Fundamentals of Optical fibre Communications*, 2nd Edition, Academic Press, New York, 1981.

[127] D. Gloge, "Weakly guided fibers", *Appl. Opt.* **10**, 2252(1971).

[128] von Bally, G., "Otoscopic investigations by holographic interferometry: a fibre endoscopic approach using a pulsed ruby laser system", in *Optics in Biomedical Sciences*, G. von Bally and P. Greguss Eds., Springer-Verlag, Berlin, 110(1982).

[129] F. Albe, H. Fagot, P. Smigielski, "Use of optical fibres in pulsed holography", *Proc. SPIE* **492**, ECOOSA'84, 324-329(1984).

[130] H. I. Bjelkhagen, "Fibre optics in holography", *Proc. SPIE* **615**, 13-17(1986).

[131] M. Yonemura, T. Nishisaka, and H. Machida, "Endoscopic hologram interferometry using fibre optics", *Appl. Opt.* **20**(9), 1664-1667(1981).

[132] J. N. Butters and J. A. Leendertz, "Application of video techniques and speckle pattern interferometry to engineering measurement", *Proc. Engineering Applications of Holography Symposium*, Los Angles, Calif., USA, 361-375(1972).

[133] R. P. Tatam, J. C. Davies, C. H. Buckbery, J. D. C. Jones, "Electronic speckle pattern surface contouring using optical fibres and wavelength modulation of laser diodes", *Proc. SPIE* **1314**, 278-283(1990).

[134] H. A. Vrooman and A . A. M. Maas, "Image processing in digital speckle interferometry," *Proc. Fringe Analysis'89*, Loughborough University of Technology, 4-5 April, 1989.

[135] M. Kujawinska, A. Spik and D. W. Robinson, "Analysis of ESPI interferograms by phase stepping techniques", *Proc. Fringe Analysis'89*, Loughborough University of Technology, 4-5 April, 1989.

[136] Newport Corporation, *The Newport Catalog No. 100*, Printed in U. S. A., J3-J19(1987)

[137] P. S. Theocaris, *Moire Fringes in Strain Analysis*, Pergamon, New York, 1969.

[138] J. D. Hovanesian and Y. Y. Hung, "Moire contour-sum contour difference, and vibration analysis of arbitray objects", *Appl. Opt.* **10**, 2734(1971).

[139] H. Takasaki, "Moire topography", *Appl. Opt.* **9**, 1467-1472(1970).

[140] D. M. Meadows, W. O. Johnson and J. B. Allen, "Generation of surface contours by moire patterns", *Appl. Opt.* **9**(4), 942-947(1970).

[141] R. Thalmann and R. Dandliker, "Holographic contouring using electronic phase measurement", *Opt. Eng.* **24**(6), 930-935(1985).

[142] Nils Abramson, "Sandwich hologram interferometry 3: Contouring," *Appl. Opt.* **15**, 200(1976).

[143] Y. Z. Dai and F. P. Chiang, "Contouring by moire interferometry", *Exp. Mech.* **31**, 76-81(1991).

[144] T. Yatagai and M. Idesawa, "Automatic fringe analysis for moire topography", *Opt. Laser Eng.* **3**, 73-83(1982).

[145] C. A. Miles and B. S. Speight, "Recording the shape of animals by moire method", *J. Phys. E: Sci. Instrum.* **8**, 773(1975).

[146] P. Benoit, E. Mathieu, J. Hormiere and A. Thomas, "Characterisation and control of three-dimensional objects using fringe projection techniques", *Nouvelle Revue D'optique* **6**, 67-86(1975).

[147] B. D. Bergguist and P. Montgomery, "Contouring by electronic speckle pattern interferometry (ESPI)", *Proc. SPIE*, December 1985.

[148] J. S. Zelenka and J. R. Varner, "Multi-index holographic contouring", *Appl. Opt.* **8**(7), 1431-1434(1969).

[149] F. Gyimesi and Z. Fuzessy, "Difference holographic interferometry (DHI): two-refractive-index contouring", *Opt. Commun.* **53**(1), 17-22(1985).

[150] J. S. Zelenka and J. R. Varner, "A new method for generating depth contours holographically", *Appl Opt.* **7**(10), 2107-2110(1968).

[151] M. Yonemura, "Wavelength-change characteristics of semiconductor lasers and their application to holographic contouring", *Opt. Lett.* **10**(1), 1-3(1985).

[152] W. Schmidt, A. Vogel and D. Preubler, "Holographic contour mapping using a dye laser", *Appl. Phys.* **1**, 103-109(1973).

[153] Yu. I. Ostrovsky, V. P. Shchepinov, V. V. Yakovlev, *Holographic Interferometry in Experimental Mechanics*, Springer-Verlag, Berlin, 190-204(1991).

[154] W. C. Gates, "Holographic phase recording by interference between reconstructed wavefronts from separate holograms", *Nature* **220**, 473-474(1968).

[155] H. Matsumoro, "Present status and problems of holographic contouring method", *Japan Soc. Prec. Eng.* **43**(5), 617-620(1977).

[156] T. Tsuruta and N. Shiotake, "Holographic generation of contour map of diffusing reflecting surface by using immersion method", *Japan. J. Appl. Phys.* **6**, 661-662(1967).

[157] E. S. Marrone and W. B. Ribbens, "Dual index holographic contour mapping over a large range of contour spacings", *Appl. Opt.* **14**(1), 23-24(1975).

[158] P. K. Rastogi and L. Pflug, "A fresh approach of phase management to obtain customized contouring of difuse object surfaces of broadly varying depths using real-time holographic interferometry", *J. Modern Opt.* **37**(7), 1233-1246(1990).

[159] C. M. Vest, *Holographic Interferometry* John Wiley & Sons. Inc., New York, 430(1979)

[160] J. Takezaki and Y. Y. Hung, "Direct measurement of flexural strains in plates by shearography," *J. Appl. Mech.* **53**, 125-129(1986).

[161] W. Tischer, "An interference pattern processing technique for optical phase measurement with applications in precision surface metrology", *Proc. SPIE* **1266**, 142-150(1990).

[162] G. Franowski, F. Schillke, "Microprofilometry using a spatial carrier frequency interferometrical technique", *Proc. SPIE* **1266**, 184-195(1990).

[163] P. Hariharan and B. S. Ramprasad, "Wavefront tilter for double-exposure holographic interferometry," *J. Phys. E: Sci. Instrum.* **6**, 173-175(1973).

[164] D. R. Matthys, T. D. Dudderar, and J. A. Gilbert, "Automated analysis of holo-interferograms for the determination of surface displacement," *Exp. Mech.* **28**, 86-91(1988).

[165] S. Nakadate and H. Saito, "Fringe scanning speckle-pattern interferometry", *Appl. Opt.* **24**(14), 2172-2180(1985).

[166] D. W. Robinson and D. C. Williams, "Digital phase stepping speckle interferometry," *Opt. Commun.* **57**(1), 26-30(1986).

[167] J. J. Gierloff, "Phase unwrapping by regions", *Proc. SPIE* **818**, 2-9(1987).

[168] R. J. Green, J. G. Walker, "Phase unwrapping using a priori knowledge about the band limits of a function", *Proc. SPIE* **1010**, 36-43(1988).

[169] R. Thalman and R. Dandliker, "High resolution video-processing for holographic interferometry applied to contouring and measuring deformations," *Proc. SPIE*, ECOOSA, 492(1984).

[170] P. C. Donovan, D. R. Burton, M. J. Lalor, "Fourier analysis of partial field fringe patterns", *Appl. Opt. Digest*, 17th-20th Sept., 325-326(1990).

[171] D. P. Towers, T. R. Judge, P. J. Bryanston-Cross, "Automatic interferogram analysis techniques applied to quasi-heterodyne holography and ESPI", *Opt. Lasers Eng.* **14**, 239-281(1991).

[172] S. Timoshenko and S. Woinowsky-Krieger, *Theory of plates and shell*, McGraw-Hill, New York, 1959.

[173] W. C. Young, *Roark's Formulas for stress and strain*, McGraw-Hill Book Company, New York, 93-100(1989).

[174] A. Lev and I. Politch, "Measuring the displacement vector by holographic interferometry", *Opt. Laser Tech.* **11**, 45-47(1979).

[175] D. G. Berghaus, "Overdetermined photoelastic solutions using least squares", *Exp. Mech.* **13**, 97-104(1973).

[176] A. S. Kobayashi, "Hybrid experimental-numerical stress analysis", *Exp. Mech.* **23**(3), 60-65(1985).

[177] R. K. Muller and L. R. Saackel, "Complete automatic analysis of photoelastic fringes", *Exp. Mech.* **19**(7), 245-251(1979).

[178] Y. Seguchi, Y. Tomita and M. Watanabe, "Computer-aided fringe-pattern analyzer–acase of photoelastic fringe", *Exp. Mech.* **19**(10), 362-370(1979).

[179] A. S. Voloshin and C. P. Burger, "Half-fringe photoelasticity: a new approach to whole-field stress analysis", *Exp. Mech.* **23**(9), 304-313(1983).

[180] F. W. Hecker and B. Morche, "Computer-aided measurement of relative retardations in photoelasticity", in *Experimental Stress Analysis*, H. Wieringa (ed), Martinus Nijhoff Publishers, Dordrecht, The Netherlands, 536-542(1986).

[181] A. G. Gillies, "Image processing approach to fringe patterns", *Opt. Eng.* **27**(10), 861-866(1988).

[182] G. M. Brown and J. L. Sullivan, "The computer-aided holo-photoelastic method", *Exp. Mech.* **30**(6), 135-144(1990).

[183] W. C. Wang, C. Y. Chen and W. H. Chen, "Half-fringe photoelastic analysis of rectangular plates subjected to transverse loading", *Opt. Eng.* **27**(8), 636-640(1988).

[184] K. Sharples, "Photoelastic stress analysis", *CME*, October, 42-46(1981).

[185] R. J. Sanford, "Photoelastic holography – A modern tool for stress analysis", *Exp. Mech.* **20**(12), 427-436(1980).

[186] M. E. Fourney and K. V. Mate, "Further applications of holography to photoelasticity", *Exp. Mech.* **10**(5), 177-186(1970).

[187] A. Ajovlast and A. Bardi, "Holographic photoelasticity: determination of absolute retardation by a single hologram", *Exp. Mech.* **16**(7), 273-275(1976).

[188] D. Post, "Holography and interferometry in photoelasticity", *Exp. Mech.* **12**(3), 113-123(1972).

[189] D. C. Holloway and R. H. Johnson, "Advancements in holographic photoelasticity", *Exp. Mech.* **11**(2), 57-63(1971).

[190] P. S. Theocaris and E. E. Gdoutos, *Matrix Theory of Photoelasticity*, Springer-Verlag, Berlin, 113-129(1979).

[191] S. P. Timoshenko and J. N. Goodier, *Theory of Elasticity*, McGraw-Hill Book Company, Auckland, 122-125(1988).

Appendix I

The Analysis of Photocarrier in Stress Measurement

This appendix describes the analysis of photocarrier fringes in a circular polariscope.

According to the Jones calculus, the Jones vector $\vec{a}'$ of the light beam that emerges from the analyser (as shown in Fig. 8.7) is

$$\vec{a}' = \vec{P}_0\ \vec{Q}_{-45}\ \vec{C}_0(\delta_1)\ \vec{R}_\alpha(\delta)\ \vec{Q}_{45}\ \vec{P}_{90} \tag{I.1}$$

For the dark-field polariscope, we have

$$\begin{aligned}\vec{a}' = &\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \frac{i+1}{2} \begin{bmatrix} 1 & -i \\ -i & 1 \end{bmatrix} \begin{bmatrix} e^{i\delta_1} & 0 \\ 0 & 1 \end{bmatrix} \\ &\begin{bmatrix} e^{i\delta}\cos^2\alpha + \sin^2\alpha & (e^{i\delta}-1)\sin\alpha\cos\alpha \\ (e^{i\delta}-1)\sin\alpha\cos\alpha & e^{i\delta}\sin^2\alpha + \cos^2\alpha \end{bmatrix} \\ &\frac{i+1}{2} \begin{bmatrix} 1 & i \\ i & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix}\end{aligned} \tag{I.2}$$

Let $x_1 = e^{i\delta}\cos^2\alpha + \sin^2\alpha$, $x_2 = (e^{i\delta}-1)\sin\alpha\cos\alpha$, $x_3 = e^{i\delta}\sin^2\alpha + \cos^2\alpha$, $x_4 =$

$e^{i\delta_1}$. By substituting x_1, x_2, x_3 and x_4 into Eq. I.2, we have

$$\begin{aligned}\vec{a}' &= \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \frac{i+1}{2} \begin{bmatrix} 1 & -i \\ -i & 1 \end{bmatrix} \begin{bmatrix} x_4 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 & x_2 \\ x_2 & x_3 \end{bmatrix} \frac{i+1}{2} \begin{bmatrix} 1 & i \\ i & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ &= \frac{i}{2} \begin{bmatrix} x_1\, x_4\, i + x_2\, x_4 + x_2 - x_3\, i \\ 0 \end{bmatrix} \\ &= \frac{1}{2} \begin{bmatrix} X \\ 0 \end{bmatrix} \end{aligned} \tag{I.3}$$

Therefore, the X can be simplified as follows,

$$\begin{aligned} X &= x_3 - x_1\, x_4 + i\,(x_2 + x_2\, x_4) \\ &= e^{i\delta} \sin^2\alpha + \cos^2\alpha - (e^{i\delta}\cos^2\alpha + \sin^2\alpha)\, e^{i\delta_1} \\ &\quad + i\,((e^{i\delta} - 1)\sin\alpha\cos\alpha + (e^{i\delta} - 1)\sin\alpha\cos\alpha\, e^{i\delta_1}) \\ &= e^{i\,\delta_1}\,(e^{i\,(\delta-\delta_1)} - 1)\,(\sin^2\alpha + i\,\sin\alpha\cos\alpha) \\ &\quad - (e^{i\,(\delta+\delta_1)} - 1)\,(\cos^2\alpha - i\,\sin\alpha\cos\alpha) \end{aligned} \tag{I.4}$$

The intensity of the fringe pattern expressed by $\vec{a}'$ is

$$\begin{aligned} I &= \vec{a}'^{*}\,\vec{a}' \\ &= \frac{1}{4}\,[e^{i\,\delta_1}\,(e^{i\,(\delta-\delta_1)} - 1)\,(\sin^2\alpha - i\,\sin\alpha\,\cos\alpha) \\ &\quad - (e^{i\,(\delta+\delta_1)} - 1)\,(\cos^2\alpha + i\,\sin\alpha\cos\alpha] \\ &\quad [e^{-i\delta_1}\,(e^{-i\,(\delta-\delta_1)} - 1)\,(\sin^2\alpha - i\,\sin\alpha\cos\alpha) \\ &\quad - (e^{-i\,(\delta+\delta_1)} - 1)\,(\cos^2\alpha + i\,\sin\alpha\cos\alpha)] \\ &= \frac{1}{4}\,[(e^{-i\,(\delta-\delta_1)} - 1)\,(e^{i\,(\delta-\delta_1)} - 1)\,(\sin^4\alpha + \sin^2\alpha\cos^2\alpha) \\ &\quad - e^{-i\delta_1}\,(e^{-i\,(\delta-\delta_1)} - 1)\,(e^{i\,(\delta+\delta_1)} - 1) \end{aligned}$$

$$(\sin^2\alpha - i\ \sin\alpha\cos\alpha)\ (\cos^2\alpha - i\ \sin\alpha\cos\alpha)$$
$$-e^{-i\delta_1}\ (e^{-i\ (\delta+\delta_1)} - 1)\ (e^{i\ (\delta-\delta_1)} - 1)$$
$$(\sin^2\alpha + i\ \sin\alpha\cos\alpha)\ (\cos^2\alpha + i\ \sin\alpha\cos\alpha)$$
$$+ (e^{-i\ (\delta+\delta_1)} - 1)\ (e^{i\ (\delta+\delta_1)} - 1)\ (\cos^4\alpha + \sin^2\alpha\cos^2\alpha)] \qquad (I.5)$$

By using values y_1, y_2, y_3 and y_4, the Eq. I.5 can be simplified as

$$I = \frac{1}{4}\ [y_1 - y_2 - y_3 + y_4] \qquad (I.6)$$

where

$$y_1 = (e^{-i\ (\delta-\delta_1)} - 1)\ (e^{i\ (\delta-\delta_1)} - 1)\ (\sin^4\alpha + \sin^2\alpha\cos^2\alpha)$$
$$= 4\ \sin^2(\frac{\delta - \delta_1}{2})\ \sin^2\alpha \qquad (I.7)$$

$$y_2 = e^{-i\delta_1}\ (e^{-i\ (\delta-\delta_1)} - 1)\ (e^{i\ (\delta+\delta_1)} - 1)$$
$$(\sin^2\alpha - i\ \sin\alpha\cos\alpha)\ (\cos^2\alpha - i\ \sin\alpha\cos\alpha)$$
$$= -2\ i\ (\cos\delta_1 - \cos\delta)\ \sin\alpha\ \cos\alpha \qquad (I.8)$$

$$y_3 = e^{-i\delta_1}\ (e^{-i\ (\delta+\delta_1)} - 1)\ (e^{i\ (\delta-\delta_1)} - 1)$$
$$(\sin^2\alpha + i\ \sin\alpha\cos\alpha)\ (\cos^2\alpha + i\ \sin\alpha\cos\alpha)$$
$$= i\ 2\ (\cos\delta_1 - \cos\delta)\ \sin\alpha\ \cos\alpha \qquad (I.9)$$

$$y_4 = (e^{-i\ (\delta+\delta_1)} - 1)\ (e^{i\ (\delta+\delta_1)} - 1)\ (\cos^4\alpha + \sin^2\alpha\cos^2\alpha)$$

$$= 4 \sin^2(\frac{\delta + \delta_1}{2}) \cos^2 \alpha \tag{I.10}$$

By taking into account Eqs. I.7, I.8, I.9, and I.10, the intensity of the fringe pattern is

$$\begin{aligned} I &= \frac{1}{4} [y_1 - y_2 - y_3 + y_4] \\ &= \sin^2(\frac{\delta - \delta_1}{2}) \sin^2 \alpha + \sin^2(\frac{\delta + \delta_1}{2}) \cos^2 \alpha \end{aligned} \tag{I.11}$$

When $\alpha = 0$, i.e. the fast axis of model disk is along x axis, the Eq. I.11 becomes

$$I = \sin^2(\frac{\delta + \delta_1}{2}) = 1 - \cos(\delta + \delta_1) \tag{I.12}$$

Appendix II

Stress in a Circular Disk

This appendix describes the analysis of stress in a circular disk.

In Fig. 8.32, two equal and opposite forces P are applied along a diameter AB. At any point M of the circumference of the disk, the compression stresses in r and r_1 directions are equal to $(2 * P/\pi)(\cos\theta/r)$ and $(2 * P/\pi)(\cos\theta_1/r_1)$, respectively. Since r and r_1 are perpendicular to each other and

$$\frac{\cos\theta}{r} = \frac{\cos\theta_1}{r_1} = \frac{1}{d} \tag{II.1}$$

where d is the diameter of the disk. Therefore, the two principal stresses at M are two equal compression stresses of magnitude $2P/\pi d$. If the boundary of the disk is free from external forces, the stress at any point is therefore by superimposing a uniform tension, in the plane of the disk, of magnitude $2P/\pi d$ on the two simple radial stress distributions.

Consider the stress on the horizontal diameter CD at point N. The normal stresses σ_x and σ_y produced by the two equal radial compressions are [191]

$$\sigma_x = 2\sigma_r \sin^2\theta_{AN} = -\frac{4}{\pi}\frac{P}{r_{AN}}\sin^2\theta_{AN}\cos\theta_{AN} \tag{II.2}$$

$$\sigma_y = 2\sigma_r \cos^2\theta_{AN} = -\frac{4}{\pi}\frac{P}{r_{AN}}\cos^2\theta_{AN}\cos\theta_{AN} \tag{II.3}$$

Superimposing on these stresses the uniform tension $2P/\pi d$, the total normal stress on the horizontal plane at N are

$$\sigma_x = -\frac{4P}{\pi}\,\frac{\sin^2\theta_{AN}\,\cos\theta_{AN}}{r_{AN}} + \frac{2P}{\pi d} \tag{II.4}$$

$$\sigma_y = -\frac{4P}{\pi}\,\frac{\cos^2\theta_{AN}\,\cos\theta_{AN}}{r_{AN}} + \frac{2P}{\pi d} \tag{II.5}$$

From geometry in Fig. 8.32,

$$\cos\theta_{AN} = \frac{d}{\sqrt{d^2 + 4\,x^2}} \tag{II.6}$$

$$\sin\theta_{AN} = \frac{2\,x}{\sqrt{d^2 + 4\,x^2}} \tag{II.7}$$

Thus, σ_x and σ_y along the horizontal plane are given by

$$\sigma_x = \frac{2P}{\pi d}\,[1 - \frac{16x^2d^2}{(d^2 + 4x^2)^2}] \tag{II.8}$$

$$\sigma_y = \frac{2P}{\pi d}\,[1 - \frac{4d^4}{(d^2 + 4x^2)^2}] \tag{II.9}$$

The maximum compressive stresses along the diameter CD are at the centre of the disk, where

$$\sigma_x = \frac{2P}{\pi d}$$

$$\sigma_y = -\frac{6P}{\pi d}$$

From symmetry it can be seen that there will be no shearing stress on the diameter CD.

Consider the case of the stresses along the diameter AB, where $x = 0, \cos\theta =$

$1, \sin\theta = 0$, then

$$\sigma_x = \frac{2P}{\pi d} \tag{II.10}$$

$$\sigma_y = \frac{2P}{\pi}\left(\frac{1}{d} - \frac{1}{d-r} - \frac{1}{r}\right) \tag{II.11}$$

Appendix III

The List of Programs

photo.c

```
#include <stdio.h>
#include <values.h>
#include <ctype.h>
#include <math.h>
#include <string.h>
#include <time.h>
#include <fcntl.h>
#include <alloc.h>
#include <stdarg.h>
#include <process.h>
#include <assert.h>
#include <io.h>
#include <dos.h>
#include <stdlib.h>
#include "photo.h"
#include "fringe.h"
#include "prepro.h"
#include "drivepip.h"
#include "cursor.h"
#include "msdos.h"
#include "salloc.h"
#include "fr4.h"
#include "memory.h"

#define MAX_MARK 30

#define TICKS ((double) CLK_TCK)

/* variables for FFT and phase steppint tan fringe computation */
FILE*  in[5];
FILE*  out_tan;
FILE*  out_mod;
float  scan[ 1025  ];
float  window_buffer[ 512 ];
float  carrier_freq;
char   prefix[255];
double scale_tan;
double tan_offset;
double critical_size;
double n_scan_lines;
double top, bottom;
int    first_time;
int    n_scans_read;
int    pip_video_gain;
int    pip_video_offset;
int    xp, yp;

/* define variable for the motor */
int count;
int control=435;
int porta=432;
int portb=433;
int portc=434;

void QWP(void);
void return_QWP(void);

/* define identity transformation matrix */

double sobel_threshold = 80.0;
double thres_percent = 50.0;

int crtc_status = 0x126c;

int minimum;
int maximum;
```

photo.c

```
char hex[] = { "0123456789ABCDEF" };

void Define_Menus();
void Snap_Shot();
void Cont_Grab();

int nleft, nright, ntop, nbottom;
int graphdriver, graphmode;

char* load_buffer;
char* load_buffer1;
char* load_buffer2;
unsigned long bufsiz;

int current_quadrant;
int pip_present = 0;

int num_frames;

MENU MA;
MENU MB;
MENU MC;
MENU MD;
MENU MF;

int last_point_marker;
int last_text_marker;
extern int dat;

IMAGE Create_Image_Buffer()
{
   IMAGE p;
   int i;

   p = (char**) malloc( 512 * (sizeof(char*)) );

   if ( p != NULL ) {

   for( i = 0 ; i < 512; i++ )
      {
        p[i] = (char*) malloc( 512 * (sizeof(char) ) );
        if ( p[i] == NULL )
           { printf("Malloc failed in image buffer allocation\n");
             exit(1); }
      }
   } else { printf("Malloc failed in image buffer allocation\n");
            exit(1); }

  return( p );
}

void Free_Image_Buffer( p )
IMAGE p;
{
   int i;

   if ( p != NULL ) {

   for( i = 0 ; i < 512; i++ )
        free( p[i] );

      free( p );
   }
}
```

photo.c

```
void Display( frame, image )
IMAGE image;
int frame;
{
  Write_Pip_Frame( frame, image );
}

main()
{
  char number[80];
  char c;

  bufsiz = 32768;

  load_buffer = (char*) malloc( bufsiz  );
  load_buffer1 = (char*) malloc( bufsiz );
  load_buffer2 = (char*) malloc( bufsiz );

  if ( load_buffer == NULL)
     {
       printf("Couldn't allocate load buffer!\n");
       exit(1);
     }

  Initialise_Pop_Up_Menus();

  Define_Menus();

  Pop_Print("Do you have the PIP board connected (Y/N) ?", 0 );

  do {
  c = toupper(getch());
     } while ( c != 'Y' && c != 'N' );

  Clear_Print();

  if ( c == 'Y' ) {
                    Pip_Init_European();
                    Pip_Set_Input_Lut();
                    Pip_Set_Output_Lut();
                    pip_present = 1;
                    current_quadrant = 0;
                    Pip_Quad( current_quadrant );
                  }
  else pip_present = 0;

  pip_video_gain   = 196;
  pip_video_offset = 128;

  Pip_Set_Video_Gain( pip_video_gain );
  Pip_Set_Video_Offset( pip_video_offset );

  for( ;; ) Main_Menu( &MA );

}

int Get_Next_Char()
{
  return( getch() );
}

void Pause()
{
   int i;

   for( i = 0; i < 5000; i++ );
```

photo.c

```
}

void Set_Output_Lut_Black_And_White()
{
  int i;

  outportb( creg0, 0x0 );
  for( i = 0; i < 256 ; i++ )
     {
       outportb( lvram, (char) i );
       outportb( olutr, (char) i );
       outportb( olutg, (char) i );
       outportb( olutb, (char) i );
     }
}

void Set_Output_Lut_Colour()
{
  int i;
  float cstep;

  cstep = 255 / 4;

  outportb( creg0, 0x0 );
  for( i = 0; i < 128 ; i++ )
     {
       outportb( lvram, (char) (i)      );
       outportb( olutr, (char) (i << 1) );
       outportb( olutg, (char) (i << 1) );
       outportb( olutb, (char) (i << 1) );
     }

  outportb( creg0, 0x0 );
  for( i = 128; i < 256 ; i++ )
     {
       outportb( lvram, (char) (i)      );
       outportb( olutr, (char) (((i-128)/32)*cstep) );
       outportb( olutg, (char) ((((i-128)/5)%5))*cstep);
       outportb( olutb, (char) (((i-128)%5)*cstep) );
     }
}

void Invert_Video()
{
  IMAGE image;
  int x,y;
  char* iline;

   if ( ( iline = Pop_Input( "Enter the frame number") )  != NULL )
     { result_frame_number = atoi( iline );

  image  = Create_Image_Buffer();
  Pop_Print( "Reading Frame", 0 );
  Read_Pip_Frame( result_frame_number, image );
  Clear_Print();
  Pop_Print( "Inverting Video", 0 );

  for( y = 0; y < 512; y++ )
     for( x = 0; x < 512; x++ )
        {
          image[x][y] = (unsigned char) ( 255 - ((int) image[x][y] ) );
        }

  Clear_Print();
  Pop_Print("Displaying Inverted Image",0);
  Display( result_frame_number, image );
```

photo.c

```
  Clear_Print();
  Pop_Print( "Deallocating PC memory", 0 );
  Free_Image_Buffer( image );
  Clear_Print();
  }
}

void Sobel( ystep )
int ystep;
{
  int x,y;
  double delta1, delta2;
  double magnitude;
  IMAGE image;
  int m;
  int ystart;
  char* in;

  ystart = 1;

  image = Create_Image_Buffer();

  Pop_Print( "Reading Frame", 0 );
  Read_Pip_Frame( 0, image );
  Clear_Print();

  Pip_Clear_Input_Lut();
  Pip_Snap( 0 );
  Pip_Set_Input_Lut();

  for( y = ystart + ystep; y < (512-ystep); y+= ystep )
     for( x = 1; x < 511; x++ )
        {
          delta1 = (double) (
          ( image[x+1][y-ystep] + 2 * image[x+1][y] + image[x+1][y+ystep] ) -
          ( image[x-1][y-ystep] + 2 * image[x-1][y] + image[x-1][y+ystep] )
                             );

          delta2 = (double) (
          ( image[x-1][y+ystep] + 2 * image[x][y+ystep] + image[x+1][y+ystep] ) -
          ( image[x-1][y-ystep] + 2 * image[x][y-ystep] + image[x+1][y-ystep] )
                             );

          magnitude = sqrt( delta1 * delta1 + delta2 * delta2 );

          m = (int) ( ( ( magnitude / 1442.0 ) * 255.0 ) + 0.5);

          if ( m > 255 ) m = 255;

          if ( magnitude > sobel_threshold ) Pip_Plot( 0, x, y, m  );
        }

  Free_Image_Buffer( image );
}


void Sobel_Only_Odd_Rasters()
{
  Sobel(2);
}

void Sobel_Edge_Detect()
{
  Sobel(1);
}
```

photo.c

```
void Exit()
{
  int c;

  do {

  Pop_Print( "Really exit ?", 0 );

  c = getch();
  c = toupper( c );

  } while ( c != 'Y' && c != 'N' );

  Clear_Print();

  if ( c == 'Y' ) {
                     fcloseall();
                        Set_Text_Page( 1 );
                     exit( 0 );
                  }
}

void View( n )
int n;
{
  Pip_Quad( n );
  current_quadrant = n;
}

void View0()
{ View(0); }

void View1()
{ View(1); }

void View2()
{ View(2); }

void View3()
{ View(3); }

void Isochromatic_Phase()
{
  int x,y;
  char* iline;
  int i;
  int i0,i1,i2,i3,i4;
  int red;
  double argument;
  double diff1,diff2,diff3;
  int choice;
  int pa;


  if ( ( iline = Pop_Input( "Enter the frame number") )  != NULL )
     { result_frame_number = atoi( iline );

  if ( ( iline = Pop_Input( "Enter choice(1, 2, 3 or 4)") )  != NULL )
     { choice = atoi( iline );

  if (choice == 1)
  {
        if ( ( iline = Pop_Input( "Enter parameter(>0)") )  != NULL )
          { pa = atoi( iline );
```

photo.c

```
        m = ( 254.0 / ( 2.0 * PIF ) );
        c = PIF;

        error = 0;

        if ( ( iline = Pop_Input("Enter file prefix") ) != NULL )
           {
            for( i = 0; i < 2; i++ )
             {
               sprintf( fname, "%s.%d", iline , i+1 );

               in[ i ] = fopen( fname, "rb" );

               if ( in [ i ] == NULL ) { error = 1; break; }
             }

            if ( error == 0 )
              {
               sprintf( fname, "%s.tan", iline );

               out = fopen( fname, "wb" );

               if ( out != NULL )
                 {

                   for( y = 0; y < PIXY; y++ )
                     for( x = 0; x < PIXX; x++ )
                       {
                         i0 = getc( in[0] );
                         i1 = getc( in[1] );

                         red = 0;

                         top    = sqrt((double) i0);

                         bottom = sqrt((double) i1);

                        if ( bottom != 0 )
                           wrapped_phase = pa * atan2( (double) top, (double) bottom)
;
                        else
                        { wrapped_phase = ( PIF / 2.0 ); red = 1; }

                    if ( red != 1 )
                      putc( (int) ( ( m * ( wrapped_phase + c ) ) + 0.5 ), out );
                    else putc( 254, out );

                        }

                  for( i = 0; i < 2; i++ ) fclose( in[i] );

                  fclose( out );

                  Load( result_frame_number, fname );

                 }
              }
           }
        }
}
     /* m = ( 254.0 / ( PIF ) );
      c = 0.0;

      error = 0;

      if ( ( iline = Pop_Input("Enter file prefix") ) != NULL )
```

photo.c

```
        {
           for( i = 0; i < 3; i++ )
             {
               sprintf( fname, "%s.%d", iline , i+1 );

               in[ i ] = fopen( fname, "rb" );

               if ( in [ i ] == NULL ) { error = 1; break; }
             }

            if ( error == 0 )
              {
               sprintf( fname, "%s.tan", iline );

               out = fopen( fname, "wb" );

               if ( out != NULL )
                 {
                   for (y = 0; y < PIXY; y++)
                      for (x = 0; x < PIXX; x++)
                       {
                         i0 = getc( in[0] );
                         i1 = getc( in[1] );
                         i2 = getc( in[2] );

                         top    = i2 - i1;
                         bottom = i2 + i1 - ( 2 * i0 );


                         if ( bottom != 0 )
                          {
                            if ( abs( top ) <= abs( bottom ) ) {

                           wrapped_phase = acos(( (double) top ) / ( (double) bottom
));

                           }
                            else {
                                    if ( (top/bottom) < 0 )

                                    { wrapped_phase = PIF; }

                                    else if( (top/bottom) > 0) {
                                    wrapped_phase = 0.0;
                                         }
                                 }
                          }
                         else
                           wrapped_phase = 0.0;


                         putc( (int) ( ( m * ( wrapped_phase + c ) ) + 0.5 ), out );
                       }

                   for( i = 0; i < 3; i++ ) fclose( in[i] );

                   fclose( out );

                   Load( result_frame_number, fname );
                 }
              }
        }
      } */

   else if (choice == 2)
   {
```

photo.c

```
m = ( 254.0 / ( 2.0 * PIF ) );
c = PIF;

error = 0;

if ( ( iline = Pop_Input("Enter file prefix") ) != NULL )
   {
    for( i = 0; i < 5; i++ )
     {
       sprintf( fname, "%s.%d", iline , i+1 );

       in[ i ] = fopen( fname, "rb" );

       if ( in [ i ] == NULL ) { error = 1; break; }
     }

    if ( error == 0 )
      {
       sprintf( fname, "%s.tan", iline );

       out = fopen( fname, "wb" );

       if ( out != NULL )
         {

           for( y = 0; y < PIXY; y++ )
             for( x = 0; x < PIXX; x++ )
               {
                 i0 = getc( in[0] );
                 i1 = getc( in[1] );
                 i2 = getc( in[2] );
                 i3 = getc( in[3] );
                 i4 = getc( in[4] );

                 diff1 = (double) (i0-i1);
                 diff2 = (double) (i1-i2);
                 diff3 = (double) (i2-i0);

                 red = 0;

                 top    = ( 2.0 * sqrt((double)2.0) *
                    sqrt(
                         (double)(diff1*diff1+diff2*diff2+diff3*diff3)
                        ) );

                 bottom = 3.0 * (double) ( i3 -i4);

               if ( bottom != 0 )
                  wrapped_phase = atan2( (double) top, (double) bottom);
               else
               { wrapped_phase = ( PIF / 2.0 ); red = 1; }

         if ( red != 1 )
           putc( (int) ( ( m * ( wrapped_phase + c ) ) + 0.5 ), out );
         else putc( 254, out );

                }

       for( i = 0; i < 5; i++ ) fclose( in[i] );

       fclose( out );

       Load( result_frame_number, fname );

       }
```

photo.c

```
                }
            }
        }

else if(choice == 3)
{
      m = ( 254.0 / ( 2.0 * PIF ) );
      c = PIF;

      error = 0;

      if ( ( iline = Pop_Input("Enter file prefix") ) != NULL )
        {
          for( i = 0; i < 3; i++ )
            {
              sprintf( fname, "%s.%d", iline , i+1 );

              in[ i ] = fopen( fname, "rb" );

              if ( in [ i ] == NULL ) { error = 1; break; }
            }

           if ( error == 0 )
             {
              sprintf( fname, "%s.tan", iline );

              out = fopen( fname, "wb" );

              if ( out != NULL )
                {
                  for( y = 0; y < PIXY; y++ )
                    for( x = 0; x < PIXX; x++ )
                      {
                        i0 = getc( in[0] );
                        i1 = getc( in[1] );
                        i2 = getc( in[2] );

                        top    = (double) sqrt(3.0) * (i2 - i0);
                        bottom = i2 + i0 - ( 2 * i1 );

                        if ( bottom != 0 )
                          wrapped_phase = one_over_four*atan2( (double) top , (doub
le) bottom);
                        else
                          wrapped_phase = one_over_four*( PIF / 2.0 );

                          putc( (int) ( ( m * ( wrapped_phase + c ) ) + 0.5 ), out
);

                       /*  principle[1] = (float) wrapped_phase;  */

                /*         fwrite( &principle, sizeof( float ), 1, out ); */
                      }

                  for( i = 0; i < 3; i++ ) fclose( in[i] );

                  fclose( out );

                  Load( result_frame_number, fname );
                }
        }
    }
}

  else if(choice == 4)
  {
```

photo.c

```
        m = ( 254.0 / ( 2.0 * PIF ) );
        c = PIF;

        error = 0;

        if ( ( iline = Pop_Input("Enter file prefix") ) != NULL )
           {
            for( i = 0; i < 4; i++ )
             {
               sprintf( fname, "%s.%d", iline , i+1 );

               in[ i ] = fopen( fname, "rb" );

               if ( in [ i ] == NULL ) { error = 1; break; }
             }

            if ( error == 0 )
              {
               sprintf( fname, "%s.tan", iline );

               out = fopen( fname, "wb" );

               if ( out != NULL )
                 {
                   for( y = 0; y < PIXY; y++ )
                     for( x = 0; x < PIXX; x++ )
                       {
                         i0 = getc( in[0] );
                         i1 = getc( in[1] );
                         i2 = getc( in[2] );
                         i3 = getc( in[3] );

                         top    = i3 - i1;
                         bottom = i2 - i0;

                       if ( bottom != 0 )
                         wrapped_phase = one_over_four*atan2( (double) top, (double
) bottom);
                       else
                         wrapped_phase = one_over_four*( PIF / 2.0 );

                    putc( (int) ( ( m * ( wrapped_phase + c ) ) + 0.5 ), out );
                       }

               for( i = 0; i < 4; i++ ) fclose( in[i] );

               fclose( out );

               Load( result_frame_number, fname );
               }
             }
           }
        }
     }
  }
}


void Isopachic_Phase()
{
  int x,y;
  char* iline;
  int i;
  int i0,i1,i2,i3;

  if ( ( iline = Pop_Input( "Enter the frame number") )  != NULL )
```

photo.c

```
    { result_frame_number = atoi( iline );

      m = ( 254.0 / ( 2.0 * PIF ) );
      c = PIF;

      error = 0;

      if ( ( iline = Pop_Input("Enter file prefix") ) != NULL )
        {
          for( i = 0; i < 3; i++ )
            {
              sprintf( fname, "%s.%d", iline , i+1 );

              in[ i ] = fopen( fname, "rb" );

              if ( in [ i ] == NULL ) { error = 1; break; }
            }

          if ( error == 0 )
            {
              sprintf( fname, "%s.tan", iline );

              out = fopen( fname, "wb" );

              if ( out != NULL )
                {
                  for( y = 0; y < PIXY; y++ )
                    for( x = 0; x < PIXX; x++ )
                      {
                        i0 = getc( in[0] );
                        i1 = getc( in[1] );
                        i2 = getc( in[2] );
                        i3 = getc( in[3] );

                        top    = i3 - i1;
                        bottom = i0 - i2;

                        if ( bottom != 0 )
                          wrapped_phase = (1/PIF) *  atan2( (double) top , (double)
bottom);
                        else
                          wrapped_phase = ( PIF / 2.0 );

                        putc( (int) ( ( m * ( wrapped_phase + c ) ) + 0.5 ), out );
                      }

                  for( i = 0; i < 4; i++ ) fclose( in[i] );

                  fclose( out );

                  Load( result_frame_number, fname );
                }
            }
        }
     }
  }


void Isoclinic_Phase()
{
  int x,y;
  char* iline;
  int i;
  int i0,i1,i2,i3;
  float principle[1];
```

photo.c

```
  if ( ( iline = Pop_Input( "Enter the frame number") )  != NULL )
     { result_frame_number = atoi( iline );

  if ( ( iline = Pop_Input( "Enter step numbers(3 or 4)") )  != NULL )
     { step_number = atoi( iline );

  if (step_number == 3)
    {

      m = ( 254.0 / ( 2.0 * PIF ) );
      c = PIF;

      error = 0;

      if ( ( iline = Pop_Input("Enter file prefix") ) != NULL )
        {
          for( i = 0; i < 3; i++ )
            {
              sprintf( fname, "%s.%d", iline , i+1 );

              in[ i ] = fopen( fname, "rb" );

              if ( in [ i ] == NULL ) { error = 1; break; }
            }

          if ( error == 0 )
            {
              sprintf( fname, "%s.tan", iline );

              out = fopen( fname, "wb" );

              if ( out != NULL )
                {
                  for( y = 0; y < PIXY; y++ )
                    for( x = 0; x < PIXX; x++ )
                      {
                        i0 = getc( in[0] );
                        i1 = getc( in[1] );
                        i2 = getc( in[2] );

                        top    = (double) sqrt(3.0) * (i2 - i0);
                        bottom = i2 + i0 - ( 2 * i1 );

                        if ( bottom != 0 )
                          wrapped_phase = one_over_four*atan2( (double) top , (doub
le) bottom);
                        else
                          wrapped_phase = one_over_four*( PIF / 2.0 );

                          putc( (int) ( ( m * ( wrapped_phase + c ) ) + 0.5 ), out
);

                       /*  principle[1] = (float) wrapped_phase;*/

                        /*  fwrite( &principle, sizeof( float ), 1, out ); */
                        }

                  for( i = 0; i < 3; i++ ) fclose( in[i] );

                  fclose( out );

                  Load( result_frame_number, fname );
                }
            }
        }
    }
  }
```

photo.c

```
  else if( step_number == 4)
       {
         m = ( 254.0 / ( 2.0 * PIF ) );
         c = PIF;

         error = 0;

         if ( ( iline = Pop_Input("Enter file prefix") ) != NULL )
            {
              for( i = 0; i < 4; i++ )
               {
                 sprintf( fname, "%s.%d", iline , i+1 );

                 in[ i ] = fopen( fname, "rb" );

                 if ( in [ i ] == NULL ) { error = 1; break; }
               }

              if ( error == 0 )
                {
                 sprintf( fname, "%s.tan", iline );

                 out = fopen( fname, "wb" );

                 if ( out != NULL )
                   {
                     for( y = 0; y < PIXY; y++ )
                       for( x = 0; x < PIXX; x++ )
                         {
                           i0 = getc( in[0] );
                           i1 = getc( in[1] );
                           i2 = getc( in[2] );
                           i3 = getc( in[3] );

                           top    = i3 - i1;
                           bottom = i2 - i0;

                         if ( bottom != 0 )
                           wrapped_phase = one_over_four*atan2( (double) top, (double
) bottom);
                         else
                           wrapped_phase = one_over_four*( PIF / 2.0 );

                          putc( (int) ( ( m * ( wrapped_phase + c ) ) + 0.5 ), out );
                           }

                   for( i = 0; i < 4; i++ ) fclose( in[i] );

                   fclose( out );

                   Load( result_frame_number, fname );
                 }
            }
       }
    }
  }
}

void Isoclinic_Angles()
{
  int x,y;
  char* iline;
  int i;
  int i0,i1;
  double isoclinic_result;
```

photo.c

```
int red;
double iso_min, iso_max;

if ( ( iline = Pop_Input( "Enter the frame number") )  != NULL )
    result_frame_number = atoi( iline );

    m = ( 254.0 / ( 2.0 * PIF ) );
    c = PIF;

    error = 0;

    if ( ( iline = Pop_Input("Enter file prefix") ) != NULL )
      {
        for( i = 0; i < 2; i++ )
          {
            sprintf( fname, "%s.%d", iline , i+1 );

            in[ i ] = fopen( fname, "rb" );

            if ( in [ i ] == NULL ) { error = 1; break; }
          }

         if ( error == 0 )
           {
            iso_min = (double) MAXDOUBLE;
            iso_max = (double) 0.0;

            for( y = 0; y < PIXY; y++ )
                  for( x = 0; x < PIXX; x++ )
                    {
                      i0 = getc( in[0] );
                      i1 = getc( in[1] );

                      top    = i0;
                      bottom = i1;

                      red = 0;

                      if ( bottom != 0 )
                         isoclinic_result =
                      ( ( (double) top ) / ( (double) bottom ) );

                      if ( iso_min > isoclinic_result )
                           iso_min = isoclinic_result;

                      if ( iso_max < isoclinic_result )
                           iso_max = isoclinic_result;
                    }

            for( i = 0; i < 2; i++ ) rewind( in[i] );

            sprintf( fname, "%s.tan", iline );

            out = fopen( fname, "wb" );

            if ( out != NULL )
              {
                for( y = 0; y < PIXY; y++ )
                  for( x = 0; x < PIXX; x++ )
                    {
                      i0 = getc( in[0] );
                      i1 = getc( in[1] );

                      top    = i0;
                      bottom = i1;
```

photo.c

```
                      red = 0;

                      if ( bottom != 0 )
                          isoclinic_result =
                      ( ( (double) top ) / ( (double) bottom ) );
                      else { isoclinic_result = 0.0; red = 1; }

                      if (red == 0 )
                        putc( (int) (254.0*((isoclinic_result-iso_min)/(iso_max-i
so_min))+0.5), out );
                      else
                        putc( (int) (255), out );

                    }

                for( i = 0; i < 2; i++ ) fclose( in[i] );

                fclose( out );

                Load( result_frame_number, fname );
              }
        }
    }
}


void Save_Image_Buffer()
{
   char* name;
   if ( ( name = Pop_Input( "Enter image file for save") ) != NULL )
      Save( 0, name );

}

void Run_Live()
{
  char* name1;
  char* name2;
  char string[80];
  char input1[80];
  char input2[80];
  clock_t start_time;
  clock_t end_time;
  clock_t elapsed_time;

  if ( ( name1 = Pop_Input("Enter 1st file for save") ) != NULL )
     {
         strcpy (input1, name1);

     if ( ( name2 = Pop_Input("Enter 2nd file for save") ) != NULL )
       {
         strcpy (input2, name2);

     start_time = clock();
     Pip_Snap ( 0 );
     Save (0 , input1); }
     {
     Pip_Cgrab( 0 );
     Pip_Snap ( 0 );
     end_time = clock();

     Save (0, input2);

     elapsed_time = (end_time - start_time ) / CLK_TCK;
     sprintf ( string, "Processing time took %d mins %d secs", (int) (elapsed_time/60
), (int) (elapsed_time % 60) );
```

photo.c

```
    Pop_Print (string, 0);

      }
    }
}

void Load_Image_Buffer()
{
   char* name;
   char* fn;
   int n;

   if ( ( fn = Pop_Input( "Enter the frame number") )  != NULL )
      n = atoi( fn );

   if ( ( name = Pop_Input( "Enter image file for load") )  != NULL )
      Load( n, name );

}

void Load_16Bit_Image_Buffer()
{
   char* name;
   char* fn;
   int n;

   if ( ( fn = Pop_Input( "Enter the frame number") )  != NULL )
      n = atoi( fn );

   if ( ( name = Pop_Input( "Enter image file for load") )  != NULL )
      Load_16bit( n, name );

}

void Cont_Grab()
{
   Pip_Cgrab( 0 );
}

void Snap_Shot()
{
   Pip_Snap( 0 );
}


#define SWAP(a,b) tempr=(a);(a)=(b);(b)=tempr

void Fourn(data,nn,ndim,isign)
float data[];
int nn[],ndim,isign;
{
        int i1,i2,i3,i2rev,i3rev,ip1,ip2,ip3,ifp1,ifp2;
        int ibit,idim,k1,k2,n,nprev,nrem,ntot;
        float tempi,tempr;
        double theta,wi,wpi,wpr,wr,wtemp;

        ntot=1;
        for (idim=1;idim<=ndim;idim++)
                ntot *= nn[idim];
        nprev=1;
        for (idim=ndim;idim>=1;idim--) {
                n=nn[idim];
                nrem=ntot/(n*nprev);
                ip1=nprev << 1;
                ip2=ip1*n;
                ip3=ip2*nrem;
```

photo.c

```
                i2rev=1;
                for (i2=1;i2<=ip2;i2+=ip1) {
                        if (i2 < i2rev) {
                                for (i1=i2;i1<=i2+ip1-2;i1+=2) {
                                        for (i3=i1;i3<=ip3;i3+=ip2) {
                                                i3rev=i2rev+i3-i2;
                                                SWAP(data[i3],data[i3rev]);
                                          SWAP(data[i3+1],data[i3rev+1]);
                                        }
                                }
                        }
                        ibit=ip2 >> 1;
                        while (ibit >= ip1 && i2rev > ibit) {
                                i2rev -= ibit;
                                ibit >>= 1;
                        }
                        i2rev += ibit;
                }
                ifp1=ip1;
                while (ifp1 < ip2) {
                        ifp2=ifp1 << 1;
                        theta=isign*6.28318530717959/(ifp2/ip1);
                        wtemp=sin(0.5*theta);
                        wpr = -2.0*wtemp*wtemp;
                        wpi=sin(theta);
                        wr=1.0;
                        wi=0.0;
                        for (i3=1;i3<=ifp1;i3+=ip1) {
                                for (i1=i3;i1<=i3+ip1-2;i1+=2) {
                                        for (i2=i1;i2<=ip3;i2+=ifp2) {
                                                k1=i2;
                                                k2=k1+ifp1;
                                        tempr=wr*data[k2]-wi*data[k2+1];
                                        tempi=wr*data[k2+1]+wi*data[k2];
                                                data[k2]=data[k1]-tempr;
                                                data[k2+1]=data[k1+1]-tempi;
                                                data[k1] += tempr;
                                                data[k1+1] += tempi;
                                        }
                                }
                                wr=(wtemp=wr)*wpr-wi*wpi+wr;
                                wi=wi*wpr+wtemp*wpi+wi;
                        }
                        ifp1=ifp2;
                }
                nprev *= n;
        }
}

#undef SWAP

void Multiply_Scan_By_Filter_Window( scan )
float* scan;
{
  int k;
  int i;

  i = 1;

  for( k = 0; k < PIXX; k++ )
      {
        scan[i] *= window_buffer[k]; i += 2;
      }
}
```

photo.c

```
void Write_Scan_Data( fft_data, output_file )
float* fft_data;
char*  output_file;
{
  FILE* out;
  int x;
  int i;

  out = fopen( output_file, "w" );

  if ( out != NULL )
     {
       i = 1;
       for( x = 0; x < PIXX; x++ )
          {
            fprintf( out, "%f\n", fft_data[i] );
            i += 2;
          }

       fclose( out );
     }
  else { Pop_Error( "Couldn't open scan output file\n" ); }
}

void Compute_Filter_Window( window_buffer )
float* window_buffer;
{
  double n;
  int    k;
  int    i;
  float wk;

  n = (double) (PIXX/2);

  for( i = 0; i < PIXX; i++ )
     {
       k = i - (PIXX/2);

       wk = (float)
        (
                (
                        1.0 +
                        cos( (double) (PIF * ((double) k) / n ) )
                )
                        / 2.0
       );


       window_buffer[i] = wk;

     }
}

void Write_Window_Data( window_buffer, output_file )
float* window_buffer;
char*  output_file;
{
  FILE* out;
  int i;

  out = fopen( output_file, "w" );

  if ( out != NULL )
     {
       for( i = 0 ; i < PIXX; i++ )
            fprintf( out, "%f\n", window_buffer[i] );
```

photo.c

```
       fprintf( out, "\n" );

       fclose( out );
     }
  else { Pop_Error( "Couldn't open window_buffer output file\n" ); }
}

void Initialise_FFT_Tan_Fringe_Computation()
{
   char string[256];

   scale_tan  = 255.0/(2.0*PIF);
   tan_offset = PIF;

   Compute_Filter_Window( window_buffer );

   sprintf( string, "%s.1", prefix );

   Pop_Print( string, 1 );

   in[0] = fopen( string, "rb" );

   if ( in[0] == NULL )
      {
        sprintf( string, "Couldn't open image file" );
        Pop_Error(string);
      }

   sprintf( string, "%s.tan", prefix );

   out_tan = fopen( string, "wb" );

   if ( out_tan == NULL )
      {
        sprintf( string, "Couldn't open tan fringe output file" );
        Pop_Error(string);
      }
}

void Perform_One_Dimensional_FFT_On_Scan_Line( scan )
float* scan;
{
  int nn[2];

  nn[1] = PIXX;

  Fourn(scan,nn,1,1);
}

void Invert_One_Dimensional_FFT_On_Scan_Line( scan )
float* scan;
{
  int nn[2];

  nn[1] = PIXX;

  Fourn(scan,nn,1,-1);
}

int Shift_Side_Lobe_To_Origin( scan, carrier_frequency )
float* scan;
float  carrier_frequency;
{
  int x;
  int fd;
```

photo.c

```
  int twice_fd;
  int r1,i1,r2,i2;
  int src,dest;
  int side_lobe_cut_off_freq;
  float d;

  fd = (int) ( (PIXX * carrier_frequency) + 0.5 );

  side_lobe_cut_off_freq = fd / 2;
/*    Locate_Side_Lobe_Cut_Off_Freq( scan, (unsigned) 2, fd, 0 ); */

  twice_fd = fd*2;

  for( x = (PIXX/2); x < PIXX; x++ )
     {
       r1 = 1 + x * 2;
       i1 = r1+1;

       scan[r1] = 0.0;
       scan[i1] = 0.0;
     }

  for( x = 0; x < side_lobe_cut_off_freq; x++ )
     {
       r1 = 1 + x * 2;
       i1 = r1+1;

       scan[r1] = 0.0;
       scan[i1] = 0.0;
     }

  for( src = 0; src < (PIXX/2); src++ )
     {
       dest = src - fd;

       if ( dest < 0 ) dest += (PIXX);

       r1 = 1 + src * 2;
       i1 = r1+1;

       r2 = 1 + dest * 2;
       i2 = r2+1;

       scan[r2] = scan[r1];
       scan[i2] = scan[i1];
     }

  for( x = ((PIXX/2)-fd); x < (PIXX/2); x++ )
     {
       r1 = 1 + x * 2;
       i1 = r1+1;

       scan[r1] = 0.0;
       scan[i1] = 0.0;
     }

  return( side_lobe_cut_off_freq );
}

void Compute_FFT_Row_Of_Tan_Fringes( ystart, out_tan )
int ystart;
FILE* out_tan;
{
  int y_start;
  int i,j;
  double real,imag;
```

photo.c

```
  int  side_lobe_cut;
  int  xp, yp;
  double wrapped_phase;

            yp = ystart;

            i = 1;

            for( xp = 0; xp < PIXX; xp++ )
               {
                 scan[i] = (float) getc(in[0]);
                 i++;
                 scan[i] = 0.0;
                 i++;
               }

            Multiply_Scan_By_Filter_Window( scan );

            Perform_One_Dimensional_FFT_On_Scan_Line( scan );

            side_lobe_cut =
               Shift_Side_Lobe_To_Origin( scan, carrier_freq );

            Invert_One_Dimensional_FFT_On_Scan_Line( scan  );

            for( xp = 0; xp < PIXX; xp++ )
               {
                 i = 2 * xp + 1;
                 j = i+1;

                 real = scan[i];
                 imag = scan[j];

                 top    = imag;

                 bottom = real;

                 if ( bottom == 0.0 )
                    wrapped_phase = (float) ( PIF / 2.0 );
                 else
                    wrapped_phase = (float) (atan2(top,bottom));

                 putc( (unsigned char) ( (scale_tan*
                        (wrapped_phase+tan_offset)) + 0.5 ), out_tan );

               }
}

void Close_FFT_Tan_Fringe_Computation()
{
   fclose( in[0] );

   fclose( out_tan );
}

void FFT_Method()
{
  int y;
  char* iline;
  char  string[80];


  if ( ( iline = Pop_Input( "Enter the frame number") )  != NULL )
     { result_frame_number = atoi( iline );

  if ( ( iline = Pop_Input("Enter carrier freq") ) != NULL )
```

photo.c

```
    {
      carrier_freq = atof( iline );

  if ( ( iline = Pop_Input("Enter file prefix") ) != NULL )
    {
      strcpy( prefix, iline );

      Pop_Print( prefix, 1 );

      Initialise_FFT_Tan_Fringe_Computation();


  for( y = 0; y < PIXY; y++ )
    {
      sprintf( string, "Processing Scan Line %d", y );
      Pop_Print( string, 0 );
      Compute_FFT_Row_Of_Tan_Fringes( y, out_tan );
    }

  Clear_Print();

  Close_FFT_Tan_Fringe_Computation();

  sprintf( string, "%s.tan", iline );

  Load( result_frame_number, string );
                }
            }
      }

}

void Do_Median()
{
  IMAGE image;
  char* in;
  int data;

  if ( ( in = Pop_Input("Enter frame number") ) != NULL )
       data = atoi(in);

  image  = Create_Image_Buffer();
  Pop_Print( "Reading Frame", 0 );
  Read_Pip_Frame( data, image );
  Clear_Print();
  Pop_Print( "Median Filtering Image", 0 );
  Median_Filter(image );
  Clear_Print();
  Pop_Print("Displaying Filterised Image",0);
  Display(data, image );
  Clear_Print();
  Pop_Print( "Deallocating PC memory", 0 );
  Free_Image_Buffer( image );
  Clear_Print();
}

void Do_Norm()
{
  IMAGE image;
  char* in;
  int dat;

  if  ( ( in = Pop_Input("Enter frame number") ) != NULL )
       { dat = atoi(in);

  image  = Create_Image_Buffer();
```

photo.c

```
  Pop_Print( "Reading Frame", 0 );
  Read_Pip_Frame( dat, image );
  Clear_Print();
  Pop_Print( "Normalising Image", 0 );
  Normalise( image );
  Clear_Print();
  Pop_Print("Displaying Normalised Image",0);
  Display( dat,image );
  Clear_Print();
  Pop_Print( "Deallocating PC memory", 0 );
  Free_Image_Buffer( image );
  Clear_Print();
        }
}

void Do_Thres()
{
  IMAGE image;
  int threshold;
  char* in;
  char* nu;
  int data;
  float n;

  if ( ( in = Pop_Input("Enter frame number") ) != NULL )
        { data = atoi(in);

  if ( (nu = Pop_Input("Enter threshold value") ) != NULL )
        { n = atof( nu );

  threshold = (int) (( ((float) (n))/100.0 ) * 255.0+0.5);
  image  = Create_Image_Buffer();
  Pop_Print( "Reading Frame", 0 );
  Read_Pip_Frame( data, image );
  Clear_Print();
  Pop_Print( "Thresholding Image", 0 );
  Threshold( threshold, image );
  Clear_Print();
  Pop_Print("Displaying Thresholded Image",0);
  Display(data, image );
  Clear_Print();
  Pop_Print( "Deallocating PC memory", 0 );
  Free_Image_Buffer( image );
  Clear_Print();
    }
  }
}

void QWP1_Rotation()
{
        int count;
        int num;
        int ratio = 200;
        int degree;
        int choice;
        char* inli;
        char* inline;

        count=0;

        outport(control,131);

        if ( ( inli = Pop_Input("Enter dir. of motor(1 or 2)") ) != NULL )
            choice = atoi(inli);

        if ( ( inline = Pop_Input("Enter degree of rotation") ) != NULL )
```

photo.c

```
            { degree  = atoi(inline);
              num = (int)(ratio * degree);}

        if(choice == 1)
           {
            while(count<num){
            outport(porta,0);
            delay(5);
            outport(porta,1);
            count=count+1;}
            delay(10);
           }

        else if(choice == 2)
            {
            while(count<num){
            outport(porta,2);
            delay(5);
            outport(porta,3);
            count=count+1;}
           }
}

void QWP2_Rotation()
{
        int count;
        int num;
        int ratio = 200;
        int degree;
        int choice;
        char* inline;

        count=0;

        outport(control,131);

        if ( ( inline = Pop_Input("Enter dir. of motor(1 or 2)") ) != NULL )
            choice = atoi(inline);

        if ( ( inline = Pop_Input("Enter degree of rotation") ) != NULL )
            { degree  = atoi(inline);
              num = (int)(ratio * degree); }

        if(choice == 1)
           {
            while(count<num){
            outport(porta,0);
            delay(5);
            outport(porta,4);
            count=count+1;}
            delay(10);
           }

        else if(choice == 2)
            {
            while(count<num){
            outport(porta,8);
            delay(5);
            outport(porta,12);
            count=count+1;}
           }
}

void Sample_Rotation()
{
        int count;
```

photo.c

```
        int num;
        float degree;
        int choice;
        float ratio = 3.3;
        char* inli;
        char* inline;

        count=0;

        outport(control,131);

        if ( ( inli = Pop_Input("Enter dir. of motor(1 or 2)") ) != NULL )
            choice = atoi(inli);

        if ( ( inline = Pop_Input("Enter degree of rotation") ) != NULL )
            { degree  = atof(inline);
              num = (int)(ratio * degree);delay(10);}

        if(choice == 1)
           {
            while(count<num){
            outport(porta,64);
            delay(50);
            outport(porta,0);
            count=count+1;}
           }

        else if(choice == 2)
            {
            while(count<num){
            outport(porta,128);
            delay(50);
            outport(porta,192);
            count=count+1;}
           }
}

void Synchronous_Rotation()
{

        int count;
        int num;
        int ratio = 200;
        int degree;
        int choice;
        char* inli;
        char* inline;

        count=0;

        outport(control,131);

        if ( ( inli = Pop_Input("Enter dir. of motor(1 or 2)") ) != NULL )
            choice = atoi(inli);

        if ( ( inline = Pop_Input("Enter degree of rotation") ) != NULL )
            { degree  = atoi(inline);
              num = (int)(ratio * degree);}

        if(choice == 1)
           {
            while(count<num){
            outport(porta,2);
            delay(5);
            outport(porta,7);
            count=count+1;}
```

photo.c

```
            delay(10);
           }

       else if(choice == 2)
           {
            while(count<num){
            outport(porta,8);
            delay(5);
            outport(porta,13);
            count=count+1;}
            }

 }

void Video_Cursor( rx, ry )
int *rx;
int *ry;
{
  unsigned char key;
  int was_zero = 0;
  int i;
  char string[256];
  char*  in;
  int    d = 1;
  int c;

  Initialise_Cursor( 0, *rx, *ry );

  do {
       sprintf( string, " x=%d y=%d", cursorx,cursory );

       Pop_Print( string, 0 );

       key = Get_Next_Char();

       if ( key == ESCAPE ) break;

       if ( key == 0 ) was_zero = 1;
       else
       {
         if ( was_zero == 1 ) {

         switch ( key ) {

         case CURSORRIGHT : Cursor_Right( 0, d ); was_zero = 0; break;
         case CURSORLEFT  : Cursor_Left( 0, d );  was_zero = 0; break;
         case CURSORUP    : Cursor_Up( 0, d );    was_zero = 0; break;
         case CURSORDOWN  : Cursor_Down( 0, d );  was_zero = 0; break;

                       }
                            }
         else {
                switch ( key ) {


                default   : break;

                            }
              }
       }
     } while( 1 );

  Clear_Print();
  Cancel_Cursor( 0 );

  *rx = cursorx;
```

photo.c

```
  *ry = cursory;
}

void Cursor_Test()
{
    int x, y;
    int x1, y1;
    int i0;
    char string[80];
    char* iline;
    int  xp[2];
    int  yp[2];
    int  t;

    if ( ( iline = Pop_Input("Enter file prefix") ) != NULL )
         {
             sprintf( fname, "%s.dat", iline );

             out = fopen( fname, "w" );

             xp[0] = yp[0] = 255;

             Video_Cursor( &xp[0], &yp[0] );

             xp[1] = xp[0];
             yp[1] = yp[0];

             Video_Cursor( &xp[1], &yp[1] );

             if ( xp[0] > xp[1] ) { t = xp[0]; xp[0] = xp[1]; xp[1] = t; }

             if ( yp[0] > yp[1] ) { t = yp[0]; yp[0] = yp[1]; yp[1] = t; }

               if ( out != NULL )
                    {
                        for (y = yp[0]; y <= yp[1]; y++)
                        {
                          for (x = xp[0]; x <= xp[1];  x++)
                           {
                             fprintf( out, "%4d", ( int ) Pip_Read( 0, x, y ) );
                           }
                          fprintf(out, "\n" );
                        }
                     fclose( out );
                    }
         }
}

void Set_Sobel_Threshold()
{
  char* num;
  char string[256];
  int n;

  sprintf( string, "Thres. %.2lf, Enter new value", sobel_threshold );

  if ( (num = Pop_Input(string) ) != NULL )
     {
       sobel_threshold = (double) atof( num );
     }
}

void Set_Gain_And_Offset()
{
  char* in;
  char string[80];
```

photo.c

```
    sprintf( string, "Gain = %d, ?", pip_video_gain );

    if ( ( in = Pop_Input(string) ) != NULL )
       {
         pip_video_gain = atoi( in );

         Pip_Set_Video_Gain( pip_video_gain );

         sprintf( string, "Offset = %d, ?", pip_video_offset );

         if ( ( in = Pop_Input(string) ) != NULL )
            {
              pip_video_offset = atoi( in );

              Pip_Set_Video_Offset( pip_video_offset );
            }
       }
}

void Define_Menus()
{
  F2_DEF  = Save_Image_Buffer;          F2_NAME = "Save";
  F3_DEF  = Load_Image_Buffer;          F3_NAME = "Load";
  F4_DEF  = Snap_Shot;                  F4_NAME  = "Snap";
  F5_DEF  = Cont_Grab;                  F5_NAME  = "Cgrab";
  F6_DEF  = Set_Sobel_Threshold;        F6_NAME  = "SetSob";
  F7_DEF  = Set_Gain_And_Offset;        F7_NAME  = "G+O";
  F8_DEF  = Load_16Bit_Image_Buffer;    F8_NAME  = "Load16";
  F9_DEF  = Sum_Frames;                 F9_NAME  = "Sum";

  /* Define menu title and option texts */

  MA.title = "Piptool";

  MA.ot[0] = "Motor";
  MA.ot[1] = "Quadr";
  MA.ot[2] = "Video";
  MA.ot[3] = "Fringes";
  MA.ot[4] = "Dos";
  MA.ot[5] = "Exit";
  MA.ot[6] = "END";

 /* Define help for each option */

  MA.oh[0] = "!This option allow!the step motors to be!controlled by PC and the!stepm
otors are used!to drive polariser,!analyser and samples.";

  MA.oh[1] = "! This option allows!the visible quadrant!to be changed there!are 4 qua
drants on!the PIP numbered 0!to 3.";

  MA.oh[2] = "! Selecting this option!causes the video camra to!capture the images!an
d pre-process the!images.";

  MA.oh[3] = "! Selecting this option!causes the package to!do the phase-stepping!ana
lysis of the fringes.";

  MA.oh[4] = "! Selecting this option!allows execution of an!opreating system command
 !such as 'dir'.";

  MA.oh[5] = "! Selecting this option!causes the package to!be exited.";

  /* Define functions or menus called by options */

   MA.op[0] = &MB;                  MA.tp[0] = CALLS_MENU;
   MA.op[1] = &MC;                  MA.tp[1] = CALLS_MENU;
```

photo.c

```
   MA.op[2] = &MD;                  MA.tp[2] = CALLS_MENU;
   MA.op[3] = &MF;                  MA.tp[3] = CALLS_MENU;
   MA.op[4] = System_Command;       MA.tp[4] = CALLS_FUNC;
   MA.op[5] = Exit;                 MA.tp[5] = CALLS_FUNC;

  /* Define option that selection bar is to appear over first */

  MA.last_selection = 0;

MB.title = "Stepper Motor Control";

MB.ot[0] = "Rotation of QWP1";
MB.ot[1] = "Rotation of QWP2";
MB.ot[2] = "Rotation of Sample";
MB.ot[3] = "Synchronours Rotation";
MB.ot[4] = "END";

/* Define help for each option */

MB.oh[0] = "Select this option!to rotate the!quterwave plate 1.";
MB.oh[1] = "Select this option!to rotate the!quterwave plate 2.";
MB.oh[2] = "Select this option!to rotate the!sample.";
MB.oh[3] = "Select this option!to both polariser!and analyser!synchronously.";

/* Define functions or menus called by options */

  MB.op[0] = QWP1_Rotation;             MB.tp[0] = CALLS_FUNC;
  MB.op[1] = QWP2_Rotation;             MB.tp[1] = CALLS_FUNC;
  MB.op[2] = Sample_Rotation;           MB.tp[2] = CALLS_FUNC;
  MB.op[3] = Synchronous_Rotation;      MB.tp[3] = CALLS_FUNC;

  /* Define option that selection bar should appear over first */

  MB.last_selection = 0;

MC.title = "Enter Quadrant Number";

MC.ot[0] = "0th quadrant";
MC.ot[1] = "1st quadrant";
MC.ot[2] = "2nd quadrant";
MC.ot[3] = "3rd quadrant";
MC.ot[4] = "END";

 /* Define help for each option */

MC.oh[0] = " View quadrant zero ?";

MC.oh[1] = " View quadrant one ?";

MC.oh[2] = " View quadrant two ?";

MC.oh[3] = " View quadrant three ?";

 /* Define functions called by options */

MC.op[0] = View0; MC.tp[0] = CALLS_FUNC;
MC.op[1] = View1; MC.tp[1] = CALLS_FUNC;
MC.op[2] = View2; MC.tp[2] = CALLS_FUNC;
MC.op[3] = View3; MC.tp[3] = CALLS_FUNC;

 /* Define option that selection bar should appear over first */

MC.last_selection = 0;

MD.title = "Video Functions";
```

photo.c

```
 MD.ot[0]  = "Continuous Grab";
 MD.ot[1]  = "Snap Shot";
 MD.ot[2]  = "Save Image";
 MD.ot[3]  = "Load Image";
 MD.ot[4]  = "Edge Detection";
 MD.ot[5]  = "Histogram";
 MD.ot[6]  = "Invert Image";
 MD.ot[7]  = "Normalise";
 MD.ot[8]  = "Threshold";
 MD.ot[9]  = "Cursor Test";
 MD.ot[10] = "Run Live";
 MD.ot[11] = "END";

  /* Define help for each option */

 MD.oh[0] = " Grab frames continuously!from input";

 MD.oh[1] = " Take a snap shot frame!from input";

 MD.oh[2] = "This option allows!an image from one of!the 4 quadrants of the!PIP board
  to be saved!to a disk file the!image saved comes from!the quadrant on display.! A fil
e name for the!image is requested.";

 MD.oh[3] = "This option allows!an image to be loaded!into one of the 4 quadrants! of
  the PIP board!the image is loaded into!the quadrant on display.!The filename for the!
image is requested.";

 MD.oh[4] = "Sobel edge detection.";
requested.";

 MD.oh[5] = "Histogram operation to !calculate the intensity !distribution.";

 MD.oh[6] = "Invert the vedio image";

 MD.oh[7] = "Normalising the image!using aveargе filter.";

 MD.oh[8] = "Threshoding the image.";

 MD.oh[9] = "Using cursor to select the data for matlab plot";

 MD.oh[10] = "Capture two frames! sequentially.";

  /* Define functions called by options */

 MD.op[0]  = Cont_Grab;          MD.tp[0] = CALLS_FUNC | KEEP_MENU;
 MD.op[1]  = Snap_Shot;          MD.tp[1] = CALLS_FUNC | KEEP_MENU;
 MD.op[2]  = Save_Image_Buffer;  MD.tp[2] = CALLS_FUNC | KEEP_MENU;
 MD.op[3]  = Load_Image_Buffer;  MD.tp[3] = CALLS_FUNC | KEEP_MENU;
 MD.op[4]  = Sobel_Edge_Detect;  MD.tp[4] = CALLS_FUNC;
 MD.op[5]  = Histogram;          MD.tp[5] = CALLS_FUNC;
 MD.op[6]  = Invert_Video;       MD.tp[6] = CALLS_FUNC;
 MD.op[7]  = Do_Norm;            MD.tp[7] = CALLS_FUNC;
 MD.op[8]  = Do_Thres;           MD.tp[8] = CALLS_FUNC;
 MD.op[9]  = Cursor_Test;        MD.tp[9] = CALLS_FUNC;
 MD.op[10] = Run_Live;           MD.tp[10] = CALLS_FUNC;

  /* Define option that selection bar should appear over first */

 MD.last_selection = 0;

 /* Define menu title and option text */

 MF.title = "Fringe Analysis";

 MF.ot[0] = "Isochromatic Phase Stepping";
```

photo.c

```
 MF.ot[1] = "Isopachic Phase Stepping";
 MF.ot[2] = "Isoclinic Phase Stepping";
 MF.ot[3] = "FFT Method";
 MF.ot[4] = "Isoclinic Image Division";
 MF.ot[5] = "END";

 /* Define help for each option */

 MF.oh[0] = "Selecting this option allows!to do isochromatic phase!stepping. Enter yo
ur choice!1: Three step acos!2: Five step actan!3: Three step actan!4: Four step actan
.";
 MF.oh[1] = "Selecting this option allows!to do isopachic phase stepping.!You need to
 capture 4 images!with holographic method!for calculation of!the phase value.";
 MF.oh[2] = "Selecting this option allows!to do isoclinic phase stepping.!You need to
 capture 3 or 4 images!with plane polariscope!for calculation of!the phase value.";
 MF.oh[3] = "Selecting this option allows!to do FFT fringe analysis. !You need to wor
k!out the carrier frequency!before the calculation.";
 MF.oh[4] = "Selecting this option allows!to extract isoclinic angles!using image div
ision method.!You need to capture!two images by plane!and circular polaricope!respecti
vely.";

 /* Define functions or menus called by options*/

 MF.op[0] = Isochromatic_Phase;          MF.tp[0] = CALLS_FUNC;
 MF.op[1] = Isopachic_Phase;             MF.tp[1] = CALLS_FUNC;
 MF.op[2] = Isoclinic_Phase;             MF.tp[2] = CALLS_FUNC;
 MF.op[3] = FFT_Method;                  MF.tp[3] = CALLS_FUNC;
 MF.op[4] = Isoclinic_Angles;            MF.tp[4] = CALLS_FUNC;

 /* Define option that selection bar is to appear over first*/

 MF.last_selection = 0;
}
```

fft2d.c

```
#include <math.h>
#include <stdio.h>
#include <values.h>
#ifdef UNIXC
#include <malloc.h>
#endif
#include <assert.h>
#ifdef TURBOC
#include <alloc.h>
#include <graphics.h>
#endif
#include "tiffio.h"
#include "fft2d.h"

#define SWAP(a,b) tempr=(a);(a)=(b);(b)=tempr
unsigned int PIXX;
unsigned int PIXY;
#define PIF 3.141592654
#define LOW_MOD ((float)(4.0*PIF))
#define THRESHOLD (51)
#define DEAD_PIXEL ((float) (1.0e+37))

typedef float** IMAGE;
typedef float** PHASE;
FILE *out_tan;
FILE *out_mod;
FILE *in;
unsigned short*  raster_tan;
unsigned short*  raster_mod;
int XSTEP,YSTEP;
float *raster;
int save_tan = 1;
int save_mod = 1;
int is_fringe_doubling = 0;

typedef struct{

        long type;
        long mrows;
        long ncols;
        long imagf;
        long namlen;
} Fmatrix;

int tif_width;
int tif_length;
int array_width;
int array_length;
unsigned char *file_name;
int no[3];
int xstep,ystep;
double xres,yres;
float* data_buffer;
PHASE phase;
IMAGE window_spatial;
IMAGE window_fourier;
double top,bottom;
float carrier_freq;
int n_data_buffer_read;
float  scale_tan;
float  tan_offset;
double critical_size;
int*   threshold_results;
int dump_ffts_begin_data_buffer;
int dump_ffts_end_data_buffer;
unsigned n_padded_zeros;
```

fft2d.c

```
unsigned n_padded_data_buffer;
float* powers;
float peak;

int    n_coeffs_fourier = 7;
float  sigma_coeffs_fourier[] = { 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0 };
int    n_coeffs_spatial = 7;
float  sigma_coeffs_spatial[] = { 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0   };


void Fourn(data,nn,ndim,isign)
float data[];
int nn[],ndim,isign;
{
        int i1,i2,i3,i2rev,i3rev,ip1,ip2,ip3,ifp1,ifp2;
        int ibit,idim,k1,k2,n,nprev,nrem,ntot;
        float tempi,tempr;
        double theta,wi,wpi,wpr,wr,wtemp;

        ntot=1;
        for (idim=1;idim<=ndim;idim++)
                ntot *= nn[idim];
        nprev=1;
        for (idim=ndim;idim>=1;idim--) {
                n=nn[idim];
                nrem=ntot/(n*nprev);
                ip1=nprev << 1;
                ip2=ip1*n;
                ip3=ip2*nrem;
                i2rev=1;
                for (i2=1;i2<=ip2;i2+=ip1) {
                        if (i2 < i2rev) {
                                for (i1=i2;i1<=i2+ip1-2;i1+=2) {
                                        for (i3=i1;i3<=ip3;i3+=ip2) {
                                                i3rev=i2rev+i3-i2;
                                                SWAP(data[i3],data[i3rev]);
                                                SWAP(data[i3+1],data[i3rev+1]);
                                        }
                                }
                        }
                        ibit=ip2 >> 1;
                        while (ibit >= ip1 && i2rev > ibit) {
                                i2rev -= ibit;
                                ibit >>= 1;
                        }
                        i2rev += ibit;
                }
                ifp1=ip1;
                while (ifp1 < ip2) {
                        ifp2=ifp1 << 1;
                        theta=isign*6.28318530717959/(ifp2/ip1);
                        wtemp=sin(0.5*theta);
                        wpr = -2.0*wtemp*wtemp;
                        wpi=sin(theta);
                        wr=1.0;
                        wi=0.0;
                        for (i3=1;i3<=ifp1;i3+=ip1) {
                                for (i1=i3;i1<=i3+ip1-2;i1+=2) {
                                        for (i2=i1;i2<=ip3;i2+=ifp2) {
                                                k1=i2;
                                                k2=k1+ifp1;
                                                tempr=wr*data[k2]-wi*data[k2+1];
                                                tempi=wr*data[k2+1]+wi*data[k2];
                                                data[k2]=data[k1]-tempr;
                                                data[k2+1]=data[k1+1]-tempi;
                                                data[k1] += tempr;
```

fft2d.c

```
                                        data[k1+1] += tempi;
                                }
                        }
                        wr=(wtemp=wr)*wpr-wi*wpi+wr;
                        wi=wi*wpr+wtemp*wpi+wi;
                    }
                    ifp1=ifp2;
                }
                nprev *= n;
        }
}

#undef SWAP

PHASE Create_Phase_Array( nx, ny )
unsigned nx, ny;
{
  PHASE phase;
  int i;

  phase = (PHASE) malloc((nx)*(sizeof(float*)));

  if ( phase != NULL )
     {
       for( i = 0; i < nx; i++ )
          {
            phase[i] = (float*) calloc(ny, (sizeof(float)));

            if (phase[i] == NULL )
              printf("Ran out of memory trying to allocate phase array\n");
          }
     }
  else printf("Ran out of memory trying to allocate phase array\n");

  return( phase );
}

void Free_Phase_Array( phase, nx )
PHASE phase;
unsigned nx;
{
  int i;

  if ( phase != NULL )
     {
       for( i = 0; i < nx; i++ )
            free( (char*) phase[i] );

       free( (char*) phase );
     }
}

IMAGE Create_Image_Array( width, length )
unsigned width;
unsigned length;
{
  IMAGE image;
  int i;

  image = (IMAGE) malloc((width)*(sizeof(float*)));

  if ( image != NULL )
     {
       for( i = 0; i < width; i++ )
          {
            image[i] = (float*) calloc(length, (sizeof(float)));
```

fft2d.c

```
            if ( image[i] == NULL )
               {
                 printf( "Out of memory trying to allocate image array\n" );
                 exit(1);
               }
          }
     }
  else
     {
       printf( "Out of memory trying to allocate image array\n" );
       exit(1);
     }

  return( image );
}


float* Allocate_Float_Array(n)
unsigned n;
{
  float* p;


  p = (float*) calloc( n , sizeof(float) );

  if ( p == NULL )
     {
       printf("Unable to allocate memory for float array\n");
       exit(1);
     }

  return(p);
}

void Free_Image_Array( image, width )
IMAGE image;
unsigned width;
{
  int i;

  if ( image != NULL )
     {
       for( i = 0; i < width; i++ )
            free( (char*) image[i] );

       free( (char*) image );
     }
}

float *Readtiff(file_name)
unsigned char *file_name;
{
        TIFF *tif_in;
        TIFFDirectory *tif_dir;
        int i;
        int x,y;
        double r;
        double d;
        int c;
        int extra_length, extra_width;
        unsigned char *buf;
        float *data_buffer;

        tif_in= TIFFOpen(file_name, "r");
        if (tif_in != NULL)
```

fft2d.c

```
        {
        tif_dir = &tif_in->tif_dir;

        xres = (double) tif_in->tif_dir.td_xresolution;
        yres = (double) tif_in->tif_dir.td_yresolution;

        buf = (unsigned char*) malloc(TIFFScanlineSize(tif_in));
        tif_width =  tif_dir->td_imagewidth;
        tif_length = tif_dir->td_imagelength;

        d = (log((double) tif_width)/(log((double) 2)));

        if ( d != ( (double) ((int) d) ) ) {

        d += 1.0;
        d = (double) ((int) d);
                                        }

        array_width = (int) ( pow( 2.0, d ) + 0.5 );
        extra_width = array_width  - tif_width;

        PIXX = array_width;

        r = (log((double)tif_length)/(log((double)2)));

        if ( r != ( (double) ((int) r) ) ) {

        r += 1.0;
        r = (double) ((int)r);
                                        }

        array_length = (int) ( pow( 2.0, r ) + 0.5 );
        extra_length = array_length  - tif_length;

        PIXY = array_length;

        if ( (data_buffer = (float*)
malloc((2*(array_width)*(array_length)+1)* sizeof(float))) == NULL)
        {
                printf ("Unable to allocate the required memory space\n");

        }
                for (y=0;y<tif_length;y++)
                {
                        TIFFReadScanline(tif_in,buf,(u_int) y,(u_int) 0);
                        for (x=0;x<tif_width;x++)
                        {
                                data_buffer[2*(array_width*y + x)+1] = (float)(buf[x]
);
                                data_buffer[2*(array_width*y + x)+2] = 0.0;

                        }
                        for (x=tif_width;x<array_width;x++)
                        {
                                data_buffer[2*(array_width*y + x)+1] = 0.0;
                                data_buffer[2*(array_width*y + x)+2] = 0.0;
                        }
                }
                for(y=tif_length;y<array_length;y++)
                {
                        for(x=0;x<array_width;x++)
                        {
                                data_buffer[2*(array_width*y + x)+1] = 0.0;
                                data_buffer[2*(array_width*y + x)+2] = 0.0;

                        }
```

fft2d.c

```
                }
        TIFFClose(tif_in);
        return(data_buffer);
        }
        else return(NULL);
}

#ifdef TURBOC
void open_graph()

{
int     GraphDriver;
int     GraphMode;

        GraphDriver = VGA;                      /* Request auto-detection       */
        GraphMode = VGAHI;
        initgraph( &GraphDriver, &GraphMode, "" );
}
#endif

void Compute_Power_Spectrum(data_buffer, power)
float *data_buffer;
float *power;
{

        float grey_diff, diff, min, max;
        int x,y;
        float real, imag, p;


        min = MAXFLOAT;
        max = 0.0;

        for (y=0; y<array_length; y++)
        {
                for (x=0; x<array_width; x++)
                {
                        real = data_buffer[2*(array_width*y+x)+1];
                        imag = data_buffer[2*(array_width*y+x)+2];

                        power[array_width*y+x] = ( p =
                        (float) sqrt((double)(real*real)+(imag*imag)) );

                }
        }
}

#ifdef TURBOC
void write_pixels(data_buffer)

float *data_buffer;

{
        float ratiox, ratioy;
        int maxx, maxy, y,x;
        float* power;

        if ( (power = (float*) malloc(array_width*array_length* sizeof(float))) == NU
LL)
        {
                printf ("Unable to allocate the required memory space\n");

        }


        compute_power_spectrum(data_buffer, power);
```

fft2d.c

```
        maxx = getmaxx();
        maxy = getmaxy();

        ratiox = ((float) maxx)/ ((float) array_width);
        ratioy = ((float) maxy)/ ((float) array_length);


        for (y=0; y<array_length; y++)
        {
                for (x=0; x<array_width; x++)
                        putpixel((int)(ratiox*(float)x+0.5), (int)(ratioy*(float)y+0
.5), (int)( power[array_width*y +x] + 0.5 ));
        }

        free( power );
}
#endif

float Compute_Peek_Frequency( fft_data )
float* fft_data;
{
  int x;
  int i,j;
  float magnitude;
  float peek_frequency;
  float peek_power;
  int   ncf;

  i = 1;

  for (y = 0; y < PIXY; y++ )
    for( x = 0; x < PIXX; x++ )
     {
       j = i+1;

       powers[x][y] = (float) sqrt( (double) ( fft_data[i]*fft_data[i] +
                                               fft_data[j]*fft_data[j] ) );
       i += 2;
     }

  peek_power = 0.0; ncfx = 0; ncfy = 0;

  for( x = 5; x < (PIXX/2); x++ )
       if ( powers[x][y]> peek_power ) { peek_power = powers[x][y]; ncfx
= x; ncfy = y;}

  peek_frequency = ((float) ncf*ncy)/((float) PIXX*PIXY);


#ifdef unixc
refresh();
#endif

  return( peek_frequency );
}

void Locate_Cut_Off_Frequencies_By_Thresholding_Power_Spectrum
( powers, carrier_index, lower_bound, upper_bound )
float* powers;
int carrier_index;
int* lower_bound;
int* upper_bound;
{
   int i,j;
   int cut_off;
```

fft2d.c

```
   int checkband;
   float power_threshold;
   float sum_power;

   checkband = ( carrier_index / 3 );

   power_threshold = 1.0;

   do {

   *upper_bound = 0;
   *lower_bound = 0;

   sum_power = 0.0;

   for( j = carrier_index; j > carrier_index-(checkband-1); j-- )
      sum_power += powers[j];

   for( i = (carrier_index -( checkband-1 ) ); i > 0; i-- )
      {
        sum_power += powers[i];
        if (sum_power<power_threshold) {*lower_bound = i; break;}
        sum_power -= powers[i+(checkband-1)];
      }


   if ( carrier_index + checkband < ( PIXX / 2 ) )
      {

   sum_power = 0.0;

   for( j = carrier_index; j < carrier_index+checkband-1; j++ )
      sum_power += powers[j];

   for( i = (carrier_index + checkband-1 );
        i < ( PIXX / 2 ); i++ )
      {
        sum_power += powers[i];
        if (sum_power<power_threshold) {*upper_bound = i; break;}
        sum_power -= powers[i-checkband-1];
      }

      }
   else *upper_bound = (( PIXX / 2 )-1);

   power_threshold += ((float) checkband * 100.0);

      } while( *lower_bound == 0 || *upper_bound == 0 );

   printf(
"Carrier %d, Check Band = %d, Lower Bound = %d, Upper Bound = %d, Power = %10.0f\n",
carrier_index,    checkband, *lower_bound, *upper_bound, power_threshold );
}

void Compute_Filter_Window( window_spatial )
float** window_spatial;
{
  double n;
  int   x,y;
  int   x1,y1;
  float wk;

  n = (double) (PIXX/2);


  for( y1 = 0; y1 < PIXY; y1++ )
```

fft2d.c

```
    for( x1 = 0; x1 < PIXX; x1++ )
     {
       x = x1 - (PIXX/2);
       y = y1 - (PIXY/2);

       wk = (float)
        (
            (0.5 + 0.5*cos( (double) (PIF * ((double) x) / n ) ))*
            (0.5 + 0.5*cos( (double) (PIF * ((double) y) / n ) ))
       );


       window_spatial[x1][y1] = wk;

     }
}

void Compute_Trig_Sum_Window( window_fourier,
                              sigma_coeffs, n_sigma_coeffs, N )
float* window_fourier;
float* sigma_coeffs;
unsigned n_sigma_coeffs;
unsigned N;
{
  unsigned t,k;
  float sum;
  double omega_t;
  double K;
  unsigned T;

  T = N - 1;

  for( t = 0 ; t <= T ; t++ )
     {
       sum = sigma_coeffs[0];

       for( k = 1; k < n_sigma_coeffs; k++ )
          {
            omega_t =
            ( (double) t ) * ( ( double ) k ) * ( (double) PIF )
            / ( (double) T );

            K = ( (double) k ) * ( (double) PIF ) / 2.0;

            sum += sigma_coeffs[ k ] * ( float ) cos( omega_t - K );
          }

       window_fourier[ t ] = sum;

     }
}

IMAGE Create_Window_Array( width, length )
unsigned width;
unsigned length;
{
  IMAGE image;
  int i;

  image = (IMAGE) malloc((width)*(sizeof(float*)));

  if ( image != NULL )
     {
       for( i = 0; i < width; i++ )
          {
            image[i] = (float*) calloc(length, (sizeof(float)));
```

fft2d.c

```
            if ( image[i] == NULL )
               {
                 printf( "Out of memory trying to allocate image array\n" );
                 exit(1);
               }
          }
     }
  else
     {
       printf( "Out of memory trying to allocate image array\n" );
       exit(1);
     }

  return( image );
}

void Free_Window_Array( image, width )
IMAGE image;
unsigned width;
{
  int i;

  if ( image != NULL )
     {
       for( i = 0; i < width; i++ )
            free( (char*) image[i] );

       free( (char*) image );
     }
}

void Multiply_Scan_By_Filter_Window( data_buffer, window_spatial )
float* data_buffer;
float** window_spatial;
{
  int x,y;
  int i;

  i = 1;

  for( y = 0; y < PIXY; y++ )
    for( x = 0; x< PIXX; x++ )
       {
         data_buffer[i] *= window_spatial[x][y]; i += 2;
       }
}

void Write_Scan_Data( fft_data, output_file )
float* fft_data;
char*  output_file;
{
  FILE* out;
  int x,y;
  int i;

  out = fopen( output_file, "w" );

  if ( out != NULL )
     {
       i = 1;
       for( x = 0; x < PIXX; x++ )
          {
            fprintf( out, "%f\n", fft_data[i]);
            i += 2;
          }
```

fft2d.c

```
        fclose( out );
      }
   else { printf( "Couldn't open data_buffer output file\n" ); }
}

void Write_FFT_Data( fft_data, output_file )
float* fft_data;
char*  output_file;
{
  FILE* out;
  int x;
  int i,j;
  float magnitude;
  float* to_swap;

  out = fopen( output_file, "w" );

  if ( out != NULL )
     {
       to_swap = Allocate_Float_Array( PIXX );

       i = 1;
       for( x = 0; x < PIXX; x++ )
          {
            j = i+1;

            magnitude = (float) sqrt
            ( (double) ( fft_data[i]*fft_data[i] +
              fft_data[j]*fft_data[j] ) );

            to_swap[x] = magnitude;

            i += 2;
          }

       for( x = (PIXX/2); x < (PIXX); x++ )
            fprintf( out, "%f\n", to_swap[x] );

       for( x = 0; x < (PIXX/2); x++ )
            fprintf( out, "%f\n", to_swap[x] );

       fprintf( out, "\n" );

       free( (char*) to_swap );

       fclose( out );
     }
   else { printf( "Couldn't open fft output file\n" ); }
}

float* Power_Buffer(data_buffer)
float* data_buffer;
{
        int x,y;
        float *power;

        if ( (power = (float*) malloc(array_width*array_length* sizeof(float))) == NU
LL)
        {
                printf ("Unable to allocate the required memory space \n");
        }

        Compute_Power_Spectrum(data_buffer, power);

        free( data_buffer );
```

fft2d.c

```
  return( power );
}

IMAGE  Rearrange_Power_Buffer(power)
float* power;
{
        int x,y;
        IMAGE image_buffer;
        int half_width;
        int half_length;

        half_width = array_width / 2;
        half_length = array_length / 2;

        image_buffer = Create_Image_Array( array_width, array_length );

        for (y=half_length; y<array_length; y++)
        {
                for(x=half_width; x<array_width; x++)
                {
                  image_buffer[x-half_width][y-half_length] =
(float) power[array_width*y + x];
                }

                for (x=0; x<half_width; x++)
                {
                  image_buffer[x+half_width][y-half_length] =
(float) power[array_width*y + x];
                }
        }

        for (y=0; y< half_length; y++ )
        {

                for (x=half_width; x<array_width; x++ )
                {
                        image_buffer[x-half_width][y+half_length] =
(float) power[array_width*y+x];
                }

                for (x=0; x<(array_width/2); x++ )
                {
                        image_buffer[x+half_width][y+half_length] =
(float) power[array_width*y+x];
                }
        }

        free( power );

        return( image_buffer );
}

void Norm(image_buffer)
IMAGE image_buffer;

{

int x,y;
float diff, min, max, scale;

        min = MAXFLOAT;
        max = 0;
        for(y = 0;y < array_length; y++)
                {
                for (x=0; x < array_width; x++)
```

fft2d.c

```
                        {
                        if (min > image_buffer[x][y])
                                min = image_buffer[x][y];
                        if (max < image_buffer[x][y])
                                max = image_buffer[x][y];
                        }
                }
        diff = max - min;
        scale = 255.0/diff;

        for(y = 0;y < array_length; y++)
                {
                for (x=0; x < array_width; x++)
                        {
                        image_buffer[x][y] = ((image_buffer[x][y] - min) * scale);
                        }
                }

}

/*void Remove_Dc_Noise(image_buffer)
IMAGE image_buffer;

{

int x,y;

for(y = ((array_length/2) - 10); y < ((array_length/2) + 10); y++)
        {
        for(x = ((array_width/2) - 10); x < ((array_width/2) + 10); x++)
                {
                image_buffer[x][y] = 0.0;
                }
        }

} */


void Write_Tiff_File(width, length, buffer, file)
int width, length;
IMAGE buffer;
unsigned char* file;
{
        TIFF *tif_out;
        unsigned char* raster;

        short bitspersample = 8 ;
        short samplesperpixel = 1;

        long bpsl;
        long rowsperstrip;

        short new_width, new_length, y;
        int x;

        raster = (unsigned char*) malloc( (unsigned) array_width );
        if ( raster == NULL ) { printf("Couldn't allocate memory for raster\n"); exit
(1); }

        tif_out = TIFFOpen (file, "w");

        if (tif_out != NULL)
        {
           new_width = width;
           new_length = length;
```

fft2d.c

```
           bpsl = ((long) new_width * (long) bitspersample * (long) samplesperpixel)
/ 8L;

           rowsperstrip = (8096L/ bpsl);

           TIFFSetField (tif_out, TIFFTAG_IMAGEWIDTH, new_width);
           TIFFSetField (tif_out, TIFFTAG_IMAGELENGTH, new_length);
           TIFFSetField (tif_out, TIFFTAG_BITSPERSAMPLE, bitspersample);
           TIFFSetField (tif_out, TIFFTAG_PHOTOMETRIC, PHOTOMETRIC_MINISBLACK );
           TIFFSetField (tif_out, TIFFTAG_HOSTCOMPUTER, "UNIX SYSTEM");
           TIFFSetField (tif_out, TIFFTAG_COMPRESSION, COMPRESSION_NONE);
           TIFFSetField (tif_out, TIFFTAG_PLANARCONFIG, PLANARCONFIG_CONTIG);
           TIFFSetField (tif_out, TIFFTAG_SAMPLESPERPIXEL, samplesperpixel);
           if (rowsperstrip > 0L)
                TIFFSetField (tif_out, TIFFTAG_ROWSPERSTRIP, rowsperstrip);
           else
                TIFFSetField (tif_out, TIFFTAG_ROWSPERSTRIP, ( (unsigned long) 0xffff
ffff));

  TIFFSetField( tif_out, TIFFTAG_XRESOLUTION, xres);
  TIFFSetField( tif_out, TIFFTAG_YRESOLUTION, yres);


           for (y = 0; y < new_length; y++)
           {
                for( x = 0; x < new_width; x++ )
                     raster[x] = buffer[x][y];

                TIFFWriteScanline(tif_out, raster, (u_int)y, (u_int)0);
           }

           TIFFClose(tif_out);
        }
}

void Write_Power_File(data_buffer, file_prefix )
float *data_buffer;
unsigned char* file_prefix;
{

        int x,y;
        float* power;
        IMAGE  image_buffer;
        TIFF *tif_out;
        char tif_file_name[80];

        sprintf( tif_file_name, "%s.pow.tif", file_prefix );

        power        = Power_Buffer(data_buffer);

        image_buffer = Rearrange_Power_Buffer(power);

      /*  Remove_Dc_Noise(image_buffer); */

        Norm(image_buffer);

        Write_Tiff_File(array_width, array_length, image_buffer, tif_file_name);

        Free_Image_Array( image_buffer, array_width );
}


void Perform_Two_Dimensional_FFT_On_Scan_Line( data_buffer )
float* data_buffer;
{
  int nn[2];
```

fft2d.c

```
  nn[1] = PIXX;
  nn[2] = PIXY;

  Fourn(data_buffer,nn,2,1);
}

void Invert_Two_Dimensional_FFT_On_Scan_Line( data_buffer)
float* data_buffer;
{
  int nn[2];

  nn[1] = PIXX;
  nn[2] = PIXY;

  Fourn(data_buffer,nn,2,-1);
}

int Shift_Side_Lobe_To_Origin( data_buffer, carrier_frequency, dump )
float* data_buffer;
float  carrier_frequency;
int dump;
{
  int x;
  int fd;
  int r1,i1,r2,i2;
  int src,dest;
  int side_lobe_cut_off_freq;
  float d;
  float window_weight;
  int   nwindow;
  int   window_fourier_flag;
  int   lower_bound;
  int   upper_bound;
  char* fname;
  char* files_prefix;

  window_fourier_flag = 1;

  fd = (int) ( ( ((float) n_padded_data_buffer) * carrier_frequency) + 0.5 );

  /* Routine called next fits a polynomial to
     part of the power spectrum and looks for the minimum
     point between the DC lobe and the side lobe */

  /* side_lobe_cut_off_freq =
    Locate_Side_Lobe_Cut_Off_Freq( data_buffer, (unsigned) 2, fd, dump ); */

    Locate_Cut_Off_Frequencies_By_Thresholding_Power_Spectrum
                  ( powers, fd, &lower_bound, &upper_bound );

  nwindow = ( upper_bound - lower_bound + 1);


  Compute_Trig_Sum_Window( window_fourier, sigma_coeffs_fourier,
                           n_coeffs_fourier, (unsigned) nwindow );

  if ( dump >= 0 ) {

  FILE* out;
  unsigned i;

  sprintf( fname, "%s.x%u.dat", files_prefix, dump );

  out = fopen( fname, "w" );
```

fft2d.c

```
  if ( out != NULL )
     {
       for( i = 0 ; i < ( n_padded_data_buffer / 2 ) ; i++ )
            fprintf( out, "%f\n", 0.0 );

       for( i = 0 ; i < lower_bound; i++ )
            fprintf( out, "%f\n", 0.0 );

       for( i = lower_bound; i < upper_bound; i++ )
            fprintf( out, "%f\n", window_fourier[i-lower_bound] );

       for( i = 0 ; i < (( n_padded_data_buffer / 2 ) - upper_bound ); i++ )
            fprintf( out, "%f\n", 0.0 );

       fclose( out );
     }
  else { printf("Couldn't open window output file\n" ); }

                          }

  for( x = (n_padded_data_buffer/2); x < n_padded_data_buffer; x++ )
     {
       r1 = 1 + x * 2;
       i1 = r1+1;

       data_buffer[r1] = 0.0;
       data_buffer[i1] = 0.0;
     }

  /* Filter Out Centre Lobe */

  for( x = 0; x < lower_bound; x++ )
     {
       r1 = 1 + x * 2;
       i1 = r1+1;

       data_buffer[r1] = 0.0;
       data_buffer[i1] = 0.0;
     }

  for( src = 0; src < (n_padded_data_buffer/2); src++ )
     {
       dest = src - fd;

       if ( dest < 0 ) dest += (n_padded_data_buffer);

       if ( window_fourier_flag == 1 )
          {

       if ( ( src  < lower_bound ) || ( src > upper_bound ) )
          window_weight = 0.0;
       else window_weight = window_fourier[ src - lower_bound ];

          }
       else window_weight = 1.0;

       r1 = 1 + src * 2;
       i1 = r1+1;

       r2 = 1 + dest * 2;
       i2 = r2+1;

       data_buffer[r2] = data_buffer[r1] * window_weight;
       data_buffer[i2] = data_buffer[i1] * window_weight;
     }
```

fft2d.c

```
   for( x = ((n_padded_data_buffer/2)-fd); x < (n_padded_data_buffer/2); x++ )
      {
        r1 = 1 + x * 2;
        i1 = r1+1;

        data_buffer[r1] = 0.0;
        data_buffer[i1] = 0.0;
      }

   return( lower_bound );
}

void Compute_FFT_Now_Of_Tan_Fringes
(save_tan,save_mod,is_fringe_doubling, dump, peak )
int save_tan;
int save_mod;
int is_fringe_doubling;
int dump;
float peak;
{
   int xp,yp;
   int i,j;
   double real,imag;
   char fname[256];
   int  side_lobe_cut;
   char *prefix;

         for( yp = 0; yp < PIXY; yp++ )
           {

             Write_Scan_Data( data_buffer, fname );

             Multiply_Scan_By_Filter_Window( data_buffer, window_spatial );

             Perform_Two_Dimensional_FFT_On_Scan_Line( data_buffer );

             side_lobe_cut =
              Shift_Side_Lobe_To_Origin( data_buffer, peak, dump );

            if ( dump >= 0 )
              {
                FILE* ct;

                sprintf( fname, "%s.t%u.dat", prefix, n_data_buffer_read );

                i = 1;
                for( xp = 0; xp < PIXX; xp++ )
                   {
                     j = i+1;

powers[xp] = (float) sqrt( (double) ( data_buffer[i]*data_buffer[i] +
                                      data_buffer[j]*data_buffer[j] ) );
                     i += 2;
                   }

                Write_FFT_Data( fname );
                sprintf( fname, "%s.c%u.dat", prefix, n_data_buffer_read );

                ct = fopen( fname, "w" );

                if ( ct != NULL )
                   {
                     fprintf( ct, "%d\n", side_lobe_cut );
                     fclose( ct );
                   }
```

fft2d.c

```
              }

            Invert_Two_Dimensional_FFT_On_Scan_Line( data_buffer  );

            for( xp = 0; xp < PIXX; xp++ )
               {
                 i = 2 * (xp+n_padded_zeros) + 1;
                 j = i+1;

                 real = data_buffer[i];
                 imag = data_buffer[j];

                 top    = imag;

                 bottom = real;

                 if ( is_fringe_doubling == 1 )
                    {
                      if ( bottom == 0.0 )
                           phase[xp][yp] = (float) PIF; /* 2 times pi/2 */
                      else
                           phase[xp][yp] = (float) (2.0*atan(top/bottom));
                    }
                 else
                    {
                      if ( bottom == 0.0 )
                           phase[xp][yp] = (float) PIF;
                      else
                           phase[xp][yp] = (float) (atan2(top,bottom));
                    }

                 if ( save_tan == 1 )
                      raster_tan[xp] = (unsigned short)
                     (scale_tan*(phase[xp][yp]+tan_offset)+0.5);


                 if ( save_mod == 1 )
                      raster_mod[xp] = (unsigned char) 0;

                 if ( in[0]->tif_dir.td_bitspersample == 8 )
                    {
                     if ( ( fabs(top) < critical_size ) &&
                          ( fabs(bottom) < critical_size ) )
                     {
                      phase[xp][yp] = DEAD_PIXEL;

                      if ( save_mod == 1 )
                           raster_mod[xp] = (unsigned char) 255;
                     }
                    }

                 if ( in[0]->tif_dir.td_bitspersample == 16 )
                    {
                      if ( ( fabs(top)    < ( critical_size * 256.0 ) ) &&
                           ( fabs(bottom) < ( critical_size * 256.0 ) ) )
                      {
                       phase[xp][yp] = DEAD_PIXEL;

                       if ( save_mod == 1 )
                           raster_mod[xp] = (unsigned char) 255;
                      }
                    }

               }

             if ( save_tan == 1 )
```

fft2d.c

```
                TIFFWriteScanline( out_tan, (u_char *) raster_tan,
                                   (u_int) n_data_buffer_read, (u_int) 0);


            if ( save_mod == 1 )
                TIFFWriteScanline( out_mod, (u_char *) raster_mod,
                                   (u_int) n_data_buffer_read, (u_int) 0);

            n_data_buffer_read++;

          /*  else {
                  for( xp = 0; xp < PIXX; xp++ )
                      phase[xp][yp] = 0.0;
                }    */
      }

}

void Close_FFT_Tan_Fringe_Computation
(save_tan,save_mod,is_fringe_doubling)
int save_tan;
int save_mod;
int is_fringe_doubling;
{

   while( n_data_buffer_read < PIXY )
        Compute_FFT_Row_Of_Tan_Fringes(save_tan,save_mod,is_fringe_doubling);

   TIFFClose( in[0] );
   free( raster[0]  );

   if ( save_tan == 1 ) {
                          TIFFFlushData(out_tan);
                          TIFFClose( out_tan );
                          free( raster_tan );
                        }

   if ( save_mod == 1 ) {
                          TIFFFlushData(out_mod);
                          TIFFClose( out_mod );
                          free( raster_mod );
                        }

   Free_Phase_Array( phase, PIXX );

   free( (char*) threshold_results );
   free( (char*) window_spatial );
   free( (char*) window_fourier );
   free( (char*) data_buffer );
}


void main(argc, argv)

int argc;
unsigned char *argv[];


{
        int dump = -1;

        xstep = 1;
        ystep = 1;

        if (argc>=1)
                file_name = argv[1];
```

fft2d.c

```
        if ((data_buffer = Readtiff(file_name)) != NULL)
        {
                no[1] = array_width;
                no[2] = array_length;

                printf("FFT Size x = %d, y = %d\n", no[1], no[2] );

                if ( argv[3] != NULL ) xstep = atoi( argv[3] );
                if ( argv[4] != NULL ) ystep = atoi( argv[4] );
                if ( argv[5] != NULL ) dump  = atoi( argv[5] );
                if ( argv[6] != NULL ) peak  = atoi( argv[6] );

                printf("Output data on grid of step size %d, %d\n", xstep,
ystep );

                phase = Create_Phase_Array(array_width,array_length);

                window_spatial
                    = Create_Window_Array(array_width, array_length);

Compute_Filter_Window( window_spatial );

Compute_FFT_Row_Of_Tan_Fringes(save_tan,save_mod,is_fringe_doubling,
dump,peak);

#ifdef TURBOC
                open_graph();
                write_pixels(data_buffer);
#endif


        }
else
printf("Usage: <fft2d> <input image> <output image> <xstep> <ystep> <dump>
<peak>\n");
 }
```

sdisk.c

```
#include <stdio.h>
#include <math.h>

#define TWO_PI (6.28318530717959)
#define PIXX 512
#define PIXY 512
#define PI (3.141592654)
#define P 10

unsigned char Spherical_Wave( f1, x, y, r)
double f1;
double x, y;
double r;

{
  double m;
  double x1,y1;
  double t1;
  double t2;
  double r1;
  double r2;
  double sx;
  double sy;

  x1 = (x - 256.0)/256.0; y1 = (y - 256.0)/256.0;

  t1 = atan((double) (x1/( r + y1)));

  t2 = atan((double) (x1/( r - y1)));

  r1 = sqrt((double)(2*y1 + r*r + x1*x1 + y1*y1));

  r2 = sqrt((double)(r*r - 2*r*y1 + x1*x1 + y1*y1));

   printf("The value of x1, y1, t1, t2, r1, r2,r are          %4.11f,%4.11f,%4.11f,%4.
11f,%4.1f,%4.1f,%4.11f\n", x1,y1,t1,t2,r1,r2,r);

  sx = (double)(2*P/PI)*( sin((double) t1) * (( cos((double) t2) /
                  r2) - 1/(2*r)) + sin((double) t2) *
                ( cos((double) t1) / r1 - 1/(2*(double)r)));
       printf("The value of sx is %4.11f\n", sx);

  sy = (double)(2*P/PI)*( cos((double) t1) * (( cos((double) t2) /
                  r2) - 1/(2*r)) + cos((double) t2) *
                ( 1/(2*r) - cos((double) t1) / r1));
       printf("The value of sy is %4.11f\n", sy);

  m = 1.0 + cos(TWO_PI*f1*((double)sx - (double)sy ));

/*  printf("The value of m is %4.1f\n", m*127.5); */
  return( (unsigned char) ( m*127.5  ) );
}

unsigned char Carrier_Spherical_Intensity( f1, f2, x, y, r )
double f1, f2;
double x, y;
double r;

{
  double m;
  double x1,y1;
  double t1;
  double t2;
  double r1;
  double r2;
  double sx;
```

sdisk.c

```
  double sy;

  x1 = (x - 256.0)/256.0; y1 = (y - 256.0)/256.0;

  t1 = atan((double) (x1/( r + y1)));

  t2 = atan((double) (x1/( r - y1)));

  r1 = sqrt((double)(2*r*y1 + r*r + x1*x1 + y1*y1));

  r2 = sqrt((double)abs((r*r - 2*r*y1 + x1*x1 + y1*y1)));

  sx = (double)(2*P / PI)*( sin((double) t1) * ( cos((double) t2) /
                  r2 - 1/(2*r)) + sin((double) t2)*
                  ( cos((double) t1) / r1 - 1/(2*(double)r )));

  sy = (double)(2*P / PI)*( cos((double) t1) * ( cos((double) t2) /
                  r2 - 1/(2*r)) + cos((double) t2) *
                  ( 1 / (2*r) - cos((double) t1) / r1 ));

  m = 1.0 + cos( TWO_PI*( f1*x + f2*((double)sx - (double)sy)));

  return( (unsigned char) ( m*127.5  ) );
}

void Spherical_Without_Carrier( image_name, f1, r )
char*  image_name;
double f1;
double r;

{
  unsigned x,y;
  FILE* out;
  int xmin, xmax;
  int ymin, ymax;
  int xc, yc;
  int xmin1, xmax1;
  int ymin1, ymax1;
  int xc1, yc1;
  int r1 = 60;

  out = fopen( image_name, "wb" );

  if ( out != NULL )
     {
       for( y = 0; y < PIXY; y++ )
         {
           yc = y - PIXY/2;

           yc1 = (int) ((r1/r) * yc);

           if (( r*r >= yc*yc ) || (r1*r1 <= yc1*yc1))
              {

           xc = (int) (sqrt((double) (( r*r ) - (yc*yc)) ) +0.5);

           xc1 = (int) ((r1/r) * xc);

           xmin = PIXX/2 - xc; xmax = PIXX/2 + xc;

           xc1 = (int) ((r1/r) * xc);

              }

           else { xmin = PIXX; xmax = 0;}
```

sdisk.c

```
          for (x = 0; x < PIXX; x++)
             {

             if ((x < xmin) || (x > xmax))

                { putc( 0, out ); }

             else
                {
           putc( Spherical_Wave( f1, (double) x, (double) y, (int)r
                               ), out );
                }
             }

        }
    }
  else { printf("Couldn't open output file\n"); exit(1); }
}

void Spherical_With_Carrier( image_name, f1, f2, r)
char*  image_name;
double f1, f2;
double r;
{
  unsigned x,y;
  FILE* out;
  int xmin, xmax;
  int ymin, ymax;
  int xc, yc;

  out = fopen( image_name, "wb" );

  if ( out != NULL )
     {
       for( y = 0; y < PIXY; y++ )
         {
           yc = y - PIXY/2;

           if ( r*r >= yc*yc )
              {

           xc = (int) (sqrt((double) (( r*r ) - (yc*yc)) ) +0.5);

           xmin = PIXX/2 - xc; xmax = PIXX/2 + xc;

              }
           else { xmin = PIXX; xmax = 0; };

           for (x = 0; x < PIXX; x++)
              {

              if (x < xmin  ||  x > xmax)
                 { putc( 0, out ); }

              else
                 {
          putc( Carrier_Spherical_Intensity( f1, f2, (double) x,
(double) y , (int)r), out );
                 }

              }
         }
    }
  else { printf("Couldn't open output file\n"); exit(1); }
}
```

sdisk.c

```
main( argc, argv )
int argc;
char* argv[];
{
  char*  image_name;
  double carrier_frequency;
  double spherical_frequency;
  double    disk_radius;
  char string[255];

  if ( argc == 5 )
     {
       image_name = argv[1];
       printf( "Output file name is %s\n", image_name );
       carrier_frequency = (double) atof( argv[2] );
       printf( "Frequency of carrier is %lf\n", carrier_frequency );
       spherical_frequency = (double) atof( argv[3] );
       printf( "Spherical Fringe Frequency is %f\n", spherical_frequency);
       disk_radius = (double) atof( argv[4] );
       printf( "Disk radius is (0-256) %d\n", disk_radius);

       sprintf( string, "%s.img", image_name );

       Spherical_Without_Carrier(
             string,
             spherical_frequency,
             disk_radius );

       sprintf( string, "%s.car", image_name );

       Spherical_With_Carrier(
             string,
             carrier_frequency,
             spherical_frequency,
             disk_radius);
     }
   else
  {
   printf("Usage: sdisk <outfile><carrier frequency>");
   printf("<fringe number><disk-radius>\n");
  }

}
```