An Investigation of Various Computational Techniques in Optical Fringe Analysis

Miguel Arevalillo Herráez

A thesis submitted in partial fulfilment of the requirements of Liverpool John Moores University for the degree of Doctor of Philosophy

December 1996

“Science can amuse and fascinate us all, but it is engineering that changes the world”

Isaac Asimov, Isaac Asimov’ Book of Science and Nature Quotations, 1988

Acknowledgements

The author would like to thank the following people:

Prof. David B. Clegg, not only for providing the funding for this research but also for the aid provided for completion of my first degree.

Prof. David R. Burton and Prof. Michael J. Lalor for their good humour, endless encouragement, guidelines, sense of direction, constant motivation, supervision and many other things that if I tried to write here would extend over the size of this thesis. It is, however, useless to attempt to express with words what is ineffable. Because without them, the completion of this work would have not been possible.

Dr. Madjid Merabti, for his good advice throughout my research. For re-motivating me every time I felt down.

My colleagues, the research students in the computing school and the engineering and technology management department. A special thanks to Dhiya Al-jumeili, Eng Huat Ng and Heather Ainsley, for their constant sincerity and useful advice.

David Guerrero López, because if there is something that everyone need is a true friend.

Finalmente, las cinco personas mas importantes en mi vida. Mis abuelos, Lorenzo Herráez Páramo y Zósima del Soto González, mi hermano Roberto Arevalillo Herráez y mis padres Miguel Arevalillo Ruiz y María Flor Herráez del Soto. Porque ellos demostraron que no hay metros, kilómetros ni millas que puedan separar el cariño, y porque siempre los lleve en el lugar mas cercano al alma, en aquel lugar donde nada jamás perece, en el corazón. Porque no puedo pensar en un solo momento en el que alguna de estas personas me haya fallado y porque los quiero mas que a nadie en este mundo. Porque ellos me dieron todo en el momento correcto, incluso la propia vida. Ninguna de estas personas tiene una carrera universitaria, ni tan siquiera un bachillerato, pero ellos me dieron la oportinudad de llevar a cabo aquello que ellos nunca pudieron. Y si no tienen estudios a nivel universitario, pueden sentirse orgullosos de ser catedráticos en cariño.

Quizás sean estos los dos únicos párrafos que entendáis en esta tésis, pero es un triunfo tan vuestro como mío propio. Poco pueden decir las palabras, ya lo dijo Friedell: "viste una idea en palabras y perderá su libertad de movimiento", pero son los sentimientos, y no las palabras, lo que quiero hacer llegar a vosotros. Gracias por estar ahí, por darme todo aquello que he necesitado durante los 23 años de mi vida.

Abstract

Fringe projection is an optical technique for three dimensional non-contact measurement of height distributions. A fringe pattern is projected onto an object's surface and, when viewed off-axis, it deforms to follow the shape of the object. The deformed fringe pattern is analysed to obtain its phase, information that is directly related to the height distribution of the surface by a proportionality constant.

This thesis analyses some key problems in fringe projection analysis. Special attention is focused on the automatisation of the process with Fourier Fringe Analysis (FFA). Unwrapping, or elimination of 2π discontinuities in a phase map, is treated in detail. Two novel unwrapping techniques are proposed, analysed and demonstrated. A new method to reduce the number of wraps in the resulting phase distribution is developed.

A number of problems related to FFA are discussed, and new techniques are presented for their resolution. In particular, a technique with better noise isolation is developed and a method to analyse non-fullfield images based on function mapping is suggested.

The use of parallel computation in the context of fringe analysis is considered. The parallelisation of cellular automata in distributed memory machines is discussed and analysed. A comparison between occam 2 and HPF, two compilers based upon a very different philosophy, is given.

A case study with implementations in occam 2 and high performance FORTRAN (HPF) is presented. The advantages and disadvantages of each solution are critically assessed.

A mi madre...

porque madre no hay mas que una,

Contents

Chapter one

Introduction

High precision measurements are very often required in many different areas of science and engineering (Atkinson et al. 1988; Bruning et al. 1974; Chung 1992; Malmo and Vikhagen 1988; Takasski 1970; Takasski 1973). Fringe projection contouring is an optical technique that allows high precision measurements of an object surface to be performed. When a straight fringe pattern is projected onto a surface and the resulting pattern is viewed off-axis, the fringe pattern will deform to follow the shape of the surface. The analysis of the deformed fringe pattern will produce an accurate reconstruction of the object's surface. This process is illustrated in Figure 1.

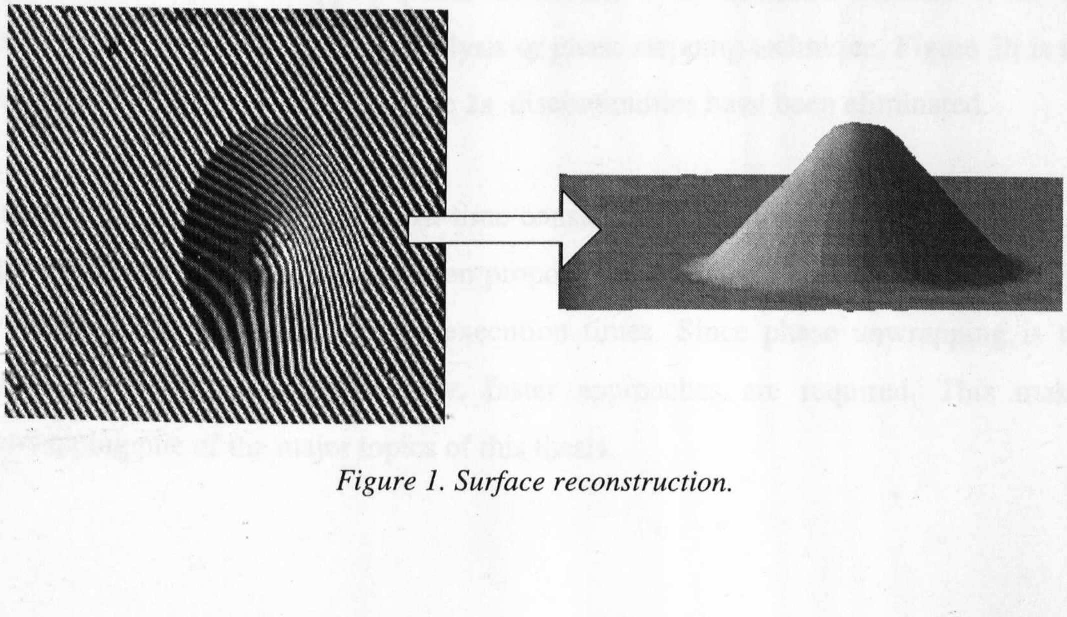
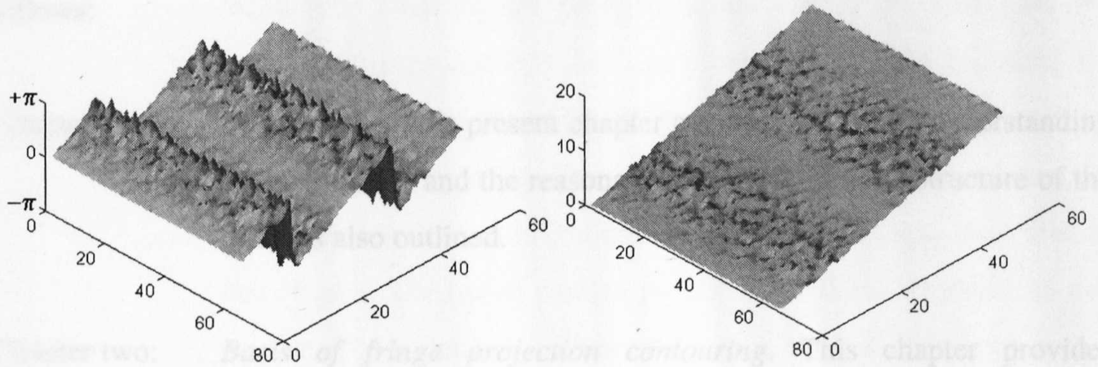


Figure 1. Surface reconstruction.

Fourier fringe analysis (FFA) and *phase stepping* are the two most common techniques used in the analysis of such fringe patterns. Each seeks to obtain the phase of the fringe pattern, information that is directly related to the height distribution of the object by a function that depends upon the parameters of the optical set-up.

A problem common to both techniques stems from the use of the *arctan* function for the calculation of the phase. Since such a function returns an angle, the result will be *wrapped* in the interval $[-\pi,\pi]$ and discontinuities will appear in the result. The process of recovering the desired continuous phase function is called *unwrapping*.



(a) (b)

Figure 2. Unwrapping process. Wrapped phase distribution (a) and Unwrapped phase distribution (b)

Figure 2 illustrates a simple unwrapping example. Figure 2a is a three dimensional representation of a wrapped phase distribution, as obtained directly from the application of the Fourier fringe analysis or phase stepping technique. Figure 2b is the unwrapped phase distribution, where 2π discontinuities have been eliminated.

Unwrapping is, generally, the most time consuming operation in the overall process. Many different unwrappers have been proposed in the past. However, high reliability is normally associated with large execution times. Since phase unwrapping is the bottleneck in fringe analysis, new, faster approaches are required. This makes unwrapping one of the major topics of this thesis.

Fourier fringe analysis is a relatively new technique based upon the Fourier transform concept. Aliasing, leakage and cross-talking, as well as the presence of high frequencies on the fringe pattern will cause inaccuracies in the final result that will concentrate on the borders of the image. The reduction of these factors will potentially increase the precision of the technique.

This thesis addresses issues relating to the Fourier fringe analysis technique. Novel improvements that increase the accuracy of the technique are presented and special attention is paid to the unwrapping problem. The thesis divides into nine chapters as follows:

Chapter one: *Introduction.* This present chapter attempts to give an understanding of the problem and the reasons for the research. The structure of the thesis is also outlined.

Chapter two: *Basis of fringe projection contouring.* This chapter provides information that will be required later in this thesis. A description of fringe projection contouring techniques as well as the equipment utilised throughout this research is given. Fourier fringe analysis is described in detail and the problems associated with the technique are discussed.

Chapter three: *Automatic phase unwrapping.* Common problems to phase unwrapping are described, and a literature review of previous attempts at phase unwrapping is included in this chapter.

Chapter four: *Two novel techniques for two dimensional phase unwrapping.* Two novel approaches to phase unwrapping are presented in this chapter. A number of both simulated and real phase distributions are analysed with the unwrappers. The results are discussed.

Chapter five: *Parallel implementation of cellular automata.* A number of different parallel paradigms for parallelisation of cellular automata are described. A comparison between the different algorithms and two parallel computation languages is undertaken. The most appropriate approach to phase unwrapping by cellular automata is also discussed.

Chapter six: *Avoiding the unwrapping*: A fractional fringe analysis-based technique. This chapter presents a new technique, based on the fractional fringe concept that can potentially remove the need for unwrapping. A description of the technique is given, and its limitations are exposed. The multichannel concept is presented, as well as its application in conjunction with the technique described in this chapter. The use of parallel processing will be introduced in this context. The technique is illustrated with a number of examples.

Chapter seven: *Reducing cross-talking in FFA.* A chapter describing the problems involved if non-fullfield images are to be analysed by Fourier fringe analysis. A pre-processing technique that allows the analysis of these type of images by Fourier fringe analysis is presented. Previous attempts are also described. This chapter also proposes a novel technique that, by using two phase stepped images, produces a single signal peak in the Fourier domain, therefore reducing the cross-talking present in Fourier fringe analysis. Advantages with respect to the FFA in precision are emphasised.

Chapter eight: *Future work.* In this short chapter, a proposal for fast non-contact measurement and a technique with inherent phase unwrapping by using colours are briefly presented.

Chapter nine: *Discussion and conclusions*. This is the last chapter of the thesis, where a global view of the research work undertaken is given. Conclusions are drawn from the results obtained in previous chapters. Guidelines for possible improvements to the techniques presented in this thesis are proposed.

For completeness, papers by the author published in relation to this thesis are included as an appendix.

Chapter two

Basis of fringe projection contouring

2.1 Introduction

The surface reconstruction process consists of two main steps, namely the production of the fringe pattern that is projected onto the object and the analysis of the resulting fringe pattern so that the required information is obtained. In this chapter the fringe pattern production systems utilised throughout the research are described. An introduction to the most common techniques used for the analysis of such fringe patterns, that is Fourier fringe analysis and phase stepping, is also given. Special attention is focused on the former because of its relevance in this research.

2.2 Fringe pattern production

The projected fringes can be produced by either the use of a grating or interferometric techniques. In this section, the optical configurations utilised in the experimentation are explained.

2.2.1 Fringes produced by a grating

If a grating is used, a light source is imaged onto the grating by means of a lens, as illustrated in Figure 3. A second lens is placed between the grating and the object so that the fringes are imaged onto the surface.

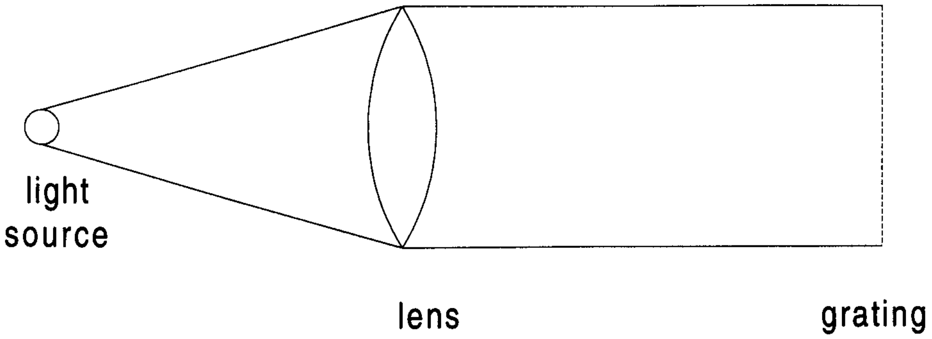


Figure 3. A light source in imaged onto the grating by means of a lens.

A grating based projection system has been set up at the Liverpool John Moores University. This system uses a single sinusoidal transmission grating that can be translated along one of the axes, allowing phase steps to be introduced into the fringe patterns. The arrangement is relatively simple and is illustrated in Figure 4 .

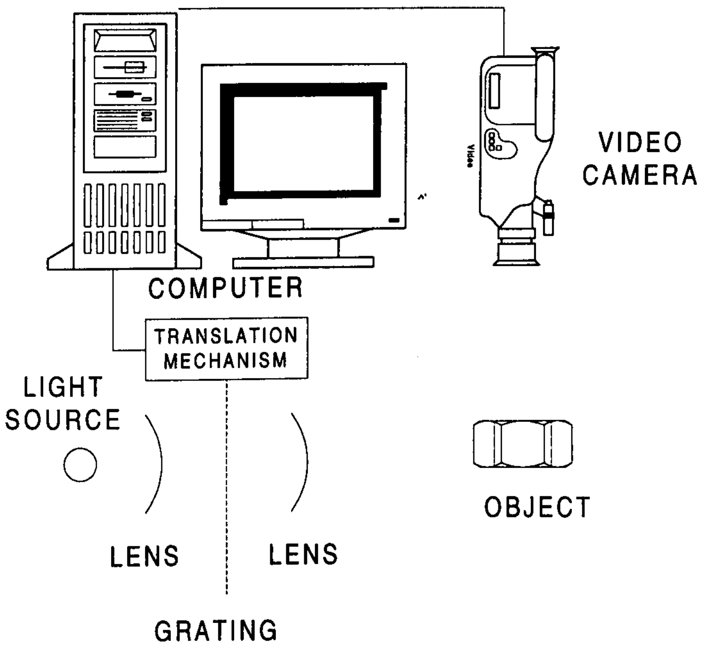


Figure 4. Phase stepping arrangement.

A light source is imaged onto the grating, fringes being projected onto the object surface. The resulting fringe pattern is captured by a video camera. A frame grabber is placed between the computer and the video camera, allowing the former to receive the fringe pattern recorded by the latter. The computer is also connected to an accurate translation mechanism that moves the grating by a distance dependent on the required phase step. A software system manages the equipment so that four fringe patterns with a phase step of $\pi/2$ are produced. An algorithm is applied to calculate the final height distribution.

Rotation of the fringes can be achieved by manual movement of the grating. However, every time a manual intervention occurs, the system needs to be re-calibrated as the translation of the grating between steps has to be recalculated.

2.2.2 Fringes produced by interferometric techniques

The use of interferometric techniques involves the division of a single laser beam into two component beams, that are made to interfere.

An interferometric projection system was also used during the new work described. The system produces Young's fringes that can be projected onto an object surface to produce an interferometric fringe pattern, analysis of which will lead to obtaining the final height distribution of the illuminated surface. The framework for this system is illustrated in Figure 5.

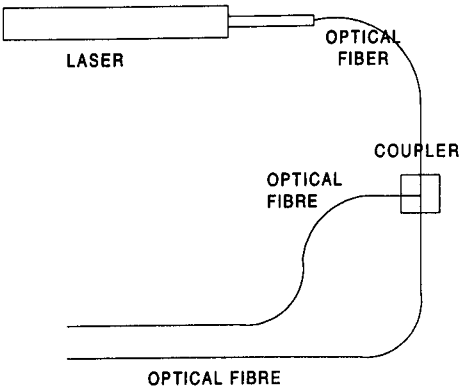


Figure 5. Optical set-up for the interferometric system

The laser is focused on one of the extremes of an optical fibre cable. A second optical fibre is joined to the first optical fibre, the original beam being divided into two beams. These two fibre optic cables carrying the two beams act as the two pin holes in the Young's experiment (Figure 6), producing what are called Young's fringes.

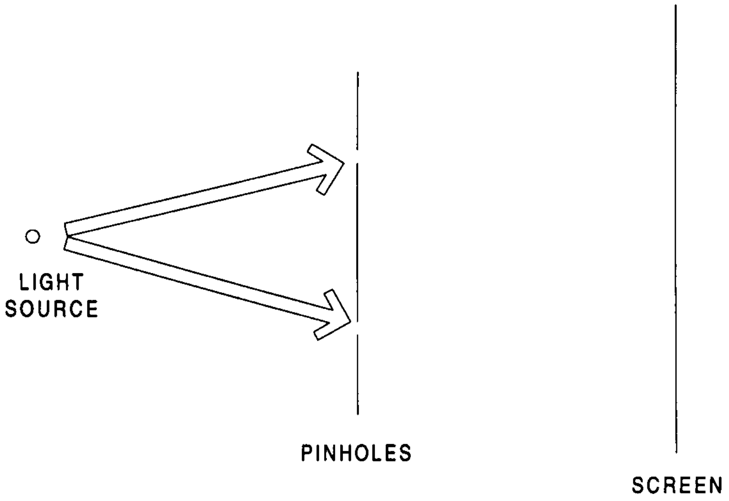


Figure 6. Young's experiment

One of the optical fibres can be moved in slight steps by a computer-controlled mechanism. A translation of this optical fibre with respect to the fixed one as illustrated in Figure 7a will vary the distance between the fibres, therefore inducing a change in the fringe spacing. A sidewards movement of the optical fibre as shown in Figure 7b will produce a variation on the fringe orientation.

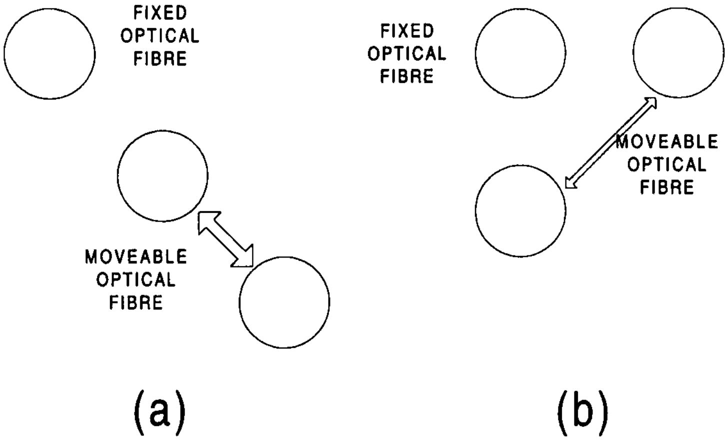


Figure 7. Movement of the optical fibres.

The two pinholes in Figure 6 act as secondary sources causing light interference that produces fringes visible on the screen. These fringes are projected onto the object's surface which is to be analysed. The fringe pattern produced is captured by a camera that is connected to a frame grabber by a set up such as illustrated in Figure 8. The data is analysed by a computer system that automatically produces the height distribution corresponding to the object's surface. The computer is also able to control the position of one of the optical fibres of the interferometer, allowing the fringe pattern projected to be rotated or the number of fringes varied.

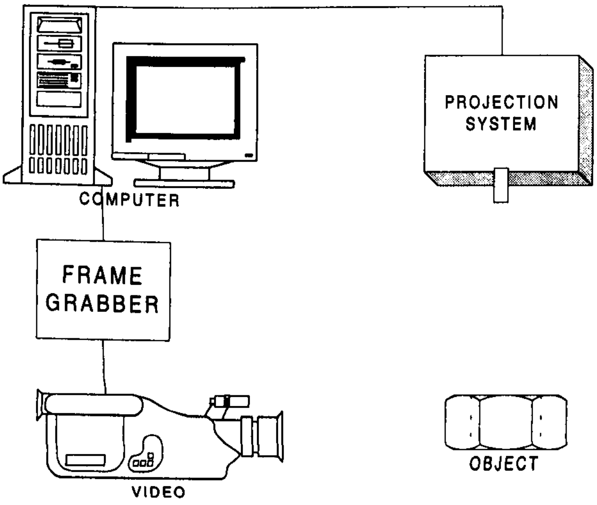


Figure 8. The interferometric system.

This interferometric projection system produces a single fringe pattern that is normally analysed by the FFA technique.

2.3 Fringe pattern analysis

Despite the existence of many established fringe pattern analysis techniques, only two of them, phase stepping and Fourier fringe analysis have become widely used. In this section, a short review of the existent techniques is given. For a more detailed review of these methods, the reader is referred to (Reid 1986). Fourier fringe analysis is the focus of the discussion because of its special relevance to the research work undertaken.

2.3.1 Fringe tracking

The most obvious technique for analysing fringe patterns is the so called fringe tracking method (Button et al. 1985; Cline et al. 1982; Yatagai and Idesawa 1982; Yatagai et al. 1984). This approach identifies the fringes and tracks the maxima and minima by an intensity based algorithm. Image processing techniques are used so as to facilitate this operation. Once the maxima and minima are determined, the rest of the image is interpolated. The main drawbacks of this technique are:

- The generally low speed of the process.
- The human interaction required in most of the cases.
- The poor data distribution obtained (specially in one direction).

2.3.2 Heterodyne interferometry

Heterodyne interferometry (Dandliker and Thalmann 1985; Indebetouw 1978; Perrin and Thomas 1979; Reid 1983) uses detectors to determine the phase electronically at a single point. By translating a fringe pattern over the object, the phase of the signal at selected points on the surface can be determined. The phase is then converted to height.

The main problems associated with this technique are:

- The phase is only determined at a single image point. If the measurement of an area is required, then the detector has to be scanned or several detectors have to be used.
- They are very sensitive to vibration and turbulence.
- High cost.

2.3.3 Phase stepping

Phase stepping (Bruning et al. 1974; Carré 1966; Cheng and Wyant 1985; Halioua et al. 1985; Robinson and Williams 1987; Srinivasan et al. 1984; Vrooman and Maas 1989b) allows phase measurement by acquiring a number of images with discrete shifts in the position of the fringes.

A single fringe pattern can be mathematically expressed by the equation such as

$$I(x,y) = a(x,y) + b(x,y)\cos(\phi(x,y))$$

where $a(x,y)$ comprises the background variations, $b(x,y)$ describes variations in the fringe visibility and $\phi(x,y)$ is the phase of the fringe pattern, which in fringe projection is related to the height distribution of the object.

In this equation, the phase is the information of interest. The unwanted terms $a(x,y)$ and $b(x,y)$ need to be eliminated. For fixed values of x and y, there are three unknowns variables. If at least two other equations could be obtained so that an independent set of at least three equations were available, the desired information $\phi(x,y)$ could be determined.

Several phase stepping algorithms have been proposed differing in the number of equations (fringe patterns acquired) and the phase step introduced amongst fringe patterns. A common algorithm makes use of four fringe patterns with a phase shift of $\pi/2$ between them. Four different equations are then obtained, namely:

$$I_1(x,y) = a(x,y) + b(x,y)\cos\left(\phi(x,y)\right)$$

$$I_2(x,y) = a(x,y) + b(x,y)\cos\left(\phi(x,y) + \frac{\pi}{2}\right)$$

$$I_3(x,y) = a(x,y) + b(x,y)\cos\left(\phi(x,y) + \pi\right)$$

$$I_4(x,y) = a(x,y) + b(x,y)\cos\left(\phi(x,y) + \frac{3\pi}{2}\right)$$

The phase of the object is finally determined by the expression:

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

The major benefits of phase stepping can be listed as follows:

- Calculations are straight forward and can be performed very quickly on a standard processor such as a desktop computer.
- It provides far better accuracy than intensity based methods such as the fringe tracking technique.

However, several drawbacks are also inherent in the technique:

- It requires extended hardware to introduce the phase steps. The optical set-up is more complex and the calibration has to be carried out very carefully so that the desired phase steps are introduced. Inaccurate introduction of phase steps is one of the main source of errors of the technique.
- The technique is more sensitive to vibration and turbulence than the FFA method described below. Better isolation is therefore required and additional storage is needed. The optical equipment is hence more expensive.

A complete review of phase stepping techniques and its accuracy can be found in (Creath 1988) and (Creath and Schmit 1994). Many compensating algorithms have been proposed in order to minimise these errors (Hariharan et al. 1987; Oreb 1996; Schwider et al. 1993).

2.3.4 Fourier fringe analysis

Fourier Fringe Analysis (FFA) was introduced by Takeda *et al.*(1982;1983). The technique has also been studied by, amongst others, Burton *et al.* (1988), O'Donovan *et al.* (1990), Halsall (1992), Bone *et al.* (Bone 1991; Bone et al. 1986), Macy (1983), Nugent (1985) and Kujawinska *et al.* (1991).

By introducing a carrier frequency, the technique produces the phase of the fringe pattern on the object utilising a single fringe pattern.

The intensity data contained in a fringe pattern can be written as

$$I(x,y) = a(x,y) + b(x,y)\cos(2\pi f_x x + 2\pi f_y y + \phi(x,y)) \tag{2.1}$$

where,

$a(x,y)$ comprises the background variations,

$b(x,y)$ describes variations is the fringe visibility,

$\phi(x,y)$ is the phase of the object,

f_x is the carrier frequency on the x-axis and

f_y is the carrier frequency on the y-axis.

This equation can be rewritten as

$$I(x,y) = a(x,y) + c(x,y)e^{i(2\pi f_x x + 2\pi f_y y)} + c^*(x,y)e^{-i(2\pi f_x x + 2\pi f_y y)}$$

where $c(x,y) = \dfrac{b(x,y)e^{i\phi(x,y)}}{2}$ and $c^*(x,y)$ represents the complex conjugate of $c(x,y)$.

When the Fourier transform of the fringe pattern is performed, three peaks are obtained: the Fourier transform of the term $a(x,y)$, placed in the centre of the spectrum and the Fourier transforms of $c(x,y)$ and $c^*(x,y)$ that will be symmetric with respect to the centre and placed at a distance that is determined by f_x and f_y.

The Fourier transform of $c(x,y)$ may be isolated and, by inverse Fourier transforming the peak, the term $c(x,y)$ itself will be obtained. It is then only necessary to divide the imaginary part by the real part and perform an *arctan* operation over the result to obtain the phase of the fringe pattern. The whole Fourier fringe analysis process is illustrated in Figure 9. Carrier removal can be undertaken at different stages of the process.

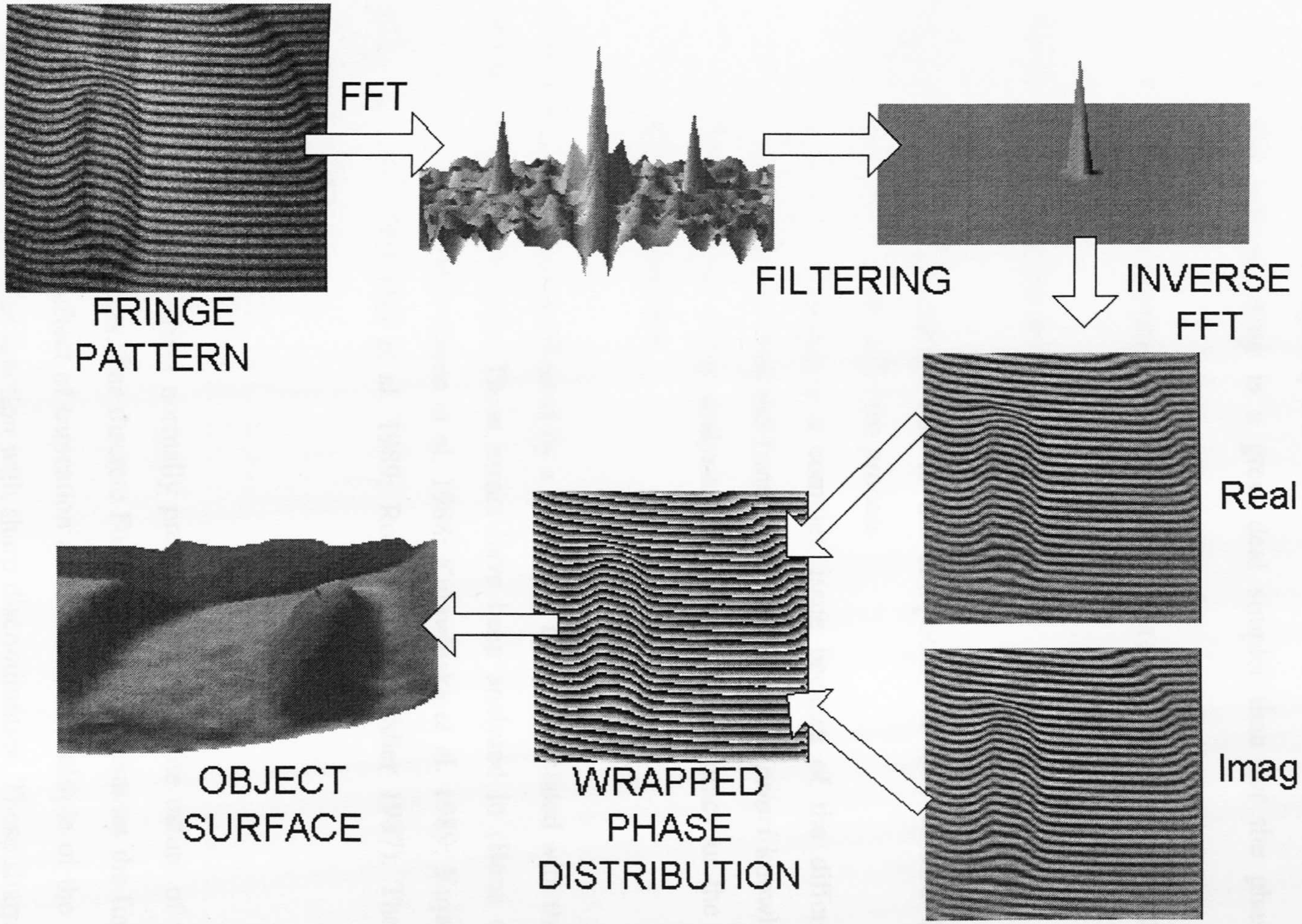


Figure 9. Fourier Fringe Analysis process.

The main advantages of Fourier fringe analysis can be summarised as follows:

- It requires the acquisition of a single fringe pattern, therefore reducing the amount of storage required.
- The optical set-up is a great deal simpler than for the phase stepping technique.
- Low frequency noise is eliminated and has no influence in the result.

The main disadvantages are:

- The slow processing due to the computational load of two fast Fourier transforms involved in the process.
- Automatic filtering is a complex issue because of the different Fourier spectra that are obtained from different images. *A-priori* knowledge of the type of object under analysis can increase the accuracy of the result by a significant amount.

Fourier fringe analysis is affected by a number of errors associated with the use of the discrete Fourier transform. These errors have been analysed in (Bone et al. 1986; Frankowski et al. 1989; Green et al. 1988; Kujawinska et al. 1989; Kujawinska and Wójciak 1991; Malcolm et al. 1989; Roddier and Roddier 1987). The errors are briefly described:

- **Leakage**

  Leakage is an effect normally produced by the finite nature of the discrete Fourier transform. The discrete Fourier transform assumes the function to be periodic. The effect of truncation at other than a multiple of the period is to create a periodic function with sharp discontinuities. These sharp changes in the truncated signal result in additional components in the frequency domain that appear as leakage in the Fourier spectrum (Brigham 1974). Two different truncations for a periodic wave are illustrated in Figure 10. The Fourier transform operation assumes periodicity and, therefore, the first truncation results in the original function. On the other hand, the second truncation produces a different signal, introducing undesired high frequencies.

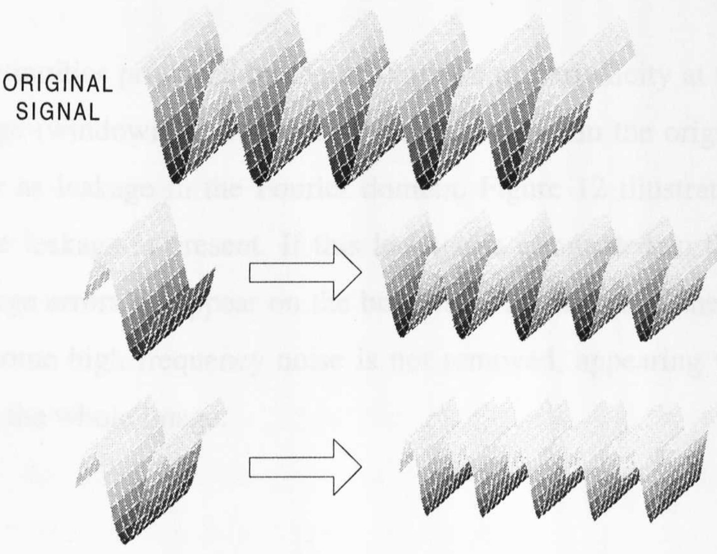


Figure 10. Windowing of a periodic signal. The first truncation results in the original signal, the second produces a signal that differs from the original significantly.

This effect is also illustrated in Figure 11, where the second truncation produces leakage in the power spectrum.

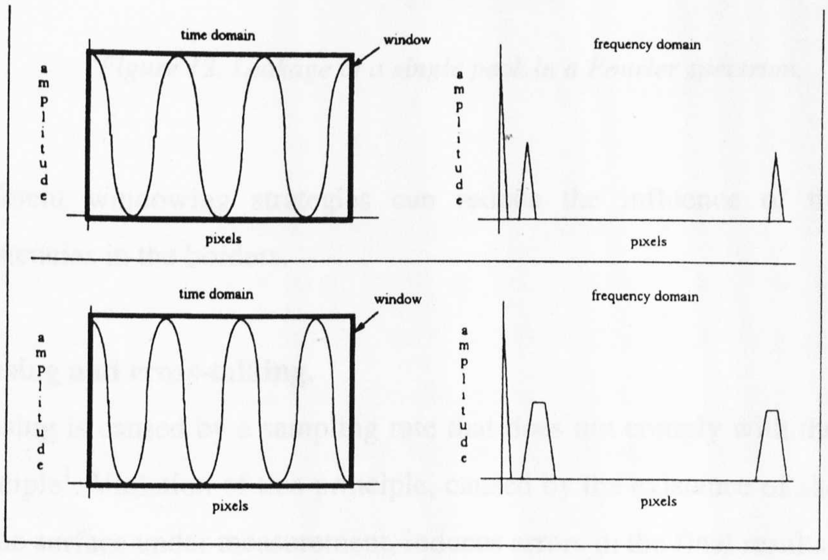


Figure 11. Fourier spectrum of truncated functions.

A fringe pattern is not a periodic function in the sense it has been defined above. If a single line or column is isolated, there may be significant differences in value between the first and last point of the line.

The discontinuities produced by the assumption of periodicity at the borders of the image (windowing), introduce high frequencies in the original image, that appear as leakage in the Fourier domain. Figure 12 illustrates a single peak where leakage is present. If this leakage is eliminated in the filtering process, large errors do appear on the borders of the image. If the leakage is included, some high frequency noise is not removed, appearing widespread throughout the whole image.

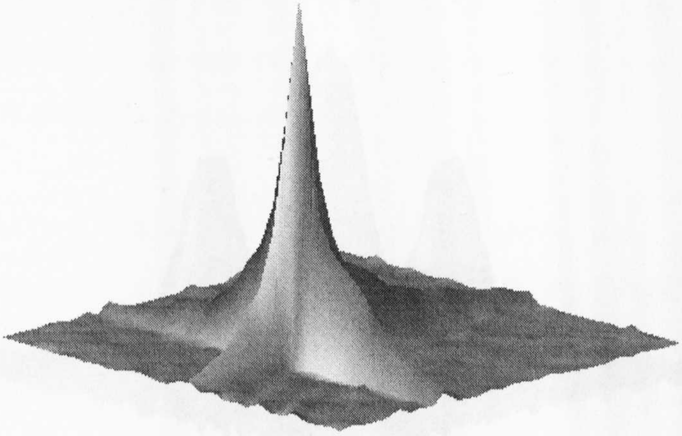


Figure 12. Leakage of a single peak in a Fourier spectrum.

Different windowing strategies can reduce the influence of these high frequencies in the borders.

- **Aliasing and cross-talking.**
  Aliasing is caused by a sampling rate that does not comply with the Nyquist principle[1]. Violation of this principle, caused by the existence of sharp edges on the surface under measurement, induces errors in the final results.

[1] The sampling frequency has to be larger than twice the maximum frequency present in the sampled signal.

Cross-talking is produced by the interference of two or more additive signals in the Fourier spectrum. Two additive signals, the phase information and the d.c. terms, are present in a fringe pattern.

A key issue in Fourier fringe analysis is an accurate detection and isolation of the signal peak. However, three different peaks co-exist in the Fourier domain, as already mentioned. Interaction or influence between these three peaks may introduce errors if they interfere as in the example illustrated in Figure 13. In this case, the isolation of a single peak is made difficult by the interference produced by the other two peaks. If a tight filter is used, frequencies that belong to the peak are lost. The use of a wider filter will introduce frequencies that do not belong to the peak.

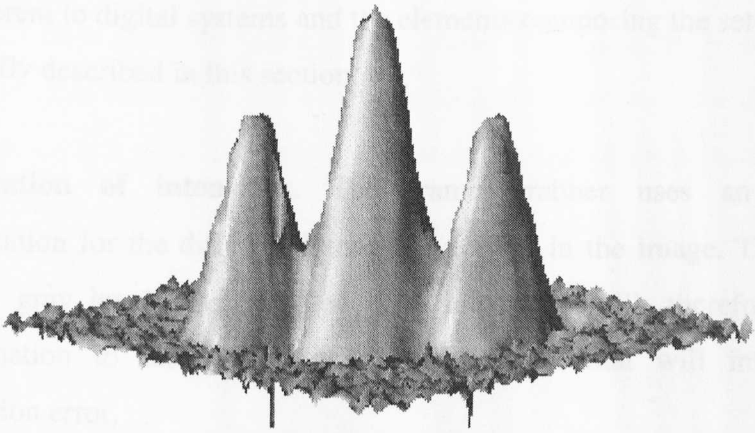


Figure 13. Example of peaks interfering each other in the Fourier domain.

Cross-talking and leakage often appear together, hindering the filtering process and introducing large errors at the borders of the resulting phase distribution. Figure 14 depicts such a case (effects are exaggerated for illustration purposes). This double effect can be reduced by the use of tilted fringes, causing the peaks to locate further apart from each other.

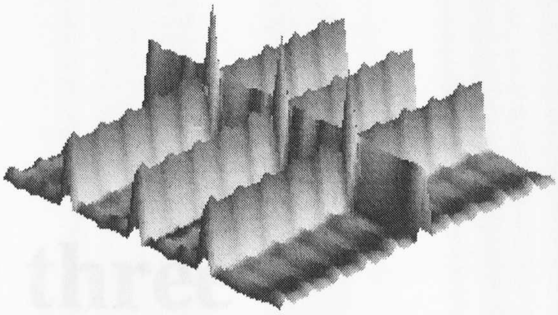


Figure 14. Leakage and cross-talking occurring at the same time.

2.4 Problems associated with the physical systems

Apart from the errors introduced by the use of one of the previous techniques, some problems are inherent to digital systems and the elements composing the set-up. These problems are briefly described in this section:

- **Digitalisation of intensities**. The frame grabber uses an eight-bit representation for the different intensities present in the image. This means that 256 gray levels are recorded. The intensities will, therefore, be an approximation to the nearest gray level, a fact that will introduce a quantisation error.
- **Rounding errors**. The limited representation of numbers in the computers can introduce slight errors in the results. These round-off errors can accumulate.
- **Determination of the optical parameters**. The non-exact determination of the projection angle and the tilt introduced by the fringes introduces errors in the final result.
- **Internal sources of noise are present in any electronic system**. e.g. camera.
- **Non linear response with intensities**. The camera does not respond linearly to the input intensities. The captured image will only be an approximation to the real image.

Chapter three

Automatic Phase Unwrapping

3.1 Introduction

Fringe analysis has been successfully applied to many different practical issues, strain measurement, fluid flow and three dimensional object measurement among others. One of the main interests focuses on the automatisation of the process to enable its use in industrial measurement systems. Therefore, the existence of an automatic phase unwrapping algorithm is essential.

In the two techniques introduced in the previous chapter, the phase is obtained by means of the arctan function. However, this mathematical function returns values that are known between the limits $\pi$and $-\pi$. Hence, the result is given modulo 2π and discontinuities of values near to 2π do appear. Unwrapping is the process by which these discontinuities are solved, the result being converted into the desired continuous function.

The unwrapping problem has been a main topic of research for the last decade. Many different algorithms do exist, but a correct solution is not guaranteed and very long execution times are often involved. At first sight, phase unwrapping seems a relatively simple problem. It actually is... if one deals with perfect simulated data. When one is concerned with real images, procedures have to be designed to deal with many different problems.

In this chapter these problems will be described, and some of the existing algorithms will be reviewed.

3.2 Problems

In this section, the problems related to phase unwrapping are described. The purpose of this section is to convince the reader that two dimensional phase unwrapping is not a simple task and that, in many cases, several possible solutions co-exist, one being as valid as the others.

The main problems involved with the unwrapping process are: noise, masked areas, undersampling, fringe breaks, holes, and choice of the threshold value. These problems are now detailed independently in more depth.

- **Noise**. Noise is a part of the real word, and it appears on images in different forms. Noise can create confusion when detecting the 2π discontinuities in the phase map.

  If we look at Figure 15, we observe a case where a one dimensional unwrapper that works just by comparing differences between consecutive points fails. Because of the noise present in (d), the unwrapper detects a second wrap that does not exist.

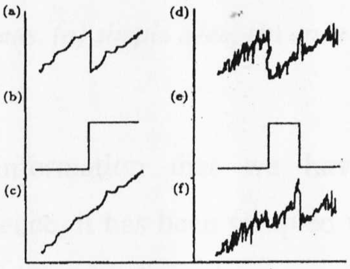


Figure 15. Incorrect unwrapping due to noise. (a) original signal - the detected discontinuities are shown in b). (c) represents the unwrapped data. If high amplitude noise is added (d), false detection of wraps can occur (e), an the incorrect unwrapping (f).

Low modulation points act in a very similar manner to noise. In addition, low modulation points tend to concentrate in the areas where wraps occur, a fact that could partially hide the existence of a wrap.

- **Masked areas**. Some noise areas can be detected before proceeding with the unwrapping. These areas are normally masked (they are set so that the unwrapper can detect them). The unwrapper has to be able to cope with masked areas in such a way that they are not taken into account when the actual unwrapping process is being carried out. The defects can have many different shapes and methods for reacting to these areas have to be developed. Figure 16 illustrates some different shapes for masked areas. (a) shows just a simple masked area; (b) shows a somewhat more complex masked area in which the unwrapper has to access the non-masked interior part; (c) is the most complex case, since a small area is isolated from the rest of the image. The unwrapper has to be able to carry out this isolation.

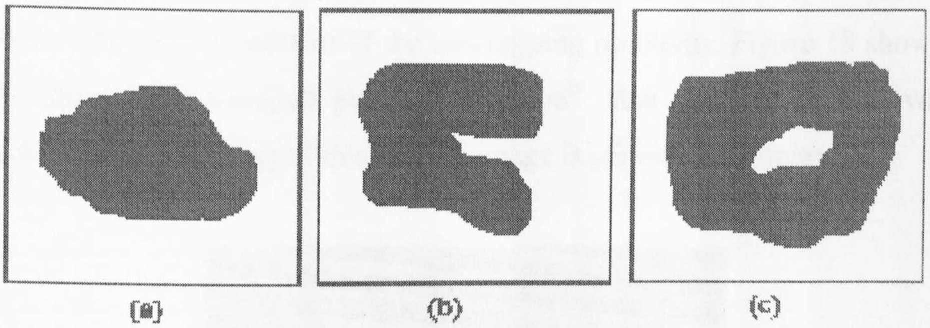


Figure 16. Masked areas. (a) simple area, (b) open area, (c) closed area.

- **Undersampling**. The information that we have (the wrapped phase distribution) is digital. Hence, it has been sampled with a certain frequency. This frequency has to fulfill several requirements. If these requirements are not accomplished, problems can arise.

Figure 17 exaggerates this effect in a one dimensional signal, illustrating a low sampling frequency that leads to large differences in phase between neighbouring points. Wraps cannot be distinguished even in the absence of noise.

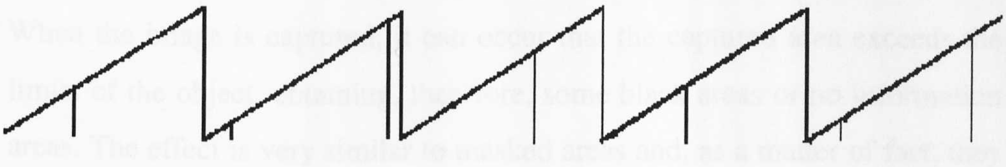


Figure 17. An undersampled wrapped phase distribution in one dimension.

- **Wrap breaks**. In a two dimensional wrapped phase distribution, wraps should be continuous. However, it does happen sometimes that, because of errors in the processing (these errors can stem, for example, from the existence of high frequencies that are reset to zero when using the FFA technique), we obtain either false wraps or discontinuities in true wraps. This is one of the most important of the unwrapping problems. Figure 18 shows a two dimensional wrapped phase distribution[2] that presents several wrap breaks. The unwrapping of this kind of image is generally complex.

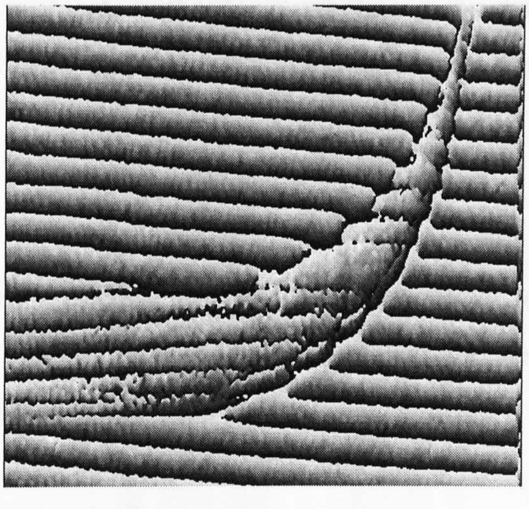


Figure 18. Two dimensional wrapped phase distribution with several wrap breaks.

[2] This figure uses a range image format. The values are represented by intensities. Those points whose phase value is near to π appear bright in the image whilst those whose value is near to -π appear dark.

The main difficulty is to distinguish when a break in a wrap is due to an error in the processing or to real discontinuities in the object surface.

- **Holes**. Holes are areas in the image where the phase is simply not defined. When the image is captured, it can occur that the captured area exceeds the limits of the object, obtaining, therefore, some black areas or no information areas. The effect is very similar to masked areas and, as a matter of fact, they can be pre-processed so that they are masked before proceeding to apply the unwrapping algorithm.

- **Choice of the threshold value**. The usual method to detect discontinuities is based on detecting pairs of neighbouring points whose difference in value is larger than a pre-set threshold value. The choice of this value is image dependent and it is not a simple task.

This section has briefly described some of the main problems that an unwrapper has to overcome. In the next section we undertake a review of current unwrapping techniques. We try to cover most of the unwrappers that have been proposed to date, giving a non-exhaustive description of them. It is important to remark that it is intended to introduce the problem, without overwhelming the reader with detailed information. References will also be included for the unwrappers depicted.

3.3 A review

Automatic phase unwrapping has been, and still is, a main topic of research. In the last few years, many different techniques have been developed and new approaches are continuously being proposed.

This section gives a description of a number of algorithms that have been proposed in the past.

3.3.1 Fixed path scanning based algorithms

Fixed path scanning based algorithms are simple extensions of a basic one dimensional unwrapper to two dimensions. A one dimensional unwrapper proceeds by comparing the phase of the pixel to the phase of the previous pixel in a sequential fashion; if the absolute value of the difference exceeds a certain threshold value (usually between π and 1.5π), 2π is added or subtracted (depending upon the sign of the difference) to the rest of the remaining values.

Fixed path scanning based algorithms are usually very fast, presenting very low execution times. On the other hand, errors propagate due to the sequential nature of the algorithms. This type of algorithm is only appropriate for extremely high quality images with low noise and no wrap discontinuities.

Several variations of this idea do exist. They differ in the manner in which the two dimensional array is converted into a one dimensional array. Some of these techniques are briefly explained below.

3.3.1.1 Linear phase scanning

The two dimensional phase distribution is converted into a single dimensional array as illustrated in Figure 19. The one dimensional array is then unwrapped.

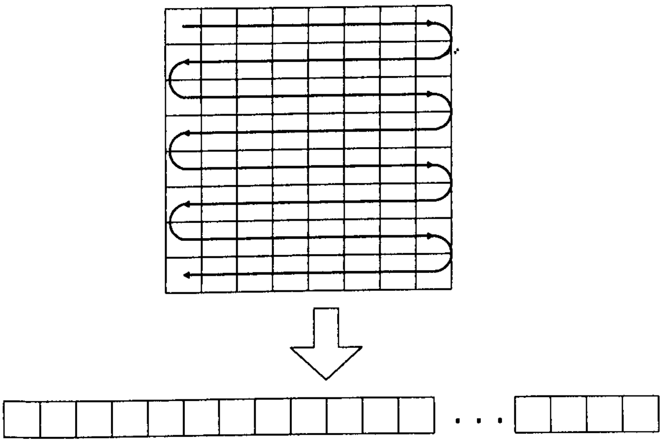


Figure 19. Linear phase scanning.

3.3.1.2 Horizontal scanning

A one dimensional unwrapper is applied to each single line independently. The final unwrapped phase distribution is obtained by using a single vertical trace. The role of rows and columns can also be reversed. The process is illustrated in Figure 20.

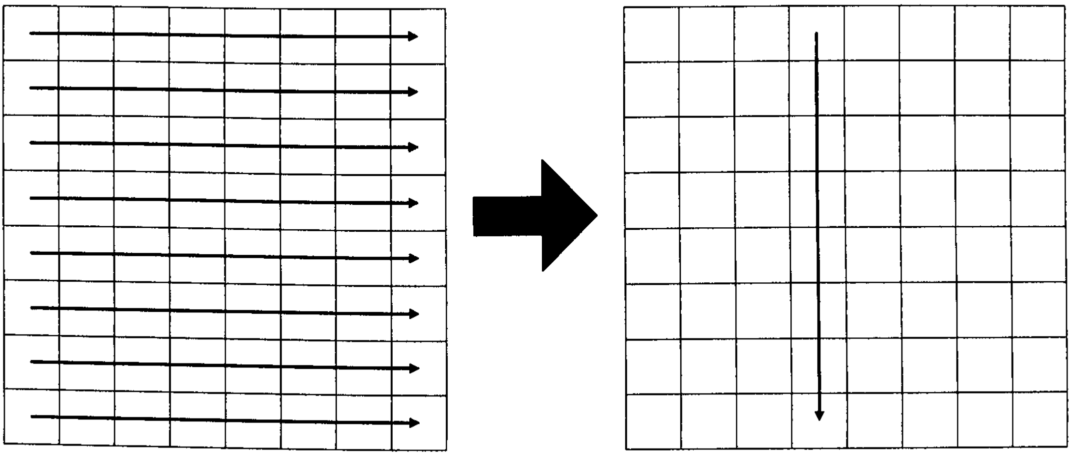


Figure 20. Horizontal scanning.

3.3.1.3 Multiple scan directions

Unwrapping is also carried out by scanning sequentially though the data. However, the unwrapped phase is calculated with respect to two different neighbours that have already been unwrapped (e.g. the one at the left and the one at the top). If the two values disagree, the pixel will be masked as incorrect and the unwrapping process will continue.

The first line unwrapped cannot use this algorithm since no previously unwrapped neighbours exist. A correct unwrapping of the first line is essential for the algorithm to perform. Therefore, the performance of the algorithm is dependent upon the capability of choosing an error free first line. Figure 21 illustrates the paths followed by the algorithm.

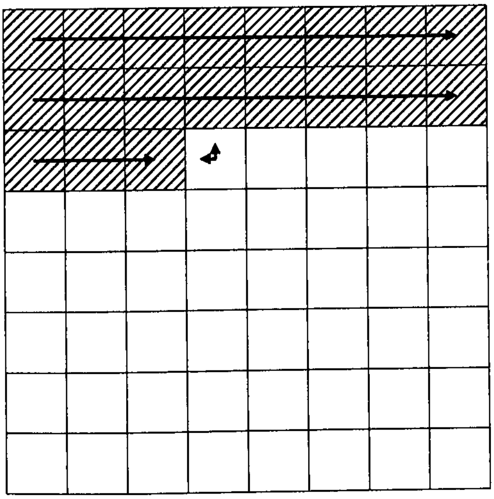


Figure 21. Multiple scan directions.

3.3.1.4 Spiral Scanning

Vrooman and Maas (1989a) used a spiral path in an attempt to produce a more reliable unwrapper. The algorithm starts with a point placed at the centre of the image, where data is generally more reliable. At every point, the algorithm calculates the unwrapped phase relative to all the available neighbours that have been previously unwrapped. The process is carried out in an spiral direction as it is shown in Figure 22.

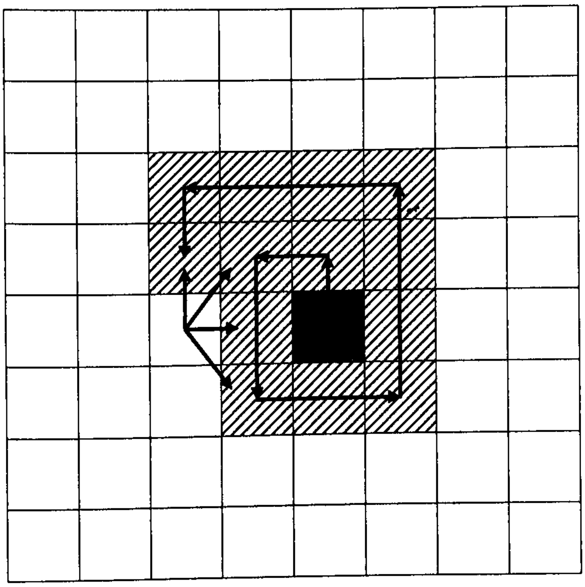


Figure 22. Spiral scanning.

3.3.2 Cut based unwrappers

Goldstein *et al.*(1988) described a noise-immune algorithm based on placing cut lines that will help to unwrap the phase distribution.

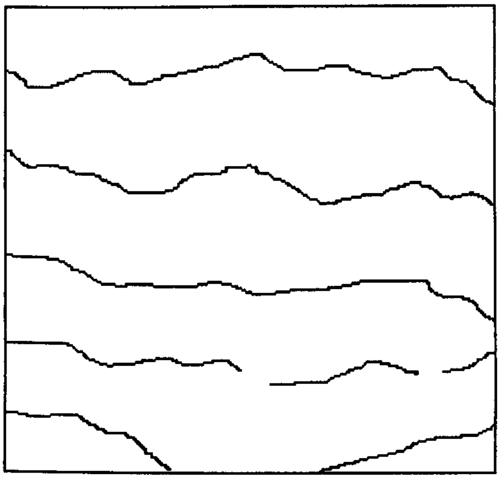


Figure 23. Wraps in a phase distribution.

The lines in Figure 23 represent the differences exceeding a fixed threshold value in the phase distribution (wraps). If no real discontinuities exist on the object's surface, these lines should be continuous. However, wraps can also break due to cross-talking or inappropriate filtering in the Fourier domain when using the FFA technique. An analysis of areas of size two by two pixels in the image will detect the position of the wrap breaks. If an unwrapping operation is performed on the direction illustrated in Figure 24, the number of positive and negative differences exceeding the threshold value must be equal. In other words, if a wrap enters the square, it also has to leave. If this principle does not hold, the area will be marked as a residue, with a sign that is dependent upon the sign of the dominating difference exceeding the threshold value. The algorithm proceeds as follows.

The phase distribution is analysed until a residue is found. A square of size three by three pixels is placed around the residue and another residue is searched. If found, a cut is placed between them. If, however, the sign of the residue is the same as the original. then the box is moved to the new residue and the search continues until either an opposite residue is located and the resulting total cut is unchanged or no residues can be found within the boxes. In the latter case, the size of the square is increased by two and the algorithm repeats from the current starting residue.

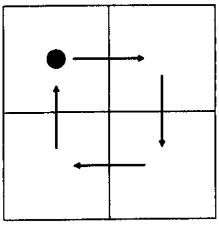


Figure 24. Unwrapping operation to locate residues.

Huntley *et al.*(Huntley 1989; Huntley 1994; Cusack *et al.* 1995) described another technique based upon a similar principle. The idea of path independent unwrapping is used (given the phase at a pixel, the phase at any other pixel should be defined uniquely, independently of the path by which the phase is unwrapped). The cut tracing is, however, more elaborate. Huntley proposed different algorithms to determine the cut lines. These cut lines will act as barriers to the phase unwrapping. Once the cut lines are set, a simple unwrapping algorithm is applied. The only condition is that the unwrapping path must not cross any of the cuts previously determined.

Bone (1991) stated that the main limitation of the Goldstein *et al.* approach stems from the fact that the network of branch cuts joining the residues is not uniquely determined. Any network of branch cuts that satisfies the criterion that the sum of the residues joined by the branch cuts is zero for all branch cuts will result in a phase unwrapping that is consistent in the sense that it is path independent. However, the distinction must be clearly made between a consistent and a correct unwrapping. Consistency does not imply correctness. Masking and cut based algorithms do generally produce a consistent result where no discontinuities greater than π occur. The result, however, may not be a reconstruction of the original phase.

He also stated that without more information than simply the positions and values of the residues, it is not possible to determine uniquely the network of branch cuts that would give the correct phase. He based his unwrapper upon the second differences, instead of the first differences used by Goldstein and Huntley.

The second differences are calculated from the locally unwrapped phase. For a given point *(i,j)*, the phase is unwrapped in a clockwise fashion around the squares formed by the *points (i,j), (i,j+1), (i+1,j+1), (i+1,j)*. Points *(i,j-1)* and *(i-1,j)* are then unwrapped with respect to *(i,j)*:

	• *(i,j+1)*	• *(i+1,j+1)*
• *(i-1,j)*	• *(i,j)*	• *(i+1,j)*
	• *(i,j-1)*	

The second differences are then simply

$$\Delta_{xy}(i,j) = \Delta_x(i,j+1) - \Delta_x(i,j)$$
$$\Delta_{x^2}(i,j) = \Delta_x(i+1,j) - \Delta_x(i-1,j)$$
$$\Delta_{y^2}(i,j) = \Delta_x(i,j+1) - \Delta_x(i,j-1)$$

where

$$\Delta_x(i,j) = \phi(i+1,j) - \phi(i,j)$$
$$\Delta_y(i,j) = \phi(i,j+1) - \phi(i,j)$$

The second differences are calculated for all the points in the field and, where any of them exceed some preset threshold, the point is masked and excluded from the unwrapping process. The unwrapping is then undertaken in a sequential fashion, evading the masked regions.

Winter *et al.*(1993) developed a different technique for phase unwrapping. They stated that the unwrapped result is not corrupted by the existence of regional discontinuities in the saw-tooth-images. The algorithm can be divided in a sequence of steps as follows:

1.- Find the phase jumps, and store them as a binary image.

2.- Find the endpoints of the phase jumps. A dilatation and thinning process removes small faults in the binary image. After bifurcations are eliminated, every pixel with only one neighbour will then be an endpoint.

3.- Produce closed areas. All endpoints are connected with each other by straight lines. All the points that are not separated by the graph of the phase jumps or a straight line are part of an area.

4.- Categorise the pixels within a number of areas.

5.- Unwrap. Starting with the largest area, adjacent areas are sorted in a stack according to the length of their respective common boundaries with the start area. The next layer of areas is then sorted under the same principle, etc. The order of the areas in the stack determines the path of the area based unwrapping.

3.3.3 Cellular automata based unwrapper

An approach based on the implementation of cellular automata was proposed by Ghiglia *et al.*(1987).

Cellular automata are simple discrete mathematical systems that can exhibit complex behaviour resulting from the collective effects of a large number of cells, each of which evolves in discrete time steps according to simple local neighbourhood rules.

As in some of the previously described algorithms, 2 x 2 areas are first examined and, whenever an inconsistency is found, the 2 x 2 region is marked as inconsistent. The cellular automaton will not consider these masked points when the rules are applied.

The algorithm can be described as follows:

1.- Each site looks at its left- and right-hand neighbours (border sites see only one neighbour). The extent of the neigbourhood is one sample in each direction with the exception of border sites. The phase differences are computed (i.e. ϕ_{site} - $\phi_{neigbour}$),

2.- The strength of each neigbour's vote is defined to be equal to the integral number of 2π rad necessary to wrap the respective phase differences.

3.- The integer strengths of each vote are accumulated, and the site is changed in value by 2π rad in a direction appropriate to the accumulated strength-of-vote.

4.- If neither neighbour differs by more that π rad from the current site, no change is made to the site value. If a tie strength of vote results, then the site is changed by +2π rad.

The main problem associated with the technique is the large execution time involved in the process. Hundreds of full field iterations are usually required.

This algorithm has also been analysed and modified by Spik *et al.*(1991). They involved the consideration of all neighbours in contrast to the orthogonal neighbours considered by the first algorithm. The marking process is also applied at the end of every global iteration instead of just at the beginning of the algorithm. This allows large-scale path inconsistencies to be resolved. They implemented the algorithm on a linear array processor consisting of 256 separate processors performing the same instruction on different data points. Even if the parallel implementation reduces the execution time by an important factor, it would still be too large in comparison to the time required to perform the rest of the fringe analysis process.

Servin *et al.* (1994) proposed a transformation of the unwrapping constraints into an energy function to be minimised. A deterministic dynamic system will lead to a minimum of this energy function.

Energy relations are established so that the only set of functions that minimise them are smooth continuous functions; i.e. continuous unwrapped phase. The updating rules are simply a deterministic gradient descent of this energy function.

Marking of phase inconsistencies or determination of cut lines are not required by the algorithm.

This last approach is considerably faster than the previous cellular automaton based approaches. The authors report that for a phase distribution containing little noise, the convergence cost of the algorithm is a single iteration. When a large number of inconsistencies are present, the proposed technique may typically take 30 to 60 whole field iterations to converge.

3.3.4 Ordering and queuing algorithms

A different approach to unwrapping is the treatment of points with high reliability first, leaving the most unreliable points to be processed last.

The key issue in this type of algorithm is the criterion used to determine the reliability of a point. This criterion is usually based upon the gradients or differences between a pixel and its neighbours. Those points with the lowest module 2π gradients with respect to their neighbours are determined to be the best points and, therefore, are processed first.

One of the major problems associated with this group of techniques is the additional memory required to store the order in which the pixels are going to be processed.

The first technique belonging to this category was proposed by Kwon *et al.*(1987). The first step of the algorithm is a search for a starting point by means of a weighting process. Even if the starting pixel is chosen to be at the centre of the image, where data are generally more reliable, the determination of this point is crucial to the algorithm. Therefore, the use of the weighting process is preferred.

The algorithm can be outlined as follows.

```
1.- Search for the best pixel.
2.- Introduce the pixel into the list.
3.- While pixels in the list.
    3.1.- Take the first pixel in the list.
    3.2.- Unwrap the pixel.
    3.3.- Add neighbours to the list according to the good pixel criteria.
```

Schörner *et al.*(1991) based their algorithm upon a similar concept. In this case, the computer begins by comparing all the neighbourhoods in the wrapped phase distribution and sorting them according to the best pixel criterion described above. The edges are formed into a network as illustrated in Figure 25, and each point is compared with its neighbours to the right and above. A table is then generated in which these edges are arranged in such a way that those with the smallest values are at the top and the values increase towards the bottom of the table. This table is used to generate an interpretation path that conforms to the principle of the 'minimal spanning tree'. With this technique, those points that have large edge values are not resolved until the end, hence errors are restricted to small areas.

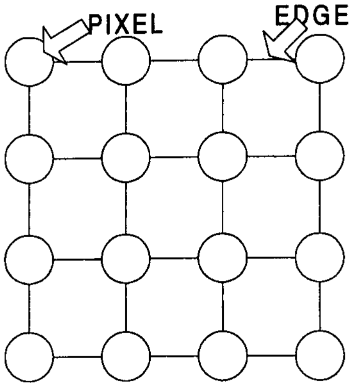


Figure 25. Illustration of pixels and edges.

Vrooman *et al.*(1991) used a queue structure to implement their algorithm so that it evaded masked areas. A check for consistency with the neighbours was also implemented so that errors could be detected and corrected as the unwrapping process is carried out.

Another technique recently proposed by Quiroga *et al.*(1994) proceeds by using a queue structure to store the points. For each point in the region of interest the phase is unwrapped with respect to a special point, called the integrator pixel. Any point, once processed, can be the integrator pixel. The algorithm performs as follows.

1.- The starting pixel is user selected and stored as an integrator pixel.

2.- Introduce the non-masked and non-processed P connected neighbours with lower gradients into the queue, sorted according to their modulo 2π with respect to the integrator. P can be any value permitting the complete processing of the image by the algorithm.

3.- Extract the first point from the queue. If it is empty the algorithm finishes.

4.- If the integrator and the pixel extracted are not connected neighbours, the processed connected neighbour with the lowest gradient modulo 2π is selected as the new integrator.

5.- The current pixel is unwrapped.

6.- Return to (2).

A modified selection rule was also proposed to avoid error propagation. If there is a pixel whose modulo 2π value is always excluded from the P selected neighbours at stage 2 in the previously outlined algorithm, it will be considered as invalid and will not be processed at all.

Quiroga *et al.*(1995) developed this algorithm further using second differences instead of gradients, obtaining a better detection of high curvature areas of the phase map that are usually associated to logical inconsistencies or discontinuities resulting in integer fringe shifts. Points with a second difference greater than a threshold value were rejected, but unlike in their first algorithm, this was an adaptive threshold that varied as the unwrapping process was carried out. The variation of the threshold is imposed by the local characteristics of the phase map. The authors state that the algorithm adapts itself to the local characteristics of the area that is processed and hence it overcomes the problem of considering too many or two few pixels in zones with different amounts of noise.

3.3.5 Division of the image based algorithms

Another approach to phase unwrapping is the division of the image into smaller areas. These areas are then unwrapped independently and brought together by a joining algorithm.

Division of the image based algorithms are normally very fast, but they normally fail at detecting large discontinuities in the wraps.

3.3.5.1 Unwrapping by regions

The first of this type of algorithms was presented by Gierloff (1987). The algorithm groups phase data into regions containing no phase ambiguities. The edges of the regions are then compared to group the numerous smaller regions into larger, simpler regions if there again is no ambiguity between regions. The edges of these large regions are then compared again to see if a given region should be phase shifted with respect to other regions. Logic is included to make the algorithm robust enough to handle logical inconsistencies in phase. All possible relationships between regions can be investigated without requiring large memory overhead due to use of memory conserving logic. Gierloff states that the decision whether a point should be phase shifted potentially may be affected by all other points in the array. The process is illustrated in Figure 26.

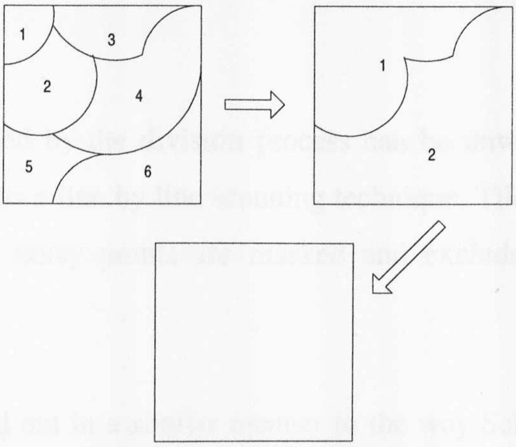


Figure 26. Unwrapping by regions. An image is divided into six different regions, that are grouped together to form two regions. Finally, the two regions are phase shifted to provide the unwrapped image.

This approach does not guarantee path independence, and does not cope with the problem of discontinuities on the wraps.

3.3.5.2 Tiled unwrapping

Towers *et al.*(1989; 1991) introduced the concept of tiled unwrapping that has also been described by *Judge et al.*(1992). In this technique, the image is divided into smaller square tiles, that are unwrapped independently. Tiles are then brought together to produce the unwrapped phase distribution of the entire image. Tile division is illustrated in Figure 27.

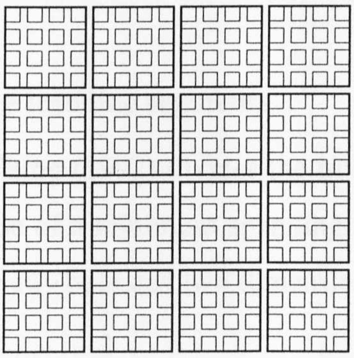


Figure 27. Division of an image into tiles. The small squares represent single pixels. Tiles composed of 16 points are illustrated in darker squares.

The small tiles produced by the division process can be unwrapped by means of a simple algorithm such as a line by line scanning technique. Tiles with a large number of low modulation or noisy points are masked and excluded from the assembly process.

Tile assembly is carried out in a similar manner to the way Schörner *et al.* processed the pixels in the image. The idea of a minimum spanning tree is also applied. The assembly process gives preference to those tiles with a low fringe density and having a high percentage of edge pixels agreeing on the relative phase difference between the tiles.

The advantage of this method is that whenever a serious error is detected on a single tile then the tile is masked and the propagation of the error though the rest of the image is avoided. However, to mask the whole tile means to completely lose all the information contained in it, hence sacrificing some useful data. This could be considered acceptable when the size of the tile is relatively small, but if we deal with bigger tiles then the masking of a single one of those tiles could mean the masking of an important percentage of the whole image. On the other hand, this algorithm is able to detect discontinuities in the wraps as long as they are smaller than the size of the tile. Thus, the choice of a small size can lead to discontinuities not being detected whilst the choice of a greater size will lead to over-masking.

3.3.5.3 Division of the image into line segments

The technique proposed by Ching (1992) divides the image into linear segments, that are brought together by the use of a minimum spanning tree. It uses a region growing approach to unwrap phase for any bounded two dimensional data.

The algorithm performs in four steps:

1.- Segmentation of the image to identify connectivity
2.- Phase unwrapping within each connected segment
3.- Phase unwrapping across disconnected segments
4.- Filling voids in the phase maps

3.3.6 Least-squares minimisation approach

Ghiglia *et al.*(1994) presented an approach based upon the use of fast cosine transforms. Three algorithms were proposed. They stated that phase noise, data inconsistencies, and other degradations are automatically accommodated by the least-squares formulation without manual intervention. All the algorithms use an efficient implementation of the fast two dimensional discrete cosine transform as the basis for the least-squares unwrapping.

3.3.7 Temporal phase unwrapping

Huntley *et al.*(1993; 1994) proposed a completely different approach to the ones described so far. Their algorithm was based upon a one dimensional unwrapping along the time axis. In this technique, the phase at each pixel is measured as a function of time as shown in Figure 28. Unwrapping is then carried out along the time axis for each pixel independently of the others. The authors stated that boundaries and regions with poor signal to noise ratios do not adversely influence good data points.

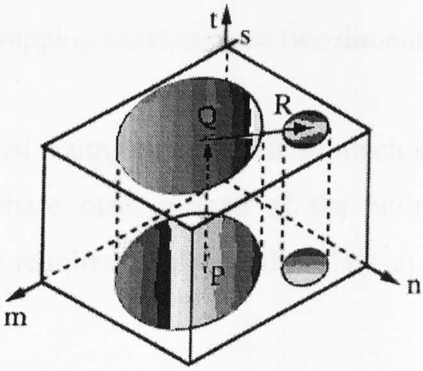


Figure 28. Temporal phase unwrapping technique. Unwrapping is carried out along the time axis t (P to Q), avoiding phase boundaries and noise present by spatial unwrapping (Q to R).

The algorithm is applicable to an important subclass of interferometric applications, in which one is interested in phase changes (rather than absolute phase values) occurring over time and in which a sequence of incremental phase maps can be obtained leading up to the final phase-difference map of interest as would be the case for deformation analysis, by moiré or speckle interferometry, in which the phase change is proportional to surface displacements.

The main advantages of the algorithm are its simplicity and the fact that it is guaranteed to produce a correct unwrapping of phase maps containing physical boundaries of the object, provided the boundaries do not change with time.

Huntley *et al.*(1996) extended the technique to be used for surface profile measurements. Using a conventional projected-fringe phase-stepping surface profilometer, a sequence of phase maps is captured. Between each phase map, the pitch of the projected fringes is changed. The phase at each pixel is then unwrapped over time, independently of the other pixels in the image, to provide an absolute measure of surface height.

3.3.8 Unwrapping by neural networks

Takeda *et al.* (1993) showed that neural networks can also be applied to the resolution of the unwrapping problem. They used a neural network consisting of a large number of neurons to perform unwrapping operations on two dimensional data.

Tipper *et al.*(1995) described a simpler approach. A much smaller network is used to process sections of the phase map, instead of the entire image as in Takeda's approach. The training time required is hence reduced by an important factor.

3.3.9 Other work on unwrapping

Carter (1992) developed an algorithm for unwrapping two dimensional phase data and produced perfect contour maps without holes or dark bands of erroneous contour lines. The technique uses local, temporary unwrapping within single grid squares of rectangularly sampled data, and the addition of extra contour levels outside the 0 to 2π range.

Another algorithm was proposed by Hedley *et al.*(1992) for Magnetic Resonance Imaging (MRI) images. The technique simply masks inconsistent points and unwraps around them. The algorithm was able to handle multiple objects in the image by a simple scanning for objects procedure. The main defect of the technique is the lack of ability to deal with discontinuities in the wraps.

Burton *et al.*(1994) have developed a technique that can serve as an aid to phase unwrapping. They introduced the multichannel Fourier fringe analysis technique and applied it to the problem of automatic phase unwrapping in the presence of surface discontinuities. Multiple fringe patterns can be separated in the frequency domain as long as they present different carrier frequencies. Since they would be placed at different locations in the frequency spectrum, they can be easily isolated. They based their technique upon the use of the full Sobel operator and the Hough transform, being able to distinguish between a real surface discontinuity and a wrap.

Burton *et al.*(1995) also described the use of carrier frequency shifting of the first order peak for elimination (or at least reduction) of phase discontinuities. This is equivalent to adding or subtracting a tilted plane from the wrapped phase distribution and re-wrapping the phase again within the original limits.

3.4 Summary

This chapter has presented a number of unwrapping techniques that have been proposed in the past. There is no unwrapper that can overcome all the problems previously described in this chapter. There is, therefore, no ideal unwrapper. Increases in performance normally induce penalties in the execution times due to an increase in the complexity of the algorithms. Human assisted unwrapping remains as the only solution to ensure a correct unwrapping in extremely complex images.

The search for a general purpose unwrapper, one of the objectives of the research, is still on, and a large volume of research concentrates on this topic.

Chapter four

Two Novel Techniques for Two Dimensional Phase Unwrapping

4.1 Introduction

Unwrapping is, commonly, the most time consuming operation in fringe analysis. A decrease in the time required to unwrap the result will, therefore, have a large repercussion on the time required to obtain the final result. Current algorithms focus on the effectiveness of the unwrapper and indeed, several effective unwrappers do exist. Unfortunately, effectiveness and speed are not always compatible. The approaches presented in this chapter deal with execution time as the main issue, whilst simultaneously, maintaining an acceptable level of effectiveness.

Because of the importance of phase unwrapping in fringe analysis, a large part of the research undertaken was focused on this topic. Two novel techniques are proposed in this chapter. The first of them, is a recursive approach based on image partitioning; the second, could be included in several of the groups described in the previous chapter. These algorithms have proven to meet the requirements for both effectiveness and execution time for our purposes.

The previous chapter reviewed some of the techniques that have been proposed in the past to deal with the phase unwrapping problem. Several of these approaches are, in fact, very effective and produce high accuracy results. However, the perfect approach to phase unwrapping has not been found yet.

This chapter presents two techniques to unwrap a two dimensional phase distribution whose values are limited to the interval [-π, +π]. Image partitioning is common to both of them, although it is performed in a different manner, as it will be shown later in this chapter.

The algorithms have been tested with images produced by fringe projection and the use of the Fourier fringe analysis technique. For such images, apparent wraps can appear due to three different causes:

- Cracks or discontinuities on the object's surface.
- Errors in the processing due to cross-talking in the Fourier domain, specially when sharp edges are present on the surface under analysis.
- Jumps in the phase distribution due to the arctan function.

The added speckle noise produced by the reflectivity and roughness of the surface and the aperture of the camera optics also complicates the process.

4.2 Recursive unwrapper

Divide and conquer is a programming technique that has been successfully applied to many problems in the past. An example is the well known quick sort algorithm, that considerably reduced the time required to sort an array of numbers (Hoare 1962; Knuth 1973; Sedgewick 1988). The technique presented in this section is also based on the divide and conquer approach.

It is important to remark at this stage that interest is not focused on recovering the whole image at any cost, but only those pixels that are error free. It is considered preferable to mask a valid point and hence lose it than to unwrap a point that contains false information, hence giving place to errors in the final result.

4.2.1 Algorithm

First of all, a description of the unwrapping process for a 2 x 2 matrix is required for later use in the development of the unwrapper.

A path independent unwrapping can be performed easily if a 2 x 2 matrix is considered. If only orthogonal paths are allowed, there are only two possible paths connecting any two pixels in the matrix, as illustrated in Figure 29, e.g. only two paths connecting point A to B exist: A-B and A-D-C-B.

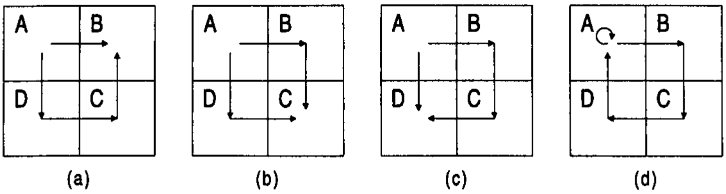


Figure 29. There are only two possible paths connecting any two pixels in the matrix. These paths are illustrated for the connection of point A to any other pixel.

An operator Ψ_{PQ} can be defined so that it denotes the integer number of 2π offsets required to unwrap point Q with respect to point P. This operator will be given by

$$\Psi_{PQ} = \begin{cases} +1 & \text{if } V_P - V_Q > tol \\ -1 & \text{if } V_Q - V_P > tol \\ 0 & \text{otherwise} \end{cases} ,$$

where V_P and V_Q are the wrapped phase values at points P and Q respectively, and *tol* is a previously set threshold value, that in some literature is called the maximum tolerated value (usually π).

For a path independent unwrapping to be achieved, the additions of the offsets along the two different paths have to be equal. Looking at Figure 29, the following four equations have to be simultaneously satisfied.

$$\Psi_{AB} = \Psi_{AD} + \Psi_{DC} + \Psi_{CB}$$
$$\Psi_{AB} + \Psi_{BC} = \Psi_{AD} + \Psi_{DC}$$
$$\Psi_{AB} + \Psi_{BC} + \Psi_{CD} = \Psi_{AD}$$
$$\Psi_{AA} = \Psi_{AB} + \Psi_{BC} + \Psi_{CD} + \Psi_{DA}$$

Considering the fact that $\Psi_{PQ}=-\Psi_{QP}$, a simple look at the set of equations above can be sufficient to notice that they are equivalent. Therefore, if one of the equations is satisfied, the others are also satisfied.

The last of the previous equations is used in the recursive unwrapper. Offsets are calculated and added together. If the result of the addition is zero, a simple unwrapping for the 2 x 2 matrix can be easily implemented as follows.

Let W_P and U_P denote the wrapped and unwrapped phase at point P respectively. The following equations are then applied.

$$U_A = W_A$$
$$U_B = W_B + \Psi_{AB}$$
$$U_C = W_C + \Psi_{AB} + \Psi_{BC}$$
$$U_D = W_D + \Psi_{AD}$$

On the other hand, if the addition of the offsets differs from zero, the matrix is considered as containing erroneous data.

We have seen that the unwrapping of a 2 x 2 matrix is a simple process. Based on this principle, a simple and effective algorithm for two dimensional phase unwrapping will be built.

Top-down approaches by partitioning are common to many different image processing problems. The image is divided into quadrants that are treated independently of each other. This process is applied recursively until small areas that can be easily analysed are obtained. Finally, the small areas are brought together to produce a consistent result for the entire image.

The same approach can be considered for unwrapping two dimensional images. Division in quadratures reduces the size of the area that is to be unwrapped and simplifies the problem of path independent unwrapping since the number of paths is reduced.

The image is subdivided into its four quadrants. These quadrants are unwrapped independently and linked together at a later stage to produce the entire unwrapped image. The same decomposition technique applied to the whole image can also be applied to each of the quadrants in turn resulting from the previous process, thus obtaining four sub-quadrants for every quadrant. These will lead to the unwrapped quadrants once the sub-quadrants are linked together. This technique can be applied recursively leading to the final unwrapping of 2 x 2 matrices, as shown in Figure 30.

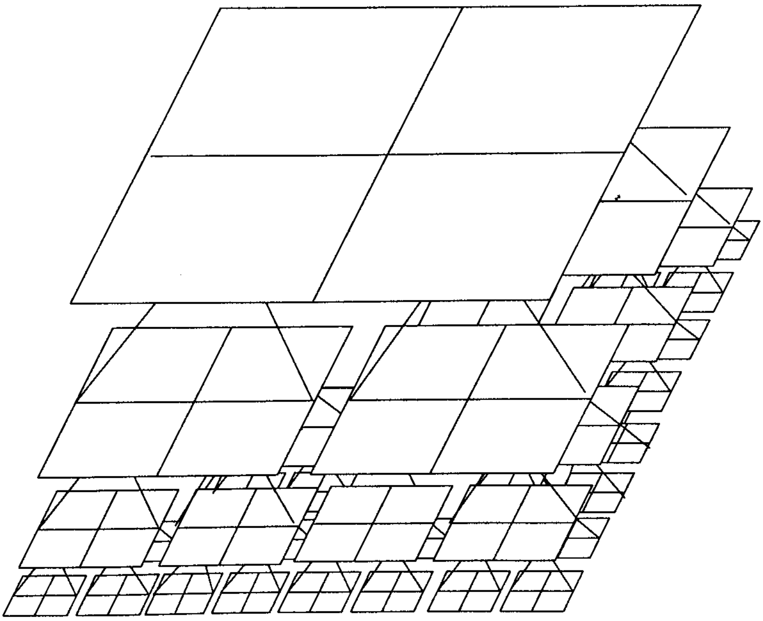


Figure 30. Decomposition process.

Once the 2 x 2 matrices have been unwrapped, it is required to bring them together to form larger matrices, starting the building-up process.

The process of joining four *n x n* sub-areas together to form a larger *2n x 2n* area proceeds by an analysis of the junctions (Figure 31) and can be described as follows.

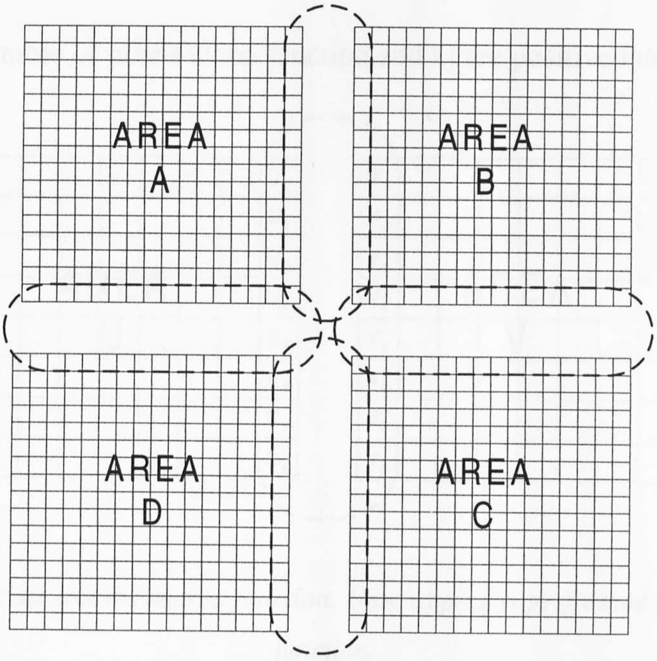


Figure 31. Junctions between two areas (dark dashed lines).

A new function $\Im_{PQ}$ is defined denoting the integer number of 2π offsets required to unwrap point Q with respect to point P. This function is defined as:

$$\Im_{PQ} = \left\langle \frac{V_P - V_Q}{2\pi} \right\rangle$$

where < > represents the nearest integer operation and the maximum tolerated value is assumed to be π.

The operator $\Re_{UV}$ between two areas U and V as represented in Figure 32 is then defined as

$$\Re_{UV} = \begin{cases} \Im_{U_iV_i} & \text{if } \Im_{U_iV_i} = \Im_{U_jV_j} \text{ for all i, j} < \text{n} \\ \\ \textit{NOT DEFINED} & \text{otherwise} \end{cases}$$

where *n* is the number of pixels at the junction and i,j are positive integers.

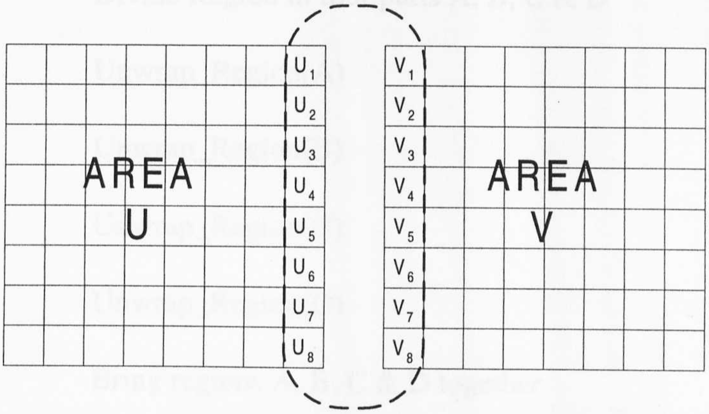


Figure 32. Two areas and the shared junction. Unwrapping is performed by analysis of the junction.

The sub-areas are brought together in a similar manner as the 2 x 2 matrix was unwrapped, but using the operator $\Re_{UV}$ instead of Ψ_{PQ} as a basis.

Looking at Figure 31, where A, B, C and D are four different regions, $\Re_{AB}$, $\Re_{BC}$, $\Re_{DC}$ and $\Re_{DA}$ are calculated. A check for path independence is then performed by summing the values. If the addition leads to a result that is different from zero, the whole area is considered as invalid. If an operation $\Re_{UV}$ returns *NOT DEFINED*, sub-areas U and V are considered as invalid and the check for path independence is not carried out, since the loop cannot be closed and only a single path exists between the two areas.

Once the decomposition and junction strategies have been described, giving a basis for the understanding of the process, a simple version of the algorithm can be outlined:

```
Unwrap_Region(Region)   /* Main procedure, entry point */
        If Region is larger than 2 x 2 Then
                Divide Region in four parts A, B, C & D
                Unwrap_Region(A)
                Unwrap_Region(B)
                Unwrap_Region(C)
                Unwrap_Region(D)
                Bring regions A, B, C & D together
            Else Unwrap_2x2_Matrix(Region)
```

This algorithm, of a recursive nature (hence its name), is somewhat simpler than the implemented algorithm, but it is self explanatory and gives a simple understanding of the general idea.

The management of masked areas is the main cause of complications in the previously outlined algorithm. If an area is masked because an error is detected, this area will no longer be valid and it should not be joined with neighbour areas at a higher level of recursion. The process of avoiding these masked regions can be cumbersome on certain occasions. The following examples describe a set of different situations that the algorithm has to cope with.

Figure 33 illustrates an area where one of the sub-areas has been completely masked as a consequence of a large error detected in it (grey color). In this case, the entire junctions placed between sub-areas A-B and B-C cannot be subjected to analysis and comparison and, therefore, it is not possible to complete the circuit. Checking for path independence cannot be undertaken at this stage and it has to be delayed (since $\Re_{AB}$ and $\Re_{BC}$ are unknown, the addition cannot be performed).

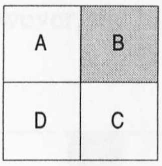


Figure 33. Area with a masked sub-area

Figure 34 shows a case in which the same junctions are masked. In this case, however, two different regions can be unwrapped, but they cannot be joined together (Figure 35). A decision is required to be taken by the unwrapper. The criteria used is to mask the smallest possible region, saving the largest of the areas.

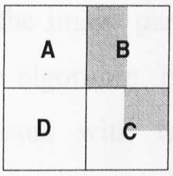


Figure 34. Masked junctions.

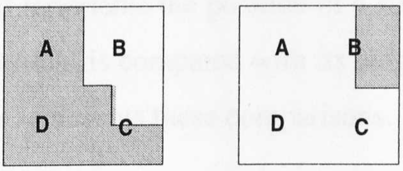


Figure 35. Two different areas can be unwrapped. A decision has to be taken by the unwrapper. Dark regions represent the different areas that can be unwrapped.

Figure 36 gives an example in which overmasking occurs. In this case, the four junctions are masked and no comparison or unwrapping process can be undertaken through them. In this kind of situation, it is considered that the area contains a great deal of noise and no information should be recovered from it. The whole area is, therefore, masked. This overmasking could be avoided by recovering the data in relation to the surrounding areas. However, the high time overhead incurred led to this alternative being disregarded.

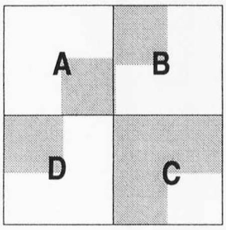


Figure 36. Overmasking is produced because connection between sub-areas cannot be established.

The algorithm is mainly based on the image partitioning concept. The multiple scan directions concept is implicit to the algorithm. Every point is initially unwrapped by comparison with its two neighbours with which it shares a junction in the corresponding 2 x 2 matrix. At a later stage, it is also compared with the remaining two orthogonal neighbours (when matrices are brought together to form larger areas).

The grey area in Figure 37(a) represents the position of a single point in a larger area. Figure 37(b) shows how that point is compared with its neighbours at each stage. The crossing lines between points represents these comparisons.

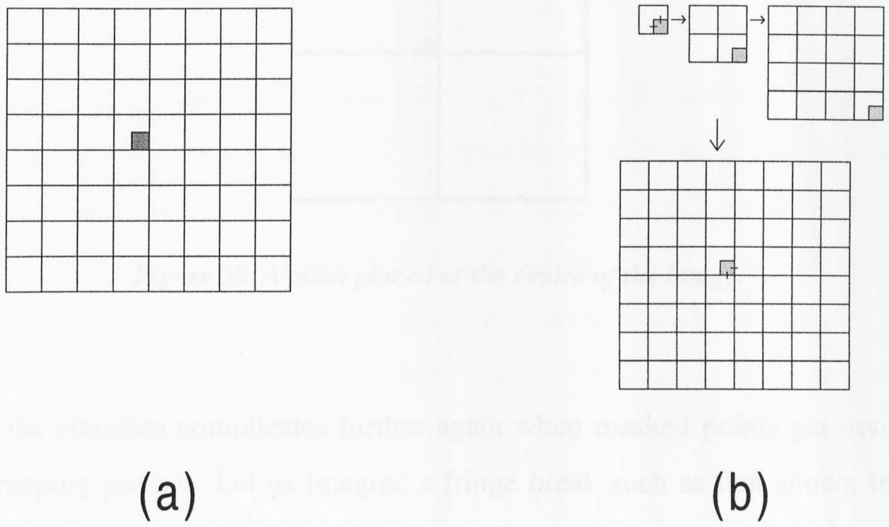


Figure 37. (a) One pixel in a larger area and (b) comparisons with its orthogonal neighbours.

Each point is clearly seen to be compared with each of its four orthogonal neighbours (if the neighbors are error free, that is, they have not been previously masked). However, the comparison with the third and fourth points will depend on the position of the point in the image. If this comparison leads to a detection of an error, it is advantageous to detect it as soon as possible, since a delay could produce overmasking.

Figure 38 is a clear example of this situation, where a point situated at the centre of the image is not compared to two of its orthogonal neighbours until the latest stage of the algorithm (when the four major quadrants are brought together to produce the entire unwrapped result). If an error is detected at this stage, the whole image would have to be masked as a result of a *non-defined* value returned from the function $\Re$.

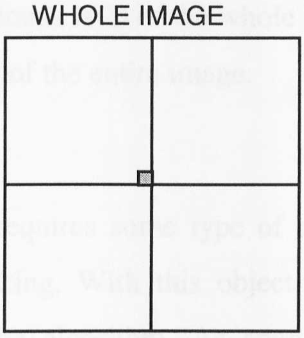


Figure 38. A point placed at the centre of the image.

Besides, the situation complicates further again when masked points get involved in the unwrapping process. Let us imagine a fringe break such as that shown in Figure 39a, where grey tones represent masked areas. Now, let us suppose that this region is placed on the dividing line between the top and bottom parts of the image, and adjusted to the left side, as shown in Figure 39b.

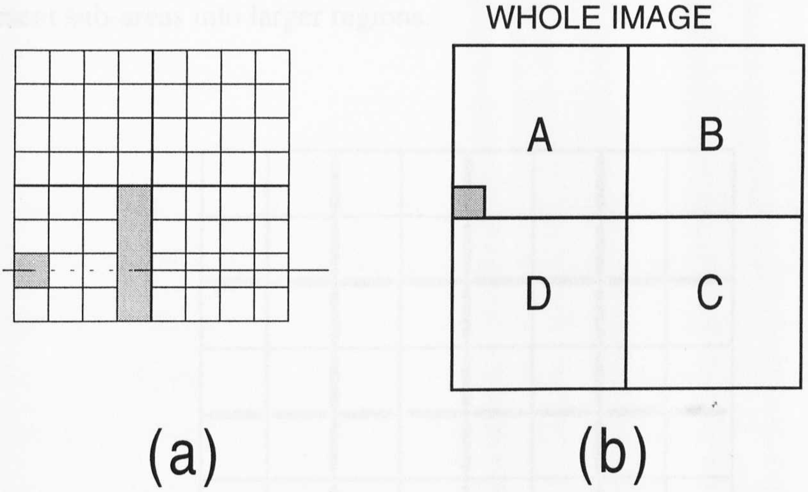


Figure 39. (a) Fringe break in an area. Discontinuous line represents the break. (b) Location of this area in the image.

When the unwrapping process is undertaken, there will not be any square at its left side, and the comparison of the sub-areas C-D, that normally would detect the error is avoided by the masked points. When the error is detected (at the end of the unwrapping process when quadrants A-D of the whole image are joined together), it is too late, leading to the masking of the entire image.

It is clear that the algorithm requires some type of modification to cope with this source of errors and overmasking. With this objective in mind, a pre-comparator process has been added to the algorithm. As soon as an area is independently unwrapped, a check for "compatibility" is undertaken before larger areas are dealt with. This test consists of a comparison of each sub-area with the adjacent areas in order to check that no problem will be found with the external junctions at a later step. If a junction is found erroneous, the sub-areas involving that junction are masked.

Figure 40 shows the junctions that are to be examined in the pre-comparator process. The squares represent the regions that have already been unwrapped. The rest of the junctions are not analysed since they will be checked in the next stage, when joining the present sub-areas into larger regions.

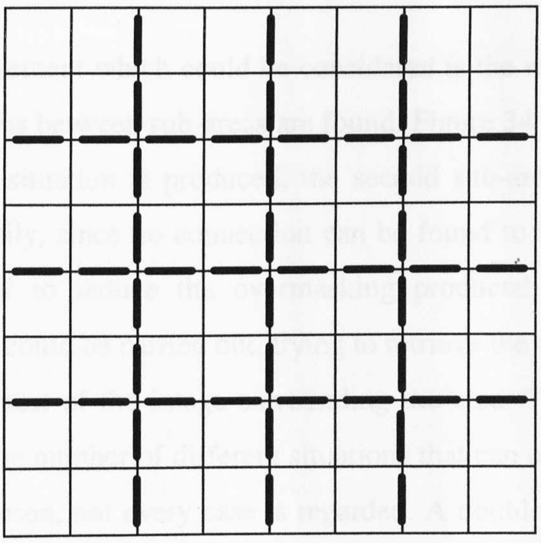


Figure 40. Junctions that are to be examined in the pre-comparator process.

4.2.2 Possible improvements

By increasing the number of checks that are carried out during the unwrapping process, many improvements could be applied to the current version of the algorithm. Basically, two main improvements were detected. Their implementation would increase the reliability of the algorithm by avoiding some of the overmasking effects previously mentioned:

- The checking of compatibility, as it was called above, by which it is intended to detect errors before the areas grow and, consequently, anticipating the masking operations could be replaced by a check similar to that undertaken with normal areas, in which it was tested if the addition of the differences is equal to zero. This testing operation could be performed with any possible set of four adjacent previously unwrapped areas at the end of each stage. The implementation of this process multiplies the execution time by an important factor, that is close to 3. This operation carries out the masking of certain areas at a former stage, hence avoiding some overmasking.

- The second improvement which could be considered is the recovering of the areas when no connections between sub-areas are found. Figure 34 is an example of such a situation. If this situation is produced, the second sub-area will be completely masked unnecessarily, since no connection can be found to link the data between sub-areas. In order to reduce the overmasking produced by the algorithm, a recovering process could be carried out, trying to retrieve the data by examining the junctions with the rest of the image surrounding the area. The implementation is made difficult by the number of different situations that can occur. In the particular implementation chosen, not every case is regarded. A double linked list is used to keep track of the areas that were masked but could be recovered. Additional memory space is therefore required and a high penalty in time is paid.

The results obtained with the application of the improvements are discussed and compared in the corresponding section.

4.2.3 Parallelisation

A transputer is a single VLSI device with processor, memory and communications links for direct connection to other transputers. Parallel systems can be constructed from a collection of transputers operating concurrently and communicating through links (INMOS 1988).

The occam 2 parallel language was intended by its designers as the assembly language of the INMOS transputer (Jones and Goldsmith 1988). A more intensive description of both the features of the language and the transputer is given in Chapter 5.

The recursive unwrapping algorithm described above has been parallelised on a four T800 transputer system using occam 2 as the programming language. The results obtained were satisfactory in terms of the speedup obtained. By optimising the algorithm, executions times for an 128 x 128 image close to 0.8 seconds in a single transputer system were achieved. When the four transputers are working in parallel, the time reduces to just one third of the sequential time. Times close to a quarter of a second are achieved with this system. These times are interesting considering that the computing power of a single transputer is of the order of a 486 processor.

Figure 41 is a diagram of the transputer system utilised in the experimentation, illustrating the communication links. A host computer (IBM compatible with an Intel 80486 SX-25 chip) is linked to the transputer system by means of a connection with the master transputer. The master transputer is linked to the rest of the network by means of transputer links that work at a speed of 10 Mbits/sec. Higher link speeds (20 Mbits/sec., as usual in transputer systems composed of T800s) would be particularly beneficial for the parallelisation, reducing the communication overheads occurring in the parallel execution and achieving greater speedups.

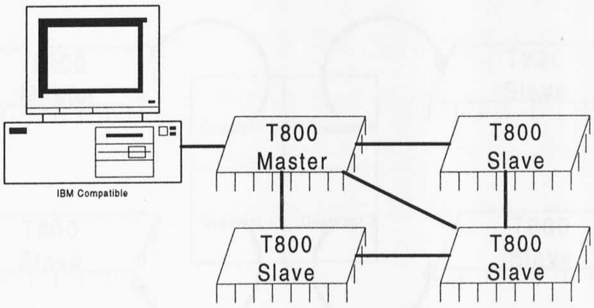


Figure 41. Transputer network. Dark lines represent the system links.

The implementation of the parallel version follows a domain decomposition approach (Ragsdale 1994), having the data equally divided between the four processors.

A master processor contains the whole set of data initially. These data have to be distributed equally through the network. A tertiary tree is used for this purpose. The links utilised in this process are illustrated in Figure 42.

The reason for using a tertiary tree is the capability of the transputers to perform parallel communication without significant degradation in performance, that is, several links can be working at a time, increasing the communication bandwidth.

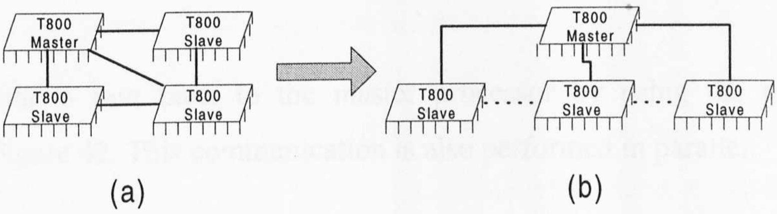


Figure 42. (a) Links in the transputer network. (b) The network from the point of view of a tertiary tree. The dashed lines represent links that are not used in the initial delivery of the data.

The data are distributed in the manner illustrated in Figure 43, each processor receiving a quadrant of the image.

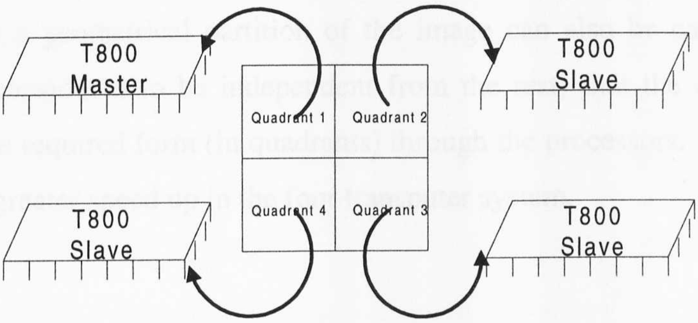


Figure 43. Distribution of the data amongst the processors

With this data distribution, every stage of the algorithm can be performed independently. Only a small part of the pre-comparator process requires communication between the processors (data at the junctions of the quadrants). However, this communication can be undertaken in parallel with the analysis of the rest of the junctions except at the latter stages, where less junctions have to be examined. Acting in this manner, the communication is hidden and little or no overhead is produced. A communication overhead takes place at the latest stage, since a different constant has to be added to each of the quadrants of the image, these constants being dependent upon the data contained on the other quadrants (the calculation of the constant is performed in a clockwise direction, and it depends on the constant added to the previous quadrant).

Finally the data is sent back to the master processor by using the tertiary tree illustrated in Figure 42. This communication is also performed in parallel.

When using a four transputer network, the main communication overhead is introduced by the delivery and gathering process. These times have been included in the calculation of the speedup[3]. However, if a parallel machine is available, it would be expected that the whole fringe analysis process (not only the unwrapping) be performed in the parallel machine. It could be the case (specially if using phase stepping, where a geometrical partition of the image can also be carried out since every point is considered to be independent from the rest) that the data is initially distributed in the required form (in quadrants) through the processors. In this case, we would obtain a greater speed up in the four transputer system.

Even if the algorithm presents a very good speedup for a four transputer system, it is not easily parallelisable when the number of processors is increased. The algorithm presents a poor isoefficiency function that indicates a fast drop in the efficiency when the number of processors in the system is augmented (Grama et al. 1993; Kumar and Gupta 1994). As more processors are in the system, so more communication is required between the processors. Another drawback stems from the geometrical decomposition of the data, since for this algorithm to be applied, the number of processors in the system must be an exact power of four.

A modification to the algorithm was also implemented to avoid the time overheads incurred because of the communication required at the end of every stage of the algorithm, increasing the potential for parallelisation in a system composed of a larger number of processors. This modification consists of delaying the pre-comparator process at the junctions of the quadrants. This operation may sound contradictory but gave good results, reducing the time required to produce the unwrapped phase and providing higher speedup. The pre-comparator process at the junctions of the quadrants is undertaken just before the four quadrants are joined together, and not at the end of every stage. This process turns into a search for areas that should have previously been masked. Once these regions have been determined, the four quadrants can be brought together to produce the final unwrapped phase distribution.

[3] The speedup is defined as the time for sequential execution (on a single processor) divided by the time consumed by the parallel execution.

4.2.4 Results

Some of the results obtained with this unwrapper are described in this section. The phase distributions are displayed in a normalised form by using a grey scale. The intensities vary from white (maximum phase value) to black (minimum phase value). Brightest or darkest regions represent those zones that have been masked by the unwrapper (the colour has been chosen so that masked areas are visible).

Figure 44 compares the performance of the recursive unwrapper with respect to linear scanning. Highly corrupted areas are produced by the latter. The recursive unwrapper avoids error propagation.

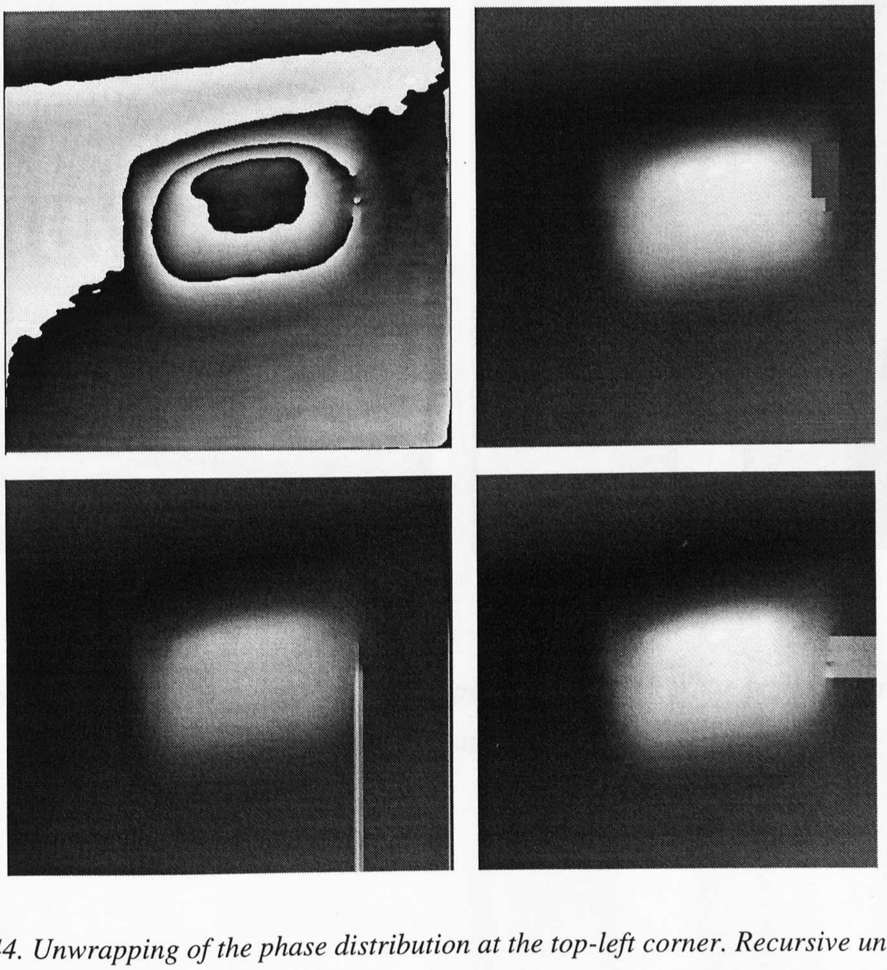


Figure 44. Unwrapping of the phase distribution at the top-left corner. Recursive unwrapper (top-right), vertical and horizontal line scanning (bottom).

Another image under analysis (Figure 45) contains a large area of erroneous data. The bad data in this region was caused by a shadow appearing in the original fringe pattern. This shadow introduced undesired high frequencies in the captured image, that, when analysed by the FFA technique, produced the observable unreliable data.

The algorithm reacts to the presence of these unreliable data, masking them out and unwrapping the rest of the phase distribution. While it is true that a high amount of information is lost, the rest of the image is not corrupted and the data remaining is reliable.

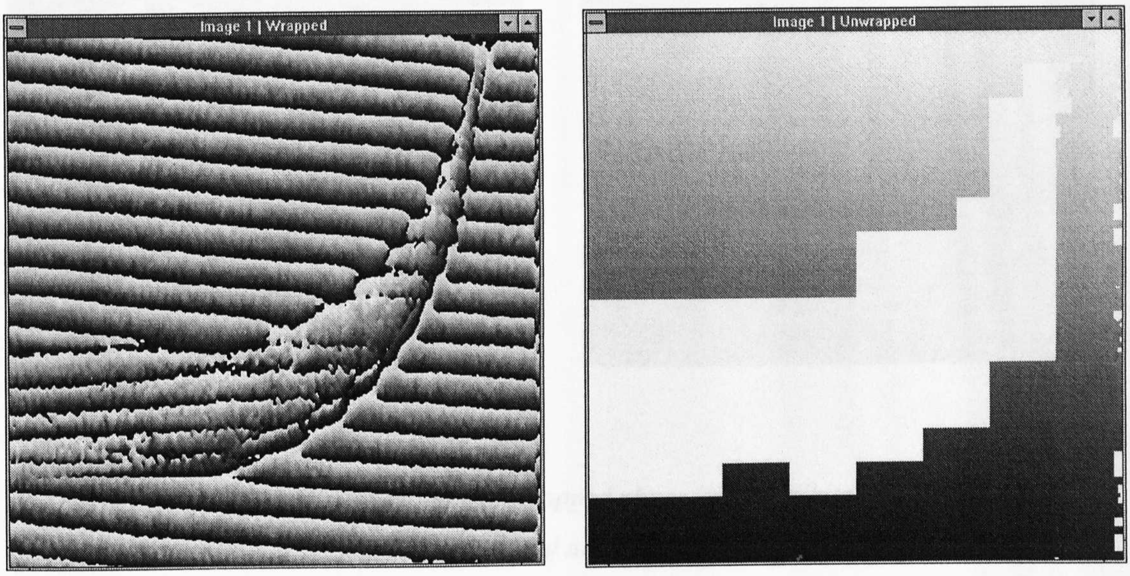


Figure 45. wrapped phase (left) and unwrapped phase (right). White regions represent masked areas.

The third case that has been analysed is a simple image that has also been obtained by using the FFA technique (Figure 46). As it was pointed out in chapter 2, this technique produces errors that tend to concentrate on the borders of the image. This unwrapper detects these errors and disregards the information contained in them, leaving the rest of the image correctly unwrapped.

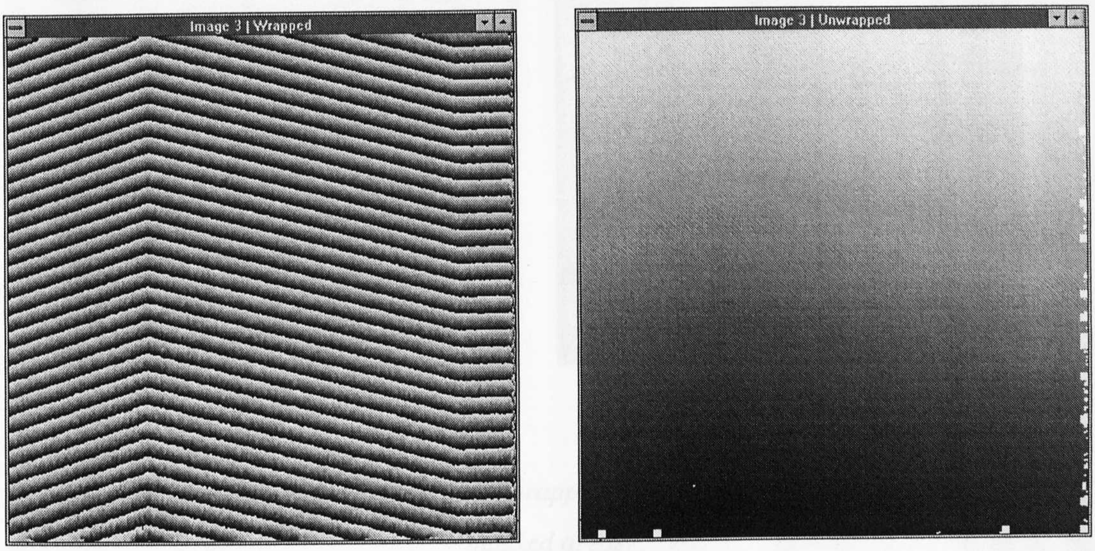


Figure 46. wrapped phase (left) and unwrapped phase (right). White regions represent masked areas.

A different image with small errors in both the borders and central areas of the image has been tested with the unwrapper (Figure 47). The behaviour of the recursive unwrapper can be observed again.

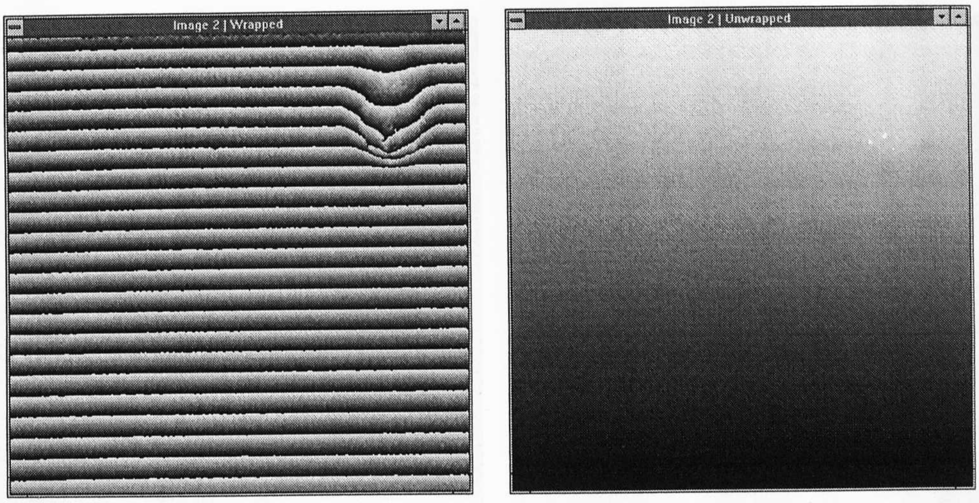


Figure 47. wrapped phase (left) and unwrapped phase (right). White regions represent masked areas.

Figure 48 is another image with a small error produced by a high frequency that has been filtered out in the Fourier domain. This error is adequately located by the unwrapper, masking the defect area. The image after carrier subtraction is also illustrated in the figure.

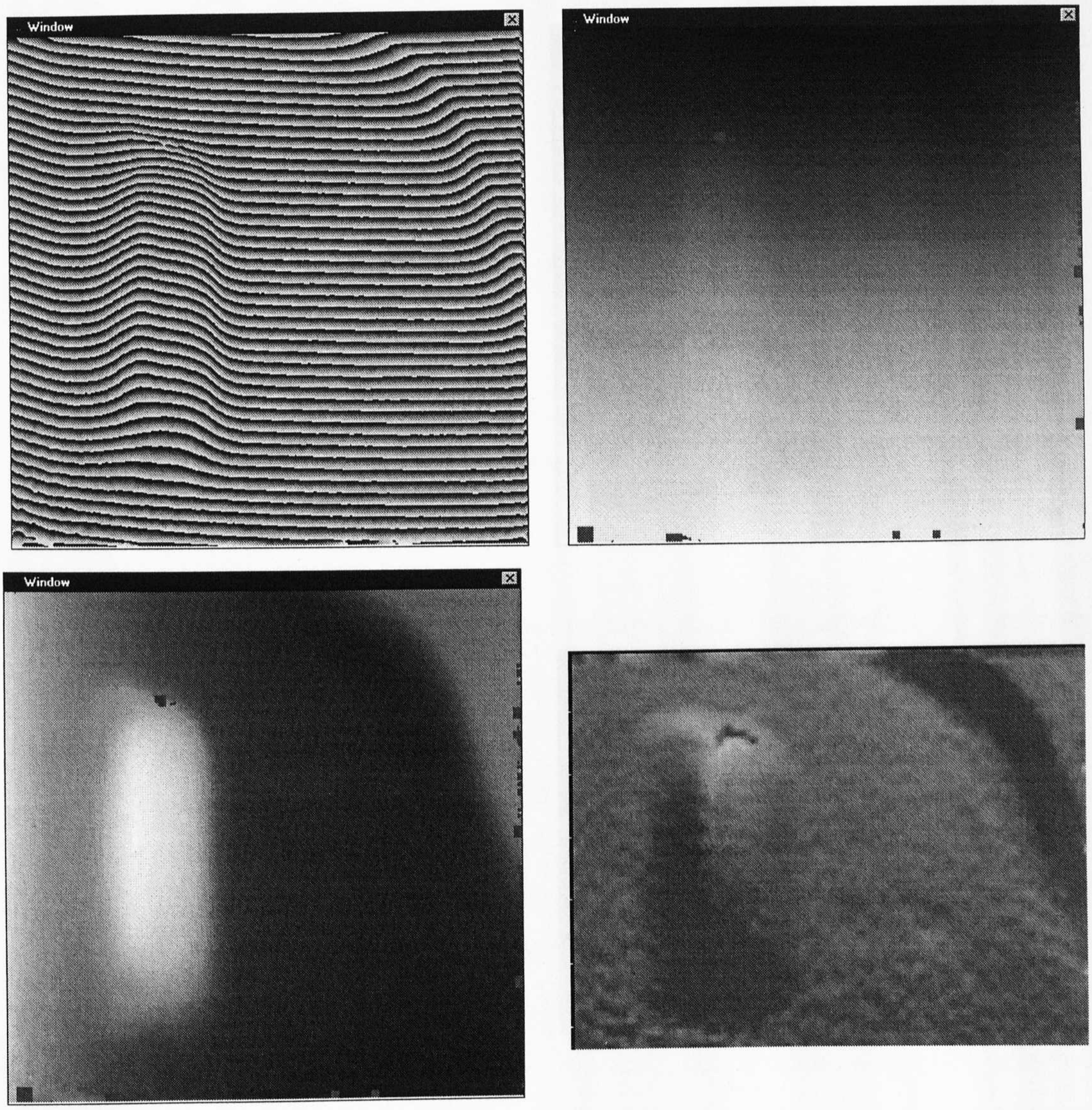


Figure 48. Wrapped phase (top-left), unwrapped phase (top-right), unwrapped phase with carrier removed (bottom-left), three dimensional representation (bottom-right). Dark regions represent masked areas

Figure 49 illustrates the unwrapping process of a far more complex wrapped phase map. Many errors can be observed at the borders of the triangular shape at the top of the image. More faulty areas appear at the bottom of the wrapped phase distribution. These erroneous values cause several discontinuities in the wraps that would usually lead to an incorrect result. The unwrapper correctly locates the faulty areas and masks them. It can be observed that the overmasking produced is not of a high magnitude and that the masked areas are of an acceptable size.

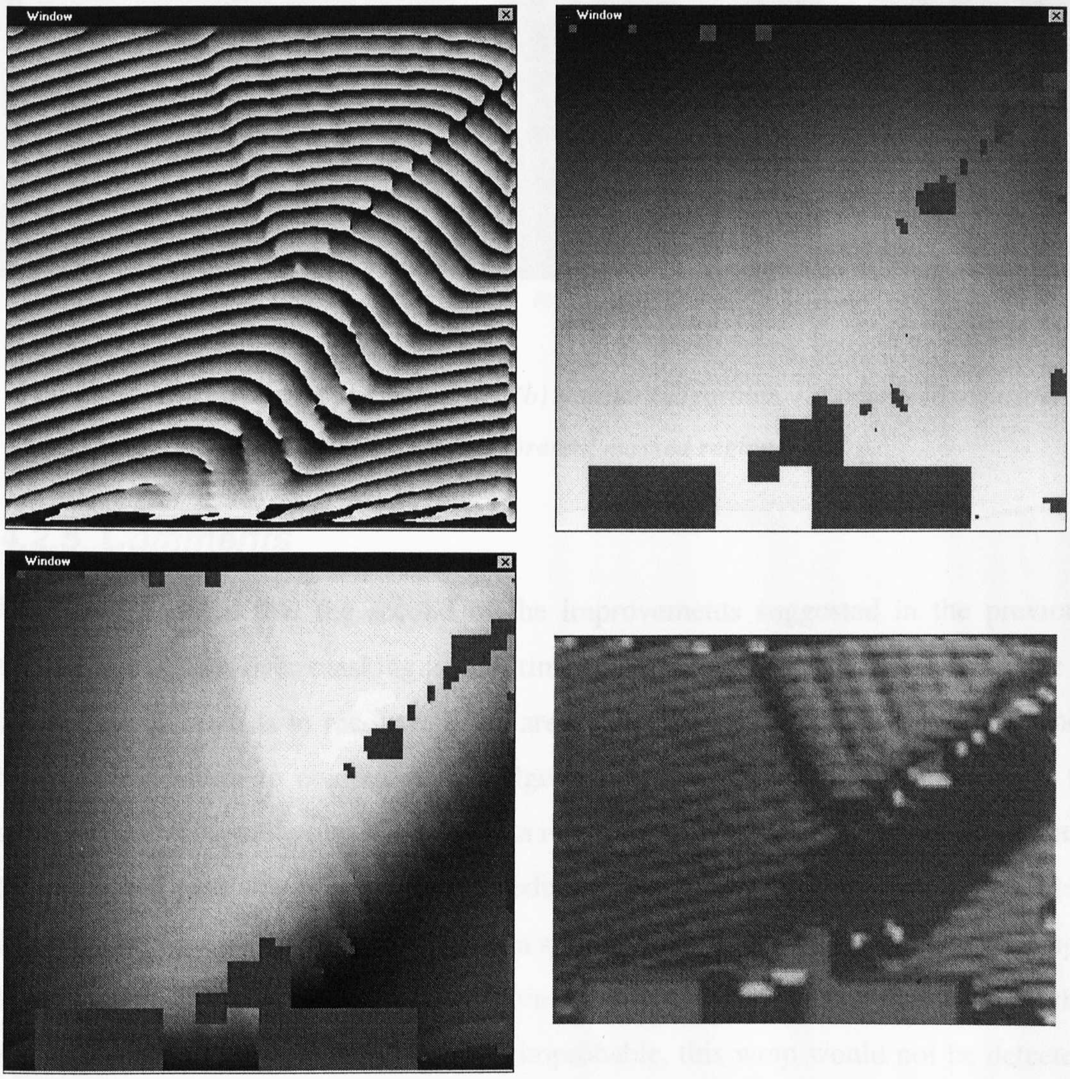


Figure 49. Wrapped phase (top-left), unwrapped phase (top-right), unwrapped phase with carrier removed (bottom-left), three dimensional representation (bottom-right). Dark regions represent masked areas

The application of the improvements described above avoids overmasking by an important factor. The results of the application of the standard and improved algorithms are compared in Figure 50. Considerably less overmasking is produced by the improved technique. No accurate comparison in time can be given for the two algorithms, since the execution time is image dependent. As a rule of thumb, the improved algorithm is over 10 times slower than the standard version.

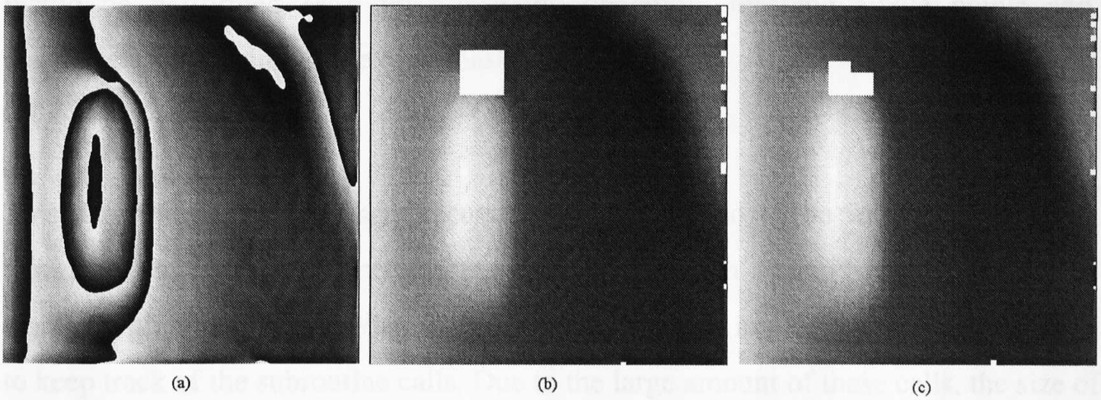


Figure 50. Unwrapping of phase map (a). (b) standard algorithm, (c) improved algorithm. Brightest areas represent masked regions.

4.2.5 Comments

It has to be stated that the second of the improvements suggested in the previous section that avoids over-masking is very time consuming and is a complicated issue if the implementation is to recover all the areas that it is possible to rescue. It was not the intention either to complicate the algorithm when the results obtained were, to some extent, acceptable regarding the data recovered and the correctness of those data. However, the presence of local errors produced by *trapping* of valid areas is not ruled out. That is, whenever an area maintains a single connection with the rest of the image by just a single border. If that area contains a false wrap of a size comparable to the size of the area inside, a situation rather improbable, this wrap would not be detected since no further checking is possible due to the loss of the connections with the rest of the image. This could again be solved by a simple check for each of the areas linked by a single connection to the rest of the image. This is not a very time-consuming process and could be worthwhile if it is required to obtain the maximum possible amount of valid data.

It is important to remark that, since the experiments had the aim of showing the validity of the algorithm, borders were not removed and low modulation tests were not performed so that more problems remain to be overcome by the unwrapper. The application of such techniques, whether in isolation or combined with validation tests such as those described by Stephenson *et al.*(1994) could improve the performance of the algorithm. Modulation tests are particularly suitable for this algorithm since they would be performed as a pre-processing operation, masking single points and, consequently, avoiding some over-masking.

This algorithm, that is clearly recursive by nature, had to be implemented in an iterative form because the memory requirements for such a recursive implementation were of a high magnitude. When a recursive algorithm is implemented, a stack is used to keep track of the subroutine calls. Due to the large amount of these calls, the size of the stack required is also large and extra memory is required for execution.

A simplified version of the implemented iterative standard algorithm is outlined below:

```
tile_size = 2 x 2
While (tile_size<whole image)
    With every_tile
        If inconsistencies between sub-tiles then
            mask corresponding tiles
        Chose the largest area that can be unwrapped
        Unwrap this area
    Pre-processing of junctions (Figure 40)
    tile_size = 2 x tile_size (the new tile includes four of the previous tiles)
```

In the pre-processing of junctions, some tiles are masked out, to avoid overmasking at a later stage.

Because of the nature of the algorithm, it is easily parallelisable. Since each partition is independently unwrapped (except the small degree of interaction required for checking on the borders), no excessive communication is required until the junction process. Because of the fact that many of the parallel machines are capable of performing communication and calculation without significant degradation in performance (e.g. transputers and C40's networks), no communication overhead will be produced. The pixels on the borders could be given priority when adding the differences, in such a way that when the differences have been added to the whole area, the required data have already been transmitted among the different processors comprising the network. It is interesting to remark that the parallelisation is particularly good when the number of processors is a power of four, because of the geometrical decomposition that takes place in this algorithm.

Another advantage of this algorithm is that the additional memory required for its execution (in the standard version), by which we mean additional to those requirements for the code and the image themselves, are negligible.

This algorithm could be considered as a variable size tiled unwrapper. The size of the tile changes dynamically to adapt to the errors present in the image. This avoids overmasking of whole large tiles because of the existence of small areas of unreliable data. The algorithm adds the advantages of using a small tile size (only small areas are masked) to the benefits of utilising a large tile size (detection of discontinuities in the wraps).

Masked areas ate normally interpolated to produce a continuous phase map. This is a difficult process if the masked areas are very large, in which case the areas are normally disregarded. If the final result contains a large number of these areas, the image is normally re-captured or re-processed to obtain an acceptable result.

Note that unwrapping has been applied before carrier removal and the borders of the wrapped phase distribution have not been cropped. This is not usually the case, since the removal of the carrier normally reduces the number of wraps and the errors produced by the windowing in the spatial domain tend to concentrate at the borders of the image. However, for illustration purposes it has been preferred to include as many errors as possible in order to test the efficiency of the algorithm.

The masking produced by the algorithm is determined by the size of the fringe discontinuities and/or the distance between inconsistent points in the phase map. If a discontinuity of a size *n* is present, an area between *n x n* and *2n x 2n* is usually masked. A version of the algorithm considering masking shapes other than a square is to be considered in the future.

It is important to emphasise that a lot of other unwrappers could have obtained similar results, but with high time and memory overheads. Thousands of iterations can be required for complete unwrapping with Ghiglia's cellular automaton; a large amount of extra memory is needed to store the pixels in queuing-based algorithms.

The speed of the algorithm is particularly noticeable. Times under a second for unwrapping of an 128 x 128 phase distribution in a single transputer system were reported above. For this size of image the recursive unwrapper is, approximately, four times slower than a linear scanning procedure.

4.3 Iso-phase unwrapper

This algorithm is termed iso-phase or perpendicular unwrapper because of its nature. The algorithm is based upon a simple principle that is explained for a single dimension. Figure 51 is a one dimensional wrapped signal. The interval [-π ,π] can be divided into smaller sub-intervals, as is shown in the same figure. This is similar to the so called *sequency* in signal processing (the concept of certain values passing certain thresholds in a sequential order). Once this division is performed it is possible to observe that the signal generally passes from one division to the previous or next division (the first and last divisions are considered contiguous). If this does not occur it is either because the frequency of the signal is too high or because the sample pixel contains an erroneous value.

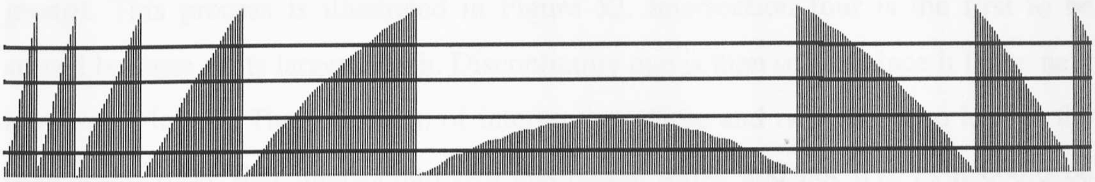


Figure 51. Divisions established as an aid to phase unwrapping.

The principle explained in the previous paragraph constitutes the basis for this two dimensional phase unwrapper: *the phase of the fringe pattern varies slowly between pixels and if it does not, it is because either there is a discontinuity in the object's surface (high frequencies), there is an erroneous point or a wrap is present.*

However, the previous statement says very little, if anything, about how the algorithm performs. When dealing with two dimensions, more information is available since data in both the *x* and *y* axis are known, and the results have to be coherent. Let us show how the principle stated can be applied to the two dimensional problem.

4.3.1 Algorithm

First of all, if one point is in the same division as another, moving though them will be reliable and, therefore, should be given a higher priority. Hence, those points that are connected and belong to the same division could be gathered together, unwrapping between them being unnecessary. The whole image can be divided into groups, in a similar manner to Gierloff's algorithm (1987). The groups are composed of connected pixels that belong to the same division. These groups are considered as already unwrapped (no operation is required between the points included inside them) and are treated as a single entity. The only further step required is the joining process between these groups. This will be performed by analysing the length of the intersections and proceeding to bring together those groups with the largest intersection (a point will lie on the intersection between two groups if it has a neighbour that belongs to the other group). This process is illustrated in Figure 52. Intersection four is the first to be solved because of its larger length. Discontinuity one is then solved since it is the next in order of length. The resolution of intersections three and two follow to lead to the final result (note that intersection two has length three, and not five as it could be expected. There are only three points on each group having a neighbour belonging to the other group).

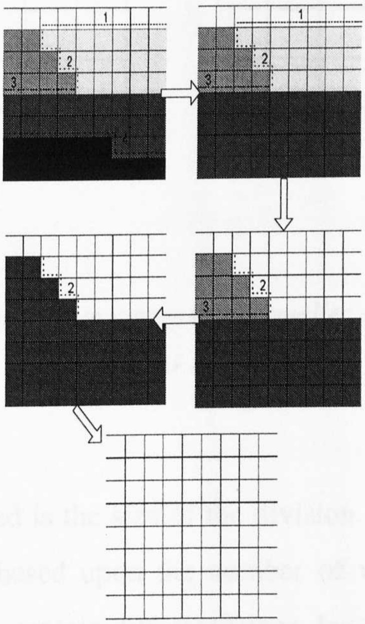


Figure 52. Joining process between the groups.

It is important to remark that intersections only exist between groups that belong to contiguous divisions and that the first and last subdivision are considered contiguous, as if they wrapped around.

Groups are joined by adding or subtracting a multiple of 2π to/from the pixels contained on the appropriate group so that the difference between the pixels placed on the intersection under analysis is lower than the size of the division (note that this multiple can be calculated with respect to a single point, since the rest of the points in the intersection belong to the same groups). Afterwards, the two groups are brought together to form a single larger group.

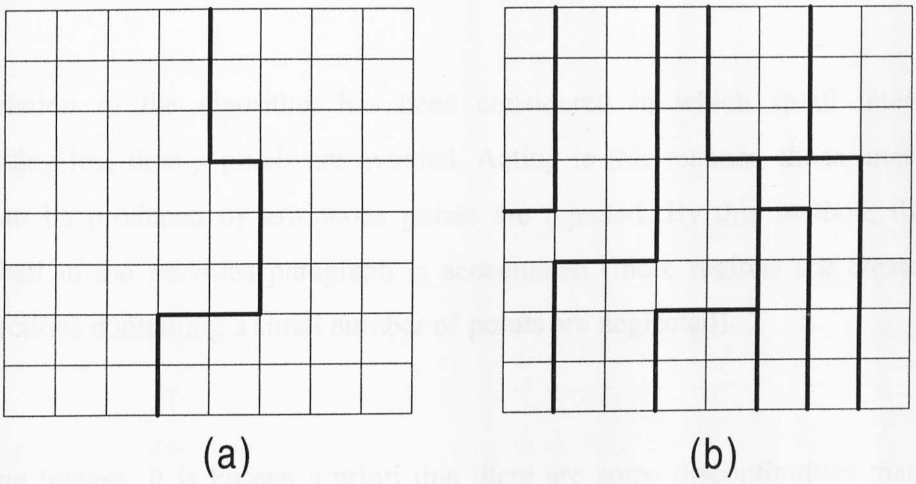


Figure 53. (a) larger division size, in contrast to a smaller division size that results in a larger number of groups (b).

Another question to be solved is the size of the division. This is an image dependent parameter and is normally based upon the number of wraps present in the image. However, a single image can contain different fringe densities in different areas. If the division size is defined too small, more groups will exist and they could be broken as illustrated in Figure 53. The unwrapping process will also be more time consuming. If it is too large, the groups could include errors that would not be corrected by the unwrapper (note that groups are considered as single entities). The best solution is to define the division size as variable through the image. If the distance between wraps is of *n* pixels a size close to $16\pi/n$ has been experimentally shown to lead to the best results. This size includes an average of 8 pixels per division. This variable parameter is defined by areas (to avoid large computation times) by a scanning method that performs a counting of the wraps in the particular area. These areas are normally squares of a pre-determined size.

It is possible that the algorithm will create several regions as a result. These are regions that cannot be joined together and, consequently, are isolated by the unwrapper (regions with no intersections or intersections of a reduced number of points).

A variation of the algorithm has been considered in which small intersections containing less than *p* pixels are avoided. Acting in this manner, those intersections that can be produced by erroneous points are rejected. By this method, the effect described in the previous paragraph is accentuated (more regions are created since intersections containing a small number of points are neglected).

In some images, it is known a priori that there are some discontinuities that can be manually determined. The algorithm can then be manually assisted easily by giving a value outside the range $[-\pi, \pi]$ to the points laying in the intersection between the two different areas at either side of the discontinuity. This will cause the algorithm not to detect any intersection joining the two areas (since points outside the interval $[-\pi, \pi]$ do not belong to any intersection).

A pseudocode version of the algorithm can be expressed as

```
Create the regions
Analyse the intersections
While intersections with more than $p$ points exist
    Choose the largest intersection
    Solve the intersection: add $2k\pi$ to one of the groups, join the groups together in a single new group
    Discard the intersection solved
Determine the different regions obtained as a result
```

The name iso-phases comes from the fact that groups containing pixels with similar phase values are formed. The name perpendicular comes from the fact that the unwrapping procedure is carried out along a direction that is perpendicular to the wraps. It involves decomposition of the image into regions, minimisation of differences and searches for the best path. There is a search for the largest intersections so that differences are lower. This can also be considered as an approximation of the best unwrapping path. A minimum spanning tree would also increase the reliability when looking for a minimum path. However, the method described for this algorithm (solve the largest intersection first) gives a very good approximation of the minimum spanning tree, in a far lower execution time.

The idea of multiple scan directions is implicit in the algorithm. At least two directions are taken into account when unwrapping a pixel: For pixels inside the group, they will have at least two other pixels joining them to the rest of the group. For pixels in the borders of the group, there will be at least one pixel joining it to its group (except in groups formed by a single pixel, rather uncommon) and another joining the pixel to another group that will be taken into account when the intersections are solved. If this last pixel does not exist it is because there is a bad area or a discontinuity in a wrap and, therefore, the unwrapping has to be carried out with a single pixel as reference (these points will be at the borders of the resulting regions).

One of the main advantages of this algorithm is its flexibility. The variation of the size of the divisions and the parameter *p* provides a high capability for adapting to different types of images. If high noise levels are present in the image (as in images obtained by speckle techniques), an increase in the size of the division (to a certain limit) will produce a better result. The parameter *p* can also be suited to the image by setting its value to the size of the maximum wrap discontinuity in the wrapped phase distribution. Acting in this manner, the algorithm will not pass through these areas since the intersection established is neglected.

4.3.2 Testing

The algorithm has been tested with a set of real and simulated images. The results obtained are illustrated in this section. The execution time is variable and it depends on the number of regions and intersections, that is, on the particular phase distribution being analysed. For most of the images analysed, the execution time was considerably less than a minute in a 80486 DX-100 processor, for a 512 x 512 image. However, the unwrapper has not been optimised and a prototype of the algorithm was used for the experimentation. This time can be expected to reduce considerably, but even this prototype performs generally in a fraction of the time consumed by the 2-D Fourier transforms that are required in Fourier Analysis.

As in the presentation of the recursive unwrapper results, a range image or intensities representation is used. In this representation, the intensity of the pixel is directly proportional to the value of the phase at that particular point. First, the image to unwrap is given. If required, the result obtained by Shafer's unwrapper (unwrapping the lines and joining them together) is shown in order to be able to illustrate the problems with the phase and, finally, the results obtained by this algorithm are presented. The final result provided by the algorithm is given in two parts: The first part is the unwrapped phase distribution; the second part is related to the different regions that the algorithm has localised. These regions are totally independent and have been isolated by the algorithm.

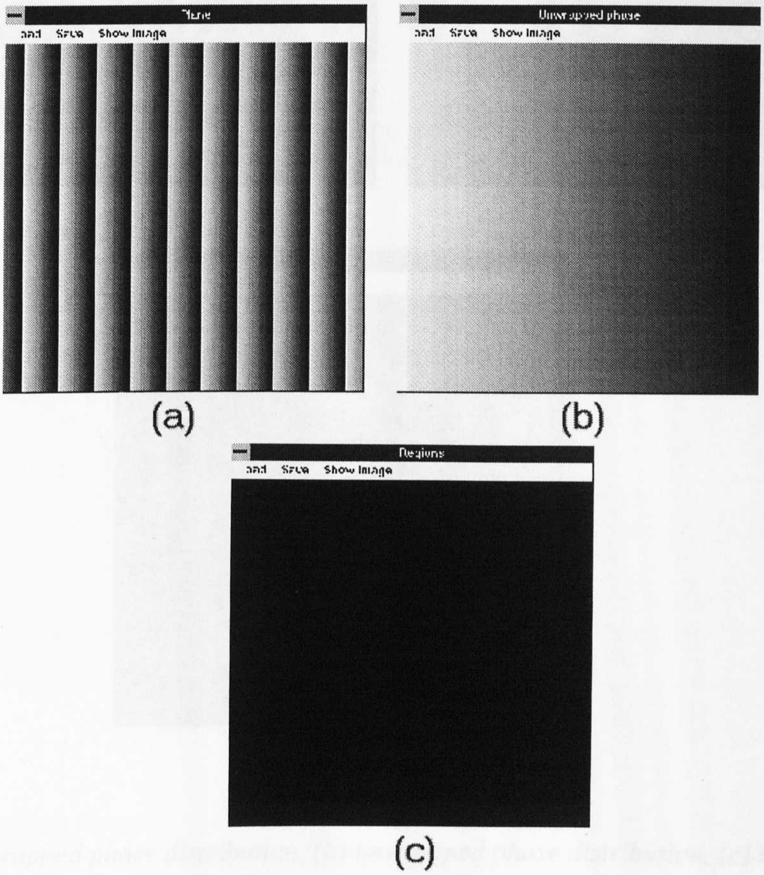


Figure 54. (a) Wrapped phase distribution, (b) unwrapped phase distribution, (c) isolated groups.

The first test has been undertaken with a simple image. The image utilised for these purposes is a simulation of the wrapped phase obtained by projection of ten vertical fringes on a plane tilted in a direction that is perpendicular to the fringes. The results obtained are clearly correct. Figure 54 illustrates the results. Figure 54(a) is the original wrapped phase distribution, Figure 54(b) is the unwrapped phase and Figure 54(c) represents the different regions detected (a single one in this case).

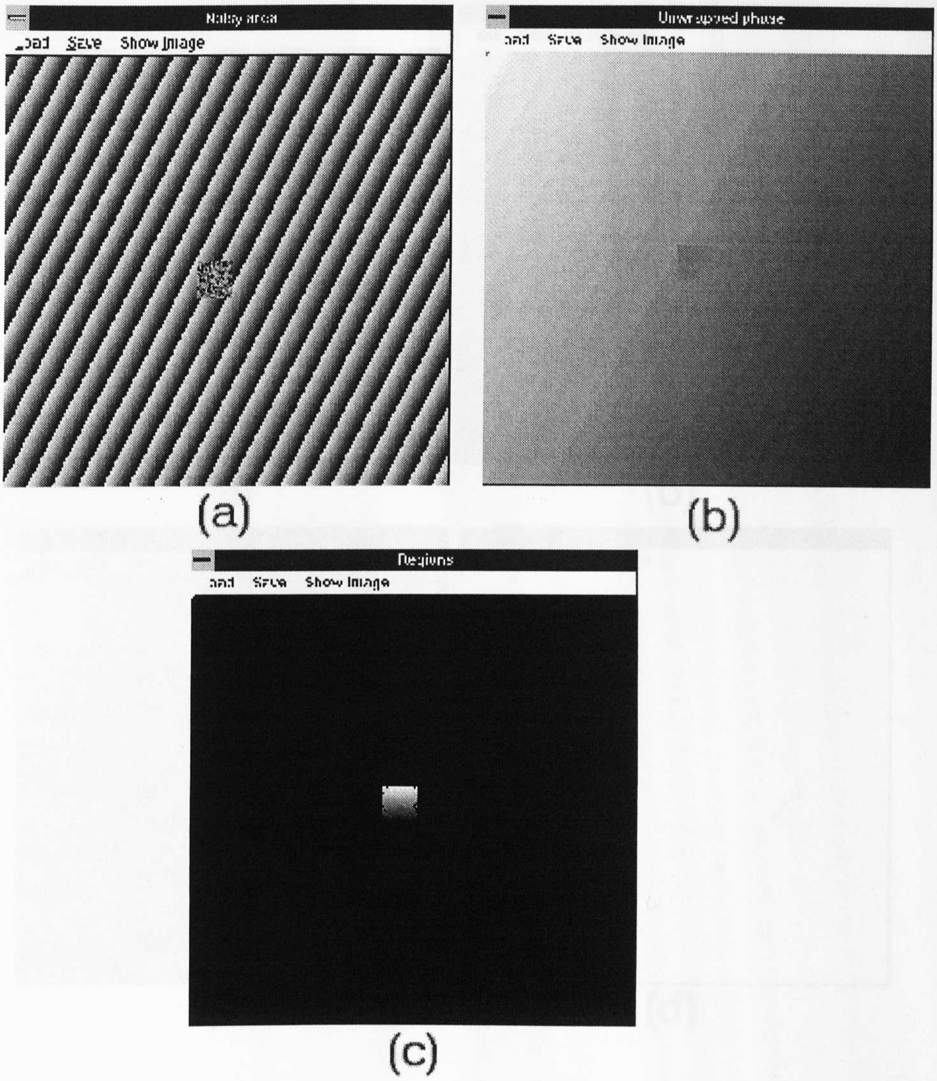


Figure 55. (a) Wrapped phase distribution, (b) unwrapped phase distribution, (c) isolated groups.

The second test involves the inclusion of a noisy area into the fringe pattern. The noise observed in Figure 55(a) is of random nature. Figure 55(b) is the unwrapped phase. The square observed in the middle is the noisy square in the initial wrapped phase distribution. Looking at Figure 55(c), it is possible to observe that this square belongs to different regions to the rest of the image and that the algorithm has isolated the noisy area.

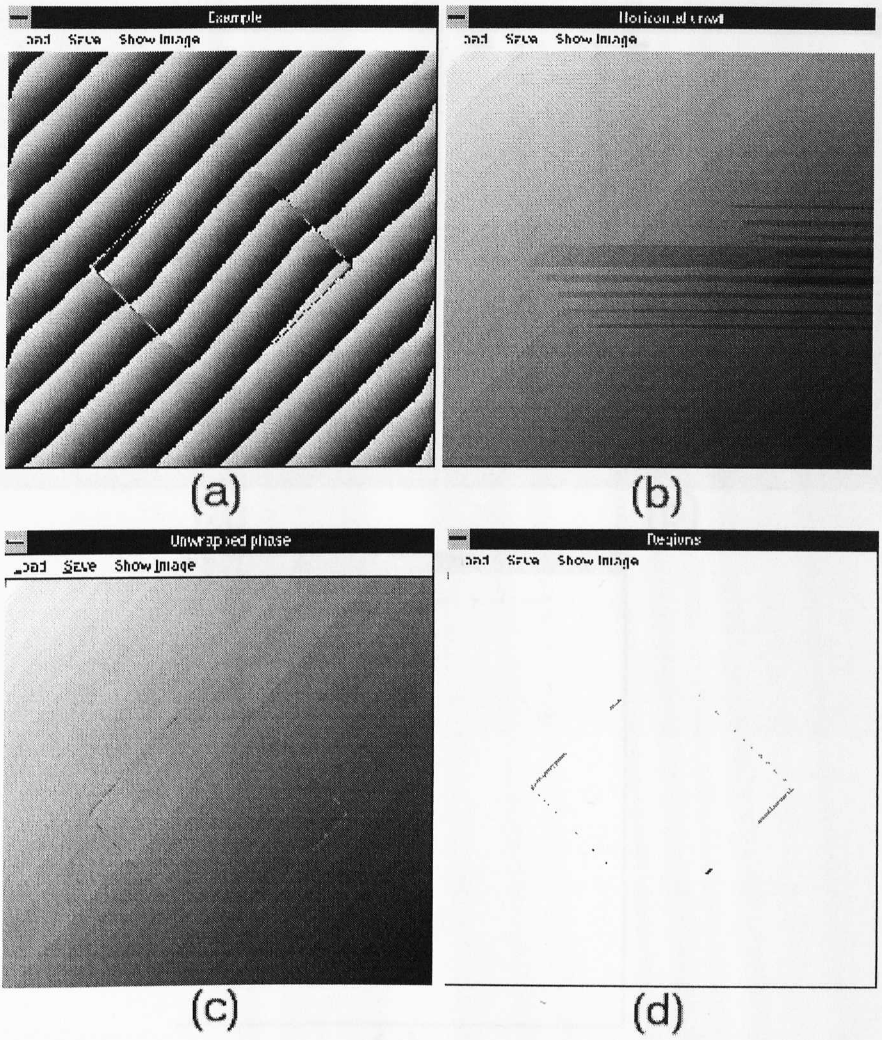


Figure 56. (a) Wrapped phase distribution, (b) horizontal unwrapping, (c) unwrapped phase distribution, (d) isolated groups.

The third image tested is a simulated image containing the results obtained by Fourier analysis when the original image has a large discontinuity in the borders of a square. The Fourier Transform performed on the image will deform the wraps as is shown in Figure 56(a). Even if it seems that the image is quite easy to unwrap, it presents serious problems in the borders. Figure 56(b) illustrates the results obtained with a typical horizontal unwrapping. Figure 56(c) is the result obtained by the unwrapper presented here. The darker and lighter areas are not errors in the unwrapping process. They are just areas belonging to different groups and that the unwrapper has qualified as independent. The different regions are illustrated in Figure 56(d).

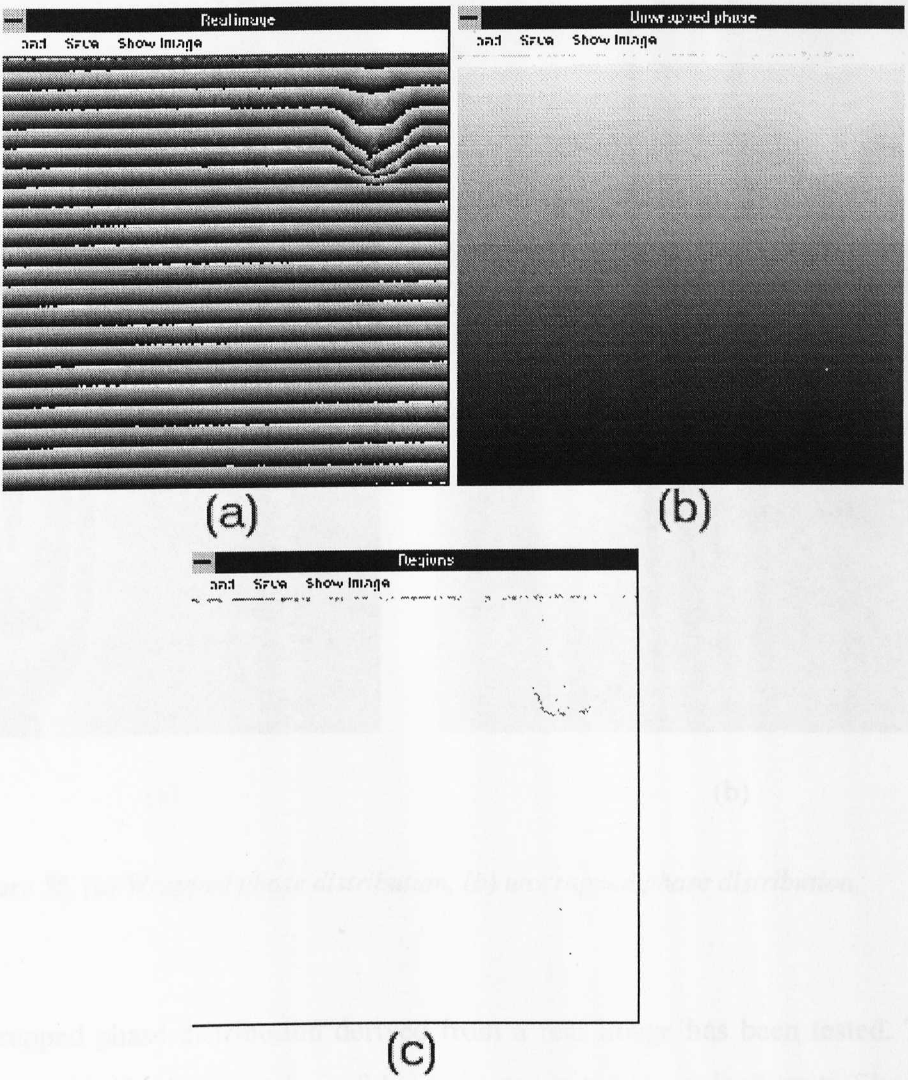


Figure 57. (a) Wrapped phase distribution, (b) unwrapped phase distribution, (c) isolated groups

A wrapped phase distribution obtained from a real image has also been unwrapped with this algorithm. The phase distribution presents some problems in the concave area (right-top area) . The algorithm detects the problems and avoids the bad areas. Errors can also be observed in the borders of the image. Figure 57(a) shows the original wrapped phase distribution. Figure 57(b) shows the unwrapped phase and, finally, Figure 57(c) shows the different regions that have been isolated.

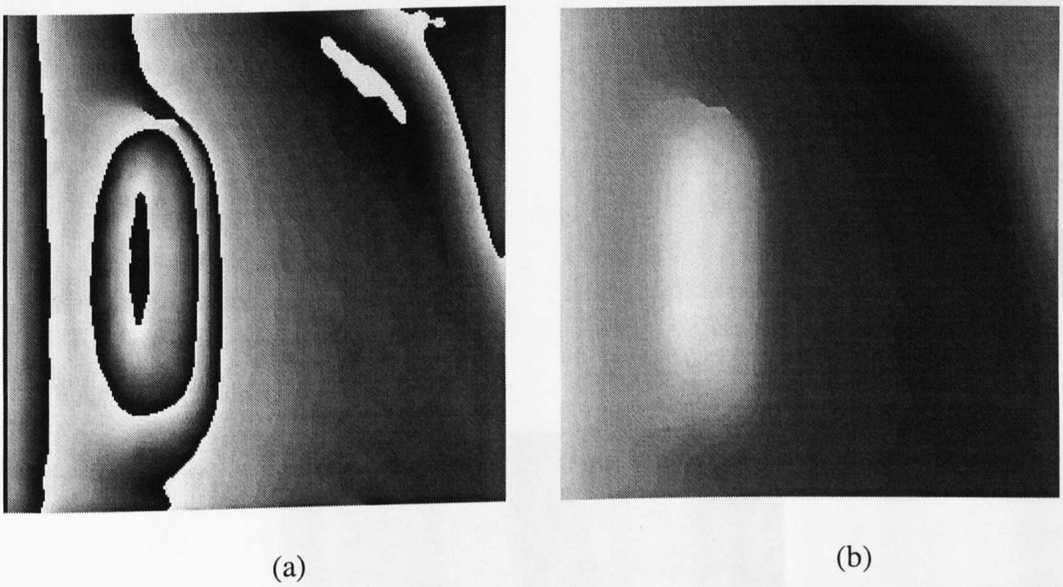
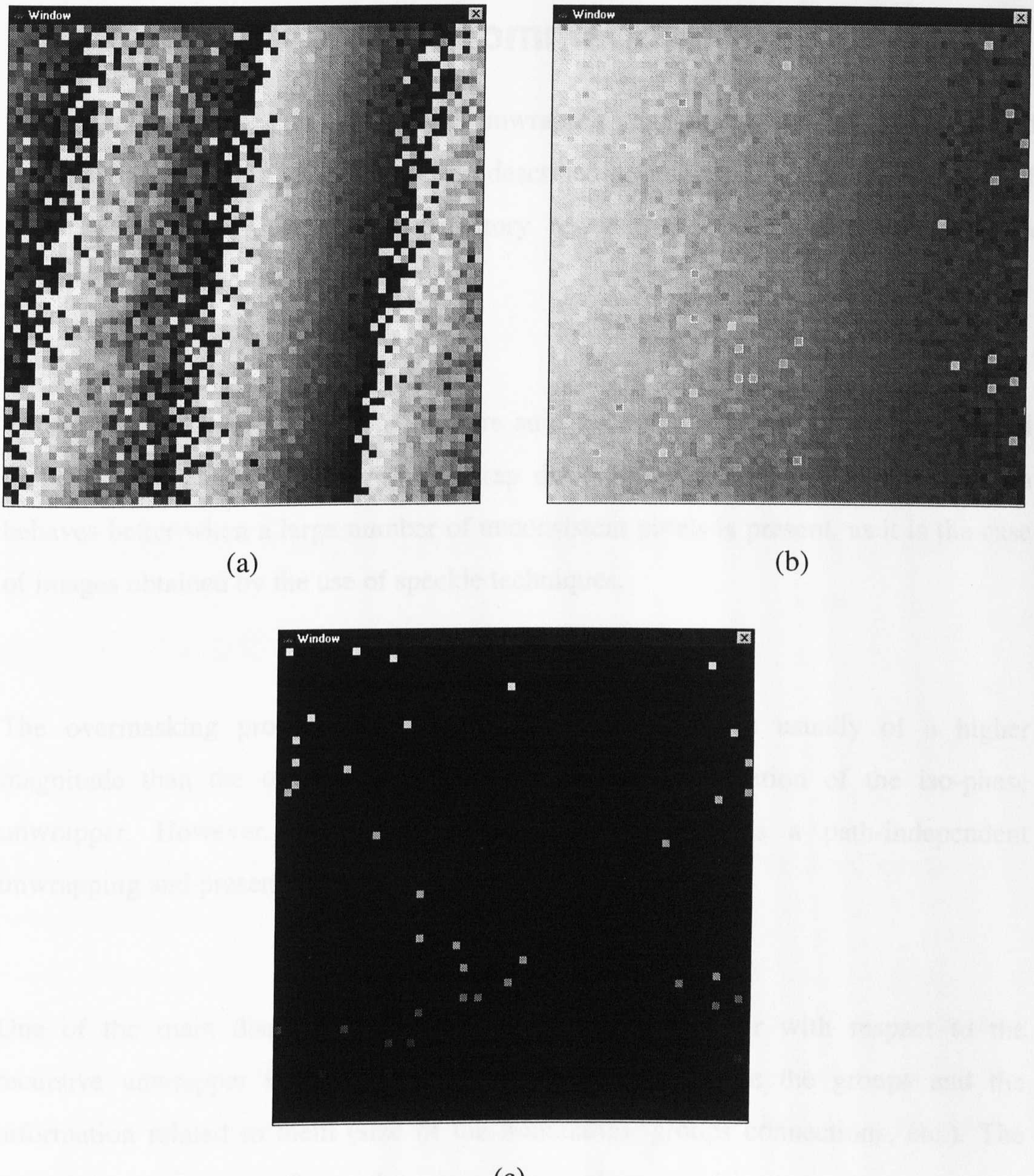


Figure 58. (a) Wrapped phase distribution, (b) unwrapped phase distribution.

Another wrapped phase distribution derived from a real image has been tested. The wrapped phase distribution contains a false wrap generated by an inaccurate filtering in the Fourier domain. Figure 58(a) shows the wrapped phase distribution. Figure 58(b) illustrates the result obtained with the iso-phase algorithm. A single group was returned as a result.



(a)

(b)

(c)

Figure 59. (a) Wrapped phase distribution, (b) unwrapped phase distribution, (c) isolated groups.

Finally, the algorithm was tested on the wrapped phase distribution illustrated in Figure 59(a). This image has high levels of noise that could produce a large amount of errors in the final unwrapped phase distribution. The algorithm produces a consistent and correct unwrapped phase distribution that is shown in Figure 59(b). Unreliable points are detected by the iso-phase unwrapper and isolated as independent regions, as illustrated in Figure 59(c). The results obtained are satisfactory.

4.4 Summary and Recommendations

This chapter has presented two novel unwrapping techniques, namely recursive and iso-phase unwrapping. They have been described in detail and tested on a number of wrapped phase distributions. Satisfactory results have been presented for both unwrapping techniques.

The recursive unwrapper is usually more suitable for images with a relatively small number of unconsistencies or short wrap discontinuities. The iso-phase algorithm behaves better when a large number of unconsistent pixels is present, as it is the case of images obtained by the use of speckle techniques.

The overmasking produced by the recursive unwrapper is usually of a higher magnitude than the overmasking resulting from the application of the iso-phase unwrapper. However, the recursive unwrappper guarantees a path-independent unwrapping and presents shorter execution times.

One of the main disadvantages of the iso-phase unwrapper with respect to the recursive unwrapper is the need for extra memory to store the groups and the information related to them (size of the boundaries, groups connections, etc.). The recursive unwrapper performs the operations in-place, requiring no extra memory to store data.

It is concluded that the decision as to which unwrapper should be utilised depends upon the particular wrapped phase map under analysis and the user requirements. The iso-phase unwrapper needs a larger amount of resources, in terms of memory and processing power. However, it may be more appropriate under certain circumstances, such as when high levels of noise are present.

Chapter five

Parallel implementation of cellular automata

5.1 Introduction

Cellular automata are dynamic systems in which space and time are discrete. Their behaviour is completely specified in terms of a local relation.

Cellular automata have been used in the past to simulate the evolution of cells in micro-biology. Very similar problems arise in the field of image analysis, where the requirement is to take an image, or a set of images, and to apply a number of operators (Sternberg 1983).

Their application has become widespread in many other areas such as physics and chemistry. Dendritic growth, simple SLANG robots, wire world, wave rule and hodge rule are examples of such automata. Their application to the development of unwrappers in the last few years is of special concern here.

Cellular automata are, normally, computationally intensive. The excessive time involved in their execution imposes a limitation to their applicability.

Parallel computing (the use of several processing units performing concurrent tasks) has been applied in the past to the resolution of many different problems that would be practically unsolvable on a single processor machine. Executing tasks on different processors at the same time can significantly reduce the computation time of a program, providing a potential for high performance at a reasonable cost.

The parallel processing principle is not a new concept. Discussions of parallel computers are found in the literature as far back as the 1920s (Denning 1986). High performance parallel computers have been constructed in the past, and many different architectures have been attempted. A number of these parallel computers are described in (Hockney and Jesshope 1981).

From a computational point of view, a cellular automaton is an array of values that evolves in discrete time steps to change their value according to certain neighbourhood rules.

Cellular automata are inherently parallel. The application of array processors[4] is obvious to the problem and it has been discussed for the development of Ghiglia's unwrapper in a paper by Spik at *al.*(1991). However, these systems are not general purpose machines and normally present an overwhelming cost. This chapter describes different implementations of a cellular automaton in a general purpose distributed memory parallel processor. A different parallel implementation paradigm is utilised for each implementation. The advantages and disadvantages of each algorithm in terms of efficiency and scalability are discussed.

5.2 Distributed memory parallel computers

Parallel computers can be divided into two main categories: on-chip and off-chip. On chip parallelism relies on architectural enhancements for improved performance,

[4] In an array processor the arithmetic unit which is equivalent to that of a serial computer is duplicated. There is a single control unit to direct activities. An example is an ICL DAP which has 4096 processors on a 64 by 64 grid.

while off-chip parallelism incorporates additional processors (TEXAS INSTRUMENTS 1994).

Examples of on-chip parallel processors include the Intel i860, the Digital Alpha and the MIPS R4000.

Distributed memory parallel computers or multicomputers are machines composed of a number of single processors, each with its own processing unit and memory. These single processors provide facilities for physical connection with other processors so that larger systems (networks) can be built.

The fact that memory is distributed limits the capability of these systems. Every processor has direct access to the data stored in its memory. If data contained in other processing unit needs to be operated on by a different processor, data have to be communicated through the network. Communication operations are generally very costly timewise.

The transputer is an off-chip distributed memory parallel computer. Its general architecture is depicted in Figure 60. It contains a floating point unit, an internal memory, four communication links and a control unit. The T800 transputer also contains an integer arithmetic unit. This unit, the links and the floating point unit can operate in parallel, enhancing the performance of the system. Transputers are widely used in embedded systems because of their high level of hardware integration. For a more detailed description of the transputer architecture, the reader is referred to (INMOS 1988; Thompson et al. 1990).

The T800 transputers used for the experimentation contain a four Kbytes internal memory and two Mbytes of external memory. The communication bandwidth for a single link is 10 Mbytes/sec.

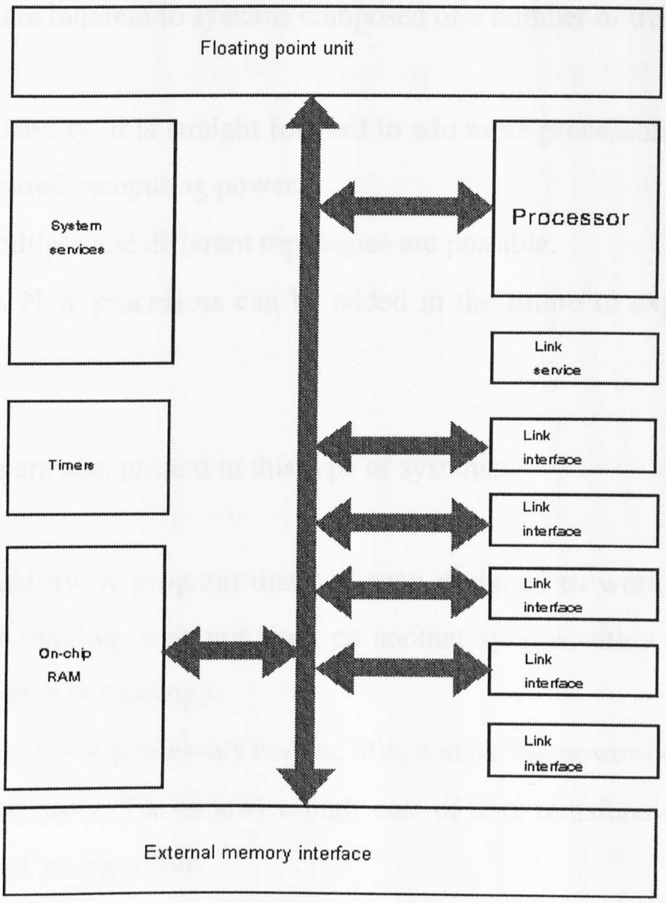


Figure 60. Transputer architecture.

It is possible to build useful, powerful and inexpensive parallel computers using transputers (Cok 1991). There are a number of features that are common to these systems:

- No transputer shares any memory with any other transputer.
- Each link has two channels that provide point-to-point synchronised communication between transputers.
- The transmission between transputers is unidirectional along each channel.
- The transputer can communicate in parallel along all eight channels whilst performing arithmetic.
- Special direct memory access hardware can move data quickly into and out of the links.

Several benefits are inherent to systems composed of a number of transputers:

- Hardware scalability. It is straight forward to add more processors to the system to obtain the required computing power.
- Flexibility. Multiple and different topologies are possible.
- Upgradability. New processors can be added in the future to expand the original system.

Several problems are also present in this type of systems:

- Software scalability. A program that has been designed to work on a number of processors and topology may not work on another system with a different number of processing units or topology.
- Load balancing. Some processors may be idle, computing power being wasted.
- Communication costs. The timewise high cost of data transferences can limit the parallelisation of an algorithm.
- Complicated organisation of data. Full control over the location of the data is required so that computation can be performed as locally as possible.
- Speed does not normally increase linearly with the system cost. A logarithmic function is more common.

Critical to the effective utilisation of a parallel computer is the ability to minimise communication latencies (Kumar and Sun 1994) and to achieve a load balance. For a given problem instance, speedup saturates at a certain number of processors either because parallel processing overheads grow with the number of processors or because the number of processors exceeds the degree of concurrency of the algorithm. These topics have been analysed in a number of papers (Amdahl 1967; Gustafson 1988; Kumar et al. 1994b; Shaffer 1989). High performance computing is an important objective of parallel systems, but it is not its only aim (El-Rewini and Lewis 1993). Other objectives so exist such as reliability, resource sharing and extensibility (Ananda and Srinivasan 1991).

The analysis of the algorithms presented in this chapter focuses on their performance and scalability. Many different parallel performance metrics have been proposed in the past (Barton and Withers 1989; Eager et al. 1989; Hockney and Curington 1989; Karp and Flatt 1990; Kumar 1988; Marinescu and Rice 1994; Sun and Gustafson 1991; Worley 1990). This work concentrated on the speedup and the isoefficiency function (Grama et al. 1993; Kumar and Gupta 1994) as metrics for the performance of the algorithms and their respective software scalability.

The isoefficiency function dictates the growth rate of the size of the problem required to keep the efficiency fixed as the number of processors increases. This function determines the ease with which a parallel system can maintain a constant efficiency and hence achieve speedups increasing in proportion to the number of processors. A small isoefficiency function value means that small increments in problem size are sufficient for the efficient utilisation of an increasing number of processors, indicating that the parallel system is highly scalable. On the other hand, a large isoeffiency function value indicates a poorly scalable system. The isoefficiency function does not exist for unscalable parallel systems, because in such systems the efficiency cannot be kept at any constant value as the number of processors increases, no matter how fast the problem size is increased (Kumar et al. 1994a).

On the software side, parallel languages can be classified into two groups:

- Expression of parallelism. The language provides the syntax to reflect the machine architecture or the demand that the programmer must explicitly encode hardware instructions in separate subroutine codes. Occam 2 and FORTRAN90 Plus belong to this group of parallel languages. Occam 2 provides a machine dependent implementation and it is very suitable for use in transputers systems. FORTRAN 90 Plus also requires the explicit expression of the parallelism involved in the programs. These type of compilers normally produce non portable code, since the parallelism is expressed for a particular architecture.

- Detection of parallelism. The programmer writes sequential code. The compiler detects the parallelism. The programmer may need to re-structure the code to suit the compiler. HPF (High Performance FORTRAN) belongs to this group of compilers. Portability is one of the major features of the language. The code only needs to be recompiled for a different architecture (although some minor changes may be required to adapt the code to the new architecture).

The use of HPF and occam 2 will be discussed later in this chapter, in the context of the implementation of the algorithms.

5.3 Description of the system

An outline of the system was given in Chapter 4. A more detailed description is given in this section. A parallel machine composed of four transputers is used for the experimentation. This is illustrated in Figure 61. Four transputer cards are connected into four different slots in a parallel tower case. Links are physically connected via serial cables.

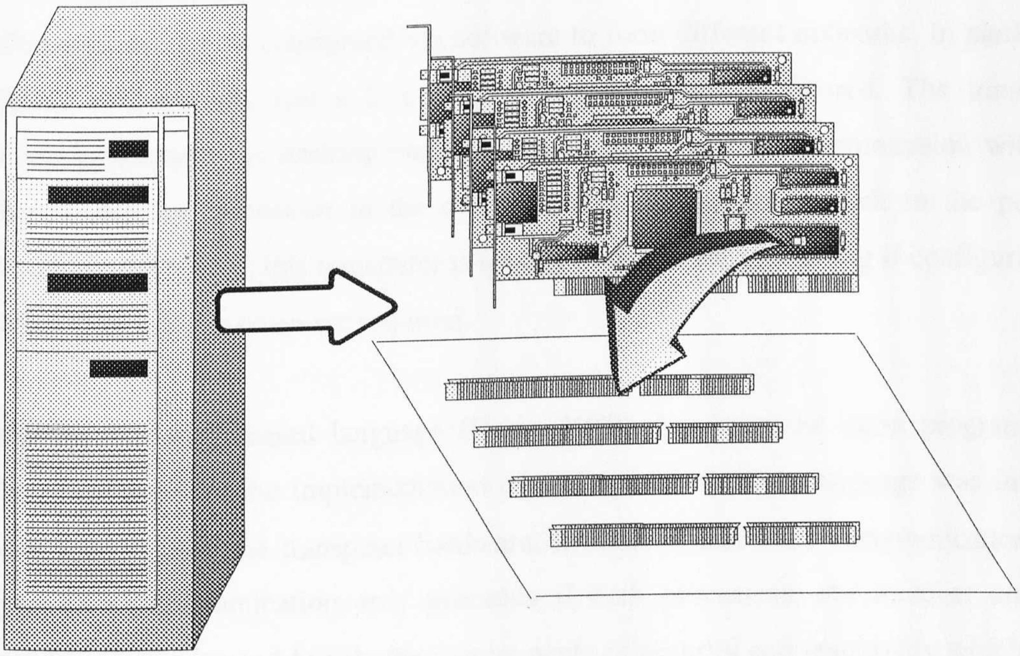


Figure 61. Four transputer cards are placed into a transputers board.

This parallel configuration is connected to a fifth transputer card installed in one of the ISA slots of a desktop computer. The computer acts as a host, allowing the input/output to the screen and providing the required filing system. It also manages the booting process. The system is illustrated in Figure 62.

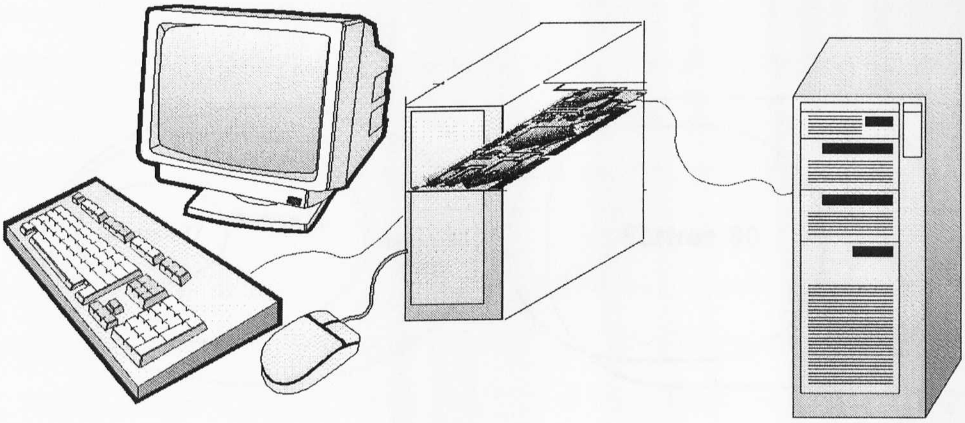


Figure 62. The parallel machine is connected to a fifth transputer card linked to a desktop computer.

The physical links established in the parallel machine allow multiple logical layouts: The system can be configured via software to form different networks. In particular, linear chains, trees and a 2 x 2 mesh can be easily configured. The transputer connected inside the desktop employs one of the links for communication with the host computer (processor in the desktop), and another one to link to the parallel machine. Therefore, this transputer is usually unusable for processing if configurations other than a linear chain are required.

Occam 2, a CSP based language (Hoare 1978), has been the main programming language used for the implementation of the algorithms. This language was initially developed to suit the transputer hardware. It offers synchronous communication as a primitive (communication only proceeds if both processors, the receiver and the sender, are ready) and fulfills the requirements of security and readability which befit a modern programming language (East 1995). Under the principle of 'keep it simple', occam 2 does not provide complex data structures such as pointers or records. A strict syntax based on specific indentations facilitate the readability of the code, forcing the programmer to write in a structured manner. Occam is highly effective for directly exploiting the speed, parallelism and communication abilities of the transputer. Code written in occam achieves an execution efficiency similar to transputer assembly code. Further information about the language can be found in (Galletly 1990; INMOS 1984; INMOS 1992; Jones 1987; Jones and Goldsmith 1988; Kerridge 1987; Wilson 1983).

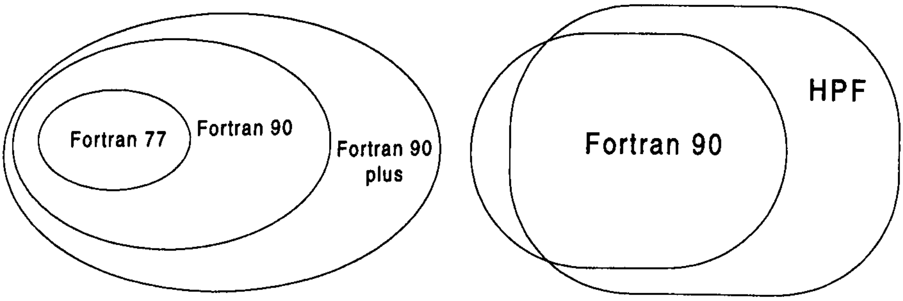


Figure 63. Relationship between FORTRAN77, FORTRAN90 and HPF. FORTRAN 90 includes the whole FORTRAN77 set of instructions as a subset. FORTRAN90 plus, the parallel version of FORTRAN90, includes FORTRAN90 as a subset. HPF is FORTRAN90 based. Not every FORTRAN90 instruction is included in HPF, but HPF provides a series of libraries that are not included in the FORTRAN90 set.

Even if the main language for implementation has been occam 2, a new HPF (High Performance FORTRAN) compiler for transputer systems has also been tested with the development of the algorithm. HPF is a new, powerful and easy to program language that has been created for the development of parallel applications. The philosophy of HPF is very different from the occam 2. HPF is FORTRAN 90 based (Figure 63) and allows data parallelism to be implemented with no difficulty (the development of a parallel application reduces to the development of a sequential application plus a short list of simple commands that describe how data is to be distributed). HPF presents a higher level of abstraction, moving the task of parallelisation from the programmer to the compiler (the programmer only has to 'help' the compiler by providing a series of hints about the distribution of the data). This language focuses on the portability of the code and ease of programming, whilst maintaining a high code efficiency. The concept is very similar to the use of assembly code in sequential programming. An expert programmer can achieve higher efficiencies with a low level language, by sacrifying portability and extendibility of the code but, is it really worthwhile? If it is, why have languages as COBOL, FORTRAN or C became the standards at their time for business, mathematics and engineering respectively? Why is the use of 4GLs (four generation languages) becoming widespread nowadays? Why is assembly code only rarely used in the development of operating system kernels, where a slight increment in efficiency results in a high increase of the overall performance of the system? Is HPF the alternative for parallel programming? Will it become the standard?

Looking at the questions above, it can seem obvious that the use of the HPF compiler is greatly advantageous. However, is not parallel programming a paradigm to achieve high speed computation? Is it affordable to lose code efficiency on a parallel algorithm? Does it not result in a high increment in cost since more processors are to be used? Is parallel processing not a problem in which the loss of speed on a single processor results on a bottleneck for the whole system? May it be that the loss of speed in the system produces a non achievable speedup (Amdahl 1967; Gustafson 1988; Heath and Worley 1989) that may be required for real time applications? Since parallelisation is performed by the compiler, would this not result in an inaccurate speedup model leading to unexpected results?

The choice between a high or low level programming language is clearly dependent upon the requirements of the application. A loss of speed may be compensated by the addition of processors to the system, that could be cheaper than the additional development cost produced by the use of a language such as occam 2. On the other hand, other problems may require a high speedup that may not be achieved by the use of a high level compiler such as HPF, independently of the number of processors used.

The investigation into the use of HPF in this work goes a small way to answering some of these questions.

Note that, although the algorithms have been designed for a four processor machine, they can be easily extended for a larger number of processors.

5.4 Implementations

This section describes three different parallel algorithms developed. They use different parallel computing programming paradigms to produce efficient code for the implementation of cellular automata.

5.4.1 Implementation 1: Geometrical decomposition

The first implementation of the algorithm follows a domain decomposition or data-partitioning approach. Data are geometrically distributed through the network. Every processor receives the same amount of data. The only communication required (apart from the initial distribution) is the transmission of the borders of every processor's domain at the end of every iteration.

The implementation of the algorithm is simple. Data are initially distributed from a master processor, a quarter of the image to each processor. Every processor also receives the points surrounding the borders of the quadrants so that the values at the borders can be calculated. Every iteration is calculated as follows:

The values at the borders of the quadrants are processed first. The calculation of the values of the rest of the points are performed in parallel with the communication of the values at the borders. In this manner, the transmission is hidden and no time overhead is produced.

After the required number of iterations have been processed, the result is sent back to the host processor.

5.4.2 Implementation 2: Functional decomposition.

A second algorithm considers functional decomposition for the implementation of the cellular automaton. Pipelining is used as the basis for constructing an algorithm in which every processor works at a different iteration and they all combine their power to process the required number of iterations. The processors are laid in a closed linear chain network as illustrated in Figure 63.

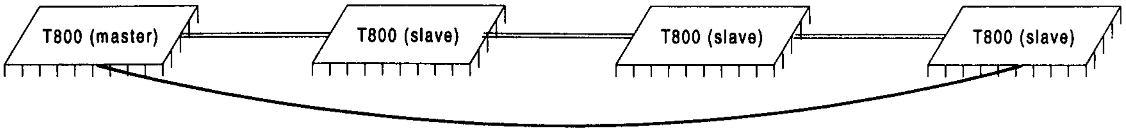


Figure 64. Linear chain configuration.

The first processor calculates the first iteration on the first three rows and the first two of these lines are sent to the second processor. Communication of the first line is carried out in parallel with the processing of the second line and so on. Processor *p* can start the calculation of its iteration as soon as it receives the two first lines from processor *p-1* (note that the neigbourhood is required to proceed to calculation). The network proceeds in a pipeline fashion. Computation of a line is performed in parallel with the transmission of the previous line. The process is repeated throughout the network so that every processor is working at a different iteration (three lines behind that of its predecessor in the linear chain as illustrated in Figure 65). Finally, the result is sent back to the first processor.

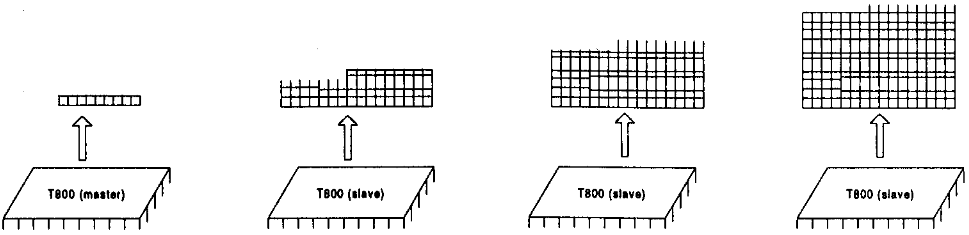


Figure 65. Pipelined process. Every processor has a delay of three lines with respect to the processor at its left.

5.4.3 Implementation 3: Farming.

Another implementation has been considered for analysis. The matrix is divided into packages. These packages include a certain region and the borders. A linear chain is considered as the network utilised for the implementation of this algorithm. This configuration is illustrated in Figure 66.

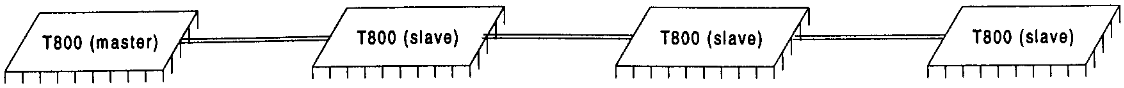


Figure 66. linear chain layout of transputers.

The algorithm follows the processor farming paradigm. Work is distributed non-uniformly across the processors. Every slave asks the master processor for packages as soon as they have finished the processing of the previous packages. Slaves work with two packages: one being processed and another one in memory so that the processor is not idle whilst receiving the next package. After the required calculations have been performed, the processed package is sent back to the master processor. When an iteration is finished, the master divides the matrix in packages again, and the same operation is performed.

Note that the algorithms have been implemented for the case in which every point only requires the information of its eight surrounding neighbours, as illustrated in Figure 67.

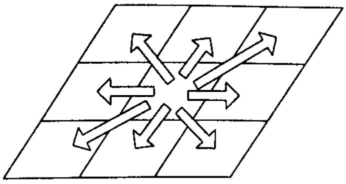


Figure 67. Every point requires its eight surrounding neighbours for calculation.

5.5 Speedup analysis of the algorithms

Speedup analyses are performed with parallel algorithms to determine if a parallelisation is worthwhile. The speedup S_u of a parallel algorithm is defined as

$$S_u = \frac{Sequential \quad execution \quad time}{Parallel \quad execution \quad time}$$

A speed up of one or lower determines that the time achieved by the parallel implementation is not better than the time consumed by the execution of the algorithm on a single processing unit. A speed up close to the number of processors in the system indicates a worthwhile parallelisation.

Another parameter, the efficiency *E,* is defined as

$$E = \frac{S_u}{Number \quad of \quad processors}$$

An efficiency function that is close to a constant indicates a good scalability. If this function drops quickly, the parallel implementation fails to provide valuable speed ups when the number of processors is augmented.
A speed up model produces an approximation to the speed up that would be achieved by a parallel implementation of the algorithm. It is also useful to evaluate the software scalability of the algorithm (how the algorithm responds to an increase on the number of processors in the system).

Speed up models follow an instruction evaluation approach. The algorithm is divided into small steps. These steps are analysed to determine their execution time with regard to three parameters:

T_C, the time required to transmit a single information item,
T_S, the time required to start a communication and

T_f, the time required to execute a single floating point operation (normally taken as an average).

These three parameters can be obtained by simple experimentation. For a T800 transputer, the processor that has been utilised throughout the experimentation, these times are given by (data is considered to be floating point and 32 bits):

$T_c = 4.4 \times 10^{-6}$ seconds

$T_s = 12.2 \times 10^{-6}$ seconds

$T_f = 2.3 \times 10^{-6}$ seconds (average value)

For the speed up estimation, the following constants are also assumed:

p x p, the number of processors in a mesh configuration (it is assumed to be a square network),
p, the number of processors in a linear chain configuration,
n x n, the size of the array of data (for simplicity, it is assumed to be a square matrix),
u, the number of floating point operations required per array element and
i, the number of whole field iterations required.

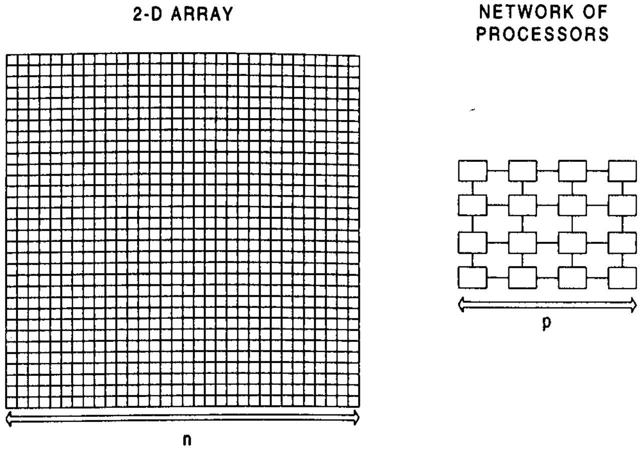


Figure 68. Illustration of the parameters p and n.

A load balanced case is also assumed (note that the speed up model is only an approximation to the real case).

These sections analyse the speedup models of the two first algorithms (mesh and pipelined approach). The speedup model of the third implementation depends heavily on the number of processors of the system (as the larger the number of processors, so the heavier the communication load in the system).

5.5.1 Mesh approach

The data contained on the initial processor is to be delivered through the network. Two links are utilised for this process by simulating two linear chains that perform a parallel communication. A total of n^2-$(n/p)^2$ data are to be distributed and $(p^2-1)/2$ packets are delivered (the number of processors in each linear chain). Every slave is to receive a packet of information.

If a parallel communication of *n* packets of information is assumed in a linear chain of *p* processors, the following table illustrates the communication.

TIME	MASTER	SLAVE 1	SLAVE 2	...	SLAVE n
0	$X_1, X_2, ..., X_n$	-	-	-	-
T	$X_1, X_2, ..., X_n$	X_n	-	-	-
2T	$X_1, X_2, ..., X_n$	X_n, X_{n-1}	X_n	-	-
...	$X_1, X_2, ..., X_n$	...	...	...	-
nT	$X_1, X_2, ..., X_n$	$X_1, X_2, ..., X_n$	$X_2, X_3, ..., X_n$	...	X_n

Table 1. Parallel communication in a linear chain.

where T is the time required to transmit a single packet between two transputers and X_k is the packet that is to be delivered to slave *k*.

The time required to perform this communication is given by

$$T_1 = \left[n^2 - \left(\frac{n}{p} \right)^2 \right] T_c + \frac{p^2 - 1}{2} T_s$$

The transmission of the borders can now be undertaken in parallel (every processor sends its elements at the same time), producing a time overhead of

$$T_2 = T_s + \frac{n}{p} T_c$$

The processing of the items is undertaken in parallel (every processor calculates the values for the next iteration concurrently) and consumes a time given by the expression

$$u \left(\frac{n}{p} \right)^2 T_f$$

As was explained in the previous section, borders are processed first so that communication of the borders can be performed in parallel with the calculation of the rest of the array. In the general case, the time to perform the communication is far smaller than the processing time of the rest of the data. Therefore, communication is hidden and iterations are performed in parallel with no degradation in performance.

The required time to perform the *i* iterations is

$$T_3 = iu \left(\frac{n}{p} \right)^2 T_f$$

Finally, the resulting data has to be sent back to the original processor, taking a time that is identical to the initial distribution

$$T_4 = \left[n^2 - \left(\frac{n}{p} \right)^2 \right] T_c + \frac{p^2 - 1}{2} T_s$$

The total parallel time is given by the expression

$$\begin{aligned}
T_{par} &= T_1 + T_2 + T_3 + T_4 \\
&= 2\left(\left[n^2 - \left(\frac{n}{p}\right)^2\right]T_c + \frac{p^2-1}{2}T_s\right) + T_s + \frac{n}{p}T_c + iu\left(\frac{n}{p}\right)^2 T_f \\
&= \left[2n^2 - 2\left(\frac{n}{p}\right)^2\right]T_c + \left(p^2-1\right)T_s + T_s + \frac{n}{p}T_c + iu\left(\frac{n}{p}\right)^2 T_f \\
&= \left[2n^2 - 2\left(\frac{n}{p}\right)^2 + \frac{n}{p}\right]T_c + p^2 T_s + iu\left(\frac{n}{p}\right)^2 T_f
\end{aligned}$$

The sequential time T_{seq} is given by the expression

$$T_{seq} = iun^2 T_f$$

and the speed up can be written as

$$S_u = \frac{T_{seq}}{T_{par}} = \frac{iun^2 T_f}{\left[2n^2 - 2\left(\frac{n}{p}\right)^2 + \frac{n}{p}\right]T_c + p^2 T_s + iu\left(\frac{n}{p}\right)^2 T_f}$$

If *iu* is considered to be much larger than *p* and *n*, the previous equation can be approximated by

$$S_u = \lim_{iu \to \infty} \frac{iun^2 T_f}{\left[2n^2 - 2\left(\frac{n}{p}\right)^2 + \frac{n}{p}\right]T_c + p^2 T_s + iu\left(\frac{n}{p}\right)^2 T_f}$$

$$S_u = \lim_{iu \to \infty} \frac{iun^2 T_f}{iu\left(\frac{n}{p}\right)^2 T_f} = \frac{n^2}{\left(\frac{n}{p}\right)^2} = p^2$$

This can be interpreted as the speed up tends to be proportional to the number of processors in the system (efficiency near to one).

However, if the number of iterations is negligible with respect to the number of data in the system, the limiting speed up is given by

$$S_u = \lim_{n\to\infty} \frac{iun^2 T_f}{\left[2n^2 - 2\left(\frac{n}{p}\right)^2 + \frac{n}{p}\right] T_c + p^2 T_s + iu\left(\frac{n}{p}\right)^2 T_f}$$

$$S_u = \lim_{n\to\infty} \frac{iun^2 T_f}{\left[2n^2 - 2\left(\frac{n}{p}\right)^2 + \frac{n}{p}\right] T_c + iu\left(\frac{n}{p}\right)^2 T_f}$$

$$S_u = \lim_{n\to\infty} \frac{iun^2 T_f}{\left[2n^2 - 2\frac{n^2}{p^2} + \frac{n}{p}\right] T_c + iu\frac{n^2}{p^2} T_f}$$

$$S_u = \lim_{n\to\infty} \frac{iun^2}{\left[2n^2 - 2\frac{n^2}{p^2} + \frac{n}{p}\right] \delta + iu\frac{n^2}{p^2}}$$

$$S_u = \lim_{n\to\infty} \frac{iun^2}{\left[2n^2 - 2\frac{n^2}{p^2}\right] \delta + iu\frac{n^2}{p^2}}$$

$$S_u = \lim_{n\to\infty} \frac{iun^2}{n^2\left(2\delta - \frac{2}{p^2}\delta + \frac{iu}{p^2}\right)} = \frac{iu}{2\delta - \frac{2}{p^2}\delta + \frac{iu}{p^2}}$$

$$S_u = \frac{iup^2}{2\delta p^2 - 2\delta + iu}$$

where δ is defined as

$$\delta = \frac{T_c}{T_f}$$

This equation reveals further rules of behaviour for the algorithm: If the number of iterations is high and the computation intensive, the speed up continues approaching p^2. On the other hand, if low computation and number of iterations are required, the speed up is dominated by the number of processors and the ratio δ. A lower speed up is then expected.

5.5.2 Pipelined approach

The pipelined approach does not require any intensive initial distribution of data. Two lines are first computed by the master and sent over to the second processor of the linear chain. The processing of the first line is performed in isolation. After the first line is computed, the rest of the lines are processed at the same time as the previous line is sent, exploiting the parallelism of the system.

The time to calculate the first line is given by

$$T_1 = nuT_f$$

and the time to compute the first iteration of a single line can be expressed as the maximum between the time to process the line and the time to send the previous line:

$$T_2 = \max(nuT_f, T_s + nT_c)$$

Assuming that the calculation time for a single line is more time consuming than the communication process (most usual case), the time required to perform the processing of the first iteration is given by

$$T_3 = nnuT_f = n^2 uT_f$$

By the time that the first processor has finished with the first iteration, the last processor will be processing the *n-3*p* row of the *p* iteration (note that every processor has a delay of three lines with respect to the previous processor). If there are more than *3(p+1)* lines to process (assumed case), the first line to be processed on the *p+1* iteration is available, and no delay would occur. Otherwise, the first processor would be idle until the last processor finishes with its last line. This would induce a delay produced by the sequential idleness of all the processors in the network (the idleness is also passed though). Therefore, the condition *n>3(p+1)* is imposed on the system. Any processor power above exceeding the condition *(n/3)-1* is wasted. This condition can be made more flexible by decreasing the packet size sent through the network (sending a fraction of the lines, instead of the entire row).

The main drawback of the algorithm is the computing power wasted at the end of the process. When the last iteration is calculated, the result has to be sent back to the first processor. The simplest implementation of this operation is to keep the rest of the processors to the right of the processor that performs the last iteration idle for computation. The only function they perform is the communication of the computed last iteration to the last processor. The worst case is to waste the whole processing power of the system during a whole iteration. The time for parallel execution is given by

$$T_{par} = n^2 u T_f \left\lfloor \frac{i}{p} \right\rfloor + n^2 u T_f = n^2 u T_f \left\lfloor \frac{i}{p} + 1 \right\rfloor$$

where $\lfloor \; \rfloor$ denotes the integer part.

and the minimum speed up achieved can be calculated as

$$S_u = \frac{T_{seq}}{T_{par}} = \frac{iun^2 T_f}{n^2 u T_f \left\lfloor \frac{i}{p} + 1 \right\rfloor} = \frac{i}{\left\lfloor \frac{i}{p} + 1 \right\rfloor}$$

This last expression reveals that the speed up will be very close to the number of processors if the number of iterations is large enough. This effect derives from the fact that as more iterations are required, so the last iteration becomes more negligible.

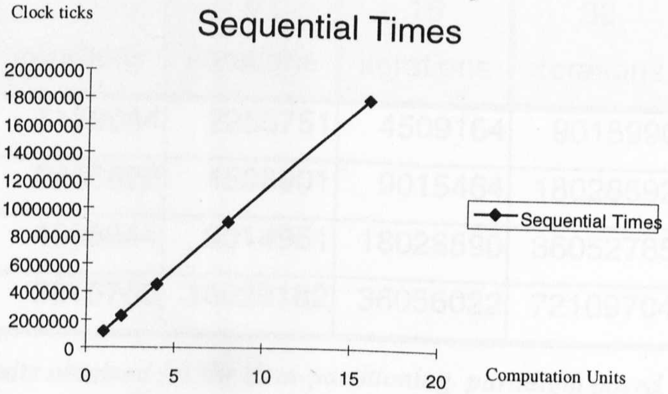
5.6 Results

The experiments have been carried out on a 128 x 128 image. A clock tick is the minimum unit of measure of time performed by the transputer[5]. A computation unit is defined as a defined amount of processing per pixel (two power operations, two divisions and two additions, all of them in 64 bits floating point format). Both the number of computation units and the number of iterations are varied so as to study the speedups as the processing requirements increase. The results obtained for the different implementations of the cellular automata are described in this section.

First of all, the sequential algorithm for the single iteration case has been implemented and executed. This algorithm consists of a loop that performs a certain amount of computation per pixel. The times obtained are illustrated in Table 2 and Graph 1.

Computation Units	Clock ticks
1	1126553
2	2253325
4	4506307
8	9014921
16	18029416

Table 2. Results for a sequential execution of a single iteration.



Graph 1. Representation of the sequential results obtained for a single iteration.

[5] 15625 clock ticks = 1 second

As it is represented in Graph 1, the time increases linearly with the number of computation units. If *n* iterations are required, the execution time will be the product of *n* by the time required to execute a single iteration. Therefore, another table with the sequential times for different numbers of iterations can be constructed (Table 3).

Computation Units	1 iteration	2 iterations	4 iterations	8 iterations	16 iterations	32 iterations
1	1126553	2253106	4506212	9012424	18024848	36049696
2	2253325	4506650	9013300	18026600	36053200	72106400
4	4506307	9012614	18025228	36050456	72100912	1.44E+08
8	9014921	18029842	36059684	72119368	1.44E+08	2.88E+08
16	18029416	36058832	72117664	1.44E+08	2.88E+08	5.77E+08

Table 3. Table generated to represent the results obtained by sequential execution of the algorithm.

These results are the basis for the calculation of the speedups obtained with the different algorithms described above.

Table 4 summarises the times obtained with the execution of the data-partitioning based algorithm. These times are expressed in clock ticks.

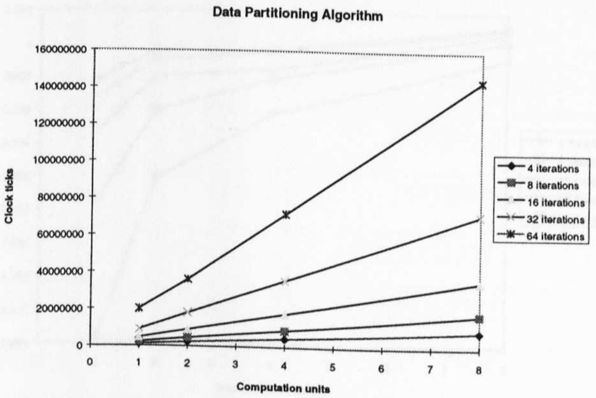
Computation Units	4 iterations	8 iterations	16 iterations	32 iterations	64 iterations
1	1129044	2255751	4509164	9015990	18029646
2	2255620	4508901	9015464	18028592	36054853
4	4508644	9014951	18028590	36052785	72103250
8	9015760	18029182	36056022	72109704	1.44E+08

Table 4. Results obtained for the data-partitioning paradigm based algorithm.

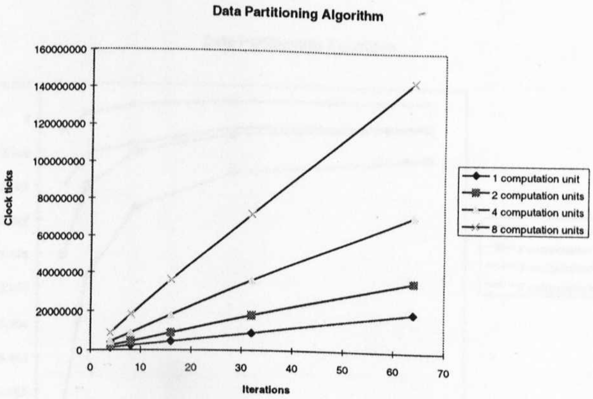
Computation Units	4 iterations	8 iterations	16 iterations	32 iterations	64 iterations
1	3.991175	3.99531	3.997381	3.998418	3.998936
2	3.99593	3.998003	3.99904	3.999558	3.999817
4	3.997927	3.998963	3.999254	3.999742	3.99987
8	3.999628	4.000146	4.000406	4.000536	4.000603

Table 5. Speedups obtained for the data-partitioning paradigm based algorithm

The graphs representing the times with respect to the computation units and the number of iterations are illustrated in Graph 2 and Graph 3.

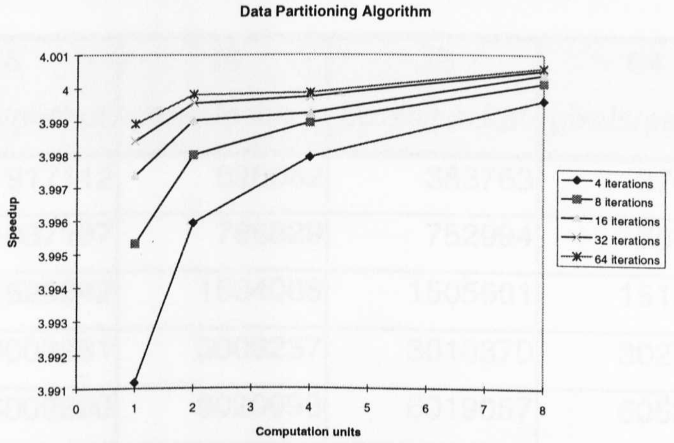


Graph 2. Results produced by the data-partitioning algorithm. Time with respect to the computation units..

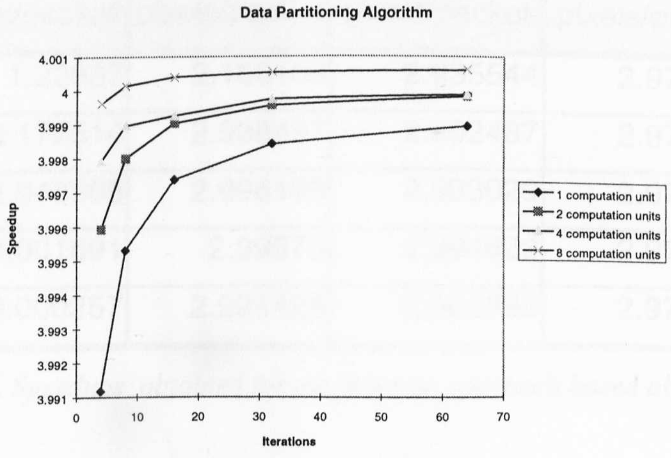


Graph 3. Results produced by the data-partitioning algorithm. Time with respect to the number of iterations required.

As it can be observed, the times obtained vary linearly with the amount of processing required (an increase in either the number of iterations or the computation units results in a linear increase of the computation time). The speedups have been calculated by dividing the values in Table 4 by the values in Table 3. These are represented in Table 5, Graph 4 and Graph 5. They are very close to the optimum value four in all the cases (efficiency close to one). It is possible to observe, however, that the speedup approaches four as the computation increases, either because the number of iterations or the computation units increase. This is produced by the fact that the initial distribution and gathering of the data becomes more insignificant as the processing becomes more intensive.



Graph 4. Results produced by the data-partitioning based algorithm. Speedups with respect to the computation units.



Graph 5. Results produced by the data-partitioning based algorithm. Speedup with respect to the number of iterations.

The second algorithm, based on the processor farming principle behaves in a different manner. The number of iterations does not affect the value of the speedup. However, the amount of processing per pixels has a major influence on the speedup. Variation of the iterations number is not described since it does not affect the speed up, as commented above. On the other hand, variation of the size of the packets sent to the slaves does affect the values of the speedup and an analysis has been carried out (the size of the packet determines the amount of communication undertaken). Table 6 illustrates the clock ticks obtained as a result. Table 7 represents the speedups.

The results have been illustrated in Graph 6 and Graph 7.

Computation	8	16	32	64	128
Units	pixels/packet	pixels/packet	pixels/packet	pixels/packet	pixels/packet
1	917112	520082	383763	378558	379845
2	1037197	766829	752994	757065	758102
4	1529342	1504008	1505601	1513858	1514672
8	3003281	3009237	3010370	3027211	3028093
16	6009290	6020993	6019057	6053889	6054563

Table 6. Results obtained for the farming approach based algorithm

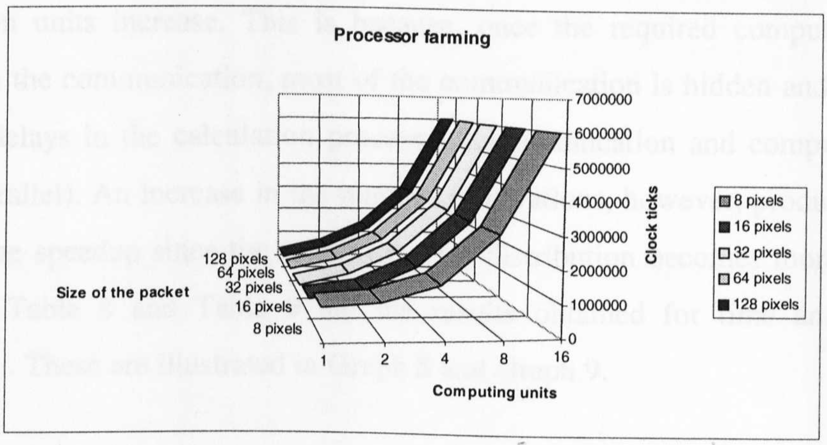
Computation	8	16	32	64	128
Units	pixels/packet	pixels/packet	pixels/packet	pixels/packet	pixels/packet
1	1.22837	2.166106	2.935544	2.975906	2.965823
2	2.172514	2.938497	2.992487	2.976396	2.972324
4	2.946566	2.996199	2.993029	2.976704	2.975104
8	3.001691	2.99575	2.994622	2.977963	2.977095
16	3.000257	2.994426	2.995389	2.978154	2.977823

Table 7. Speedups obtained for the farming approach based algorithm

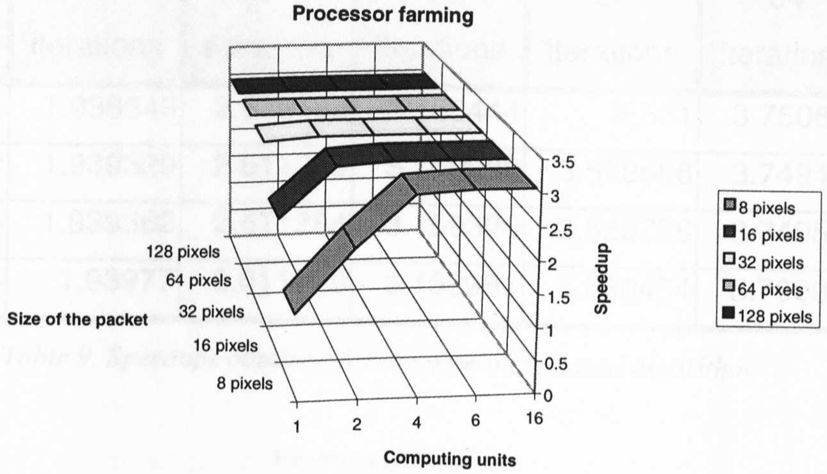
It can be observed that the packet size has a major influence in the result. The optimum packet size is determined by the computation units. Different processing requirement present a different optimum number of pixels per packet. This number can be determined easily by analysing the speedup model.

The maximum achievable speedup with this system is three, since the first processor does not perform any data computation (it is only utilised to control the communication). It can be observed from Graph 7 that, once the computation required is high enough to hide the communication overheads, the speedup is constant and very close to the optimum value three. This can be appreciated in Graph 7, for a packet size greater that 16 pixels. When the packet size is small, more communication overheads are produced, since a higher number of communication processes need to be started.

.



Graph 6. Results produced by the farming based algorithm. Time with respect to the number of computing units.



Graph 7. Results produced by the farming based algorithm. Speedup with respect to the number of computing units.

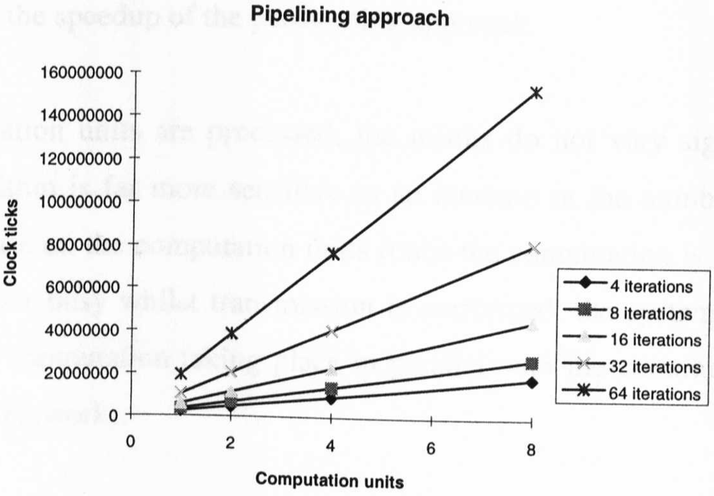
In the last approach (pipelining), the speedup does not vary considerably as the computation units increase. This is because, once the required computation takes longer than the communication, most of the communication is hidden and it does not introduce delays in the calculation processes (communication and computation take place in parallel). An increase in the number of iterations, however, produces a better value for the speedup since the initial time for distribution becomes more and more negligible. Table 8 and Table 9 are the results obtained for time and speedups respectively. These are illustrated in Graph 8 and Graph 9.

Computation Units	4 iterations	8 iterations	16 iterations	32 iterations	64 iterations
1	2324415	3450152	5703264	10209487	19221935
2	4647638	6903441	11411812	20428549	38462025
4	9294413	13805643	22821611	40853550	76917436
8	18589672	27612593	45645451	81711165	1.54E+08

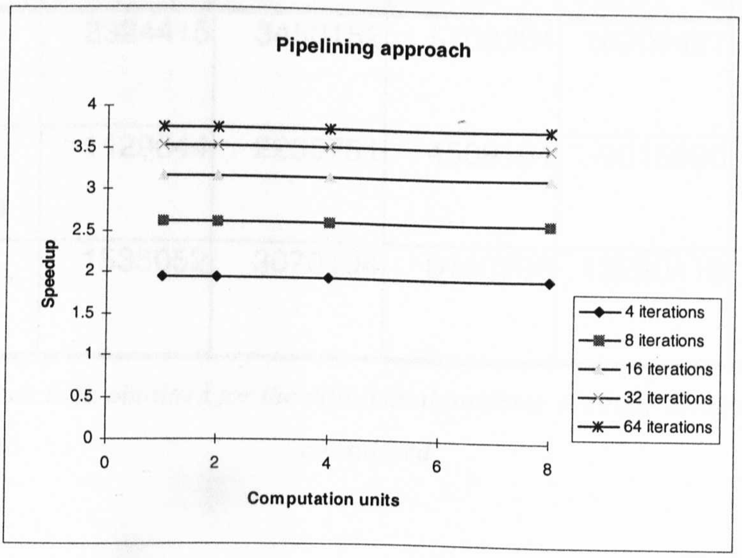
Table 8. Times obtained for the pipelining based algorithm

Computation Units	4 iterations	8 iterations	16 iterations	32 iterations	64 iterations
1	1.938643	2.612182	3.160444	3.531	3.750891
2	1.939329	2.611249	3.159288	3.529688	3.749485
4	1.939362	2.611284	3.159326	3.529726	3.749522
8	1.93977	2.611829	3.159981	3.530454	3.750052

Table 9. Speedups obtained for the pipelining based algorithm



Graph 8. Results produced by the pipelining based algorithm. Time with respect to the number of computing units.



Graph 9. Results produced by the pipelining based algorithm. Speedup with respect to the number of computing units.

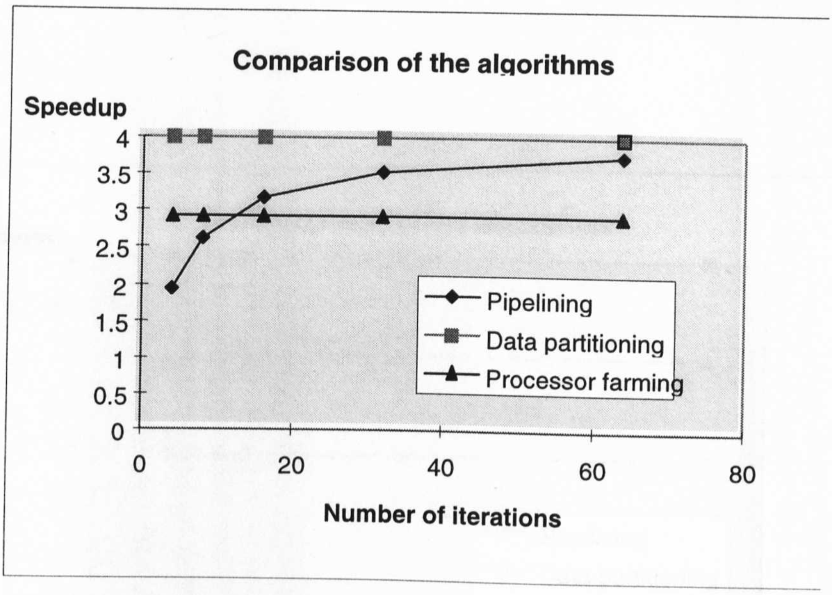
5.6.1 Comparison of the results

A brief comparison of the results obtained by the three algorithms is performed in this section. Table 10 and Graph 10 show a comparison of the times and speedups obtained when a single computation unit is to be processed. The speedups obtained for the data-partitioning based and the processor farming algorithms are close to four and three respectively and the number of iterations does not affect the results significantly. For the pipelining algorithm, the number of iterations is a major factor and the behaviour of the pipeline based algorithm improves as this number is increased, tending to equal the speedup of the partitioning approach.

If more computation units are processed, the results do not vary significantly. The pipelining algorithm is far more sensitive to an increase in the number of iterations than to an increase on the computation units (once the computation is high enough to keep the processor busy whilst transmission is performed, the system performs in a parallel fashion, computation taking place in parallel with the communication of the data through the network).

	4 iterations	8 iterations	16 iterations	32 iterations	64 iterations
Pipelining	2324415	3450152	5703264	10209487	19221935
Data-partitioning	1129044	2255751	4509164	9015990	18029646
Processor farming	1535052	3070104	6140208	12280416	24560832

Table 10. Clock ticks obtained for the different algorithms. A single computation unit is considered.

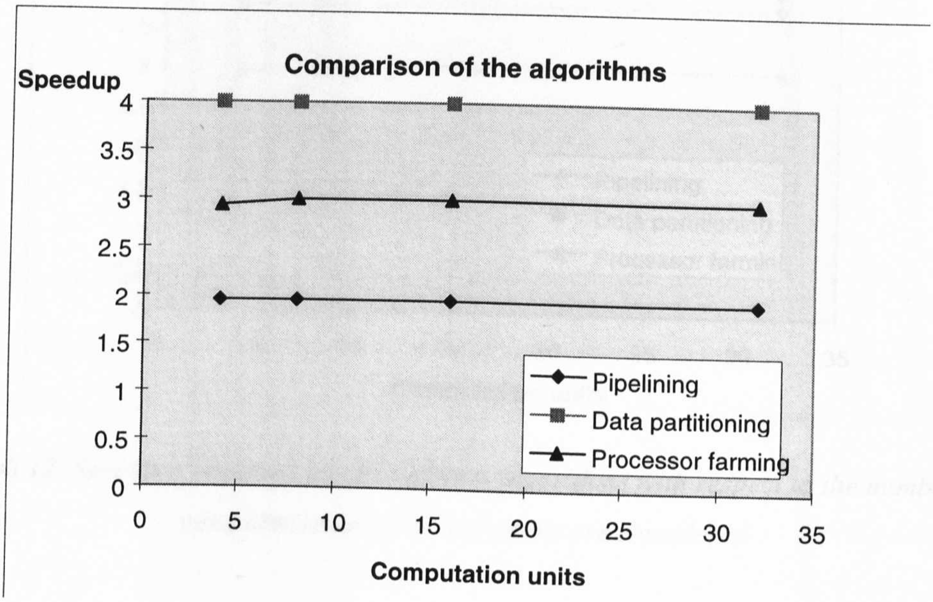


Graph 10. Speedups obtained for the different algorithms with respect to the number of iterations. A single computation unit is considered

Table 11 and Table 12 present the speedups obtained for the three algorithms for an execution of 4 and 64 iterations respectively. Graph 11 and Graph 12 are the graphs corresponding to the previously mentioned tables. It can be observed that the results are nearly constant for the partitioning and processor farming approaches. The high variation in the speedup of the pipelining algorithm supports the above mentioned comment about the sensitivity of this algorithm to the number of iterations.

	4 Computation Units	8 Computation Units	16 Computation Units	32 Computation Units
Pipelining	1.938643	1.939329	1.939362	1.93977
Data-partitioning	3.991175	3.997927	3.997927	3.999628
Processor farming	2.935544	2.992487	2.993029	2.994622

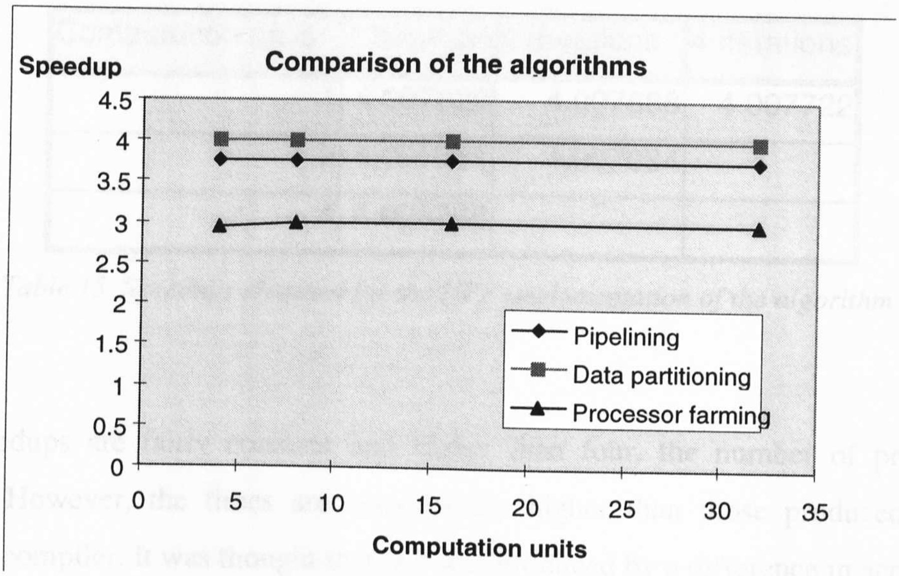
Table 11. Speedups obtained for the different algorithms. Four iterations are considered.



Graph 11. Speedups obtained for the different algorithms with respect to the number of computation units. Four iterations are considered

	4 Computation Units	8 Computation Units	16 Computation Units	32 Computation Units
Pipelining	3.750891	3.749485	3.749522	3.750052
Data-partitioning	3.998936	3.999817	3.99987	4.000603
Processor farming	2.935544	2.992487	2.993029	2.994622

Table 12. Speedups obtained for the different algorithms. 64 iterations are considered



Graph 12. Speedups obtained for the different algorithms with respect to the number of computation units. 64 iterations are considered

The same program has been implemented in HPF. The results for sequential and parallel executions are presented in Table 13 and Table 14 respectively. The speedups obtained are shown in Table 15. Note the reduced amount of results. The timing function in HPF does not allow for the measurement of large execution times.

Computation units	Clock ticks
1	1414313
2	2828176
4	5655901

Table 13. Results for the sequential execution of the HPF implementation.

Computation units	4 Iterations	8 Iterations	16 Iterations
1	1411623	2823200	5646352
2	2823195	5646352	-
4	5646350	-	-

Table 14. Time in clock ticks obtained for the parallel implementation.

Computation units	1 Iteration	2 Iterations	4 Iterations
1	4.007622	4.007688	4.007722
2	4.007057	4.007084	
4	4.006766		

Table 15. Speedups obtained for the HPF implementation of the algorithm.

The speedups are fairly constant and higher than four, the number of processors utilised. However, the times are considerably higher than those produced by the occam 2 compiler. It was thought that this was produced by a difference in accuracy in favor of the HPF compiler. However, after a precision check, it has been proven that the accuracy obtained with the occam 2 compiler is far better than the obtained with HPF, specially when the power function is used. The results are clearly favourable to the occam 2 compiler, that, being implemented in the 80's, still gives a far better performance not only in computation time but also in accuracy.

5.7 Comments and conclusions

From the speedup models developed and the results presented in the previous section, several conclusions can be extracted:

- When the data is load balanced, that is, when the same amount of computation is required with every single element of the array, the mesh configuration performs better in the normal case since less communication is undertaken. The pipelined approach, however, is very close in performance.

- If the data is not load balanced, the pipelined implementation performs better than the mesh if the iterations are load balanced (every iteration requires the same amount of computation). Load balancing algorithms could be applied to reduce the time overheads. If a processor finishes the calculations, it asks for more data to calculate from the nearest processor in the network. When data are processed, they are sent back to the original processor. However, the communication times would still be too high to counteract the load unbalance unless the difference in load between the processors is of a great magnitude.

- The farming approach produces poor results except when the data are highly load unbalanced and a large amount of computation is required, in which case, this implementation may be preferred.

Note that in the processor farming approach there are only three processors calculating the final result. It could be argued that the master processor could also be processing part of the array. However, a high amount of communication is performed on this processor, and a slight decrease on the data transmission undertaken by this processor would have important repercussions on the rest of the system (processors would have to wait idle to receive the packages). For a four processor system this is not a major problem, but if larger configurations are considered, the master processor may become the bottleneck of the system. Figure 69 illustrates a typical configuration for the processor farming approach with a large number of processors. Only processors at the bottom layer/layers are usually used for processing. The rest of the processors are usually assigned the communication task (communication is very intensive at the very top layers and it is important not to slow down the data transference. The communication process becomes less important as the last layer is approached, and some processing is possible at intermediate layers). For this reason, it has been preferred in the illustration of the algorithm to lose one of the four processors of the system.

- The inactivity of a number of processors in the system has an important effect in the resulting speedup. If a system such as that described in Figure 69 is utilised and only the bottom layer is dedicated to processing, the number of processors that are not performing any calculation is *(n-1)/3, n* being the total number of processing units in the system. This reduces the maximum efficiency achievable by a factor that approaches the value 2/3 as the number of layers increases.

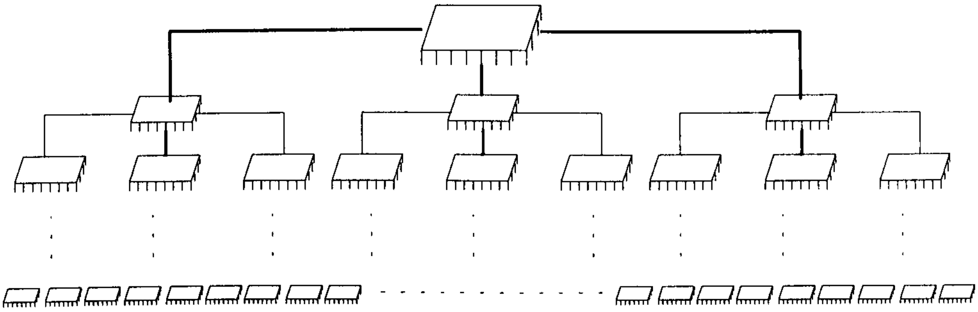


Figure 69. Typical configuration used for the processor farming approach.

- Even if the pipelined approach generally produces worse results than the mesh (specially for a low number of iterations), its implementation can be advantageous if expandability requirements are a major concern. A single processor can be added to the system in order to reduce the computation time. If a mesh of processors needed to be extended, a high number of processors could be required depending on the current size of the mesh. The flexibility of the pipelined version of the algorithm is one of its main strengths.

- The best algorithm depends upon the particular cellular automaton under analysis. The main factors to consider are load balancing, amount of calculation, number of iterations and expandability required. Therefore, different paradigms suit a different type of automaton. The optimum implementation is application dependent. There is no algorithm that is better than the others. Every algorithm presents different performance for different cases.

- In any case, the parallelisation of the algorithm in off-chip parallel distributed memory machines can be considered worthwhile in terms of efficiency. The efficiency only starts dropping if the number of processors is very high. The pipelined and mesh approaches for implementation present extremely good isoefficiency functions. If twice the number of processors are used, the efficiency is maintained by enlarging the array to twice its size. This is a linear isoefficiency function that implies a good software scalability of the algorithm. The farming approach, however, presents a poor isoefficiency function. As processors are added, the communication starts to be the most time consuming operation. This type of implementation is only preferred for highly load unbalanced cases if a low number of processors is to be used.

- The loss in accuracy and the increase of the computation time in a fairly simple program, induce the believe that the HPF compiler used for the experimentation[6] is not an alternative to expression of parallelism parallel compilers as occam 2, C or FORTRAN 90 plus. The HPF concept, however, is interesting enough to be a serious alternative to the above cited languages. It reduces the implementation effort by an important factor.

The version of HPF used during this research however, is incomplete. It does not provide some of the major facilities of the HPF standard, as a general FORALL statement including a block of instructions. It has to be stated, that not only the compiler but the technical support provided is insufficient, if any. Functions such as the power do not work and in some occasions produce a random number, rather than a result. A general error displayed as "unknown error. Please report" appears constantly when compiling code. The programmer has to be aware that some general instructions perform faster than others and no documentation is supplied. Either the lack of knowledge produced by the lack of both documentation and technical support or a bad implementation of the compiler hindered the implementation of the programs in this work. It is important to remark that the author agrees with the principle and concepts

[6] NASofware Limited

of HPF and supports the idea that a detection of parallelism type compiler is very useful to implement data parallelism based code. At this stage and after the experimentation, however, the author does not support the use of the current version of HPF for transputer systems (an HPF subset).

There is an issue that has a very important influence on the results, and that should be taken into account when determining the algorithm to be used in a particular case. In Table 5 and Table 7, that represent the speedups for the first two algorithms, some speedups are slightly greater than the number of processors in use. The same case occurs with the HPF implementation. In some bibliography this effect is referred to as super-speedup and it is produced by the use of the internal memory in the individual transputers. Transputers contain two types of memory. Internal memory, of only a few Kbytes with a fast access mechanism and external memory, with slower capabilities but a much larger capacity, of the order of Mbytes. When data is split between the processors, less data has to be processed and a larger percentage of the array can be located in internal memory, data access being, therefore, faster. This fact is very important in the farming approach since, for small packet sizes, the whole computation is performed in the internal memory. The other two approaches do not benefit as much from this fact. In the data-partitioning approach, this effect will become more important as the number of processors is increased (if a large number of processors is used and a large amount of computation is required, a high super-speedup can be easily achieved). In the pipelining approach, this effect is not usually relevant. Even if the slaves can be programmed to work on a single line located in internal memory, the master processor has to keep the whole data set on its memory and, because all the processors are working in parallel in a pipelined fashion, the master becomes the bottleneck of the process (the slaves have to be idle until the master finishes the processing of its line). Control of the internal memory is not explicitly available with occam 2. However, knowledge about the compiler allows the programmer to declare the variables so that they are likely to be in internal or external memory as desired. Nevertheless, no certain knowledge is available about where variables are allocated and, therefore, modeling of this effect is usually very complex.

5.8 Implementation of a cellular automata based unwrapper

The cellular automaton presented by Ghiglia *et al.*(1987) is load balanced, since every pixel requires the same amount of computation in every iteration. The computation required with each pixel is very low, and therefore, the communication is relatively expensive.

The processor-farming approach is completely ineffective for parallelisation of this automaton. Communicating a pixel would consume more time than performing the required computation. A parallel implementation following this approach would give a decrease in speed, i.e. the parallel algorithm would run slower than the sequential version executed in a single processor, independently of the number of processors utilised.

The data-partitioning based approach shows the best behaviour for the implementation. However, this algorithm is forced to run in a mesh network and the number of processors and extendibility of the system are not flexible. The algorithm presents a high hardware scalability, but the extendibility is limited to create a larger mesh, and a large number of processors may be required for these purposes.

The application of the pipelined approach is advantageous. It allows the addition of any number of processors to the system (with the limitations that the size of the image imposes) and, since the number of iterations is significantly large, the 'load' of the pipeline is virtually hidden. The main problem of this implementation is the initialisation of the pipeline every time an stable state is reached. This time is usually negligible due to the large number of iterations between stable states.

The use of the processor farming approach is not appropriate for the implementation of the automaton under analysis. The decision of which of the remaining two algorithms should be used should be based on extendibility interests. If no future expansion will be required, the data-partitioning approach performs better. If extendibility matters are of concern, the pipeline based approach is the most appropriate.

Chapter six

Avoiding the unwrapping

6.1 Introduction

The last three chapters have been completely dedicated to the unwrapping problem. Chapter 3 introduced the problem and gave a literature review of previous attempts at phase unwrapping. Chapter 4 described two novel techniques presenting a satisfactory behaviour for a number of images. A correct solution is, however, not guaranteed even with no time constraints. It is believed that no unwrapper has been implemented yet that is able to cope with all the different problems that can appear in a wrapped phase distribution. The present chapter describes a technique that avoids the phase unwrapping process, or at least reduces the complexity. A second image is captured. The new set of data is utilised so that the number of wraps is largely reduced. This reduction can eliminate the wraps under certain conditions.

Fractional interferometry (Bourdet and Orszag 1979; Tilford 1977) and two-wavelength interferometry (Polhemus 1973; Yokozeky and Suzuki 1972) were initially proposed to achieve larger beam wavelengths from two different lasers with a much shorter wavelength.

Wraps are formed because of the existence of several fringes on the fringe pattern. Hence, it could be argued that if a projected fringe pattern is created by interferometric techniques, the utilisation of a large wavelength, that would usually translate into a low spatial frequency[7] in the fringe pattern created, would leave the final result with no wraps. Nevertheless, the projection of a single fringe onto an object leads to very poor results. This effect is now discussed:

- First, the digitisation process would only allow a certain set of values to be stored that is determined by the different intensities that can be detected and represented by the system. Therefore, if a single fringe is available, all the different heights in the object have to be represented by this set of values (typically 256 grey scales), producing an inaccurate result.

- If the FFA technique is utilised, the signal peak will lie very near to the centre, at just one pixel. The cross-talking with the symmetric signal peaks will be very large. In addition, the d.c. peak will be concentrated very near to the signal peak and its elimination would not be possible.

However, it is possible to notice certain similarities between achieving an increase in the wavelength and eliminating the wraps in a phase distribution. It will be these similarities that compose the basis of this technique.

[7] The term spatial frequency is used here to refer to the number of fringes that are present in the image.

6.2 Background

The technique is explained for the FFA technique. It is important to remark that the experimentation has been undertaken working with FFA exclusively. Similar work for phase stepping has been developed in (Cheng and Wyant 1984) and (Creath et al. 1985).

There are, however, advantages in flexibility that are inherent to the FFA technique. This flexibility is related to the ease of varying the frequencies and the rotational angle of the fringe patterns.

Related work using FFA has also been published. In (Zhao et al. 1994), two images with different sensitivities are utilised for unwrapping. By using a reduced sensitivity, it is possible to achieve a phase distribution that has a range of less than 2π. This phase distribution is utilised as an aid to unwrapping a wrapped phase distribution with a higher sensitivity. The main problem associated with the technique is the achievement of a phase distribution with a range of less than 2π. In some occasions, this would imply the projection of a very low number of fringes. The signal peak is then very close to the d.c. peak and its isolation is not possible.

It was showed in chapter 2 that a fringe pattern can be mathematically represented by an expression such as

$$I(x,y) = a(x,y) + b(x,y)\cos(2\pi f_x x + 2\pi f_y y + \phi(x,y))$$

where,

$a(x,y)$ comprises the background variations,

$b(x,y)$ describes variations is the fringe visibility,

$\phi(x,y)$ is the phase of the object,

f_x is the carrier frequency on the x-axis and

f_y is the carrier frequency on the y-axis.

The unwrapped phase of the object $\phi(x,y)$ relates to the relative heights[8] of the object by the expression

$$h(x,y) = c\phi(x,y) \tag{6.1}$$

where c is a constant that depends on both the camera angle and the frequency of the projected fringe pattern. The value of this constant is given by:

$$c = \frac{1}{2\pi f_0 \sin\theta} \tag{6.2}$$

where f_0 is the spatial frequency of the projected fringe pattern ($\frac{1}{p_0}$ in Figure 70) and θ is defined as the view to projection angle. Figure 69 illustrates the parameters. $\bar{i}$ is the direction of the illumination and $\bar{v}$ is determined by the position of the camera.

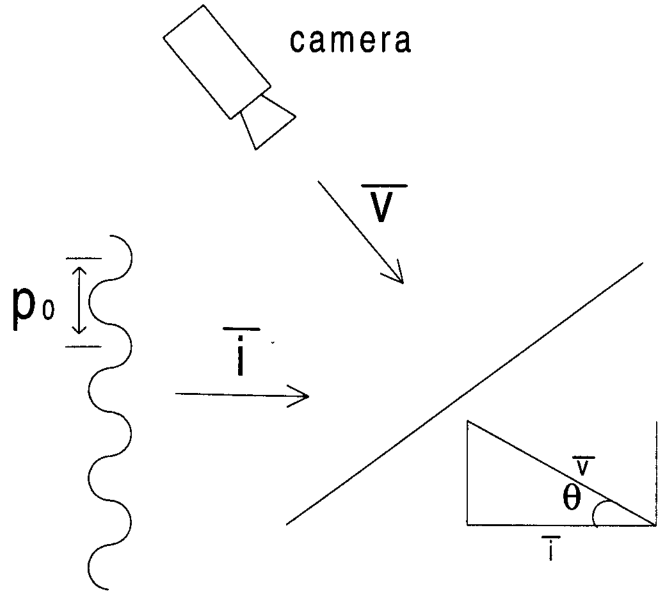


Figure 70. Illustration of the parameter θ.

[8] Absolute heights are obtained by adding a constant to the relative heights.

Because the phase is finally obtained by means of the *arctan* function, it will be bounded in the range [$-\pi$, π] as it was explained previously in this thesis. This means that a value of unwrapped phase that differs by 2π from another value in the same fringe pattern will be represented by the same intensity. By using equation (6.1), it is easily derived that an increment in height of an integer multiple of $\frac{1}{f_0 \sin\theta}$ would not be detected unless additional information is available[9]. When unwrapping a wrapped phase distribution, it is normally assumed that the surface is continuous or smooth. However this assumption is often erroneous and leads to errors in the reconstructed object's surface.

Figure 70 represents a real discontinuity in a signal where the spatial frequency of the projected fringe pattern is f_o and the camera to projection angle is θ. If k is an integer number, this step will not be detected and the result obtained will be a plane surface.

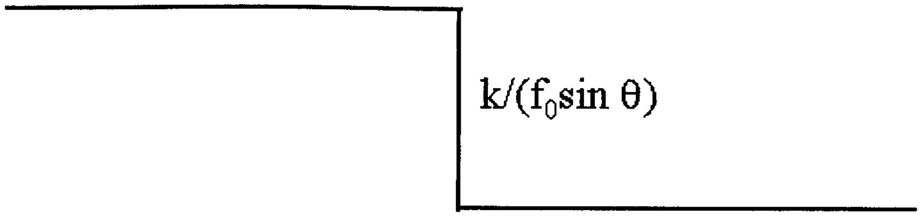


Figure 71. Real discontinuity.

Therefore, it would be advantageous to have a second image to complement the information provided by the first fringe pattern. With a single fringe pattern it is not possible to detect any discontinuity on an object's surface if the height of the discontinuity is a multiple of a constant defined by $\frac{1}{f_0 \sin\theta}$. If two fringe patterns are available with different values for this constant, the problem is partially overcome. To

[9] Note that such an increment in height corresponds to a 2π increment in phase.

achieve a different value for this constant one of the parameters, either the spatial frequency of the projected fringe pattern f_0 or the angle of the camera with respect to the fringe projection system θ, is required to change.

However, a variation of the camera angle implies a change in the view of the object. Hence, the heights represented by the two images would be different and no comparison would be feasible. Therefore, the remaining parameter, the frequency of the projected fringe pattern f_o, is the only option left for our purposes.

If the real height on the x-axis and the obtained wrapped height (the height obtained when phase unwrapping is not carried out) on the y-axis are represented, the resultant graph will be as shown in Figure 72. Note that this is not the typical graph representing the wrapped phase with respect to the continuous phase.

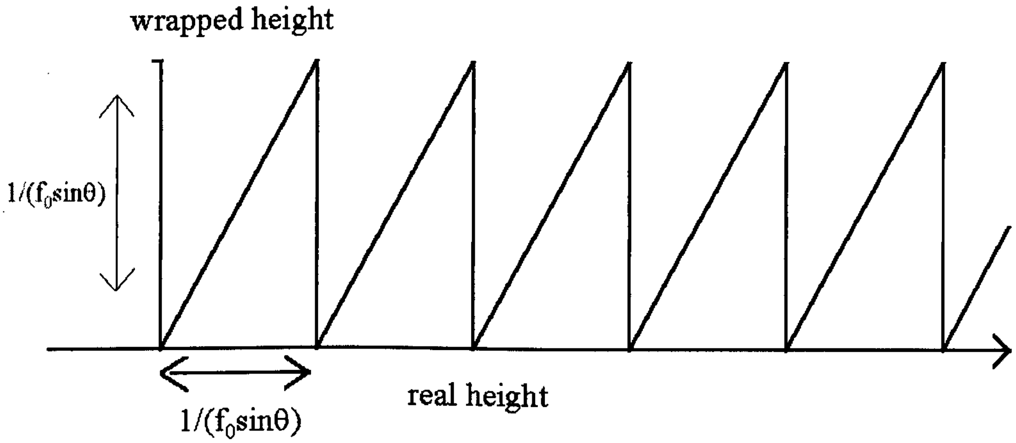


Figure 72. Wrapped height distribution: Observed vs. real.

This is a periodic function with a period given by $\frac{1}{f_0 \sin\theta}$. It will be of major interest to increase this period so that there are not two points in the real height that correspond to the same wrapped height value. That is, it is a major concern that a wrapped height value corresponds to an unique real height value.

In order to satisfy this condition, a second image is to be captured. This process is carried out with the camera in the same position so that the angle is held and, therefore, the view of the object is not changed. It will be the spatial frequency of the projected fringe pattern that is modified in order to obtain a different constant relating the obtained phase distribution to the relative height of the object.

Let us denote the spatial frequency of the projected fringe pattern on the second image by f_1. If the functions representing the wrapped height value with respect to the actual height value for both images are represented, then plots similar to those of Figure 73 are obtained.

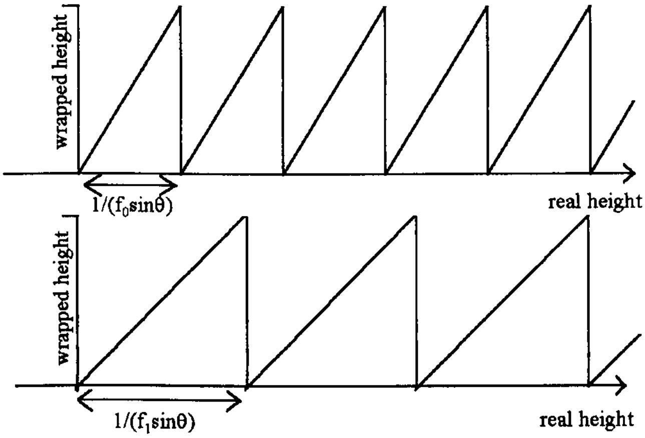


Figure 73. Two fringe patterns with different frequencies. Real height vs. wrapped height.

Two different measurements are then available, one on each image, for every point on the object's surface. The relative height is related to the wrapped height by an expression of the form

$$h = h_w + \frac{k}{f_0 \sin\theta}$$

where k is an integer value, h is the relative height and h_w is the wrapped height. Note the use of the term wrapped height, unusual in fringe analysis. The wrapped height is referred to as the value obtained by application of Equation (6.1) to the wrapped phase.

As two images with different spatial frequencies are available, two different equations are obtained,

$$h = h_{w1} + \frac{k_1}{f_1 \sin\theta} \qquad\qquad h = h_{w2} + \frac{k_2}{f_2 \sin\theta} \tag{6.3}$$

where f_1 and f_2 represent the spatial frequencies of the two fringe patterns and h_{w1} and h_{w2} represent the wrapped height values on the two images.

Bringing these two equations together the following expression is obtained,

$$h_{w1} + \frac{k_1}{f_1 \sin\theta} = h_{w2} + \frac{k_2}{f_2 \sin\theta} \tag{6.4}$$

This is a single equation containing two unknowns variables. It will have, therefore, infinite solutions. However, the solutions of interest are only those in which both k_1 and k_2 take integer values.

Unfortunately, there are still infinitely many solutions to Equation (6.4), being periodic in the sense that the solutions will have a form:

k_1	a	$a + p_1$	$a + 2\,p_1$	$a + 3\,p_1$	. . .	$a + n\,p_1$
k_2	b	$b + p_2$	$b + 2\,p_2$	$b + 3\,p_2$	. . .	$b + n\,p_2$

where a, b, p_1 and p_2 are integer values.

Therefore, the achievement of mapping the wrapped height values to an unique real height value has not been achieved. However, the period of the function has increased. This implies an increase in the minimum distance between two wrapped heights that leads to the same relative height. The period will now be

$$\frac{1}{(f_2 - f_1)\sin\theta} \tag{6.5}$$

By looking at Expression (6.5), the reader could think of decreasing the value of the difference between the frequencies, therefore increasing the period. However, as the value of the period is increased, the influence of the noise also increases. Therefore, a balance has to be reached where the influence of the noise does not significantly affect the results.

This technique can also be used with a greater number of images. The period can be so increased to a greater value and, therefore, the results potentially improved.

Finally, it is worthwhile to remark on the fact that because the value of the height can be calculated by any expression of the type given in (6.3), there is a replication of the data, a fact that can be used to reduce the noise by simply taking the mean of both results.

This procedure has similarities with the Fractional Fringe Analysis technique in the sense that we are obtaining a synthetic fringe. Therefore, the theory developed for Fractional Fringe Analysis will clearly be useful for the calculation of the synthetic phase distribution from the two original phase distributions.

6.3 Technique

As was outlined in the previous section, this technique has clear links to Fractional Fringe Analysis concepts. Therefore, the theory underlying Fractional Fringe Analysis can be used as a basis for our work.

The equations in (6.3) can be rewritten as

$$hf_1 \sin\theta = h_{w1} f_1 \sin\theta + k_1$$

$$hf_2 \sin\theta = h_{w2} f_2 \sin\theta + k_2$$

If these two equations are subtracted from each other, the result can be expressed as

$$h(f_2 - f_1)\sin\theta = h_{w2} f_2 \sin\theta - h_{w1} f_1 \sin\theta + k_2 - k_1$$

Since k_1 and k_2 are integers, this equation becomes

$$h(f_2 - f_1)\sin\theta = h_{w2} f_2 \sin\theta - h_{w1} f_1 \sin\theta + I,$$

where I is an integer number. The relative height h can be easily derived from the previous equation,

$$h = \frac{h_{w2} f_2 \sin\theta - h_{w1} f_1 \sin\theta + I}{(f_2 - f_1)\sin\theta} \tag{6.6}$$

The only parameter that is left to determine is I, which is known to be an integer value. It is also known that the value of the period is given by the expression

$$\frac{1}{(f_2 - f_1)\sin\theta}$$

Since the value of the new height should be within a single period, a further constraint is imposed on Equation (6.6). This condition is given by the equation

$$-\frac{1}{2(f_2-f_1)\sin\theta}<h<\frac{1}{2(f_2-f_1)\sin\theta}$$

(6.7)

that combined with Equation (6.6) produces

$$-\frac{1}{2(f_2-f_1)\sin\theta}<\frac{h_{w2}f_2\sin\theta-h_{w1}f_1\sin\theta+I}{(f_2-f_1)\sin\theta}<\frac{1}{2(f_2-f_1)\sin\theta},$$

that can be simplified to

$$-\frac{1}{2}<h_{w2}f_2\sin\theta-h_{w1}f_1\sin\theta+I<\frac{1}{2}$$

(6.8)

By translating Equation (6.6) into Fractional Fringe terms, we could state that the fractional part of the new height obtained is the subtraction of the two fractional parts of the available original heights (plus the addition of an integer). This can be written as

$$F(h)=F(h_2)-F(h_1)+I$$

(6.9)

where the function *F* denotes the fractional part.

This conversion in notation has been introduced in order to analyse the errors as a fraction of a fringe, that is as they are normally given in fringe analysis.

The error in the new fractional part obtained can be easily derived from equation (6.9) being,

$$\Delta F(h) \leq \Delta F(h_2) + \Delta F(h_1),$$

If the errors in both height distributions are assumed to be the same, this equation can be written as

$$\Delta F(h) \leq 2\Delta F(h_1),$$

this can be enunciated as: the maximum error in the fractional part of the height distribution is double that of the greater of the errors in the fractional part of the two original height distributions.

This last equation can be misleading since the new period can be much larger than the previous two periods. This fact makes the maximum error in the new fringe pattern much larger if heights are considered instead of fractional parts. This effect can be illustrated with a simple example: suppose that the two original height distributions contained an error of one tenth of a fringe. The error in the resulting height distribution will be of one fifth of a fringe. However, the fringe is now much larger. Therefore, this is a much larger error in absolute terms.

Nevertheless, even if the maximum error has increased, in practice the component errors normally tend to compensate each other giving an acceptable result.

It is important to remark at this stage that the resulting fringe pattern may only be utilised to detect the phase wraps. The final data would be obtained from either of the two original height distributions. Hence, as long as the maximum error does not exceed the size of half the lowest of the original periods, the results would be completely reliable.

6.4 Further development

The previous method implies the calculation of two different heights distributions, the calculation of their respective fractional parts and, finally the determination of the integer *I*. However, this process can be largely simplified by operating with the wrapped phase distributions instead of the wrapped height distributions.

By using Equations (6.1) and (6.2), equation (6.6) can be rewritten as

$$h = \frac{I + \dfrac{\phi_{w2} f_2 \sin\theta}{2\pi f_2 \sin\theta} - \dfrac{\phi_{w1} f_1 \sin\theta}{2\pi f_1 \sin\theta}}{(f_2 - f_1)\sin\theta}$$

where ϕ_{wl} represents the first wrapped phase distribution and ϕ_{w2} represents the second wrapped phase distribution.

This equation can be further simplified to

$$h = \frac{I + \dfrac{\phi_{w2} - \phi_{w1}}{2\pi}}{(f_2 - f_1)\sin\theta}$$

which in turn can be re-written as

$$h = \frac{2\pi I + \phi_{w2} - \phi_{w1}}{2\pi(f_2 - f_1)\sin\theta}$$

(6.10)

and the condition stated in Equation (6.7) can now be written as

$$-\frac{1}{2(f_2-f_1)\sin\theta}<\frac{2\pi I+\phi_{w2}-\phi_{w1}}{2\pi(f_2-f_1)\sin\theta}<\frac{1}{2(f_2-f_1)\sin\theta},$$

that is simplified to

$$-\frac{1}{2}<\frac{2\pi I+\phi_{w2}-\phi_{w1}}{2\pi}<\frac{1}{2},$$

from where

$$-\frac{1}{2}<I+\frac{\phi_{w2}-\phi_{w1}}{2\pi}<\frac{1}{2}$$

and so finally

$$-\pi<2\pi I+\phi_{w2}-\phi_{w1}<\pi \tag{6.11}$$

This condition implies that if the phases of the two original fringe patterns are subtracted, the resulting phase has to be re-wrapped modulo 2π before being converted into heights.

This effect is easily understood by examining Figure 74, where the *x*-axis represents the real heights of the object and the *y*-axis is the phase obtained in the wrapped phase distribution. The subtraction of the first two functions result in a signal with a much larger period, obtaining a decrease in the number of wraps in the wrapped phase distribution.

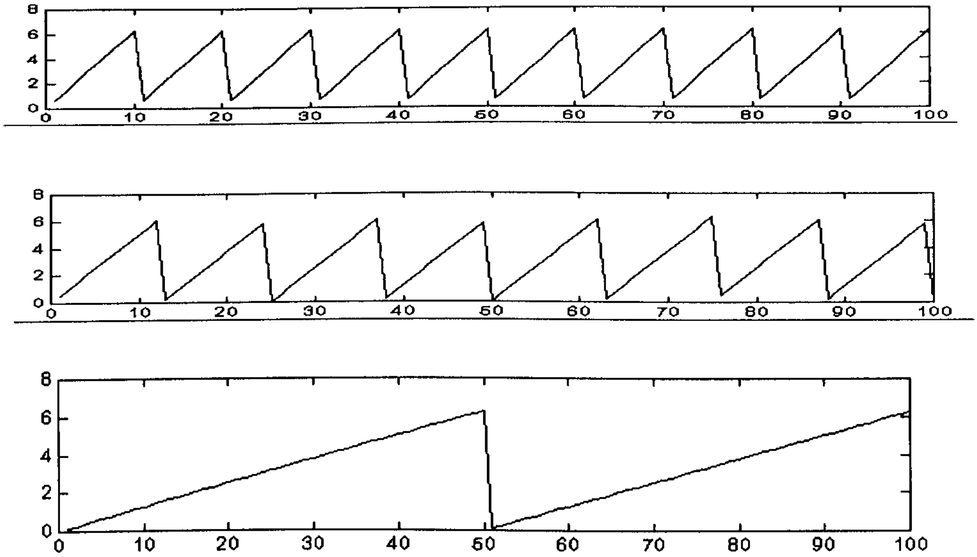


Figure 74. A wrapped phase distribution (top), a wrapped phase distribution with a different frequency (middle) and the subtraction of the previous two signals re-wrapped (bottom). The x-axis represents real heights. The y-axis, the wrapped phase obtained.

Once I has been determined, and the subtraction of the two wrapped phase distributions performed (that is equivalent to subtracting the phases and re-wrapping the result modulo 2π), the term $2\pi I sin\theta + \phi_{w2} - \phi_{w1}$ in Equation (6.11) has been obtained, and the process to obtain the heights is the multiplication by the constant c defined in Equation (6.2), where the frequency f_0 has now changed its value to $f_2 - f_1$. The reader can compare this set of results to equation (6.10).

This process is illustrated in Figure 75. Two fringe patterns with a different spatial frequency are captured. The phase of the two images obtained is calculated independently following the same procedure as in conventional FFA. After carrier removal, the two wrapped phase distributions are subtracted to give the final unwrapped phase distribution.

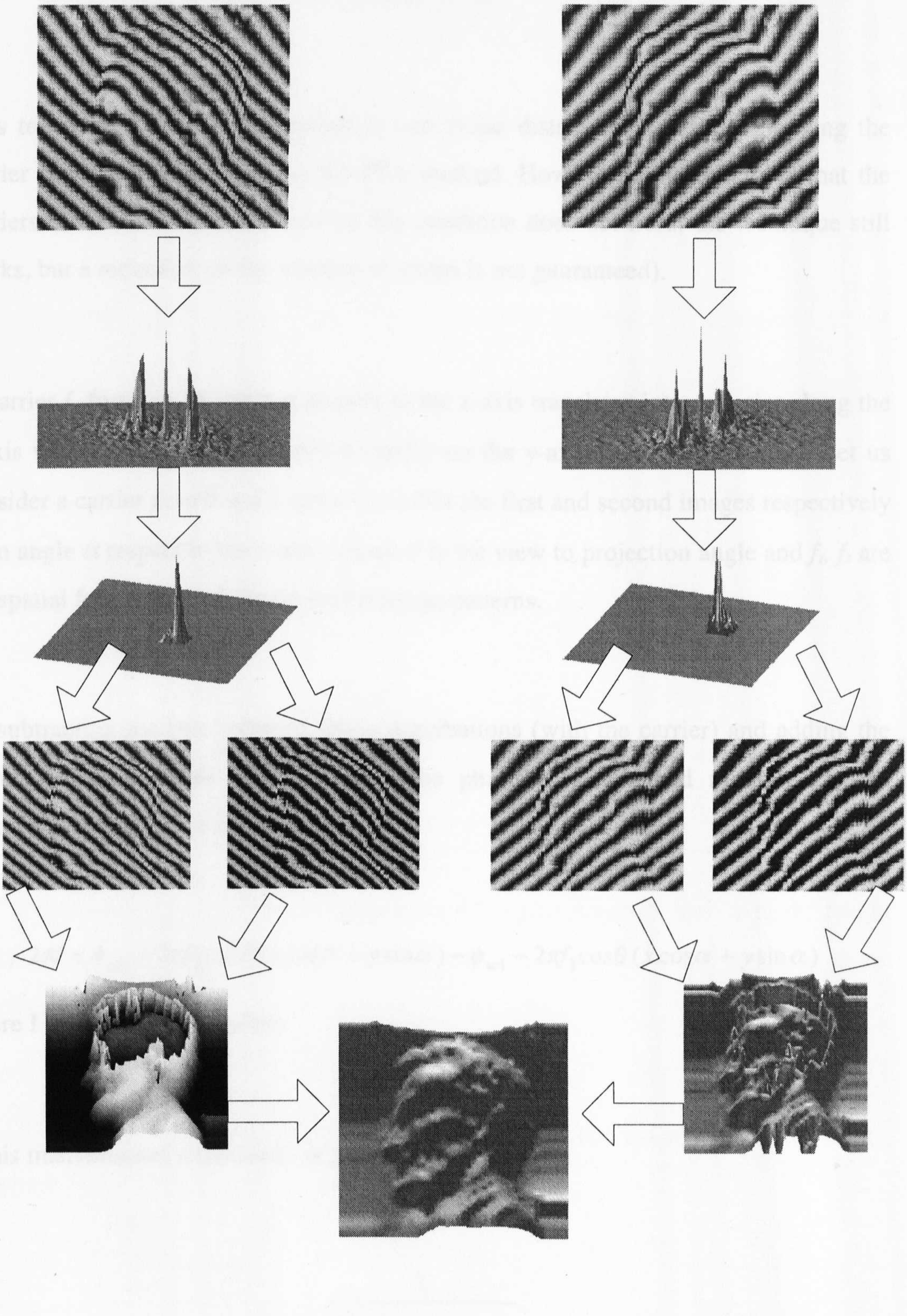


Figure 75. Use of two fringe patterns to eliminate the wraps.

6.5 Introduction of a carrier

This technique can also be applied to two phase distributions before removing the carrier frequency introduced by the FFA method. However, it is preferable that the carriers are in the same direction (if this condition does not hold, the technique still works, but a reduction on the number of wraps is not guaranteed).

A carrier f_c forming an angle α respect to the x-axis translates into a carrier along the x-axis with a value of $f_c cos\alpha$ and a carrier on the y-axis that will be $f_c sin\alpha$. Let us consider a carrier $f_1 cos\theta$ and a carrier $f_2 cos\theta$ in the first and second images respectively at an angle α respect to the x-axis, where θ is the view to projection angle and f_1, f_2 are the spatial frequencies of the projected fringe patterns.

By subtracting the two wrapped phase distributions (with the carrier) and adding the corresponding multiple of 2π so that the phase is re-wrapped modulo 2π, the following expression is obtained

$$2\pi I + \phi_{w2} + 2\pi f_2 \cos\theta(x\cos\alpha + y\sin\alpha) - \phi_{w1} - 2\pi f_1 \cos\theta(x\cos\alpha + y\sin\alpha)$$

where I takes an integer value.

If this mathematical expression is multiplied by the term

$$\frac{1}{2\pi(f_2 - f_1)\sin\theta}$$

the resulting expression is given by

$$\frac{I}{(f_2-f_1)\sin\theta}+\frac{\phi_{w2}-\phi_{w1}}{2\pi(f_2-f_1)\sin\theta}+\frac{f_2\cos\theta(x\cos\alpha+y\sin\alpha)\ -f_1\cos\theta(x\cos\alpha+y\sin\alpha)}{(f_2-f_1)\sin\theta}$$

Using Equations (6.1) and (6.2) this can also be expressed as a function of the heights as

$$\frac{I}{(f_2-f_1)\sin\theta}+\frac{f_2h_{w2}-f_1h_{w1}}{(f_2-f_1)}+\frac{(x\cos\alpha+y\sin\alpha)\cos\theta}{\sin\theta},$$

that can be simplified to

$$\frac{I+f_2h_{w2}\sin\theta-f_1h_{w1}\sin\theta}{(f_2-f_1)\sin\theta}+(x\cos\alpha+y\sin\alpha)\cot\theta\ ,$$

that is the expression outlined in Equation (6.6) plus a spatial carrier in the height distributions that is introduced by the presence of a carrier in the original phase distributions.

Because the period of the heights distribution has increased, less wraps will appear on the phase distribution even in the presence of a carrier.

6.6 Results

The technique described in this chapter has been successfully applied to a number of fringe patterns. The images in the example given were obtained by projecting an interferometric fringe pattern onto a simple object. It should be noted that the images had a small speckle effect, produced by the coherence of the laser and the reflectivity of the surface. This fact increases the errors in both the original phase distributions and hence the resulting phase distribution. However, and despite the speckle nature of the images, the results obtained are satisfactory.

The technique is applied to two fringe patterns of the same object but with different carrier frequencies. The technique will increases the phase to height conversion factor therefore reducing the number of wraps with respect to the original images. Figure 76(a) and (b) are the original fringe patterns captured. Figure 76(c) and (d) illustrates the wrapped phase distribution corresponding to Figure 76(a) and (b). Finally, Figure 76(e) represents the final phase distribution obtained after the subtraction of Figure 76(c) and (d). From the result, it can be observed that the errors introduced are not very high.

If the reduction factor (the ratio between the least of the number of wraps in the wrapped phase distribution of the original image and the number of wraps in the resulting phase distribution) is very high, the errors obtained can also be very large. In these cases, the use of this technique as explained in this chapter could lead to unacceptable results. However, it could be advantageous to use the resulting image as a reference for the unwrapping. In this case, the definitive height data is obtained by the analysis of one of the original fringe patterns and the subtraction is used as merely additional data that is useful for the unwrapping process.

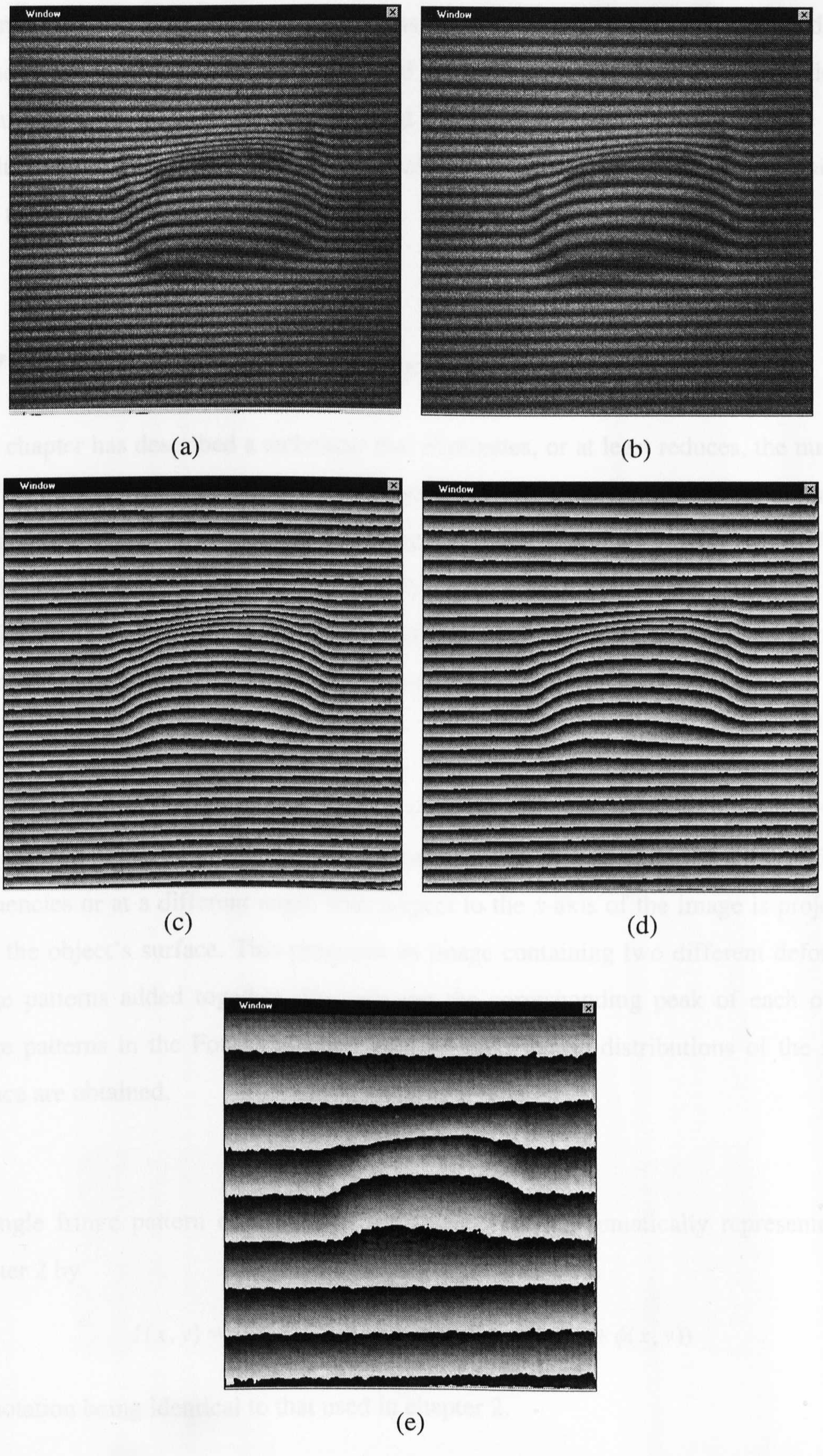


Figure 76. Wraps reduction. (a) and (b) original fringe pattern, (c) and (d) wrapped phase distribution, (e) final result.

An important advantage of this technique is that every point is analysed independently of the others. Therefore, errors are limited to particular points and no propagation is produced. A process can be implemented to search for inconsistencies in the final result. This process would detect those isolated points whose values are inconsistent with their neighbourhoods.

6.7 Multichannel Fourier Transform

This chapter has described a technique that eliminates, or at least reduces, the number of wraps in the resulting phase distribution. The main disadvantage of the method if compared to conventional FFA is the need to capture two images with two different numbers of fringes projected. The use of the multichannel Fourier transform has been previously described by Takeda (1993) and Burton *et al.*(1994). Its application to the technique presented in this chapter eliminates the need for the second image.

The multichannel Fourier fringe analysis (MFFA) principle is based upon a straight forward principle. A single pattern containing two fringe patterns with different frequencies or at a different angle with respect to the x-axis of the image is projected onto the object's surface. This produces an image containing two different deformed fringe patterns added together. By isolating the corresponding peak of each of the fringe patterns in the Fourier domain, two different phase distributions of the same surface are obtained.

A single fringe pattern deformed by the object was mathematically represented in chapter 2 by

$$I(x,y) = a(x,y) + b(x,y)\cos(2\pi f_x x + 2\pi f_y y + \phi(x,y))$$

the notation being identical to that used in chapter 2.

If two of these fringe patterns are added together, the intensity obtained is given by

$$I(x,y) = a_1(x,y) + b_1(x,y)\cos(2\pi f_{x1}x + 2\pi f_{y1}y + \phi_1(x,y)) +$$
$$+ a_2(x,y) + b_2(x,y)\cos(2\pi f_{x2}x + 2\pi f_{y2}y + \phi_2(x,y))$$

where subscripts denote the corresponding fringe pattern. This equation can be rewritten using complex number notation as

$$I(x,y) = a_1(x,y) + c_1(x,y)e^{i(2\pi f_{x1}x + 2\pi f_{y1}y)} + c_1^*(x,y)e^{-i(2\pi f_{x1}x + 2\pi f_{y1}y)} +$$
$$a_2(x,y) + c_2(x,y)e^{i(2\pi f_{x2}x + 2\pi f_{y2}y)} + c_2^*(x,y)e^{-i(2\pi f_{x2}x + 2\pi f_{y2}y)}$$

where

$$c_1(x,y) = \frac{b_1(x,y)e^{i\phi_1(x,y)}}{2} \text{ and } c_2(x,y) = \frac{b_2(x,y)e^{i\phi_2(x,y)}}{2}$$

If the Fourier transform of a single fringe pattern projected onto an object at a certain angle respect to the *x*-axis is performed, three different peaks are obtained, as it was explained in chapter two. The position of the peaks depends upon the frequency of the fringe pattern and the angle of the projection system with respect to the camera. An example is illustrated in Figure 77.

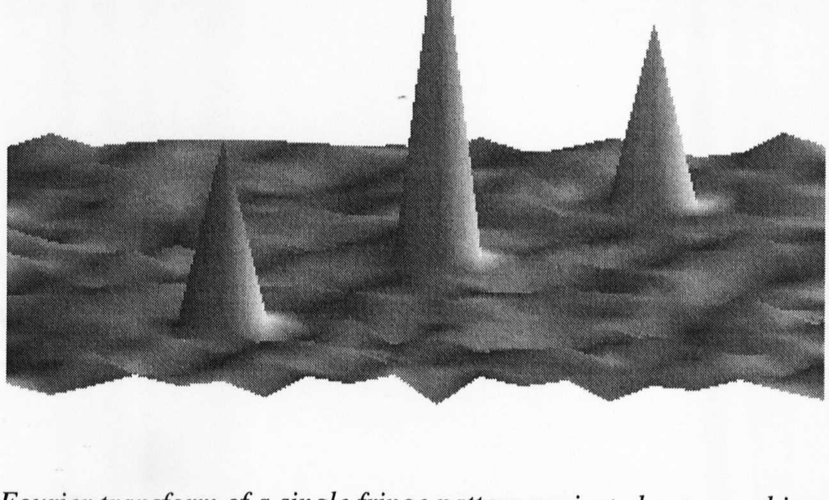


Figure 77. Fourier transform of a single fringe pattern projected onto an object at an angle π/4 with respect to the x-axis.

If a pattern composed of two fringe patterns at a determinate mutual angle is projected onto an object, a power spectrum similar to that illustrated in Figure 78 is obtained. If two of the peaks, one for each of the fringe patterns projected, are isolated, two different wrapped phase distributions can be obtained by the same procedure as explained in chapter 2.

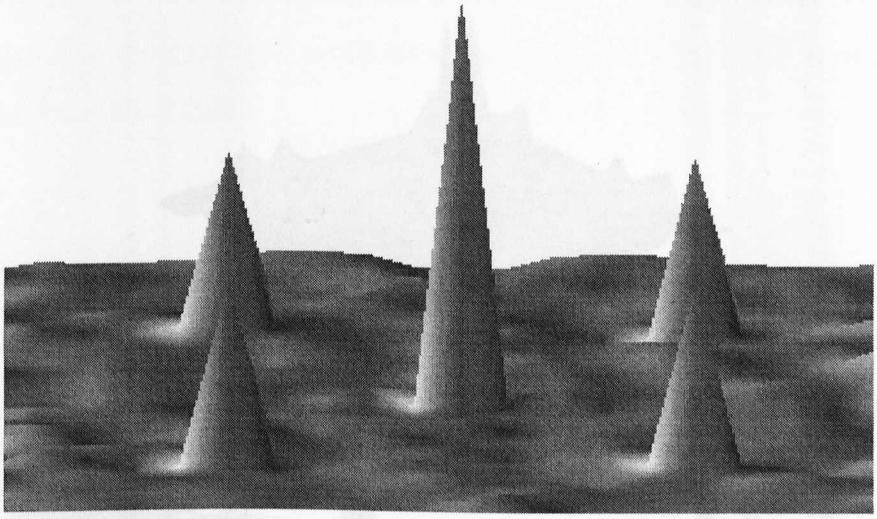


Figure 78. Fourier transform of a pattern composed of two fringe patterns projected onto an object at an angle π/4 and -π/4 respectively with the x-axis.

The angle between the two fringe patterns is particularly relevant. It is of major importance to keep the two signal peaks as far as possible from each other, so that cross-talking (interferences between the two peaks) be reduced to a minimum.

Figure 79 illustrates a series of different Fourier spectra that can be obtained by the MFFA technique. Note that if the number of fringes in the fringe patterns is the same, an exact angle of $\pi/2$ would not be the best since the leakage produced by the high frequencies at the border of the image would interfere each other at the other peak (Figure 79(a)). Different frequencies achieve a better separation of the peaks (Figure 79(b)). If the two fringe patterns are orientated in the same direction (angle 0), the leakage has a major influence and the results are very unreliable (Figure 79(c)).

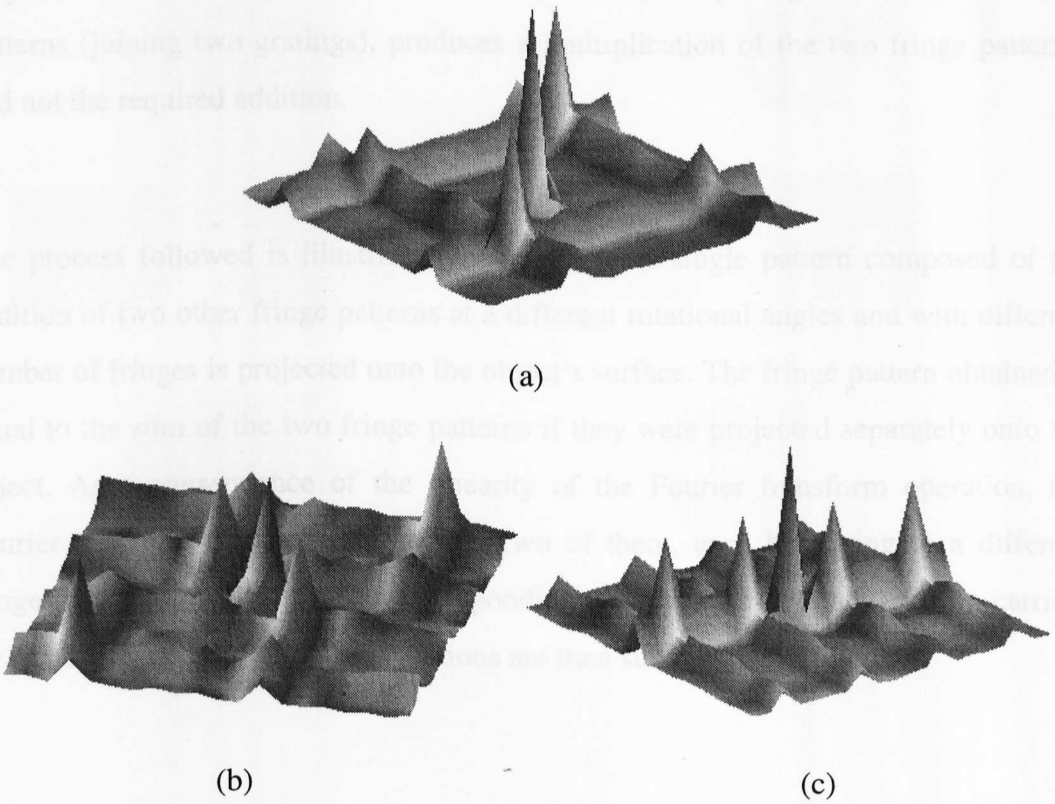


Figure 79. Examples of power spectra obtained with the MFFA technique, depending on the orientation of the fringes. (a) same frequency and an angle of $\pi/2$ rad.(b) different frequencies and (c) different frequencies but in the same orientation.

Small angles with respect to the *x* or *y* axis of the image should also be avoided, since the signal peaks would suffer from the leakage of the d.c. peak.

The grating based projection system described in chapter 2 and whose set-up is illustrated in Figure 4 has been utilised for the experimentation. A special grating has been installed so that a pattern composed by the addition of two fringe patterns is projected onto the object's surface. Note that superimposing two different fringe patterns (joining two gratings), produces a multiplication of the two fringe patterns, and not the required addition.

The process followed is illustrated in Figure 80. A single pattern composed of the addition of two other fringe patterns at a different rotational angles and with different number of fringes is projected onto the object's surface. The fringe pattern obtained is equal to the sum of the two fringe patterns if they were projected separately onto the object. As a consequence of the linearity of the Fourier transform operation, the Fourier spectrum contains four peaks. Two of them, each belonging to a different fringe pattern, are filtered. The corresponding phases are calculated and the carriers are removed. The two phase distributions are then subtracted.

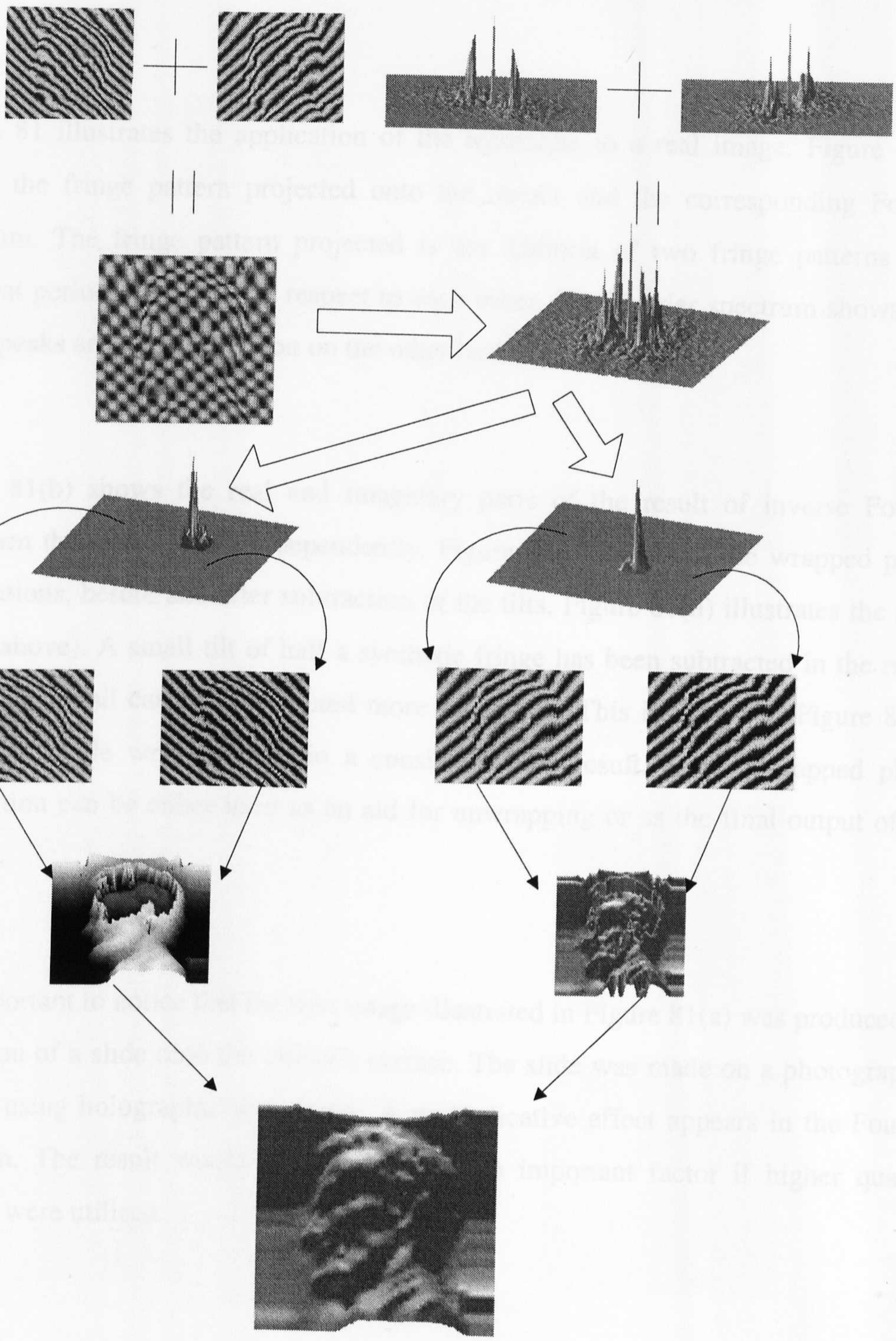


Figure 80. Use of Multichannel Fourier Fringe Analysis to reduce the number of images required by the technique.

Figure 81 illustrates the application of the technique to a real image. Figure 81(a) shows the fringe pattern projected onto the object and the corresponding Fourier spectrum. The fringe pattern projected is the addition of two fringe patterns with different periods rotated with respect to each other. The Fourier spectrum shows two signal peaks and their reflection on the other half of the spectrum.

Figure 81(b) shows the real and imaginary parts of the result of inverse Fourier transform the signal peaks independently. Figure 81(c) represent the wrapped phase distributions, before and after subtraction of the tilts. Figure 81(d) illustrates the final result (above). A small tilt of half a synthetic fringe has been subtracted in the result so that the detail can be appreciated more accurately. This is shown in Figure 81(d) below. A single wrap appears in a consistent final result. This unwrapped phase distribution can be either used as an aid for unwrapping or as the final output of the process.

It is important to notice that the first image illustrated in Figure 81(a) was produced by projection of a slide onto the object's surface. The slide was made on a photographic film by using holographic techniques. A multiplicative effect appears in the Fourier spectrum. The result would be improved by an important factor if higher quality gratings were utilised.

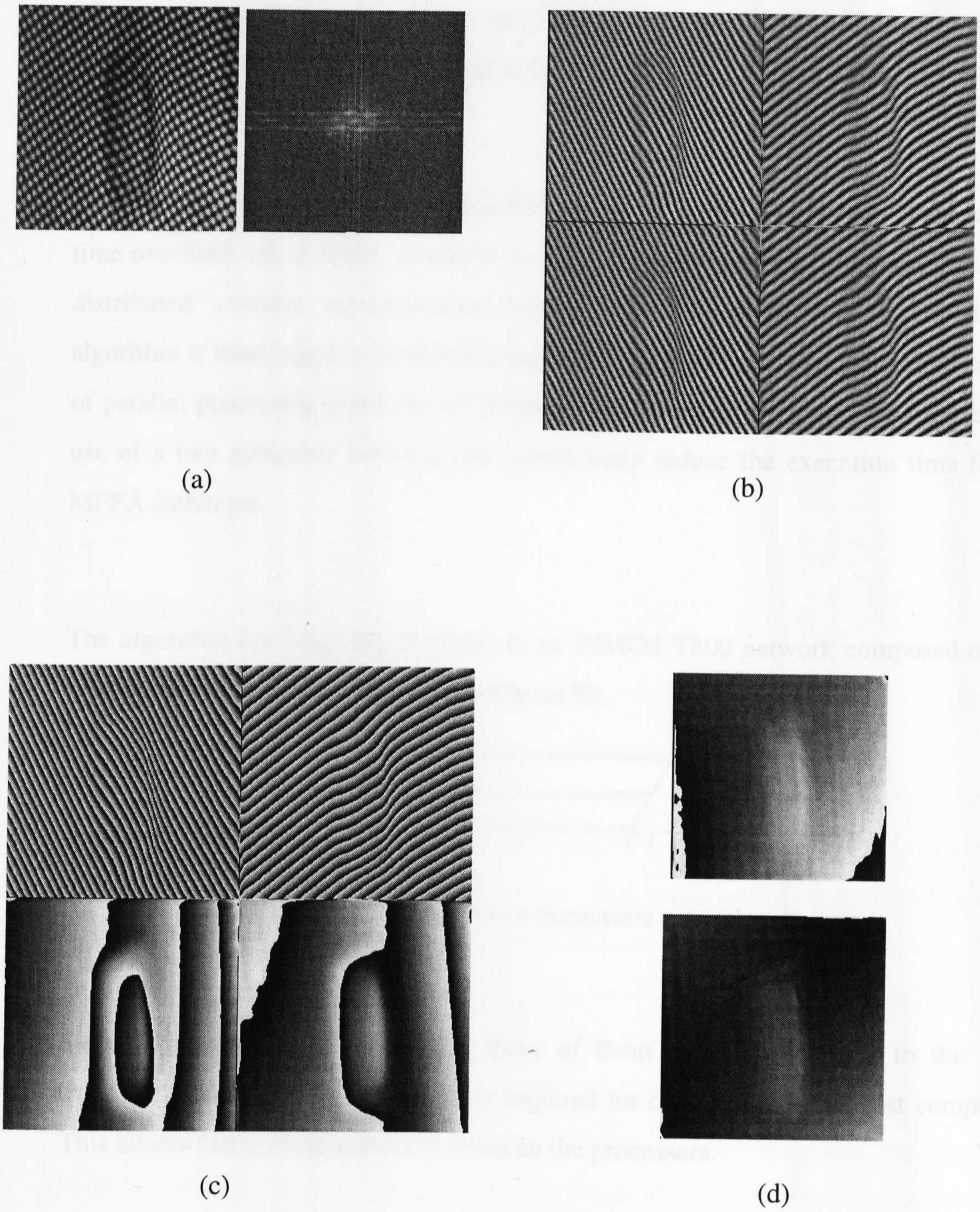


Figure 81. The multichannel technique used to reduce the number of wraps. (a) Original image and Fourier spectrum, (b) Real and imaginary parts after filtering, (c) Wrapped phase distributions (above), carried removed (below), (d) final result.

6.8 Parallel implementation

The extra processing required in the MFFA technique presented in this chapter produces an overhead in time respect to the conventional FFA technique.

This section presents a parallel implementation of the algorithm that translates this time overhead into a slight increment in monetary cost. A simple implementation of a distributed memory multiprocessor version of the algorithm is described. The algorithm is based upon a domain decomposition approach, so typical in the resolution of parallel processing problems for general image processing. It will be seen that the use of a two processor network can considerably reduce the execution time for the MFFA technique.

The algorithm has been implemented in an INMOS T800 network composed of two processors connected as illustrated in Figure 82.

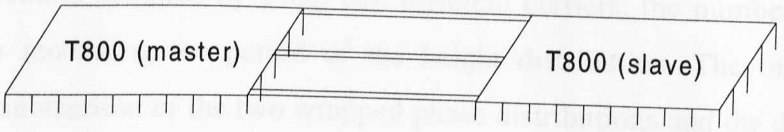


Figure 82. A two-transputers network.

Since a transputer has four links, three of them can be connected to the other transputer (note that the fourth link is required for connection to the host computer). This allows faster communication between the processors.

The information contained in the initial fringe pattern is initially placed on one of the processors (master) . Parts of these data have to be delivered to the second transputer (slave) for processing. Fortunately, the cost of this transmission is negligible if performed in parallel with the calculation of the first two dimensional FFTs. The communication of row *i* can be undertaken in parallel with the calculation of the FFT of row *i-1* and the exchange of the previous FFT calculated by each processor. This exchange is required to proceed to the calculation of the FFTs of the columns.

Once the two dimensional FFT has been calculated, each processor contains two peaks; one processor contains the peaks at positive frequencies and the other the peaks belonging to negative frequencies. Hence, each processor can proceed to the filtering of a different peak, and the processing required to obtain the wrapped phase distribution. The subtraction of the two fringe patterns, the multiplication by the phase to height conversion factor and the return of the whole set of data to the master processor can be performed in a similar manner to the initial calculation of the two dimensional FFT.

6.9 Summary

A new technique based upon the FFA has been presented. By using two different phase distributions obtained by using two different carriers, the number of wraps is decreased by increasing the period of the height distribution. The only operation required is a subtraction of the two wrapped phase distributions and the determination of the new phase to height conversion factor, that is performed by the substitution in Equation (6.2) of the frequency f_0 by the subtraction of the carrier frequencies in the two different fringe pattern captured. A reduction of the number of wraps by a factor of five can easily be obtained. In the absence of a carrier, that is, when the carrier removal is performed before the unwrapping process (either by frequency shifting or in the spatial domain) this factor can be sufficient to remove completely the wraps from the phase distribution (if less that five fringes are present in the original wrapped phase distributions). If the carrier is not removed, just a reduction of the number of wraps is possible since the errors involved in the new phase distribution can lead to unacceptable results if the period is to be increased by a large factor.

The technique is effective with most of the fringe patterns and, in a perfect case without noise, the technique completely eliminates the wraps whatever high the carriers are. This is achieved by setting the carriers so that the difference between them is less than one. This is a good feature of the algorithm. As new methods are developed and new improvements reduce noise, the applicability of the method can also increase.

Note that multichannel could also be applied to the technique presented in (Zhao et al. 1994), producing the result with a single image.

Chapter seven

Reducing cross-talking in FFA

7.1 Introduction

Fourier Fringe Analysis was introduced by Takeda *et al.*(1982; 1983) as a technique for optical measurement. As described in chapter 2, the technique relies on the two-dimensional Fourier transform of the original fringe pattern to separate the amplitude and phase components. This technique requires a single fringe pattern therefore it has an advantage with respect to the so called phase stepping technique.

When working with non-fullfield images, the process is more complicated. The borders of the data appear as high frequencies in the Fourier domain whilst the areas with the same intensity (background) appear as low frequency components. This information interferes with the signal peak and leads to large errors in the phase result throughout the image.

A different technique, or an additional step is hence required in order to analyse these images. This chapter presents a mapping process that potentially allows non-fullfield images to be analysed. A simple example is also presented to verify the theory.

A novel technique that makes use of two fringe patterns and exhibits a higher precision than Fourier Fringe Analysis is also proposed in this chapter.

This technique is also based on the Fourier transform concept, but that avoids some of the problems related to the FFA method. It produces high precision results with less errors in the borders of the image than the FFA technique. Because only two fringe patterns are required, the use of the technique may be preferred to phase stepping that makes use of at least three fringe patterns.

Advantages and drawbacks of the technique will be discussed and a simple example will be given.

7.2 Solving the problem of non-fullfield images

As was explained in the previous section, the presence of borders will introduce additional high frequencies in the Fourier transform at the same time that constant intensities produce low frequencies in the power spectrum. Therefore, in order to obtain a separable spectrum in which the information concerning the object (signal peak) and the d.c. peak lie far enough apart to be isolated with no or little interference or cross-talking, these borders need to be eliminated.

This section proposes a direct mapping of the non-fullfield image into a full-field image, normally a square. The analysis can then be carried out on this full-field image, translating the final result into the original shape by the inverse of the mapping function.

Standard functions do already exist to translate geometric figures into different geometric figures. If the shape does not correspond to any geometric figure, a first mapping function is to be designed so that the original shape is translated into a first geometric figure, that will be translated into a square shape that fulfills the image.

This process can be rather complex depending on the specific shape, especially if it is taken into account that the functions must have an inverse so that the original shape can be recovered.

Two types of mapping functions are possible:

- **Direct mapping**. A point in the original shape has an integer number of correspondences in the fullfield image (larger than zero), and the value of the point is replicated into the created new image. The number of pixels on the square should ideally be equal to the number of points on the original shape. Otherwise, conflicts in the values either in the application of the function or its inverse can arise. However, these conflicts can be insignificant and neglected if the original shape and the square contain a similar number of points. The advantage of this mapping (Figure 83) is that the inverse of the function is implicit, specially in the one to one mapping.

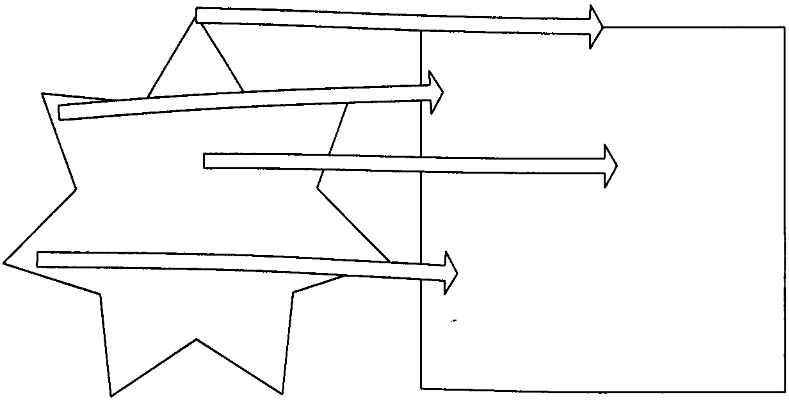


Figure 83. Direct Mapping.

- **Indirect mapping**. A function is designed so that given a pair of coordinates, another pair of coordinates is obtained as a result. The pair of coordinates obtained does not need to be integers and the value of the new points will be calculated in a neighbourhood weight basis.

The first approach has been successfully applied in experiments reported here. These kind of functions are easier to design and have shorter execution times. On the other hand, the second type of functions lead to more accurate results. Direct mapping functions can produce false high frequencies in the case of very irregular shapes.

In both cases, the distortion of the shape of the fringes has to be kept to a minimum, so that no closed fringes are obtained. The Fourier transform technique does not respond accurately to these type of fringes, since they are also reflected as high frequencies in the power spectrum, that are normally filtered out. Closed fringes do not give an isolatable peak.

In a normal square fringe pattern errors are obtained due to the assumed periodic nature of the Fourier transform. These errors concentrate on the borders of the image. When the method presented in this chapter is applied, this problem can be partially overcome.

The solution is to map the original shape into a square that is slightly smaller than the size of the image as illustrated in Figure 84. This process allows the creation of a simulated border. The values that will be given to the borders depend on the nature of the particular image under analysis and the orientation of the fringes but, as long as these values do not produce high frequencies in the Fourier spectrum, the errors will concentrate in the simulated border, that contains no information and can hence be discarded.

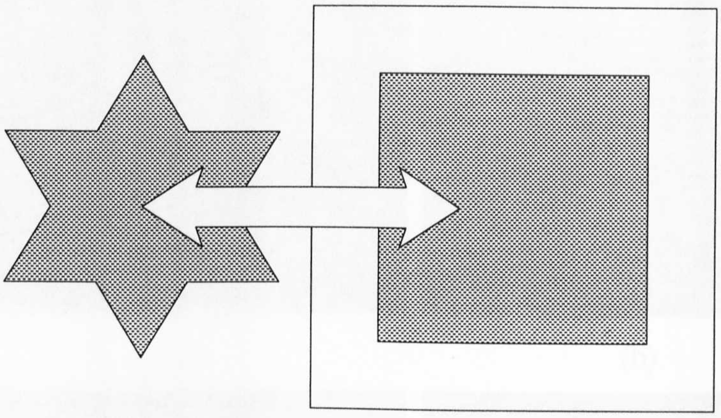
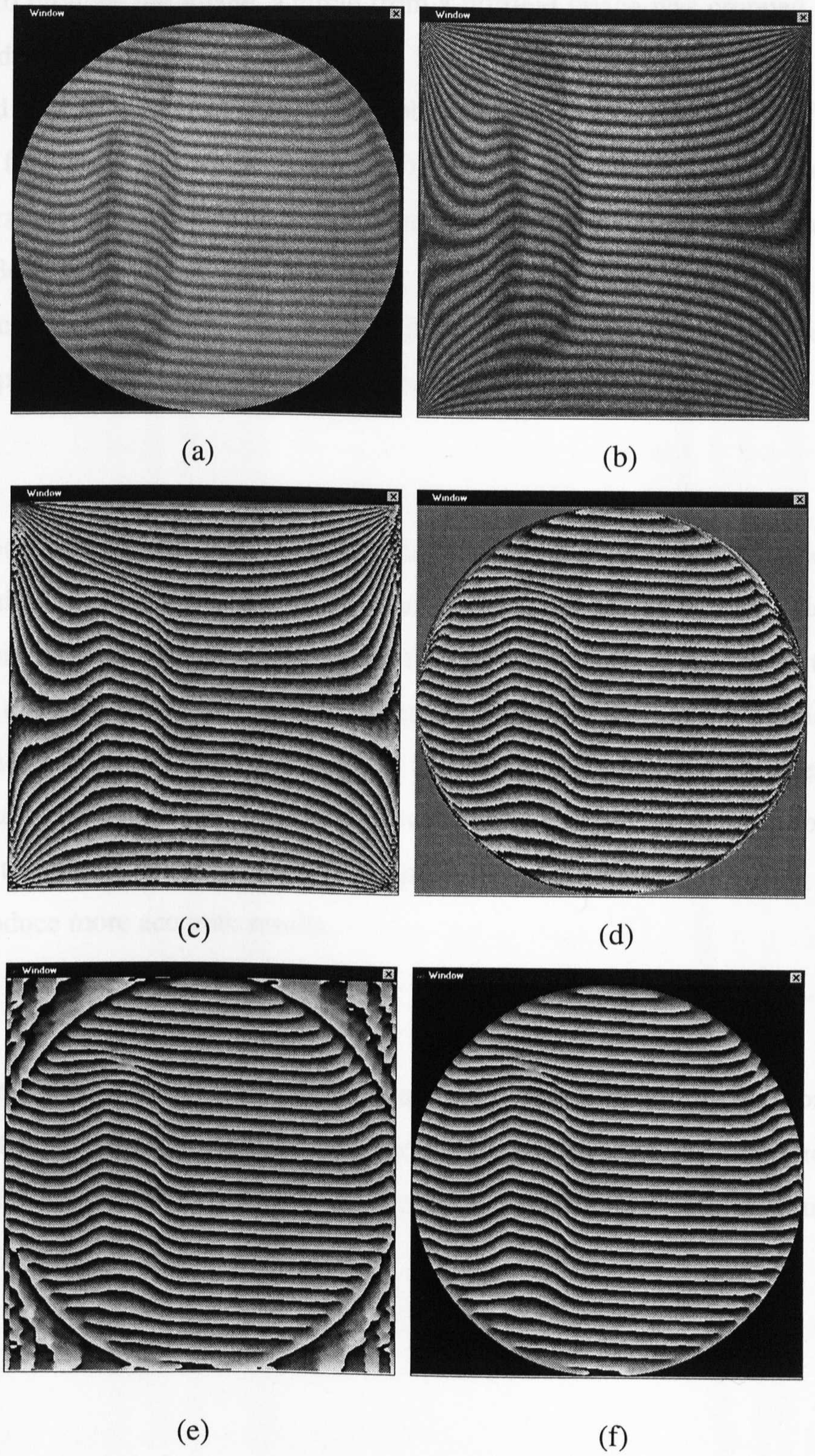


Figure 84. Mapping into a smaller square.

This leaves the result with little or no distortion, since the errors produced in the borders have been avoided.

7.2.1 Results

In this section, an example is given and the results are analysed. In order to test the technique, a circle shape has been chosen as the non-fullfield image. This conversion has a simple mapping. However, the mapping of a circle into a square is a function that causes a high distortion of the fringes shape, leading to errors in the borders. Other more regular mappings, causing less distortion, would lead to better results around the borders of the image.



(a) (b)

(c) (d)

(e) (f)

Figure 85. (a) Original cropped image, (b) mapping, (c) wrapped phase of the mapped image, (d) final wrapped phase of the original cropped imag. (e) and (f) results obtained with conventional FFA.

In order to produce the image, a circle from a fullfield image was cropped and it was proceeded to analyse the non-fullfield image obtained. The cropped image is illustrated in Figure 83(a). The image obtained after the application of a simple mapping function is shown in Figure 83(b). The image illustrated in Figure 83(b) is Fourier transformed, filtered in the Fourier domain and inverse Fourier transformed. Figure 83(c) is the result obtained when the wrapped phase distribution of Figure 83(b) is calculated. Finally, the inverse of the mapping function is applied to this wrapped phase distribution, obtaining the result exposed in Figure 83(d).

Figure 83(e) and Figure 83(f) represent the results obtained with conventional FFA, before and after cropping. Wrap discontinuities can be observed at the borders, where wraps seem to wash out and at the top of the convex surface at the left of the figure. Note that different filter sizes and shapes could have been used in the Fourier domain to decrease these errors. Figure 83(e) and Figure 83(f) can hence be misleading and better results could have been obtained. Nevertheless, the same principle applies to the mapping technique presented in this chapter. Better filters and mapping functions would produce more accurate results.

It is notable that although no technique to improve the results in the borders (the mapping filled the whole image) has been used and despite the distortion of the fringes, the areas on the border that contains misleading information are not particularly large.

7.3 A Fourier phase stepping technique

FFA has been proven to be an effective technique for three dimensional object measurement. However the technique suffers from two main problems stemming from the operations performed in the Fourier domain, that the technique presented in this chapter partially overcomes:

- Cross-talking. Because the distance between the peaks is not infinite, the peaks influence each other and complete isolation is not possible. Errors are produced when either frequencies belonging to the peak are cropped or when frequencies belonging to a different peak are included.

- Leakage. That is the main cause of the errors produced in the borders by this technique. It is produced because of the high frequencies on the borders of the image.

The proposed technique, that is explained for a single dimension (note that the extension to two dimensions is straight forward), makes use of two fringe patterns that are phase shifted by $\frac{3\pi}{2}$ rad. with respect to each other. In this way, the two different fringe patterns can be mathematically expressed by equations such as

$$I_1(x) = a(x) + b(x)\cos(2\pi fx + \phi(x))$$

and

$$I_2(x) = a(x) + b(x)\sin(2\pi fx + \phi(x))$$

If the second equation is multiplied by the complex number i and added to the first equation, a new expression is obtained:

$$I(x) = a(x) + b(x)\cos(2\pi fx + \phi(x)) + ia(x) + ib(x)\sin(2\pi fx + \phi(x))$$

that can be simplified as

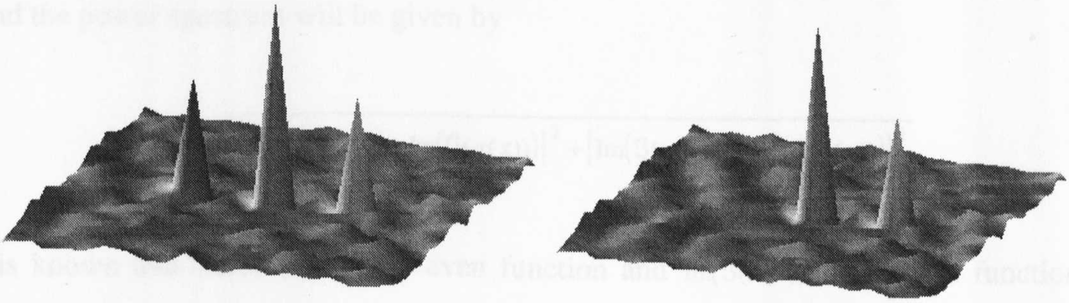
$$I(x) = a(x) + ia(x) + b(x)\left[\cos(2\pi fx + \phi(x)) + i\sin(2\pi fx + \phi(x))\right]$$

$$I(x) = a(x) + ia(x) + b(x)\left[e^{i(2\pi fx + \phi(x))}\right]$$

$$I(x) = a(x) + ia(x) + c(x)\left[e^{i2\pi fx}\right] \tag{7.1}$$

where $c(x) = b(x)e^{i\phi(x)}$

Therefore, the Fourier spectrum will present only two peaks instead of the three in conventional FFA. This is illustrated in Figure 86.



(a) (b)

Figure 86. The power spectrum presents only two peaks (b) in contrast to the three peaks that appear in conventional FFA (a).

Cross-talking is hence reduced. There is no cross-talking between two signal peaks as in the FFA technique, since there is a single signal peak. The phase of the wavefront can be obtained by exactly the same procedure applied in the FFA technique, isolating the signal peak.

The main advantage of the technique presented is in the filtering process.

Let us return to expression (7.1) and analyse the Fourier transform of the term $a(x)+ia(x)$. Because of the linearity of the Fourier transform,

$$\Im(a(x)+ia(x))=\Im(a(x))+i\Im(a(x))=(1+i)\Im(a(x))$$

where $\Im$ denotes the Fourier transform operation

This equation can be expressed in terms of real and imaginary parts as follows:

$$\Im(a(x)+ia(x))=(1+i)\left[\mathrm{Re}(\Im(a(x)))+i\,\mathrm{Im}(\Im(a(x)))\right]$$
$$\Im(a(x)+ia(x))=\mathrm{Re}(\Im(a(x)))+i\,\mathrm{Im}(\Im(a(x)))+i\,\mathrm{Re}(\Im(a(x)))-\mathrm{Im}(\Im(a(x)))$$
$$\Im(a(x)+ia(x))=\mathrm{Re}(\Im(a(x)))-\mathrm{Im}(\Im(a(x)))+i\left[\mathrm{Im}(\Im(a(x)))+\mathrm{Re}(\Im(a(x)))\right]$$

And the power spectrum will be given by

$$P(x)=\sqrt{\left[\mathrm{Re}(\Im(a(x)))-\mathrm{Im}(\Im(a(x)))\right]^2+\left[\mathrm{Im}(\Im(a(x)))+\mathrm{Re}(\Im(a(x)))\right]^2}$$

It is known that $\mathrm{Re}(\Im(a(x)))$ is an even function and $\mathrm{Im}(\Im(a(x)))$ is an odd function. Hence,

$$\mathrm{Re}(\Im(a(-x)))=\mathrm{Re}(\Im(a(x)))$$
$$\mathrm{Im}(\Im(a(-x)))=-\mathrm{Im}(\Im(a(x)))$$

and

$$P(-x)=\sqrt{\left[\mathrm{Re}(\Im(a(-x)))-\mathrm{Im}(\Im(a(-x)))\right]^2+\left[\mathrm{Im}(\Im(a(-x)))+\mathrm{Re}(\Im(a(-x)))\right]^2}$$
$$P(x)=\sqrt{\left[\mathrm{Re}(\Im(a(x)))+\mathrm{Im}(\Im(a(x)))\right]^2+\left[-\mathrm{Im}(\Im(a(x)))+\mathrm{Re}(\Im(a(x)))\right]^2}$$

Comparing the previous two equations, it is possible to conclude that $P(x)=P(-x)$ and, therefore, the power spectrum will be an even function, symmetric with respect to the centre. This characteristic is one of the main strengths of the technique. The power spectrum, except the phase of the object located at a distance *f* from the centre, will be symmetric with respect to the centre. This property makes it easy to distinguish the d.c. component from the signal peak in the frequency domain allowing a more selective filter to be applied. The cross-talking between the two signal peaks in the FFA technique is completely avoided and the only cross-talking remaining is introduced by the d.c. peak. However, the effect can be greatly reduced by the use of special filters and the use of tilted fringes.

The information about the noise, based on its symmetric nature, is merely subjective. However, the application of a star filter (as the one illustrated in Figure 87) has been proven to produce good results. The use of tilted fringes will cause the signal peak to lay further away from the d.c. peak and will reduce the influences of the cross-talking interference between the amplitude and signal peaks. If the fringes are not tilted, the leakage of the signal peak cross-talk with the d.c. peak, the application of the star band-pass filter being simply not convenient.

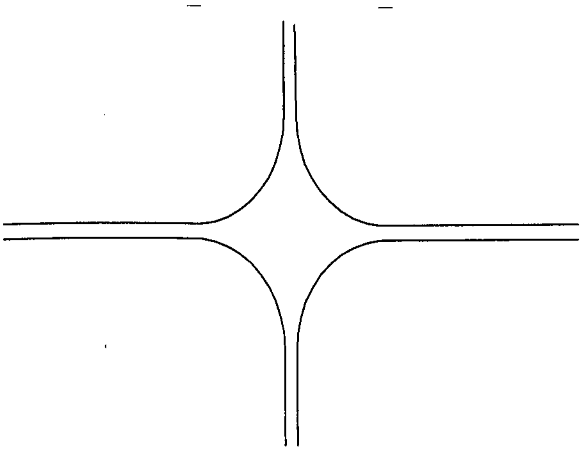


Figure 87. Star filter. This filter includes the leakage that produces the errors in the borders.

This filter is positioned at the centre of the signal peak. Errors at the borders of the image are greatly reduced because the application of the star filter includes the leakage. Speckle images are also suited to the techniques, since the speckle effect is reduced by the use of the Fourier domain filtering.

7.3.1 Experiments and results

The production of the fringe patterns is simple by using the phase stepping system described in chapter 2 (Figure 4).

Calibration of the system can be carried out as a phase shifting system. Errors in the calibration required to produce the phase step will appear very clearly in the Fourier domain as a 'shadow' that is symmetric to the signal peak and of less intensity. This is illustrated in Figure 88. If the system is properly calibrated, the signal peak should not appear with a symmetric partner.

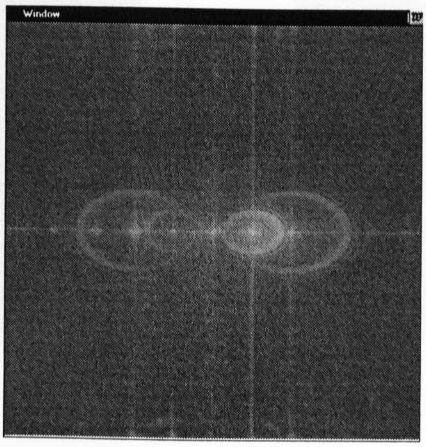
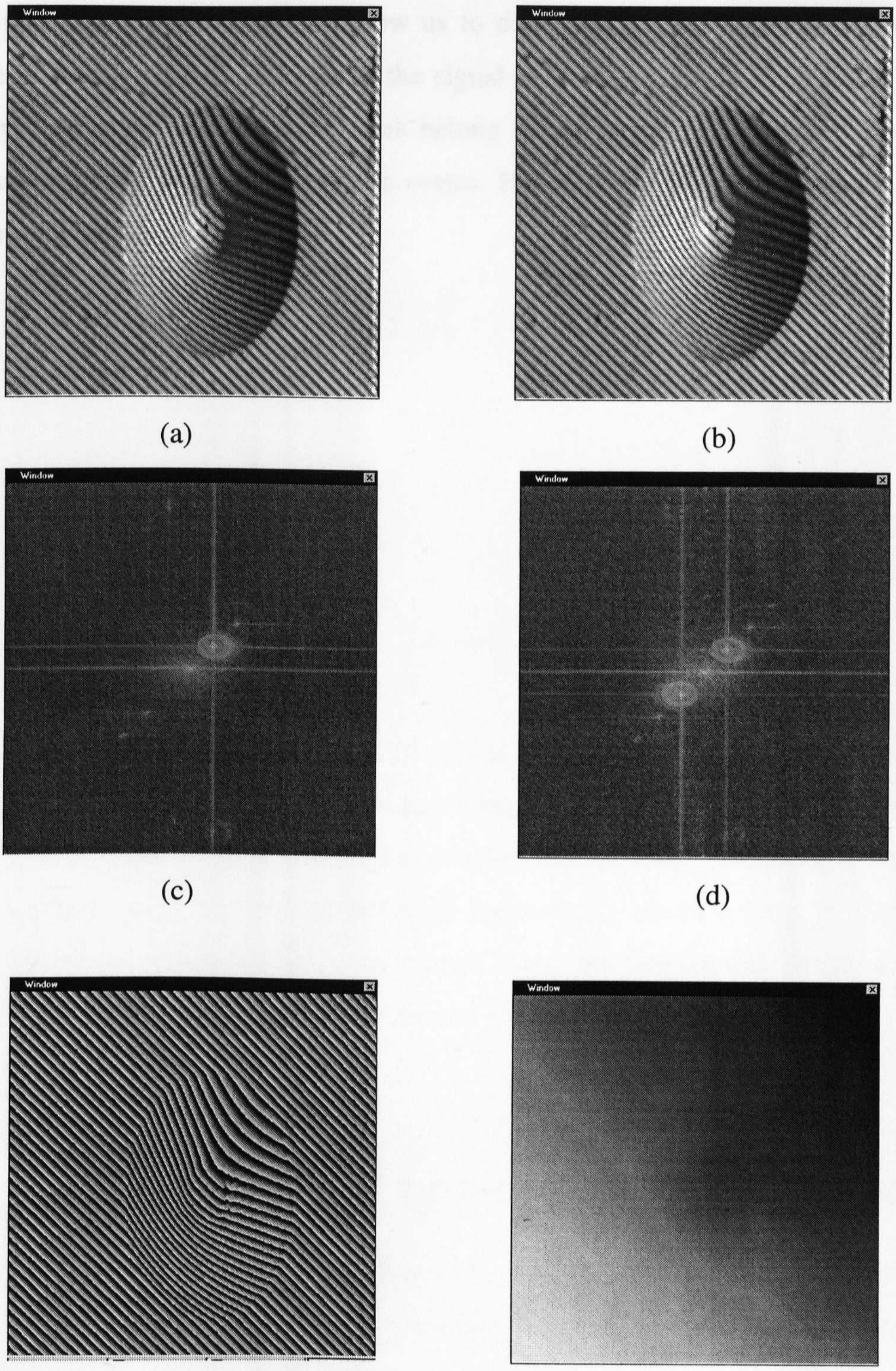


Figure 88. Symmetrical reflectance produced by mis-calibration of the system.

The technique presented in this paper has been tested with a number of images. The technique is illustrated with an image captured by this system.

Figure 89(a) is the original fringe pattern. Figure 89(b) represents the fringe pattern that is obtained when we introduce a phase shift of $\frac{2\pi}{3}$ rad. Figure 89(c) represents the Fourier spectrum when the technique explained in this chapter is applied, in contrast to Figure 89(d), which illustrates the Fourier spectrum when conventional Fourier fringe analysis is applied on Figure 89(a). Figure 89(e) represents the wrapped phase distribution and, finally, Figure 89(f) illustrates the final result. Filtering was undertaken by using the star filter previously described. No errors were observed in the borders and they are free of inconsistencies when the unwrapping is undertaken.



(a) (b)

(c) (d)

(e) (f)

Figure 89. The technique illustrated. (a) and (b) are two images captured with the required phase step, (c) is the Fourier spectrum produced by our technique, (d) is the power spectrum produced by conventional FFA, (e) is the wrapped phase distribution and (f) the unwrapped phase distribution produced by the recursive unwrapper described in chapter 4.

Note that Figure 89(d) does not allow us to determine if the hoop surrounding the centre of the signal peak belongs to the signal peak or not. From Figure 89(c), it is clearly determined that the hoop does belong to the signal peak, since it does not appear symmetric with respect to the centre. Hence it should not be cropped when performing the filtering process.

Chapter eight

Future work

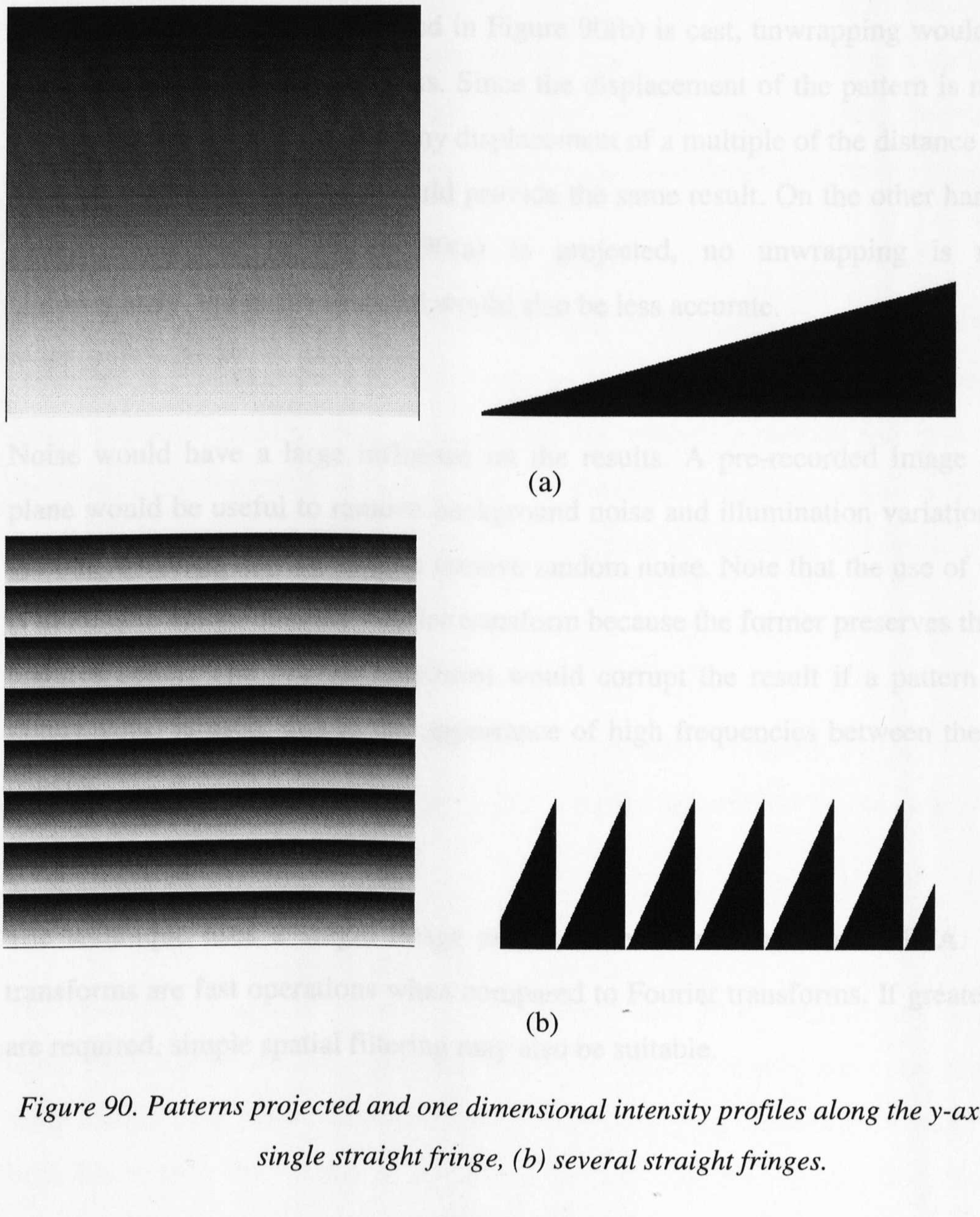
8.1 Introduction

This chapter introduces two further techniques that have not yet been implemented. The first technique, a proposal for fast non-contact measurement is based on a special grating to produce a fast and accurate measurement of three dimensional surfaces using a single image. The second technique proposes the use of colours to determine the positions of the wraps in a technique based on the Fourier fringe analysis principles. This work is intended to be carried out in the near future.

8.2 A proposal for fast non-contact measurement

This thesis has concentrated on techniques based on fringe projection for non-contact measurement of object surfaces. The method described in this section varies the pattern projected onto the object, so that a fast measurement can be performed.

Conventional fringe pattern projection is based on casting a sinusoidal fringe pattern onto a surface. The method suggested here relies on the projection of a saw-tooth fringe pattern as those illustrated in Figure 90.


(a)

(b)

Figure 90. Patterns projected and one dimensional intensity profiles along the y-axis. (a) a single straight fringe, (b) several straight fringes.

These fringe patterns can be decomposed on a series of lines with the same intensity along the *x*-axis. When they are projected onto a surface they appear deformed, if observed off-axis. The distance between the initial position and the location of the projected lines provides the required information about the shape of the object. The displacement has to be multiplied by a constant, that depends on the parameters of the optical configuration, to produce the height distribution of the object under analysis.

If a pattern such as that illustrated in Figure 90(b) is cast, unwrapping would also be required at the end of the analysis. Since the displacement of the pattern is measured with respect to the nearest line, any displacement of a multiple of the distance between lines with the same intensity would provide the same result. On the other hand, if the pattern illustrated in Figure 90(a) is projected, no unwrapping is required. Unfortunately, the result obtained would also be less accurate.

Noise would have a large influence on the results. A pre-recorded image of a flat plane would be useful to remove background noise and illumination variations in the system. Wavelets may be used to remove random noise. Note that the use of wavelets is more convenient than the Fourier transform because the former preserves the spatial features better. The Fourier transform would corrupt the result if a pattern such as Figure 90(a) is used, due to the appearance of high frequencies between the straight fringes.

The technique uses a single image and is faster than conventional FFA. Wavelet transforms are fast operations when compared to Fourier transforms. If greater speeds are required, simple spatial filtering may also be suitable.

8.3 A technique with inherent phase unwrapping

The problem of phase unwrapping reduces to the determination of the fringe orders of a fringe pattern. This section presents a technique that allows phase unwrapping in a simple manner by using the projection of colours onto the object's surface. The colours serve as a method for identification of the fringe orders. Unwrapping is performed by addition of a multiple of 2π whose value depends upon the fringe order.

It has been observed that in FFA the position of the wraps in a phase distribution corresponds to the position of the minima of the fringes in the original modulated fringe pattern. Tracking these minima does not produce a method for unwrapping in the case of discontinuities or sharp edges on the object's surface, since the minima are not continuous in those circumstances. Unwrapping is, therefore, limited to the ability of an algorithm to count the fringes.

If a fringe pattern in which the minima are replaced by different colours is projected onto the object's surface, the fringe order is easily determined by the colour of these minima.

The resulting image is processed by conventional FFA. Before any analysis is undertaken a preprocessing operation would be required to produce a completely grey scale image. This preprocessing consists on replacing the colours by a value that is a little lower than the values of the pixels surrounding the colour pixel, that is, the colours are replaced by an approximation to the minima. Processing by FFA is then undertaken in the conventional manner. For unwrapping, the fringe orders are provided by the colours at every phase wrap. In other words, the position of the wraps is provided by the location of the colours on the initial modulated fringe pattern.

Errors would be produced whenever there are discontinuities on the object surface that correspond to a difference in phase greater than 2π and if they occur in the same direction than the fringes, in which case some minima are not present. In this case, uncertainties occur in the areas surrounding the discontinuity. However, the error does not propagate further than the extension of a single fringe and the areas can be localised and some kind of post-processing can be carried out. This processing consists of the analysis of colours from the original sequence that do not appear on the acquired fringe pattern. Special attention would need to be paid to this area and discontinuities tracked. This process constitutes a problem due to the unreliability of the data around the discontinuity area. Projection of the fringes perpendicular to the discontinuities in the object surface is hence preferred to avoid the problems.

The technique is limited to the ability of the camera to accurately distinguish the projected colours. However, it is not required to recognise as many colours as there are fringes in the original projected fringe pattern. Sequenced combinations of colours could be used. In this case, the fringe order will be determined by taking into account the previous sequence of colours.

Chapter nine

Discussion and Conclusions

9.1 Discussion

This thesis has presented a number of novel techniques concerned with the resolution of several different key problems related to fringe analysis.

Several topics have been discussed, FFA being the most used technique through the experimentation process.

Many problems have been described as inherent to the technique. Subtraction of the carrier or tilt can be performed in either the spatial or the frequency domain. Problems are associated with this operation if it is performed in the Fourier domain. If the number of fringes is not an exact integer number the displacement of the signal peak to the centre will not be an accurate process because of the digitisation of the image and the use of the discrete Fourier transform. Carrier removal is therefore an operation that should be carried out in the spatial domain. The stage at which the tilt removal is implemented is also relevant. If the carrier is removed in the wrapped phase distribution, the number of wraps usually decreases by a large factor. Note that in this case the phase distribution has to be re-wrapped after the elimination of the tilt.

Filtering in the Fourier domain is also a difficult process. Its complexity can be attenuated by the choice of the appropriate orientation and number of fringes in the fringe pattern projected. Introducing an angle in the projected fringe pattern induces a larger distance between the amplitude and phase terms in the Fourier spectrum. Cross-talking is also decreased along with a separation of the leakage influences between the peaks. A priori information, such as the smoothness of the surface under analysis, is also an aid to the determination of the appropriate frequency for the projected fringe pattern. The use of a high number of fringes is usually advantageous for filtering as it increases the distance between the phase and amplitude terms. However, it increases the difficulty of the unwrapping process since more wraps will appear in the final phase distribution. In many cases, it is possible to determine a reasonable number of fringes so that no wraps appear in the final phase distribution (or at most a single wrap that can be resolved by the addition of a constant and the re-wrapping of the resulting phase distribution). If the difference between the maximum and minimum heights in the object's surface is lower than

$$\frac{1}{f \sin\theta}$$

where θ is the angle the angle between the actual camera position with respect to the orthogonal camera position and f is the frequency of the fringe pattern projected, at most one wrap appears in the resulting wrapped phase distribution. From the last expression, it is obvious that a low number of fringes is beneficial for wrap reduction purposes. Nevertheless, if the number of fringes is too low, cross-talking does occur between the amplitude and signal peaks, and between the two signal peaks. This leads to unreliable information spread amongst the final result, and possibly false wrap discontinuities that complicate the unwrapping process.

Two contrary effects have been described for the use of the appropriated number of fringes: A high number of fringes facilitates filtering and minimises cross-talking between the signal and d.c. peaks in the Fourier domain. On the other hand, a low number of fringes leads to less wraps in the wrapped phase distribution. A balance has to be reached so that little or no cross-talking occurs whilst, at the same time, the number of fringes allows simple unwrapping.

A novel Fourier filter with a star shape has been proposed that includes the leakage of the signal peak. The non-elimination of this leakage produces less errors at the borders. Its shape usually suits the signal peak shape in the Fourier domain. The application of this filter is dependent upon the introduction of a rotation to the projected fringe pattern so that a minimal influence between the leakages of the peaks occurs. The size of filter, however, has to be carefully chosen. A large size of the filter will lead to a sinusoidal error appearing in the final result. A very small filter size induces a lose of definition in the resulting phase distribution as important first order peak information is eliminated and the phase cannot be correctly reconstructed. The optimum size of the filter depends upon the particular image under analysis and its calculation has to be undertaken with regard to a particular Fourier spectrum.

The experience with the two systems described in Chapter 2 shows that phase stepping performs better than FFA in terms of accuracy under stable environment conditions. However, the requirement for a single image in FFA and its flexibility makes this a valuable technique for fringe analysis. A variation on the number of fringes projected or a rotation of the fringe pattern are straight forward if an interferometric system is utilised. If phase stepping is used with a grating based projection system, a rotation of the fringe pattern usually implies a re-calculation of the phase step. If a variation of the number of fringes is required, the grating needs to be replaced and the system re-calibrated.

The requirement of a single image makes FFA less susceptible to environmental conditions such as vibration, being more suitable for unstable environments or surfaces in slow motion. When industrial inspection is considered as a main application of fringe projection analysis, this feature is a major advantage of the FFA technique since there are situations in which the achievement of stable environmental conditions is not possible.

Unwrapping is a problem that is common to both phase stepping and FFA techniques. Whichever of these techniques is applied to the analysis of a modulated fringe pattern, the phase is finally obtained in the interval [-π, π] and discontinuities in the phase map do usually occur. The process of solving these discontinuities to obtain the desired continuous phase distribution is called unwrapping. Two dimensional unwrapping has been a major topic of the research described in this thesis. Two different novel techniques have been proposed. However, perfect unwrapping still remains as an unsolved topic and further research is to be considered in the future.

The first approach is based upon a recursive algorithm that could be considered as an adaptive tile size unwrapper. Areas containing erroneous data are masked and disregarded for further processing. Because the tile size is not defined but grows as the unwrapping process evolves, masking is limited to minimal areas. The algorithm is also able to cope with discontinuities in the wraps as long as they are shorter than half the length of a single line of the image (note that if the size of the largest discontinuity permitted is larger than this length, the algorithm disregards the wrap, rather than the discontinuity). A main feature of the algorithm is that it guarantees path consistency. The main limitation of the technique is that masking is limited to square areas. This fact can lead to some overmasking under certain conditions.

A second approach to unwrapping has also been proposed. The concepts of multiple scan directions, minimal path search and division of the phase distribution into regions have been combined to produce a relatively fast unwrapper. Its flexibility to suit different kind of images and its ability to isolate regions make this approach particularly attractive as an accurate algorithm for phase unwrapping. No masking is undertaken by the algorithm. Erroneous areas are rather given as different regions. Its masking, if required, is a simple process: small regions can be disregarded and masked out.

Both algorithms are suitable to *a priori* masking of points or regions. This is a useful feature if low modulation tests are performed during the calculation of the wrapped phase. Non-reliable pixels can then be disregarded prior to the unwrapping process.

A simple two-dimensional interpolation algorithm can be applied to resolve the masked areas so that a fullfield phase map is obtained. However, interpolation of large areas is not accurate and in certain cases, it may be preferred to consider these areas as invalid.

Two dimensional unwrapping has been an important topic of research during the last decade. A number of previous attempts at phase unwrapping has been gathered in Chapter 3. A great deal of effort has been dedicated to the problem. However, a perfect unwrapper has not been yet developed, to our knowledge. This fact has motivated the development of a technique that reduces the complexity of the unwrapping process. This technique has been described in Chapter 6. Its potential to eliminate the wraps in a two dimensional phase distribution makes this technique particularly interesting.

By subtracting two phase distributions of the same object's surface obtained independently with the projection of straight fringe patterns containing a different number of fringes, a larger period is obtained and a decrease in the number of wraps is achieved. Under determinate circumstances, this can be sufficient to completely eliminate the wraps in the phase distribution. Since every point is obtained independently from the others, no propagation of errors occurs and these are limited to single points. Non-consistent points can be detected and masked at the end of the process.

Problems related to this technique are the magnifications of the errors in the resulting height distribution. The decrease on the number of wraps is achieved by an increase of the phase to height conversion factor. Therefore, even if the errors in the phase distribution do not increase, they will be multiplied by a new, larger phases to height conversion factor. A solution for this is to look at the height differences rather than to the height values in the new height distribution and use this information to unwrap one of the original phase maps.

One problem is inherent in the use of the Fourier transform. When objects with sharp edges or discontinuities are under analysis, the high frequencies produced in the areas close to the edges, create disruptions in the final result. When subtracting the phases, these errors increase by a large factor and unreliable or invalid data are obtained in these areas. However, these regions are usually small and an interpolation is possible in order to recover the lost data.

The Multichannel Fourier Fringe Analysis technique has also been introduced in the context of the previously described technique. The use of this method allows the wrap reduction technique to be implemented so that the final phase is extracted with the requirement of a single image. A further benefit is associated with the multichannel strategy when the technique is to work under unstable environmental conditions. The use of multichannel ensures that the object is exactly at the same position in the two fringe patterns, a condition that needs to be accomplished.

Multichannel Fourier transforms require a previous analysis of the position of the peaks so that cross-talking and leakage influences are minimised. By appropriately defining the frequencies of the projected fringe patterns the peaks can be isolated and a different phase distribution can be obtained from each of them. The maximum errors in the new height distribution are much larger. However, it has been observed that, in practice, the errors from the two initial distributions tend to subtract and the results obtained are acceptable for most situations, if the reduction factor is not too high. A reduction of the number of wraps by a factor of five is easily achieved.

Problems with the FFA technique have been briefly described. Cross-talking, leakage and aliasing have been presented as usual problems to the technique. Corruption of the borders and edges is a general problem of the application of FFA. A novel technique that eliminates or at least attenuates these problems has been proposed.

With the use of two phase shifted fringe patterns, a single signal peak can be made to appear in the Fourier domain, filtering being a simpler and more accurate task. The application of the star filter has been illustrated in the context of this technique. The technique allows a visual separation of noise and signal information in the Fourier domain as unwanted components appear symmetric in the frequency spectrum. Since a single signal peak is present in the frequency spectrum, no cross-talking occurs as is the case in conventional FFA. This allows the use of a lower number of fringes that leads to a complexity reduction in the phase unwrapping process.

By introducing a rotational angle in the projected fringe pattern, cross-talking between the leakage's of the phase and d.c. peaks is considerably reduced, making the application of the star filter viable.

The main problem with this technique is that most of the information in the Fourier spectrum is merely visual, that is, even if the information can be easily isolated from the noise by looking at the Fourier spectrum, the automatisation of the algorithm is not simple.

Non-fullfield images are usually problematic if the FFA technique is used for analysis. High frequencies at the borders of the image (windowing effect) tend to corrupt the whole set of data under analysis. Iterative methods to extrapolate fringes to the empty areas have been proposed in the past (Stephenson 1994). Chapter 7 describes a simple and fast two dimensional mapping into a square shape as a solution to the analysis of non-fullfield images. This mapping produces a fullfield image that can be analysed by conventional FFA. This mapping, however, can introduce high curvature in the fringes (specially if the original non-fullfield area has a circular shape), that is most intensive at the corners of the image. This does not usually constitute a problem since borders are often cropped when conventional FFA is used. A mapping into a smaller area, i.e. a square that does not fill the entire image, and an extension of the borders has also been proposed to eliminate the errors at the edges. This extension is useless if a high curvature effect occurs. In this case, errors in the borders are present in the final result, and they should be regarded as invalid or non-reliable information.

Parallel computation has been applied in the past to the resolution of many different problems in the area of image processing (Helzle 1993). The use of parallel computation has been noted in different sections of this thesis. Parallel versions of several algorithms have been proposed and implemented.

The recursive unwrapper has been implemented on a four transputer network showing a good performance relative to the execution of the algorithm on a single transputer. The main problem is the load unbalance that can be produced by the existence of unreliable data that concentrate on a particular area of the image. The application of parallel computing also translated the excess of time produced by the use of multichannel Fourier fringe analysis into a slight increment in cost. Finally, an analysis of the implementation of cellular automata on a multicomputer network was undertaken, demonstrating the potential for high performance computation of this type of parallel machine.

The INMOS T800 transputer has been use to illustrate the potential of parallel computation to fringe analysis. Many other processors are commercially available at present. A new version of the transputer chip, the INMOS T9000, has been commercialised. The communication capabilities of the INMOS T9000 are far superior to that of first generation transputers (INMOS 1993). Texas Instruments also produces very fast DSP machines that also allow parallel processing via links connecting the processors (Platzner et al. 1992). A good performance comparison with the T800 transputer can be found in (Platzner et al. 1993), where the far greater performance of the DSP machine is demonstrated.

The results show the high scope of applicability of parallel computation to fringe analysis. Parallel processing allows a conversion of time into monetary cost. Multicomputers have been introduced as a relatively inexpensive alternative to produce fast processing of data. As processing power increases the scope for new, more numerically intensive techniques and solutions appear that were inapplicable in the past but can be of interest in the future.

9.2 Conclusions

A number of conclusions can be extracted from the work described in this thesis:

- FFA does not produce accurate results when objects with sharp edges or discontinuities are under analysis.

- Conventional FFA produce unreliable results on the borders of the image.

- Carrier removal is an operation that should be carried out in the spatial domain.

- Tilt removal eases the unwrapping process.

- Introducing an angle in the projected fringe pattern induces a larger distance between the amplitude and signal peaks in the Fourier spectrum. Cross-talking and leakage influences between the peaks are also reduced.

- A priori information, such as the smoothness of the surface under analysis, is an aid to choosing the appropriate number of fringes in the projected fringe pattern.

- The use of a high number of fringes increases the distance between the phase and amplitude terms. However, more wraps will appear in the final phase distribution. A low number of fringes is beneficial for wrap reduction purposes, but the effects of cross-talking and leakage influences are increased. A balance has to be reached.

- The optimum size of the filter depends upon the particular image under analysis and its calculation has to be undertaken with regard to a particular Fourier spectrum.

- The application of the Fourier filter with a star shape introduced in this thesis reduces the errors on the borders of the image, by including the leakage of the signal peak in the filtering process.

- Automatic filtering is not a simple issue.

- A mapping process can be utilised to analyse non-fullfield images by means of the FFA technique.

- The recursive and iso-phase unwrapping techniques are two effective and fast approaches for two dimensional phase unwrapping. Both algorithms are suitable to *a priori* masking of points or regions. The use of low modulation tests is beneficial to both techniques.

- In some cases, overmasking can be a major problem for recursive unwrapping. Simple two-dimensional interpolation algorithms can be applied to resolve the masked areas. Interpolation of large areas is not accurate.

- A perfect unwrapper has not been yet developed

- A decrease in the number of wraps can be achieved by subtracting two phase distributions of the same object surface obtained independently with the projection of straight fringe patterns containing a different number of fringes.

- The magnification of the errors in the resulting height distribution is a major problem for the wrap reduction technique.

- The wrap reduction technique is best used as an aid to phase unwrapping.

- The multichannel technique is applicable to the wrap reduction method.

- Multichannel Fourier transforms benefit from a previous analysis of the appropriate multichannel fringe pattern.

- The Fourier phase stepping technique can be used to reduce leakage influence and cross-talking between the peaks in the Fourier spectrum.

- Parallel computation can be successfully applied to fringe analysis.

REFERENCES

Amdahl (1967), G. M. “Validity of the single-processor approach to achieving large scale computing capabilities.” *Proc. AFIPS Spring Joint Computer Conference*, 30, Atlantic City, NJ, 483-485.

Ananda, A., and Srinivasan, B. (1991). *Distributed computing systems: concepts and structures*, IEEE Computer society Press, Los Alamitos, California. ISBN 0818689750.

Atkinson, J. T., Burton, D. R., Lalor, M. J., and O'Donovan, P. C. (1988). “Opto/computer methods applied to the evaluation of a range of acetabular cups.” *Eng. in medicine*, 17(3), 105-110.

Barton, M., and Withers, G. (1989) “Computing performance as a function of the speed, quantity and cost of the processors.” *Proc. Supercomputing '89*, 759-764.

Bone, D. J. (1991). “Fourier fringe analysis: the two-dimensional phase unwrapping problem.” *Applied optics*, 30(25), 3627-3632.

Bone, D. J., Bachor, H. A., and Sanderman, J. (1986). “Fringe-pattern analysis using a 2-D Fourier transform.” *Applied Optics*, 25(10), 1653-1660.

Bourdet, L., and Orszag, A. G. (1979). “Absolute distance measurements by CO_2 laser multiwavelength interferometry.” *Applied optics*, 18(2), 225-227.

Brigham, E. O. (1974). *The Fast Fourier Transform*, Prentice Hall. ISBN 013307496x.

Bruning, J. H., Herriot, D. R., Gallagher, J. E., Rosenfeld, D. P., White, A. D., and Bragaccio, D. J. (1974). "Digital waveform measuring interferometer for testing optical surfaces and lenses." *Applied optics*, 13(11), 2693-2703.

Burton, D. R., Goodall, A. J., Atkinson, J. T., and Lalor, M. J. (1995). "The use of carrier frequency shifting for the elimination of phase discontinuities in Fourier transform profilometry." *Optics and lasers in engineering*(23), 245-257.

Burton, D. R., and Lalor, M. J. (1988). "Precision measurement of engineering form by computer analysis of optically generated contours." *Proc. SPIE*, 1010, 17-24.

Burton, D. R., and Lalor, M. J. (1994). "Multichannel Fourier fringe analysis as an aid to automatic phase unwrapping." *Applied optics*, 33(14), 2939-2948.

Button, B. L., Cutts, J., Dobbins, B. N., Moxon, C. J., and Wykes, C. (1985). "The identification of fringe positions in speckle patterns." *Opt. and laser technol.*, 17, 189-192.

Carré, P. (1966). "Installation et utilisation du comparateur photoélectrique et interférentiel du Bureau International des Poids et Mesures." *Metrologia* 2, 13-23.

Carter, W. H. (1992). "On unwrapping two-dimensional phase data in contour maps." *Optics communications*, 94, 1-7.

Cheng, Y. Y., and Wyant, J. C. (1985). “Multiple wavelength phase shifting interferometry.” *Applied optics*, 24, 804-807.

Cheng, Y.-Y., and Wyant, J. C. (1984). “Two-wavelength phase shifting interferometry.” *Applied optics*, 23(24), 4539-43.

Ching, N. H. (1992). “Two dimensional phase unwrapping using a minimum spanning tree algorithm.” *IEEE transactions on image processing*, 1(3), 355-365.

Chung, R. Y. (1992). “Non-contact surface inspection. ” PhD Thesis, John Moores University, Liverpool.

Cline, H. E., Holik, A. S., and Lorensen, W. E. (1982). “Computer aided surface reconstruction of interference contours.” *Applied optics*, 21, 4481-4488.

Cok, R. S. (1991). *Parallel programs for the transputer*, Prentice Hall.

Creath, K. (1988) “Phase measurement interferometry techniques.” *Progress in optics*, 26, 349-369

Creath, K., Cheng, Y.-Y., and Wyant, J. C. (1985). “Contouring aspheric surfaces using two-wavelength phase-shifting interferometry.” *Optical Apta*, 32(12), 1455-64.

Creath, K., and Schmit, J. (1994). “Errors in spatial phase-stepping techniques.” *SPIE*, 2340, 170-176.

Cusack, R., Huntley, J. M., and Goldstein, H. T. (1995). "Improved noise-immune phase-unwrapping algorithm." *Applied optics*, 34(5), 781-789.

Dandliker, R., and Thalmann, R. (1985). "Heterodyne and quasi-heterodyne holographic interferometry." *Opt. engineering*, 24, 824-831.

Denning, P. J. (1986). "Parallel computing and its evolution." *Communications of the ACM*, 29(12), 1163-1167.

Eager, D., Zahorjan, J., and Lazowska, E. (1989). "Speedup versus efficiency in parallel systems." *IEEE Trans. Comput.*, 3, 403-423.

East, I. (1995). *Parallel processing with communicating process architecture*, UCL Press, London, U.K.

El-Rewini, H., and Lewis, T. (1993). "Parallel and distributed systems. From theory to practice." *IEEE Parallel and distributed technology*, 8, 7-11.

Frankowski, G., Stobbe, I., Tischer, W., and Schillke, F. (1989). "Investigation of surface shapes using carrier frequency based analysing systems." *Proc. SPIE*, 1122, 89-100.

Galletly, J. (1990). *Occam 2*, Pitman, London, UK. ISBN 0273030671.

Ghiglia, D. C., Mastin, G. A., and Romero, L. A. (1987). "Cellular-automata method for phase unwrapping." *J. Opt. Soc. Am.*(4A), 267-280.

Ghiglia, D. C., and Romero, L. A. (1994). "Robust two-dimensional weighted and unweighted phase unwrapping that uses fast transforms and iterative methods." *J. opt. soc. Am. A*, 11(1), 107-117.

Gierloff, J. J. (1987) "Phase unwrapping by regions." *Current developments in optical engineering II*, SPIE 818, 2-9.

Goldstein, R. M., Zebker, H. A., and L.Werner, C. (1988). "Satellite radar interferometry: two-dimensional phase unwrapping." *Radio Science*, 23(4), 713-720.

Grama, A. Y., Gupta, A., and Kumar, V. (1993). "Isoefficiency: measuring the scalability of parallel algorithms and architectures." *IEEE Parallel and distributed technology*, 8, 12-21.

Green, R. J., Walker, J. G., and Robinson, D. W. (1988). "Investigation of the Fourier transform method of fringe pattern analysis." *Optics and lasers in engineering*, 8, 29-44.

Gustafson, J. (1988). "Reevaluating Amdahl's law." *Commun. ACM*, 31, 5, May, 532-533.

Gustafson, J. L. (1989). "Response to once again Amdahl's law." *Comm. ACM*, 32(2), 263-4.

Halioua, M., Krishnamurthy, R. S., Liu, H. C., and Chiang, F. P. (1985). "Automated 360 degree profilometry of 3-D diffuse objects." *Applied optics*, 24, 3105-3108.

Halsall, G. R. (1992). "Real time non-contact profilometry." PhD Thesis, Liverpool John Moores University, Liverpool.

Hariharan, P., Oreb, B. F., and Eiju, T. (1987). "Digital phase-shifting interferometry: a simple error compensating phase calculation algorithm." *Applied optics*, 26(13), 2504-2505.

Heath, M., and Worley, P. (1989). "Once again Amdahl's law." *Comm. ACM*, 32(2), 262-263.

Hedley, M., and Rosenfeld, D. (1992). "A new two-dimensional phase unwrapping algorithm for MRI Images." *Magnetic resonance in medicine*, 24, 177-181.

Helzle, M. (1993) "New ways in image processing with parallel DSPs." *Proc. of the 4th international conference on signal processing applications and technology*, vol. 2, UK 1134-43.

Hoare, C. A. R. (1962). "Quicksort." *Comput. J.*(5), 10-15.

Hoare, C. A. R. (1978). "Communicating sequential processes." *Comm. ACM*, 21(8), 666-677.

Hockney, R. W., and Curington, I. J. (1989). "f 1/2: A parameter to characterize memory and communication bottlenecks." *Parallel computing*, 10(3), 277-286.

Hockney, R. W., and Jesshope, C. R. (1981). *Parallel computers*, Hilger Publishing Co., Bristol, U.K.

Huntley, J.M., and Saldner, H.O. (1996). "Shape measurement of discontinuous objects using projected fringes and temporal phase unwrapping." , *Proceedings of the applied optics divisional conference of physics*, Reading, 16-19 September, K.T.V. Grattan, Institute of Physics publishing, ISBN 0-7503-0382-4

Huntley, J. M. (1994). "New methods for unwrapping noisy phase maps." *SPIE*, 2340, 110-122.

Huntley, J. M., and Saldner, H. O. (1993). "Temporal phase-unwrapping algorithm for automated interferogram analysis." *Applied optics*, 32(17), 3047-3052.

Huntley, J. M. (1989). "Noise-immune phase unwrapping algorithm." *Applied optics*, 28(15), 3268-3270.

Indebetouw, G. (1978). "Profile measurement using projection of running fringes." *Applied optics*, 17, 2930-2933.

INMOS. (1984). *occam programming manual*, Prentice Hall, New Yersey.

INMOS. (1988). *Transputer Reference Manual*, Prentice Hall, New Yersey.

INMOS. (1992). *The occam 2 reference manual,* Prentice Hall, New Yersey.

INMOS. (1993). *The T9000 transputer hardware reference manual. First edition,* Prentice Hall, New Yersey.

Jones, G. (1987). *Programming in occam,* Prentice Hall. ISBN 0137297734.

Jones, G., and Goldsmith, M. (1988). *Programming in occam 2,* Prentice Hall. ISBN 0137303343.

Judge, T. R., Quan, C., and Bryanston-Cross, P. J. (1992). "Holographic deformation measurements by Fourier transform technique with automatic phase unwrapping." *Optical engineering,* 31(3), 533-543.

Karp, A. H., and Flatt, H. P. (1990). "Measuring parallel processor performance." *CACM,* 33, 539-543.

Kerridge, J. (1987). *occam programming: a practical approach,* Blackwell scientific publications, Computer science texts series. ISBN 0632016582.

Knuth, D. E. (1973). The art of computer programming, Addison-Wesley, Reading, 115-25. ISBN 020103803x

Kujawinska, M., Spik, A., and Wójciak, J. (1989). "Fringe pattern analysis using Fourier transform techniques." *Proc. SPIE,* 1121, 130-6.

Kujawinska, M., and Wójciak, J. (1991). “High accuracy Fourier transform fringe pattern analysis.” *Optics and lasers in engineering*(14), 325-339.

Kumar, V. (1988). “Measuring parallelism in computation intensive scientific/engineering applications.” *IEEE-Transactions on computers*, 37(9), 1088-1098.

Kumar, V., Grama, A., Gupta, A., and Karypis, G. (1994a). *Introduction to parallel computing. Design and analysis of algorithms*, Benjamin/Cummings, Redwood city, California.

Kumar, V., Grama, A. Y., and Vempaty, N. R. (1994b). “Scalable load balancing techniques for parallel computers.” *Journal of parallel and distributed computing*, 22, 60-79.

Kumar, V., and Gupta, A. (1994). “Analyzing scalability of parallel algorithms and architectures.” *Journal of parallel and distributed computing*, 22, 379-391.

Kumar, V., and Sun, X.-H. (1994). “Special issue on scalability of parallel algorithms and architectures.” *Parallel and distributed computing*, 22, 375-378.

Kwon, O. Y., Shough, D. M., and Williams, R. A. (1987). “Stroboscopic phase-shifting interferometry.” *Optics Letters*, 12(11), 855-857.

Macy, W. (1983). “Two-dimensional fringe pattern analysis.” *Applied optics*, 22(23), 3898-3901.

Malcolm, A. A., Burton, D. R., and Lalor, M. J. (1989) "A study of the effects of windowing on the accuracy of surface measurements from Fourier analysis fringe patterns." *Proceedings applied optics and opto-electronics FASIG/IOP fringe analysis 89*, Loughborough, UK

Malmo, J. T., and Vikhagen, E. (1988). "Vibration analysis of car body by means of TV holography." *Experimental techniques*, 28-30.

Marinescu, D. C., and Rice, J. R. (1994). "On the scalability of asynchronous parallel computations." *Journal of parallel and distributed computing*, 22, 538-546.

Nugent, K. A. (1985). "Interferogram analysis using an accurate fully automatic algorithm." *Applied optics*, 24(18), 3101-3105.

O'Donovan, P. C., Burton, D. R., and Lalor, M. J. (1990). "Fourier analysis of partial field fringe patterns." *Proceedings applied optics and opto-electronics FASIG/IOP*, Nottingam, UK, 325-326.

Oreb, B. F. (1996) "A quest for the elusive phase-shifting algorithm." *Proceedings applied optics and opto-electronics FASIG/IOP*, Reading, U.K., 197-204.

Perrin, J. C., and Thomas, A. (1979). "Electronic processing of moiré fringes: application to moiré topography and comparison with photogrammetry." *Applied optics*, 18, 563-574.

Platzner, M., Steger, C., and Weiss, R. (1992). "Parallele Digitale Signalprozessoren." Technical report no. *92/01*, Institut für Technische Informatik, Graz.

Platzner, M., Steger, C., and Weiss, R. (1993) "Performance measurements on a multi-DSP architecture with TMS320C40." *Proc. of the 4th international conference on signal processing. Applications and tech.*, vol. 2, UK, 1144-52

Polhemus, C. (1973). "Two-wavelength interferometry." *Applied optics*, 12(9), 2071-2074.

Quiroga, J. A., and Bernabeu, E. (1994). "Phase-unwrapping algorithm for noisy phase-map processing." *Applied optics*, 33(29), 6725-6731.

Quiroga, J. A., Gonzâlez-Cano, A., and Bernabeu, E. (1995). "Phase-unwrapping algorithm based on an adaptive criterion." *Applied Optics*, 34(14), 2560-2563.

Ragsdale, S. (1994). *Parallel Programming*, McGraw-Hill, New York. ISBN 0070511861.

Reid, G. T. (1983). "Surface form measurement by heterodyne moiré contouring." *Proc. SPIE*, 376, 8-14.

Reid, G. T. (1986). "Automatic fringe pattern analysis: a review." *Optics and lasers in engineering*, 7, 37-68.

Robinson, D. W., and Williams, D. C. (1987). "Digital phase stepping speckle interferometry." *Optics communications*, 57(1), 26-29.

Roddier, C., and Roddier, F. (1987). "Interferogram analysis using Fourier transform techniques." *Applied Optics*, 26, 1668-73.

Schörner, J., Ettemeyer, A., Neupert, U., Rottenkolber, H., Winter, C., and Obermeier, P. (1991). "New approaches in interpreting holographic images." *Optics and Lasers in Engineering*, 14, 283-91.

Schwider, J., Falkendtorfer, O., Schreiber, H., Zoller, A., and Streibi, N. (1993). "New compensating four-phase algorithm for phase shift interferometry." *Optical engineering*, 32(8), 1883-1885.

Sedgewick, R. (1988). Algorithms, Addison-Wesley, 2nd edition, New York, 115-30. ISBN 0201066734.

Servin, M., Rodriguez-Vera, R., and Moore, A. J. (1994). "A robust cellular processor for phase unwrapping." *Journal of modern optics*, 41(1), 119-127.

Spik, A., and Robinson, D. W. (1991). "Investigation of the cellular automata method for phase unwrapping and its implementation on an array processor." *Optics and lasers in engineering*(14), 25-37.

Srinivasan, V., Liu, H. C., and Halioua, M. (1984). "Automated phase measurement profilometry of 3-D diffuse objects." *Applied optics*, 23, 3105-3108.

Stephenson, P. R. (1994). "Evaluation and solutions of key problems in Fourier fringe analysis." , PhD Thesis, John Moores University, Liverpool.

Stephenson, P. R., Burton, D. R., and Lalor, M. J. (1994). "Data Validation Techniques in a Tiled Phase Unwrapping Algorithm." *Optical engineering*, 33, 3703-3708.

Sternberg, S. R. (1983). "Biomedical image processing." *Computer*, 16, 22-34.

Sun, X.-H., and Gustafson, J. L. (1991). "Toward a better parallel performance metric." *Parallel computing*, 17, 1093-1109.

Takasski, H. (1970). "Moiré topography." *Applied optics*, 9(6), 1457-1472.

Takasski, H. (1973). "Moiré topography." *Applied optics*, 12(4), 845-850.

Takeda, M. (1993). "Phase unwrapping by neural network", *Proc. Fringe'93*, akademie Verlay.

Takeda, M., Ina, H., and Kobayashi, S. (1982). "Fourier-transform method of fringe-pattern analysis for computer-based topography and interferometry." *J. Opt. Soc. Am.*, 72(1), 156-160.

Takeda M., and Kitoh, M. (1992) "Spatiotemporal frequency multiplex heterodyne interferometry." *J. opt. soc. Am.* A, 9, 1607-1614

Takeda, M., and Mutoh, K. (1983). "Fourier-transform profilometry for the automatic measurement of 3-D object shapes." *Applied optics*, 22(24), 3977-3982.

TEXAS INSTRUMENTS. (1994). *Parallel processing with the TMS320C4x. Application Guide*, Texas Instruments.

Thompson, D. A., Michell, G. A., Manson, G. A., and Brookes, G. R. (1990). *Inside the transputer*, Mackays of Chatham plc., Computer science texts series. ISBN 0632016892.

Tilford, C. R. (1977). "Analytical procedures for determining lengths from fractional fringes." *Applied optics*, 16(7), 1857-60.

Tipper, D. J. (1995). "A neural network approach to the phase unwrapping problem in fringe analysis." *Nondestr. test. eval.*, 12, 391-400

Towers, D. P., Judge, T. R., and Bryanston-Cross, P. J. (1989). "A quasi heterodyne holographic technique and automatic algorithms for phase unwrapping." *Proc. SPIE*, 1163, 95-119.

Towers, D. P., Judge, T. R., and Bryanston-Cross, P. J. (1991). "Automatic interferogram analysis techniques applied to quasi-heterodyne holography and ESP." *Optics and lasers in engineering*(14), 239-281.

Vrooman, H. A., and Maas, A. A. M. (1989) "Image processing in digital speckle interferometry", *Proceedings applied optics and opto-electronics FASIG/IOP fringe analysis 89*, Loughborough, UK

Vrooman, H. A., and Maas, A. A. M. (1989b). "New image processing algorithms for the analysis of speckle interference patterns." *SPIE*, 1163, 51-61.

Vrooman, H. A., and Maas, A. M. (1991). "Image processing algorithms for the analysis of phase-shifted speckle interference patterns." *Applied optics*, 30(13), 1636-1641.

Wilson, P. (1983). "OCCAM architecture eases system design." *Computer Design*, 11, 107-115.

Winter, D., Ritter, R., and Sadewasser, H. (1993). "Evaluation of modulo 2π phase-images with regional discontinuities: areas based unwrapping." *Proc. 2nd. int. workshop on automatic processing of fringe patterns*, 157-9.

Worley, P. T. (1990). "The effect of time constraints on scaled speedup." *SIAM Journal Science Stat. Comput.*, 11, 838-858.

Yatagai, T., and Idesawa, M. (1982). "Automatic fringe analysis for moiré topography." *Opt. and lasers in engineering*, 3, 73-83.

Yatagai, T., Inaba, S., Nakano, H., and Suzuki, M. (1984). "Automatic flatness tester for very large scale integrated circuit wafers." *Opt. Engineering*, 23, 401-405.

Yokozeky, S., and Suzuki, T. (1972). "Modified double-beam interferometer using the moiré method." *Applied optics*, 11(2), 446-448.

Zhao, H., Chen, W., and Tan, Y. (1994). "Phase-unwrapping algorithm for the measurement of three-dimensional object shapes." *Applied optics*, 33(20), 4497-4500.

APPENDIX

PAPERS BY THE AUTHOR

Arevalillo Herráez, M., Burton, D.R., Lalor, M.J., Clegg, D.B., (1996) "Robust, Simple and Fast Algorithm for Phase Unwrapping", Applied Optics, Vol. 35, No. 29

Arevalillo Herráez, M., Burton, D.R., Lalor, M.J., Clegg, D.B., (1996) "Robust Unwrapper for Two Dimensional Images" , Proc. Laser Optics Vision for Productivity in Manufacturing. Proc. SPIE. 2784 106-110. Besançon. June

Arevalillo Herráez, M., Burton, D.R., Lalor, M.J., (1996) "Phase Wrap Reduction Using Fringe Pattern Projection", Appl. Opt. Div. Conf., Reading, 16-19 September, ISBN 0-7503-0382-4

Arevalillo Herráez, M., Burton, D.R., Lalor, M.J., Clegg, D.B., (1996) "Robust, Simple and Fast Algorithm for Phase Unwrapping", Applied Optics, Vol. 35, No. 29, 5847-5852

Robust, simple, and fast algorithm for phase unwrapping

Miguel Arevalillo Herráez, David R. Burton, Michael J. Lalor, and David B. Clegg

Phase unwrapping has been and still is a cumbersome concern that involves the resolution of several different problems. When dealing with two-dimensional phase unwrapping in fringe analysis, the final objective is, in many cases, the realization of that analysis in real time. Many algorithms have been developed to carry out the unwrapping process, with some giving satisfactory results even when high levels of noise are present in the image. However, these algorithms are often time consuming and far removed from the goal of real-time fringe analysis. A new approach to the construction of a simple and fast algorithm for two-dimensional unwrapping that has considerable potential for parallel implementation is presented. © 1996 Optical Society of America

1. Introduction

Phase unwrapping is not a *sui generis* concern to fringe analysis. Because it is present in many different fields, it has been the object of research for a long time, and many different algorithms have been developed for its resolution. Different orientations and approaches have been tried, from simple algorithms, such as those developed by Oppenheim and Schafer,[1] to complex and effective techniques, such as the cellular automata developed by Ghiglia *et al.*[2] Unfortunately, speed and accuracy are not always compatible, and reliable unwrappers can lead to time-consuming processing.

Another important issue regarding these algorithms is the propagation of errors. This issue was the main reason why unwrappers based on tiles[3] or regions[4] were developed. Such unwrappers are based on the partitioning of the image into smaller areas, with these being independently unwrapped. Thus errors are limited to relatively small areas and propagation of errors beyond tile boundaries is avoided.

Different approaches include those tried by Huntley and co-workers.[5,6] The first of these approaches[5] was based on the detection of discontinuities and an attempt to join them with cut lines to look for the shortest possible paths between the points marking the discontinuities. The second algorithm, developed by Huntley and Soldner,[6] was based on recording several time-lapsed fringe patterns. Unwrapping then proceeds along the time axis by the use of the information contained in all the patterns. However, neither algorithm was free of problems: The first had the disadvantage of a time delay incurred when high levels of discontinuities were present in the image: the second suffered from the number of images required.

Ghiglia *et al.*[7] developed yet another set of unwrappers based on the concept of least-squares unwrapping and the use of fast cosine transforms. If no additional information is available (e.g., localization of bad areas), the unweighted unwrapper can lead to deviations of the wraps that translate into local errors in the final unwrapped phase map. If *a priori* information is available, hundreds of iterations can lead to very effective unwrapping (with a high penalty in time).

For a more extensive description and a review of existing phase-unwrapping techniques the reader is referred to Ref. 8. In our approach we attempt to reduce the time required to a minimum but at the same time to maintain a reasonable level of efficiency and reliability. It is important to notice that many aspects of the algorithm could be improved if an undesirable increase in the execution time were incurred. Those improvements could include an increase in the number of checks undertaken or a reduction in size of some masked areas. This potential is described in Section 3, following the description of the algorithm.

The authors are with Liverpool John Moores University, Byrom Street, L3 3AF Liverpool, United Kingdom: M. A. Herráez and D. B. Clegg are with the School of Computing and Mathematical Sciences and D. R. Burton and M. J. Lalor are with the School of Engineering and Technology.

Received 25 August 1995; revised manuscript received 8 April 1996.
0003-6935/96/295847-06$10.00/0

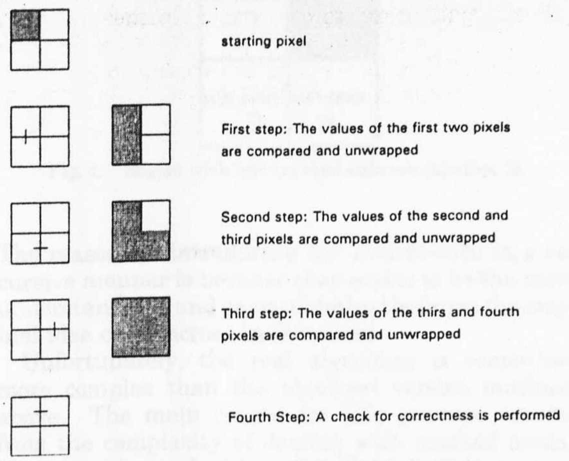

starting pixel
First step: The values of the first two pixels are compared and unwrapped
Second step: The values of the second and third pixels are compared and unwrapped
Third step: The values of the thirs and fourth pixels are compared and unwrapped
Fourth Step: A check for correctness is performed

Fig. 1. Unwrapping of a 2 × 2 pixel matrix.

Another important point to bear in mind is that we are not interested in recovering the whole image at any cost, but only those pixels that are error free. We consider that it is better to mask one valid point, hence to lose it, than to unwrap one point containing erroneous information, hence creating errors in the final result.

2. Presentation of the Algorithm

Divide and conquer is a programming technique that has been successfully applied to many problems in the past. An example is the well-known QUICK SORT algorithm, which considerably reduced the time required to sort an array of numbers.[9] The technique presented in this paper is based on divide and conquer, and it seems to behave in a very satisfactory way for most types of images. On the basis of two facts—that unwrapping a 2 × 2 pixel matrix is a simple concern and that two areas independently unwrapped can easily be connected—we build up a simple and effective algorithm for phase unwrapping.

Unwrapping a 2 × 2 pixel matrix proceeds in an anticlockwise direction by means of working out the difference between two adjacent points and comparing it with a previously set threshold value that in some literature is called the maximum tolerated value (usually π). If the absolute value of the calculated difference is greater than the tolerated value, 2π is added or subtracted from the furthest area in the clockwise direction, taking as the reference point, the starting point. If the addition of the differences applied around the 2 × 2 matrix differs from zero, the whole matrix will be masked. Figure 1 illustrates this process. The small lines crossing some pixels represent the calculation of the differences, whereas the gray pixels are those that have already been unwrapped.

If we are able to unwrap 2 × 2 pixel areas in a simple and effective way, why is it so difficult to unwrap a larger image? Any image can be subdivided into four sections (Fig. 2), which leads to the individual unwrapping of each of those sections. After unwrapping occurs, these sections can be considered in the same way as are single points and can be linked together in a process similar to that explained above for single pixels, to yield an unwrapped version from the initial image.

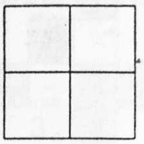


Fig. 2. Image subdivided into four parts. Each part is independently unwrapped.

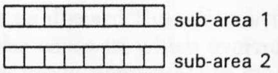


Fig. 3. Pixels at the junction of two sections.

The obvious question that immediately arises from the image-splitting process is how to deal with the junctions. Let us suppose that we have two sections whose junction is composed of n points, as illustrated in Fig. 3. For each point in the function, the number of 2π differences that must be added to the second subarea, so that the difference between the two points is lower than a tolerated value, needs to be worked out. The most common difference (that with which more points agree) is added to every point in the second subarea. Both areas will be masked if the appearance rate of that common difference is not higher than a certain percentage of the checked points.

The process followed to unwrap the whole area is similar to that for unwrapping the 2 × 2 matrix. After the subareas have been unwrapped, they are joined together, and the subareas are dealt with as if they were single points. Hence the area will be totally masked for the case in which the addition of the differences is different from zero.

If we can divide the whole image into four parts and perform the unwrapping process independently in each of them, there is no reason why we cannot perform the same decomposition process with each of those parts, giving place to a recursive decomposition and to construction of the image.

We have described how the decomposition and junction are performed, providing the basis for understanding the process. Now a simple version of the algorithm can be outlined:

```
(1) UNWRAP_REGION (region) /*main procedure, entry point*/
    (a) IF region is larger than 2 × 2 pixels THEN
        (i) DIVIDE region into four parts: A, B, C, D
        (ii) UNWRAP_REGION(A)
        (iii) UNWRAP_REGION(B)
        (iv) UNWRAP_REGION(C)
        (v) UNWRAP_REGION(D)
        (vi) BRING together A, B, C, and D
    (b) ELSE
    (a) UNWRAP_2 × 2_MATRIX(region)
```

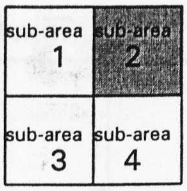


Fig. 4. Region with one masked subarea (number 2).

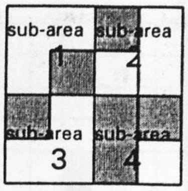


Fig. 7. Example of overmasking produced because the connections between subareas cannot be established.

The reason for introducing the pseudo-code in a recursive manner is because that seems to be the most understandable and easiest method because the original idea came across in this form.

Unfortunately, the real algorithm is somewhat more complex than the idealized version outlined above. The main reason for this increase results from the complexity of dealing with masked areas. If an area is masked because of the detection of an error, this area is no longer valid and should not be joined with the neighboring areas in a higher level of the recursion. As an example, let us assume that we have an area as shown in Fig. 4, where the dark color represents a masked subarea. In this case, an entire junction placed between subareas 1–2 and 2–4 has been masked. Because of this masking, it is impossible to complete the circuit of this area. Hence only the junctions between subareas 1–3 and 3–4 can be unwrapped. Figure 5 shows another case in which the junctions are completely masked and further unwrapping is not possible through them. However, this case reveals one of the problems we have to deal with: Two different regions can be unwrapped, and a decision as to which one it should be is required. The criteria used for deciding is to take the largest possible area. Figure 6 shows the two different regions that can be unwrapped.

For the case in which the four junctions are masked, the whole area will be masked because it is believed that the area contains a great deal of noise, and no information should be recovered from it. We could have avoided this overmasking, but the time overhead incurred made us discard this alternative. Figure 7 shows one of the cases in which overmasking is produced (the gray areas correspond to masked points).

The algorithm is based mainly on the image-partitioning concept. This concept directly implies multiple scan directions because initially a point is unwrapped by a comparative process with the two neighbors with which it shares a junction in a 2 × 2 matrix. It is compared with the remaining two orthogonal neighbors in a later stage of the algorithm (when the matrices are brought together). The gray area in Fig. 8(a) represents the position of a single point in a bigger area. Figure 8(b) shows how that point is compared with its neighbors in each stage. The lines crossing between points represent these comparisons.

Each point is clearly seen to be compared with each of its four neighbors with which it shares a junction. However, the comparison of a point with the third and fourth orthogonal neighbors depends on the sit-

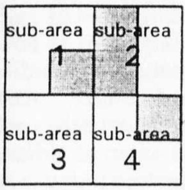


Fig. 5. Masked junctions. The shaded areas that border subarea adjacencies represent the masked junctions.

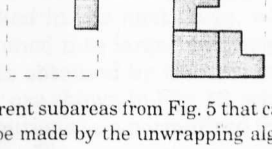


Fig. 6. Two different subareas from Fig. 5 that can be unwrapped. A decision must be made by the unwrapping algorithm.

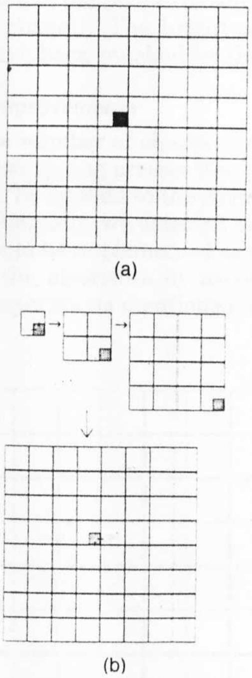


Fig. 8. Points within larger areas: (a) one pixel (shaded square) within a larger area, and (b) a large area that is being built up from its recursive components [the shaded square represents the pixel from (a)].

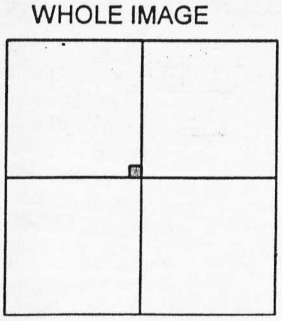


Fig. 9. Pixel (shaded square) placed between the first junctions to be decomposed.

uation of the point and, in some cases, can be performed when the last junction is carried out, i.e., when the initial four subareas are brought back together to form the whole image. However, this stage is too late for the detection of the error and could lead to the ultimate masking of the whole image or to errors in the final solution. This situation is shown in Fig. 9. The point placed between the initial junctions will not be compared with the third and fourth neighbors until the last stage of the unwrapping process.

In addition, the situation becomes complicated again when masked points are involved in the unwrapping process. Let us suppose that a fringe break exists, such as that shown in Fig. 10(a), in which the gray tones represent masked areas. Now let us suppose that this region is placed on the divisory line between the top and bottom parts of the image and that it is adjusted toward the left-hand side, as shown in Fig. 10(b). When the unwrapping process is undertaken, there will be no square at its left-hand side, and the comparison of subareas 2 and 3, which would normally detect the error, misses the masked points. When the error is detected (at the end of the unwrapping process when areas 1 and 2 of the whole image are joined together), it is too late. Either the whole image is masked or the error remains.

Because of this error-detection failure, a precomparator process had to be included in the algorithm. As soon as an area is independently unwrapped, a check for compatibility is undertaken before larger areas are dealt with. This test consists of a comparison of each subarea with its adjacent areas to ensure that it will be possible to make the junction cut at a later step without causing problems. If a junction is found to be in error, the subareas involving that junction are masked. Figure 11 shows those junctions that are to be examined. The squares represent the regions that have already been unwrapped. The rest of the junctions are not analyzed because they will be checked in the next stage, when the current subarea is joined into larger subareas.

The results obtained by the unwrapper presented in this paper are shown in Fig. 12, which displays the phase distributions in a normalized form by the use of a gray scale. The intensities vary from black (the minimum value of the phase) to white (the maximum value of the phase). The bright regions represent zones that have been masked by the unwrapper.

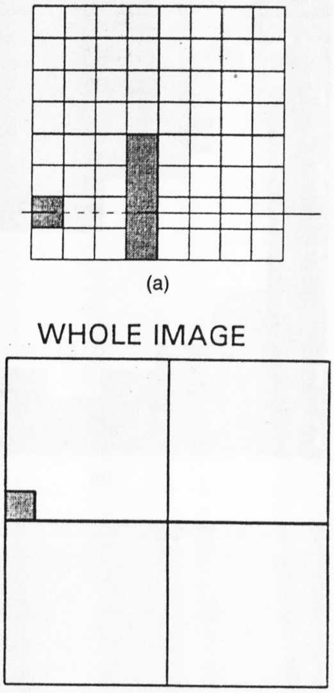


Fig. 10. (a) Representation of a fringe break within an area. The dashed part of the line represents the break. (b) Location of the fringe-break area in (a) within the whole image.

3. Possible Improvements

Increasing the number of checks that are carried out during the unwrapping process would allow many improvements to be applied to the current version of the algorithm. Basically, we detected two main improvements that could be implemented so as to increase the reliability of the algorithm by avoiding some of the overmasking operations mentioned above:

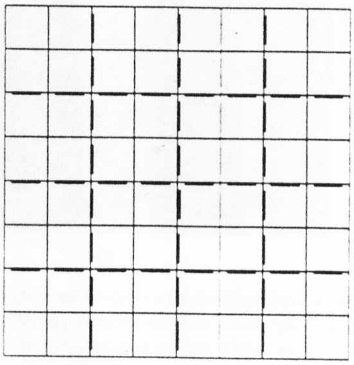
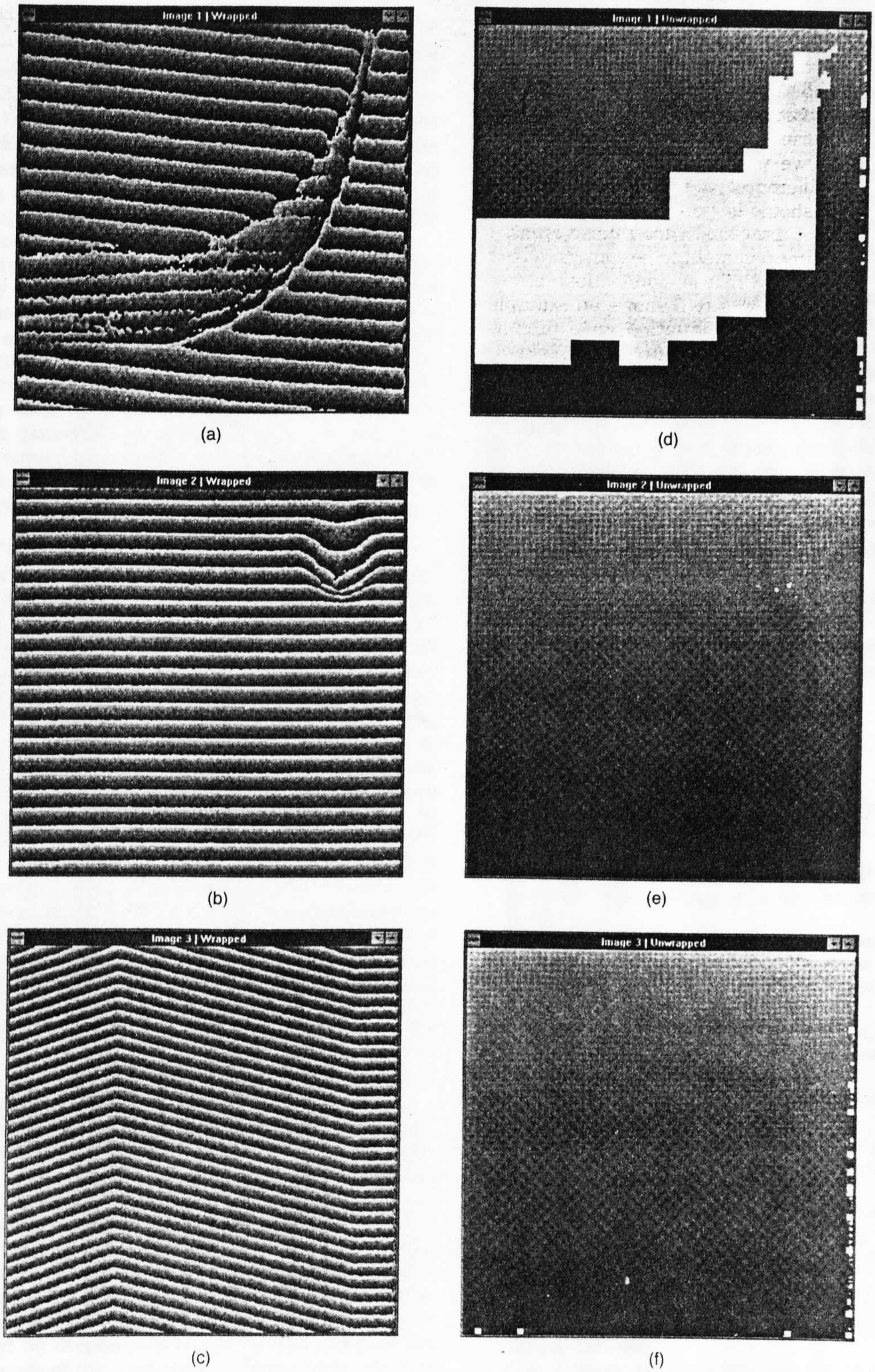


Fig. 11. Representation (boldface line segments) of the junctions that are to be checked.



(a) (d)

(b) (e)

(c) (f)

Fig. 12. Results obtained by the use of the unwrapping algorithm presented here: (a), (b), and (c) depict the wrapped phase distributions, whereas (d), (e), and (f) present their phase distributions, respectively, after the unwrapping process has been applied. The phase distributions are displayed in normalized form with a gray scale, in which black represents the minimum and white the maximum phase values.

- The first improvement to be considered is to check for compatibility, as defined in Section 2, during which we would try to detect errors before the subareas could grow and, consequently, to anticipate replacing the masking operation by a check similar to that undertaken for normal areas. In this anticipated replacement check we would test to determine if the addition of the differences were equal to zero. This testing operation should be performed on any possible set of four adjacent, previously unwrapped areas.
- The second improvement to be considered is the recovery of areas for which no connections between subareas was found. Figure 5 shows an example of such a situation. If this situation were produced, the second subarea would be completely masked unnecessarily, because no connection could be found to link the data between subareas. To reduce overmasking generated by the algorithm, we can carry out a recovery process to try to retrieve the data by examining the junctions with the rest of the image that surrounds the area.

4. Comments

We must state that the second suggested improvement listed in Section 3 to avoid overmasking when it is not required is time consuming. It is also a complicated issue if the implementation is intended to recover all the areas that could possibly be rescued from the overmasking problem. Nor was it our intention to complicate the algorithm when the results obtained were, to some extent, more than acceptable with respect to the data recovered and the correctness obtained for those data. We do not ignore, however, the presence of local errors produced by trapping valid areas, that is, whenever an area maintains a single connection with the rest of the image by only a single border. If that area contains a wrap of a size comparable with the size of the area inside, a situation that is rather improbable, this wrap would not be detected because no further checking is possible as a result of the loss of connection with the rest of the image. This problem could again be solved by a simple check of each of the areas linked by means of two or more connections to the rest of the image. This process is not very time consuming and could be worthwhile if a completely error-free unwrapping were required.

It is important to remark that, because the experiments aimed to show the validity of the algorithm, borders were not removed; nor were low-modulation tests (low-modulation points normally appear in low-visibility areas as rapid fluctuations of the values in the wrapped-phase distribution) performed. Thus, more errors need to be overcome by the unwrapper. The application of such techniques, whether in isolation or combined with validation tests such as those described by Stephenson *et al.*,[10] could improve the performance of the algorithm. These tests are particularly suited to the algorithm because they would function as a preprocessing operation, masking single points and consequently avoiding some overmasking.

The algorithm presented here, which is clearly recursive in nature, had to be implemented in an iterative form because the memory requirements for such a recursive implementation were of a high magnitude. Because of the nature of the algorithm, it is easily made parallel. Because each partition is independently unwrapped—except for the interaction produced for checking on the borders—no excessive communication is required until the junction process. Because of the fact that many parallel machines are capable of performing communications and calculations without significant degradation in performance, no communication overhead will be produced. Communication overhead is avoided because the pixels on the borders can be given priority when the differences are added such that, when the differences have been added to the whole area, the required data have already been transmitted among the different processors constituting the network. It is interesting to note that the parallelization is particularly good when the number of processors is a power of 4 because of the geometrical decomposition that takes place in our algorithm.

Another advantage of the algorithm presented in this paper is that the additional memory required for its execution, by which we mean memory additional to the requirements for the code and for the images themselves, is minimal.

References

1. A. Oppenheim and R. Schafer, *Digital Signal Processing* (Prentice-Hall, Englewood Cliffs, N.J., 1975), Chap. 10, pp. 507–511.
2. D. Ghiglia, G. Masting, and L. Romero, "Cellular-automata method for phase unwrapping," J. Opt. Soc. Am. A **4**, 267–280 (1987).
3. D. Towers, T. Judge, and P. J. Bryanston-Cross, "A quasi heterodyne holographic technique and automatic algorithms for phase unwrapping," in *Fringe Pattern Analysis*, Proc. SPIE **1163**, 95–119 (1989).
4. J. Geirlöff, "Phase unwrapping by regions," in *Current Developments in Optical Engineering II*, R. E. Fischer and W. J. Smith, eds., Proc. SPIE **818**, 2–9 (1987).
5. J. M. Huntley, "Noise-immune phase unwrapping algorithm," Appl. Opt. **28**, 3268–3270 (1989).
6. J. M. Huntley and H. Saldner, "Temporal phase unwrapping algorithm for automated interferogram analysis," Appl. Opt. **17**, 3047–3052 (1993).
7. D. C. Ghiglia and L. A. Romero, "Robust two-dimensional weighted and unweighted phase unwrapping that uses fast transforms and iterative methods," J. Opt. Soc. Am. A **11**, 107–117 (1994).
8. T. R. Judge and P. J. Bryanston-Cross, "A review of phase unwrapping techniques in fringe analysis," Opt. Lasers Eng. **21**, 199–239 (1994).
9. C. A. R. Hoare, "Quicksort," Comput. J. **5**, 10–15 (1962).
10. P. R. Stephenson, D. R. Burton, and M. J. Lalor, "Data validation techniques in a tiled phase unwrapping algorithm," Opt. Eng. **33**, 3703–3708 (1994).

Arevalillo Herráez, M., Burton, D.R., Lalor, M.J., Clegg, D.B., (1996) “Robust Unwrapper for Two Dimensional Images” , Proc. Laser Optics Vision for Productivity in Manufacturing. Proc. SPIE. 2784 106-110. Besançon. June

Robust unwrapper for two dimensional images

Miguel Arevalillo Herráez[†], David R. Burton[‡], Michael J. Lalor[‡], David B. Clegg[†]
Liverpool John Moores University
[†]School of Computing and Mathematical Sciences
[‡]School of Engineering and Technology Management
Byrom Street, L3 3AF, Liverpool, UK

ABSTRACT

The unwrapping of a two dimensional image in a reasonable time is still an unsolved question. Many algorithms have been proposed in the past, but all of them failed to some extent: Either they do not meet the requirements for accuracy or the amount of time required for execution makes them unacceptable because of the time constraints involved in certain kinds of problem. Towers, amongst others, introduced the concept of tiled unwrapping, obtaining good results in a reasonable amount of time. From the idea of dividing the image into tiles, an algorithm for two-dimensional phase unwrapping that allows the achievement of high precision in an acceptable time is proposed.

1. INTRODUCTION

Towers unwrapper[1] is based on the division of the image into smaller parts or tiles that are to be unwrapped independently and joined together at a subsequently stage

The advantage of this method is that whenever a serious error is detected on a single tile then the tile is masked and the propagation of the error though the rest of the image is avoided. However, to mask the whole tile means to completely lose all the information contained in it, hence sacrificing some useful data. This could be considered acceptable when the size of the tile is relatively small, but if we deal with bigger tiles then the masking of a single one of those tiles could mean the masking of an important percentage of the whole image. On the other hand, this algorithm is able to detect discontinuities in the wraps as long as they are smaller than the size of the tile. Thus, the choice of a small size can lead to discontinuities not being detected whilst the choice of a greater size will lead to over-masking.

A solution to this problem could be to execute the algorithm with several tile sizes on the same image. For instance, we could execute the algorithm for a tile size of four and, if the result is incorrect (differences greater than 2π remain on the image) then apply the algorithm with a tile size of 8 on the initial image, and so on until the optimum size for the tile is found appropriate to the image. However, this process involves the execution of the whole algorithm for each tile size that is to be tried, consequently leading to an unacceptable increase in the time for the resolution of the problem. It also can happen that the best tile size varies along the image.

Obviously, the approach described above is not attractive at all, and does not guarantee that a solution will be found, since it is possible that there is no size leading to an acceptable balance between the over-masking produced and the detection of discontinuities in the wraps (The minimum size for the tile that guarantees the detection of all the discontinuities existing on the image could produce an unacceptable overmasking).

Fortunately, this is not the only procedure that can be applied in order to obtain a correctly unwrapped solution without producing excessive overmasking The idea of the algorithm presented in this paper is to combine the benefits of both small and large tiles whilst not incurring the drawbacks previously outlined.

2. ALGORITHM

Basically, the problem involved with the tiles unwrapper was the loss of the view of the image as a whole. When an image is split into several parts, whether they are regions, tiles or of any other form, by performing an independent unwrapping process on each one of them, the global view of the image will be lost. This does not mean, of course, that the image should not be split and that we have to deal with the whole image as a single entity. On the contrary, the division of an image will simplify the process since the unwrapping of small areas is, obviously, an easier operation. Nevertheless, whenever a partition of an image is undertaken, a method to recover the view of the image as a whole has to be developed.

In order to avoid over-masking, the first step was to divide the image into small tiles, proceeding then to the unwrapping of each of these areas independently.

This first operation is, basically, the application of Tower's unwrappers for a small tile size, except that the areas are not brought together yet, and does not solve the problem of discontinuities in the wraps in the case in which those breaks are greater than the size of the tile.

This issue is usually dealt with by using tiles of a greater size that the wrap discontinuities. Although the choice of the tile size has already been carried out and we have chosen a small size, it is interesting to note that, in the same manner that a tile can be decomposed in order to produce smaller areas, tiles can be grouped together so as to make up tiles of a greater size.

Once the image has been split into tiles and these tiles have been unwrapped independently, they can be dealt with as if they were single points, applying the tiled unwrapper again and forming bigger tiles. Those tiles will be generalised again as if they were single points and so on until a single tile remains, at which point the whole image has been unwrapped. If the size of the initial tile is n * m, the tile size in the second passing of the algorithm will be (n * m) (n * m) = $n^2 * m^{2\cdot}$, in the third passing $n^3 * m^3$ and so on, being in the x passing n^x x m^x . It is because tiles are to be treated as if they were single points in a subsequent stage that we use the terms point and sub-tile or sub-area as synonyms in the rest of this paper.

The tiled unwrapping itself will not be dealt with, since there could be many different methods to handle that problem and all of them are perfectly valid as long as they ensure that were the tile a whole image, the unwrapping procedure would lead to a correct unwrapping of the image.

One of the questions of concern is how to deal with small areas in the same way as when dealing with single pixels. In order to perform this operation, it is necessary to analyse a junction (border shared by two areas as illustrated in figure 1) instead of a point.

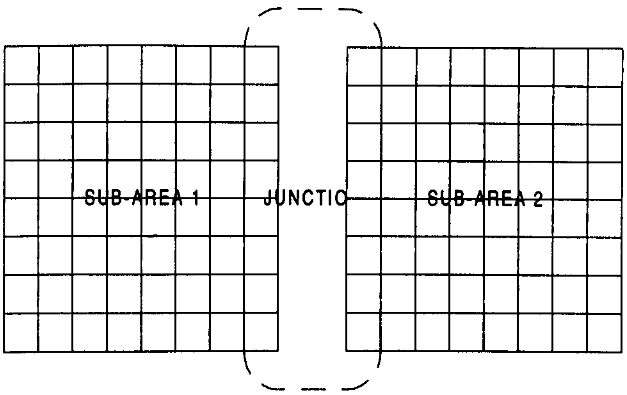


Figure 1. Junction between two tiles

Consider two tiles whose junction in composed of n points. For every point in the junction, the number of 2π differences that are required to be added to the point on the second tile placed just opposite to it so that the difference between the two points is lower than a tolerance value has to be worked out. After this, a testing process is undertaken checking that the value found is the same for every point or sub-area on the junction. If it is not, several approaches could be considered. For example both tiles could be completely masked; a small percentage of errors may be considered valid - masking those points where the errors are produced-, a solution which could help avoiding overmasking areas; or, finally, taking as a reference the most common difference, whenever the difference between two points on the junctions differs from that calculated, those two points -or sub-tiles in the case- are to be masked .

The algorithm is easily understandable when there are not erroneous points on the image. However, when errors do exist, the algorithm complicates because of the existence of masked areas. This will tend, to some extent, to increase the complexity of the problem

Take for example a tile size of 4 x 4 and the situation illustrated in figure 2, where grey squares represent masked sub-areas and the tile marked with an x character is a sub-area with an error that should be detected when joining this tile to the rest of the tiles.

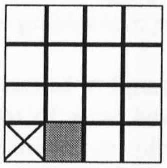


Figure 2. Special situation.

The sub-area with the error will be joined to the rest, with the error not detected since there is a single junction to compare (at least two junctions are required to determine an inconsistency).

Let us place this small area as illustrated in figure 3.

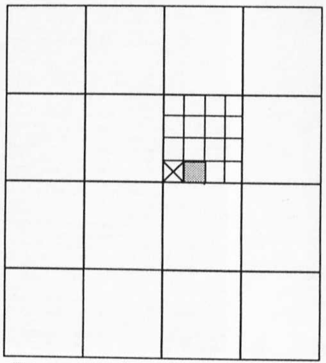


Figure 3. Previous situation in a larger area.

The next comparison with the rest of the neighbourhood is only performed at the end of the unwrapping process (when bringing the last 16 tiles together). The whole image could then be masked due to a small inconsistency.

This kind of problem takes place when a point or sub-area is only compared to a single neighbour. Therefore, whenever this situation occurs there is a potential danger of producing an erroneous unwrapping. These circumstances turn up most frequently on the borders of the tile, where points or sub-tiles have just two or three neighbours in contrast to those points located on the centre of the tile that are surrounded by four neighbours.

In order to counteract this drawback checks have to be performed not only between the points or areas inside the tile but also between the frontiers between tiles. This check will complete the neighbourhood of points or sub-tiles placed on the borders of the tiles so that every point will have been compared to four other points.

Nevertheless, if a sub-tile is erroneous and three of the sub-tiles belonging to its neighbourhood are masked, there is a possibility that an error is not detected, leading to a local error in that area. In order to correct this error a check could be undertaken with each subtile so as to ensure that at least two neighbours are available for comparison or the sub-tile is masked.

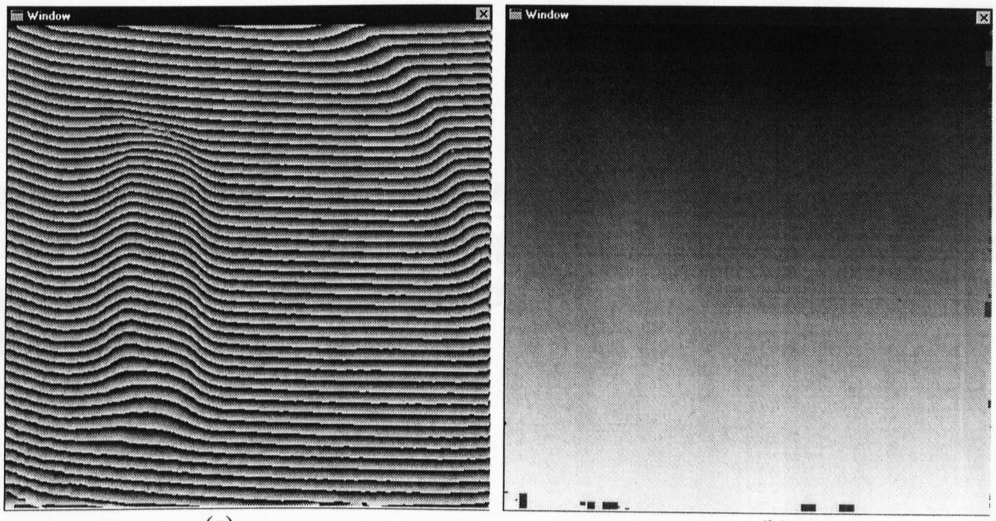
In the case in which errors are found on the frontiers between tiles, both tiles are to be masked in those areas where errors have been found.

3. RESULTS

In this section the performance of the algorithm is illustrated by applying it to two different images. The sampling of the images is 512 x 512 pixels.

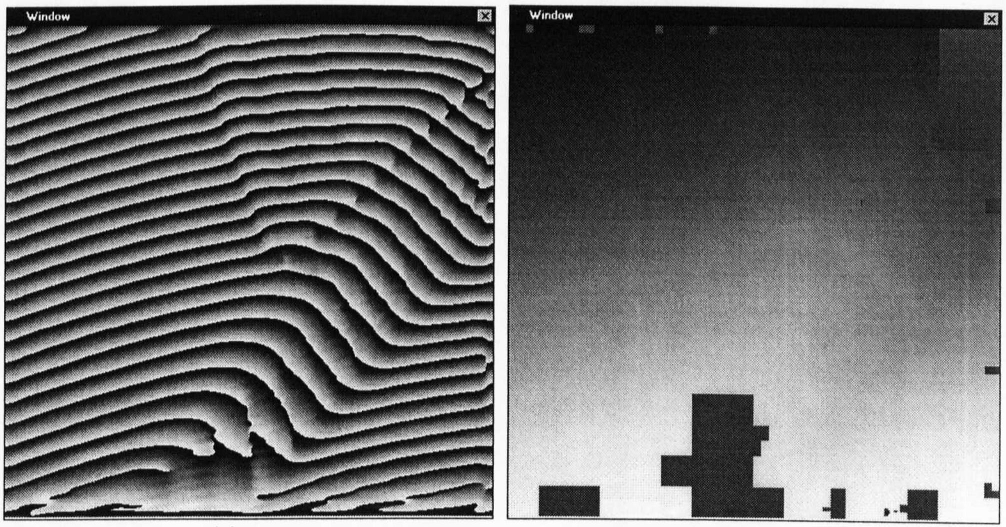
The first wrapped phase distribution that is to be analysed was obtained by the application of the FFA method to measure a car door handle. The wrapped phase distribution is illustrated in figure 4a. The final unwrapped phase distribution can be seen in figure 4b. The results are very satisfactory in the sense that only small areas are masked where problematic points are localised.

The second wrapped phase distribution, illustrated in figure 5a, was obtained by the same technique as the previous example applied to a small part of a car headlight. Masked areas clearly correspond to uncertainties in the wraps (whether a wrap vanishes or splits into two). The results prove the technique to be effective and to overcome the wrap discontinuities problem.



(a) (b)

Figure 4. (a) wrapped phase distribution, (b) unwrapped phase distribution.



(a) (b)

Figure 5. (a) wrapped phase distribution, (b) unwrapped phase distribution.

4. REFERENCES

1. D.P. Towers, T. R. judge, P.J. Bryanston-Cross, "Automatic interferogram analysis techniques applied to quasi-heterodyne holography and ESP", *Optics and lasers in engineering*, 14, pp. 239-281, 1991

Arevalillo Herráez, M., Burton, D.R., Lalor, M.J., (1996) "Phase Wrap Reduction Using Fringe Pattern Projection", Appl. Opt. Div. Conf., Reading, 16-19 September, ISBN 0-7503-0382-4

Paper presented at Appl. Opt. Div. Conf., Reading, 16-19 September 1996
(c) 1996 IOP Publishing Ltd

Phase wrap reduction using fringe pattern projection

Miguel Arevalillo Herráez[†], **David R. Burton**[‡] **and Michael J. Lalor**[‡]

Liverpool John Moores University, [†]School of Computing and Mathematical Sciences, [‡]School of Engineering and Technology Management, Byrom Street, L3 3AF, Liverpool, UK

Abstract This paper describes a technique that, by using two fringe patterns with different carrier frequencies can produce a height distribution over a larger range than that obtained by the use of a single fringe pattern. This decreases the number of wraps in the image, making the process of unwrapping simpler and providing detection of object discontinuities that are outside the scope of a single fringe pattern system

1. Introduction

As higher precision components are required, more accurate measurements are needed. Because of this, optical measurement is becoming essential as a technique for industrial inspection. Optical measurement techniques include fringe analysis, where a set of fringes are either created by interferometric methods or directly projected onto the objects surface. The analysis of these patterns results in a high resolution measurement of the object surface.

Nowadays, there are two main techniques that can be utilised to perform the fringe analysis process, and a high research effort has been put into improving them. The first technique, called phase stepping, was proposed by Carré[1]. Several variations have been proposed. The technique makes use of at least three fringe patterns.

The second technique, called Fourier Fringe Analysis (FFA), was introduced by Takeda et al. [2,3]. The technique proceeds by introducing a carrier frequency. This techniques utilises a single fringe pattern. This technique has also been studied in [4,5].

A common problem to both techniques is the unwrapping. Because the phase of the object is obtained by means of the arctan function, the result is limited to the intervale $]-\pi, +\pi[$. Therefore, the phase of the object is obtained modulo 2π. The process of obtaining the desired continuous funtion from the modulo 2π phase distribution is called unwrapping. In an ideal world, this is a simple process, but many difficulties arise from a real fringe pattern. Noise, real discontinuities in the objects surface, areas of invalid data, shadows, etc. make phase unwrapping a cumbersome concern. In fact, it is normally the most time-consumming operation in the fringe analysis process and a correct solution is not guaranteed.

Many unwrappers were developed in the past[6,7], but none of them were robust or fast enough to be used with total reliability. We propose a technique, based upon the fractional fringe analyis and synthetic fringe theory, that can potentially remove the need for unwrapping in many situations or, at least, reduce the number of wraps present in the image.

2. Technique

We explain the technique for a single dimension and for Fourier fringe analysis. Its generalisation to the two dimensional problem is straight forward.

Once the phase of the object is obtained by Fourier fringe analysis, it is converted to a height distribution, the equation that relates phase to height being given by

$$height(x) = c \cdot phase(x)$$

where c is the so called magnification factor that is given by:

$$\frac{1}{2\pi f \sin\theta}$$

where f is the carrier frequency and θ is the angle of the camera with respect to the projection system.

It is obvious that for a fixed object and camera to view angle, the lower the carrier frequency the lower the number of wraps in the phase distribution (since the same increment in height corresponds to a smaller increment in phase). However, when utilising the Fourier analysis technique, small carrier frequencies produce more errors since the noise peak and the phase peak lay closer to each other. Therefore, the use of a very small carrier frequency is not feasible.

The technique described in this paper, utilises two fringe patterns projected onto the same object with different carrier frequencies. Since the carriers are different, so are the magnification factors, and a new frequency of the difference between the two frequencies in the initial fringe patterns can be obtained. The process creates a synthetic fringe whose period is somewhat larger that the originals.

Since the phase is wrapped modulo 2π, if we do not perform the unwrapping process, the heights will be the wrapped modulo

$$\frac{1}{f \sin\theta}$$

This can be called the period of the signal.

The fractional fringe theory states that the fractional part of the new signal will be equal to the substraction of the fractional parts of the two initial signals, and the frequency of the new signal obtained is the substraction of the frequencies of the two initial fringe patterns.

This new fringe pattern will have a period given by

$$\frac{1}{(f_1 - f_2)\sin\theta},$$

that can be much larger than the period of the two original fringe patterns if the frequencies f_1 and f_2 are properly selected . If the whole object surface can be mapped into this period, no wraps will appear (it could happen to be a single wrap that can be removed by simply adding a constant and re-wrapping the heights distribution).

The fractional part of a signal can be expresed as the signal divided by the period. Consequently, the fractional part of the new signal can be expressed as

$$h_1(x)f_1\sin\theta - h_2(x)f_2\sin\theta$$

and the new height distribution will be given by

$$(h_1(x)f_1\sin\theta - h_2(x)f_2\sin\theta)\frac{1}{(f_1 - f_2)\sin\theta} \qquad (1)$$

where h_1 is the function representing the first height distribution, h_2 is the function representing the second height distribution, f_1 and f_2 are the carrier frequencies for the first and second fringe patterns respectively and θ is the angle of the camera respect to the projection system.

The technique clearly reduces the number of wraps by a factor of

$$r = \frac{f_1}{f_1 - f_2}$$

where f_1 is the carrier frequency of the higher of the carrier frequencies utilised in the original fringe patterns. The main problem with the technique is noise. Even if errors tend to compesate, the maximum error increases as does r. Therefore if a reduction of the number of wraps by a high factor is desired, noise will limit the application of the technique.

It is important to remark that the removal of the carrier frequencies before or after the application of the technique does not influence the results, but if carrier removal is performed after equation (1) is applied, the new carrier will be the substraction of the initial two carriers. For the technique to work, the carriers in the original two signals have to be oriented in the same direction. Otherwise, the calculation of the new carrier frequency will be somewhat more cumbersome and the reduction factor of the number of wraps will not be as great.

The main advantages of the technique described in this section are two:

- The number of wraps is potentially reduced.
- Heights that expand further than the highest period of the original signals can be detected.

3. Application of the technique

In this section, we present some simulations in order to illustrate how the technique performes. It is important to notice that the only source of errors in these images are those introduced by the filtering in the Fourier domain and no noise has been added.

The same simulation object has been utilised in the two examples illustrated in this section. The object is a tilted plane with a discontinuity as shown in figure 1, a one dimensional representation of an object slice. The first discontinuity in figure 1 has a height value that is greater than

$$\frac{1}{f \sin\theta},$$

f being the carrier frequency in the first fringe pattern. Therefore, the discontinuity is not quantitatively detected by an independent analysis of any of the two fringe patterns.

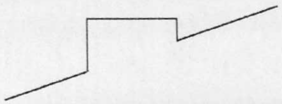
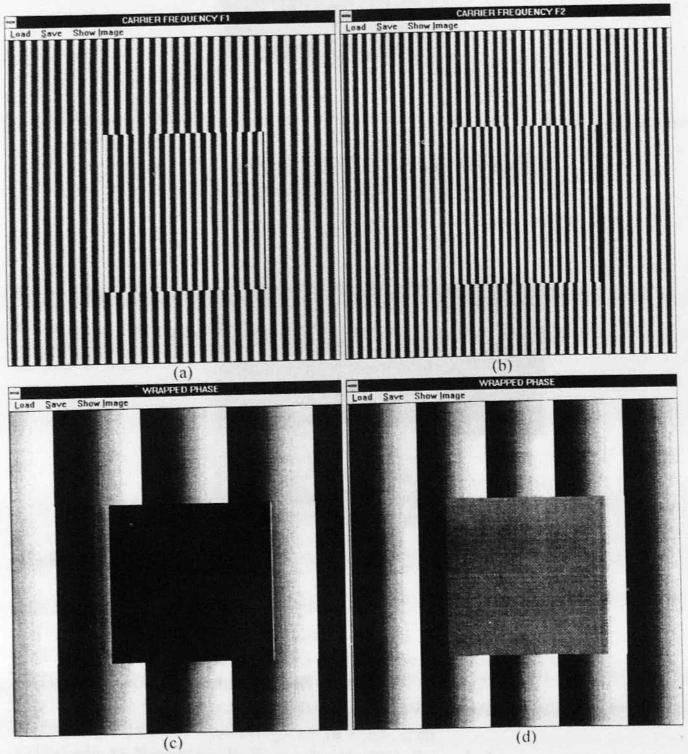


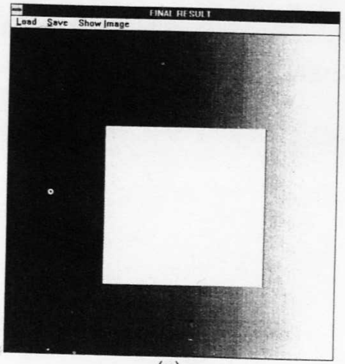
Figure 1. One dimensional representation of an object slide on the x-axis.

In the first example two images with carrier frequencies $f_1 = 40$ and $f_2 = 32$ have been used. Carrier removal has been carried out after the phase calculation by substracting it and rewrapping the resulting phase distribution. Figures 2(a) and 2(b) represent the original fringe patterns. Figures 2(c) and 2(d) represent the heights after carrier removal and, finally, Figure 2(e) represent the final result where no wraps are present.

In the second example, the carrier is not removed and two images with carrier frequencies $f_1 = 33$ and $f_2 = 32$ have been used. Figures 3(a) and 3(b) represent the original fringe patterns. Figures 2(c) and 2(d) represent the heights with the carrier removal and, finally, Figure 2(e) represent the final result where no wraps are present. The reduction factor in this example would be too high to work in a noisy real image. We only try to illustrate the potential of the technique. In a real image, the approach adopted in the first example will lead to better results.

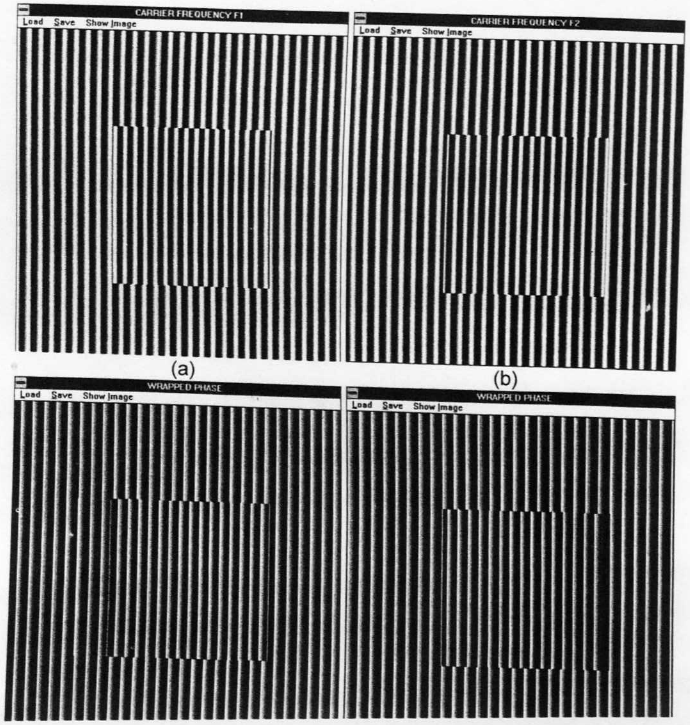
As we stated at the beginning of the section an independent analysis of a single fringe pattern will not detect the correct value of the discontinuity and a unwrapped images as that shown in figure 4 will be obtained.

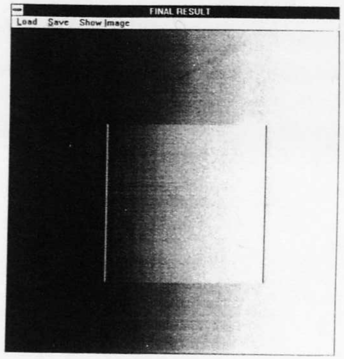




(e)

Figure 2. (a), (b) original fringe patterns. (c), (d) wrapped heights distributions, (e) final result,





(c)

Figure 1. (a), (b) original fringe patterns. (c), (d) wrapped heights distributions, (c) final result

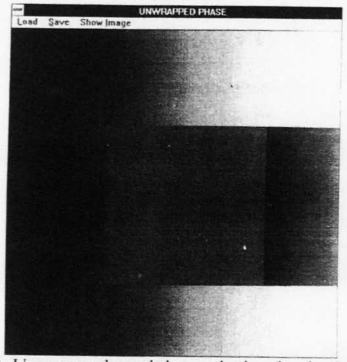


Figure 2. Unwrapped result by analysis of a single image.

References

[1]Carré P 1966 *Metrologia* 13-23
[2]Takeda M and Mutoh K 1983 *Applied optics* **22** 3977-3982
[3]Takeda M Ina I and Kobayashi S 1982 *J. Opt. Soc. Am.* **72** 156-160
[4]Burton D R and Lalor M J 1988 *Proc. SPIE* **1010** 17-24
[5]O'Donovan P C Lalor M J and Burton D R 1990 *Applied optics* 325-326
[6]Huntley J M 1989 *Applied optics* **28** 3268-3270
[7]Takeda M Nagatome K Watanabe Y 1993 *Proc. 2nd int. workshop on automatic processing of fringe patterns* 136-141.

“I know you think you understood what I said, but I’m not sure I said what you think I meant”

Anonymous