

# Three-Dimensional Fourier Fringe Analysis and Phase Unwrapping

Hussein Abdul-Rahman

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

General Engineering Research Institute (GERI),  
Liverpool John Moores University.

June 2007

**“Allah will exalt in degree those of you who believe and those who have been granted knowledge. And Allah is well-acquainted with what you do”**

*Holy Quran*

*Chapter 58. Verse 11*

**The Prophet Mohammad, peace be upon him, Said “Whoever takes a path in search for knowledge, Allah will facilitate for him a path to Paradise”**

*Prophet Teachings*



*Dedicated to:*  
*My beloved parents*

## Abstract

### Three-Dimensional Fourier Fringe Analysis and Phase Unwrapping

Hussein Sudqi Abdul-Rahman

Ph.D. Thesis

For many years the two-dimensional Fourier Fringe Analysis (FFA) technique has been regarded as a fast and reliable technique for the analysis of fringe patterns projected onto static objects. Today, two-dimensional FFA is seen as a fast and flexible method for processing fringe patterns of a dynamic object. But it is still inherently a two-dimensional approach, *i.e.* it deals with three-dimensional data (a video sequence) on the basis of regarding the data-set as a series of individual 2D images. The analysis of each 2D image is performed completely independently, no information from the previous images, or following images, is available at the time of processing the current image.

In the case of dynamic objects, we need new powerful techniques capable of processing the whole video sequence at once, instead of seeing it as a series of disconnected two-dimensional images. Regarding the data as a single unit means involving all variations of the fringe pattern, in space and time, in the processing procedure. In other words, three-dimensional processing may give us great potential benefits by taking into account the time variation of the fringe pattern, which was previously ignored.

The extension of the two-dimensional FFA technique into three-dimensions requires the extension of the current two-dimensional phase unwrapping algorithms into three-dimensional form. Phase unwrapping can be simply defined to be the process of solving the ambiguity problem caused by the fact that the absolute phase is typically wrapped into the interval  $(-\pi, \pi)$ . Phase unwrapping is considered to be a real challenge in fringe pattern analysis and many other applications, even in its two-dimensional form.

In this thesis, a novel three-dimensional FFA system has been implemented and used for demodulating fringe pattern sequences of dynamic objects. In addition, this thesis illustrates two novel three-dimensional phase unwrapping algorithms. The first algorithm attempts to find the best unwrapping path in the three-dimensional wrapped-phase volume. The second algorithm follows a best path approach to unwrap the phase volume, but takes into account the effect of singularity loops. Singularity loops are defined here as the source of noise that must be avoided during unwrapping. The two different algorithms have been tested on both simulated and real objects. The results show outstanding performance for these algorithms when unwrapping fringe volumes with very high levels of noise. This thesis also compares the performance of the proposed algorithms with other existing two-dimensional and three-dimensional phase unwrapping algorithms.

## Acknowledgement

I would like to thank Doctor Munther Gdeisat for the intellectual support, encouragement, continuing guidance and motivation, modesty, friendship and good humour throughout, which made this thesis possible, and for his patience in correcting both my stylistic and scientific errors. Thanks Munther.

I would like to express my deep gratitude to Professor Michael Lalor and Professor David Burton for their professional guidance, inspiration, invaluable advice, patience, friendly supervision, constant support and encouragement throughout the different stages of the project. I also wish to thank Doctor Francis Lilley for his support and help in editing and correcting this thesis. Thanks for every thing.

I am also grateful to Bashar Rajoub, Abdulbasit Abid, Salah Karout, Mahmoud Al-Ghreify, Mohammad AlS'ad and all other colleagues for their advice, interesting discussions and helpful suggestions and for the good time I spent with them.

A special thank to my parents for their endless support throughout my studies. Without their constant assurance and assistance, completion of this project would have not been possible. Their enthusiasm, wisdom, drive, strength of character, tenacity and determination have inspired me to carry out this research work. And as a sign of my love, gratitude and affection I dedicate this work to them. I want to thank my lovely wife for her support and motivation in the completion of my PhD degree. Also, I would like to thank all of my brothers and sisters for their help and encouragement.

Finally, I acknowledge Doctor Mohammad Bataineh for his support and help during the early stages of this research.

Thank you all.

Hussein Abdul-Rahman

Table of Contents

Abstract..... i

Acknowledgement..... ii

Table of Contents..... iii

1. Introduction..... 1

    1.1. Introduction..... 2

    1.2. Fringe Projection and Analysis..... 4

    1.3. Phase Unwrapping and its Applications..... 8

    1.4. Research Objective..... 10

    1.5. Contributions..... 10

    1.6. Thesis Structure..... 11

    References..... 13

2. Fringe Pattern Analysis Techniques..... 16

    2.1. Introduction..... 17

    2.2. Direct Phase Demodulation..... 17

    2.3. Phase Stepping..... 23

    2.4. Fourier Transform Fringe Analysis..... 28

    2.5 Wavelet Transform Fringe Analysis..... 35

    2.6. Summary..... 38

    References..... 39

3. A Review of Two-Dimensional Phase Unwrapping Algorithms... 42

    3.1. Introduction..... 43

    3.2. Local Phase Unwrapping Algorithms..... 44

        3.2.1. Quality-Guided Phase Unwrapping Algorithms..... 48



3.2.2. Two-Dimensional Quality Maps.....	49
3.2.3. Residue-Balancing Phase Unwrapping Algorithms.....	52
3.2.4. Other Local Phase Unwrapping Algorithms.....	57
3.3. Global Phase Unwrapping Algorithms.....	57
3.3.1. Unweighted Least-Squares.....	59
3.3.2. Weighted Least-Squares.....	61
3.3.3 $L^p$ -Norm Method.....	63
3.3.4. Other Global Phase Unwrapping Methods.....	64
3.4. Summary.....	65
References.....	66
 4. Three-Dimensional Fourier Fringe Analysis.....	 69
4.1. Introduction.....	70
4.2. Theoretical Background.....	72
4.3. Filtering in Frequency Domain.....	75
4.4. Computer Simulation Results.....	77
4.5. Experimental Results.....	87
4.6. Summary.....	101
References.....	103
 5. Three-Dimensional Phase Unwrapping Algorithms.....	 105
5.1. Introduction.....	106
5.2. A Review of the Existing Three-Dimensional Phase Unwrapping Algorithms.....	107
5.3. Three-Dimensional Best Path (BP) Phase Unwrapping Algorithm...	109
5.3.1. Three-Dimensional Quality Maps.....	109
5.3.2. Unwrapping Path.....	112
5.4. Three-Dimensional Best Path Avoiding Singularity Loops (BPASL) Phase Unwrapping Algorithm.....	117

5.4.1. Identifying Singularity Loops.....	119
5.4.2. Closing Partial Loops.....	123
5.4.3. Identifying Zero-Weighted Edges.....	126
5.4.4. Calculating Edges' Qualities.....	145
5.4.5. Best Path Unwrapping.....	145
5.5. Results and Comparisons.....	146
5.5.1. Simulated Results.....	146
5.5.2. Experimental Results.....	151
5.5.3. The Effect of the Quality Map on the Unwrapping Path.....	163
5.5.4. Comparisons Between Two-Dimensional and Three- Dimensional Phase Unwrapping Algorithms.....	165
5.6. Summary.....	177
References.....	180
6. Conclusion and Future Work.....	182
6.1. Conclusion.....	183
6.2. Recommendations for Future Work.....	186
References.....	190
Appendix A: The Simulated Objects Used in the Thesis.....	191
Appendix B: List of the Author's Publications.....	200

# *Chapter One*

## *Introduction*

# Chapter One

## Introduction

---

### 1.1 Introduction

Optical non-contact measurement methods are very effective techniques for measuring the height of an object without affecting its surface. Indeed, optical measurements play a much more important role today, in the field of metrology, than they ever did in the past. More precise optical equipment, very fast computers and reliable image processing software increase the total reliability and accuracy of the measurement system, making optical metrology the first choice across many engineering and scientific disciplines, such as, medical applications (Woisetschlager *et al.*, 1994; Lilley, 1999; Lilley *et al.*, 2000; Engelsman *et al.*, 2003; Moore *et al.*, 2003), engineering and industrial applications (Kujawinska and Sitnik, 2000; Haist and Tiziani, 2002; Leopold *et al.*, 2003; Quan *et al.*, 2003), robot and machine vision (Tao-Xian and Xianyu-Su, 2001; Chen and Li, 2003; Smith and Smith, 2003) and in many other applications.

One of the most effective non-contact measurement techniques involves the use of structured lighting patterns, which are projected onto the object's surface to obtain its 3D height. Perhaps the commonest form of structured-light is that of projected fringe patterns. The different methods used to demodulate fringe patterns in order to calculate 3D object height are referred as **fringe pattern analysis techniques**. Several algorithms have been proposed to analyse these fringe patterns, including phase stepping techniques (Chan *et al.*, 1995), **Fourier fringe analysis (FFA)** (Takeda *et al.*, 1982; Bone *et al.*, 1986), direct phase detection (DPD) (Ichioka and Inuiya, 1972), wavelet transform fringe analysis (Dursun *et al.*, 2004; Abid *et al.*, 2006; Gdeisat *et al.*, 2006) and many other algorithms.



Takeda's original implementation of Fourier fringe analysis was one-dimensional, in that it only analysed a single line of the image at a time. However, it was not very long before the technique was extended into two-dimensions (Bone *et al.*, 1986; Gorecki, 1992) and, it has undergone considerable development over time (Burton and Lalor, 1989a; Burton and Lalor, 1989b; Burton and Lalor, 1994; Burton *et al.*, 1995; Su and Chen, 2001; Su *et al.*, 2001).

For many years the two-dimensional Fourier transform fringe analysis technique has been regarded as being a fast and reliable technique for the analysis of fringe patterns projected onto static objects. Today, two-dimensional Fourier fringe analysis is seen as a fast and flexible method for processing fringe patterns of a dynamic object. But it is still inherently a two-dimensional approach, *i.e.* it deals with three-dimensional data (a video sequence) on the basis of an unconnected series of individual two-dimensional images. The analysis of each image is completely independent, no information from the previous images, or following images, is available at the time of processing the current image. Consequently, the use of this technique to analyse a sequence of fringe patterns for a dynamic object does not utilise any relationship between consecutive fringe patterns.

In this thesis, the author proposes a novel approach to analyse dynamic fringe pattern video sequences. The proposed technique extends 2D-FFA into three dimensions. This new proposed algorithm analyses the sequence of fringe patterns together as an entire volume, not as a series of individual images. This has the advantage of making the algorithm more robust, accurate and suitable for the measurement of dynamic objects when compared to the 2D-FFA algorithm.

Any of the Fourier fringe analysis algorithms can be divided into two main processing stages: phase extraction and phase unwrapping. The first stage extracts the phase of a fringe pattern by using a Fourier transform and carrying out filtering in the frequency domain. The phase produced by the first stage is wrapped and is typically limited to the  $-\pi$  and  $\pi$  range. The wrapped phase contains  $2\pi$  jumps, which should be removed by using a phase unwrapping algorithm. This constitutes the second main stage in any FFA

algorithm. On that basis, the extension of the 2D-FFA into three dimensions requires the extension of both stages into 3D, as will be shown in this thesis.

The suitability of the algorithm for measuring dynamic objects could benefit many applications. For example, in medicine the 3D-FFA algorithm could be used for the dynamic monitoring of patients during radiotherapy treatment (Lilley *et al.*, 2000).

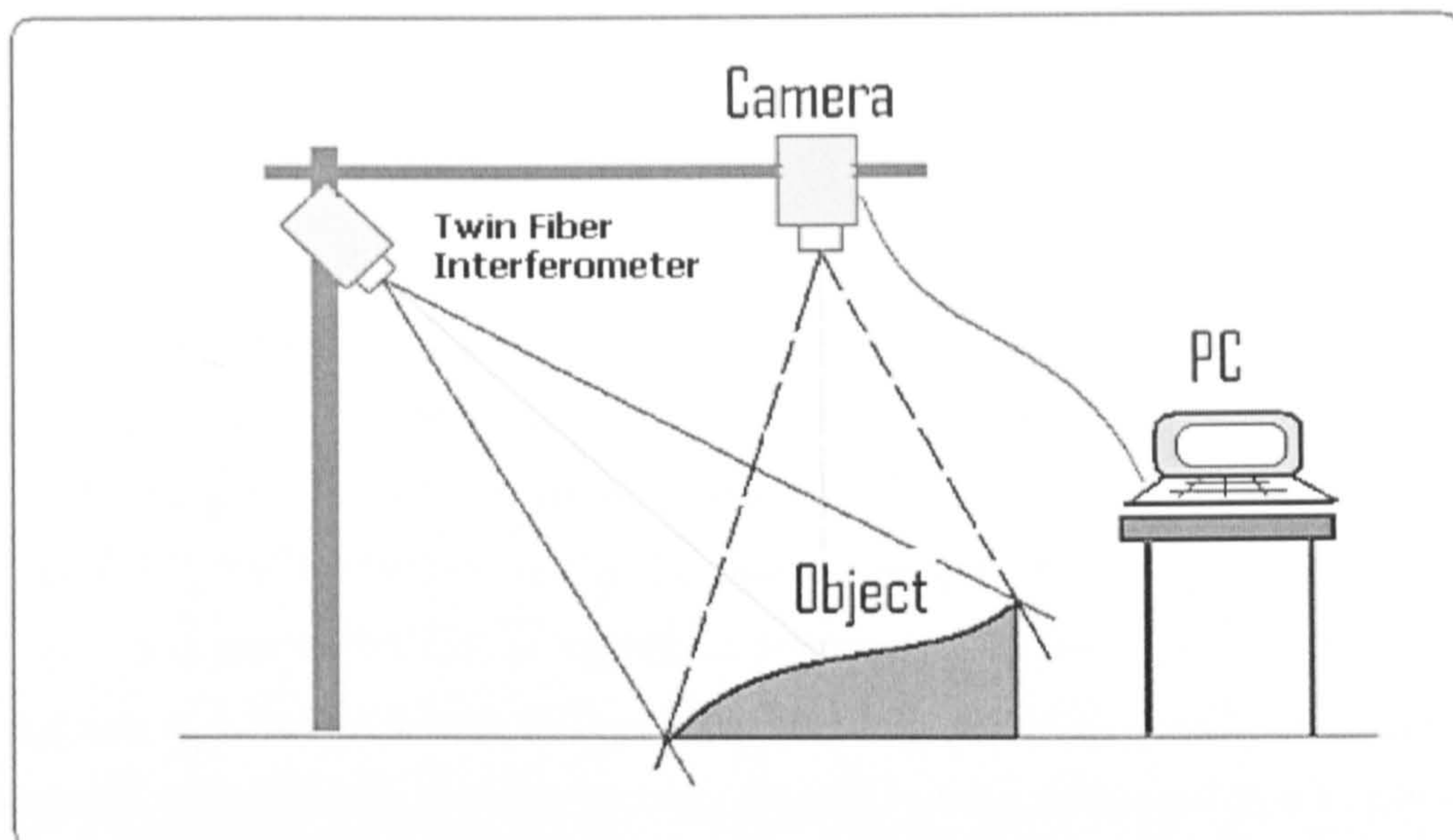
## 1.2 Fringe Production and Analysis

Fringe Analysis techniques (Rastogi, 1997) are considered to be an effective, reliable and robust optical non-contact method for measuring 3D surface height. In these methods; a structured lighting pattern is projected onto the object's surface. Due to the surface shape of the object, the projected pattern will be modified according to the object's 3D height. This pattern is typically captured by a CCD camera and then stored in computer memory by the use of a frame-grabber. The image is then analysed by one of a range of fringe analysis algorithms. Finally, the phase is extracted and related to actual object height by use of some sort of phase-height relationship that has been previously determined during system calibration. This type of typical fringe projection system is shown in Figure 1.1.

In the case of dynamic objects, the CCD camera captures a sequence of fringe patterns. This sequence of fringe patterns represents the 3D movement of the object during the measurement period.

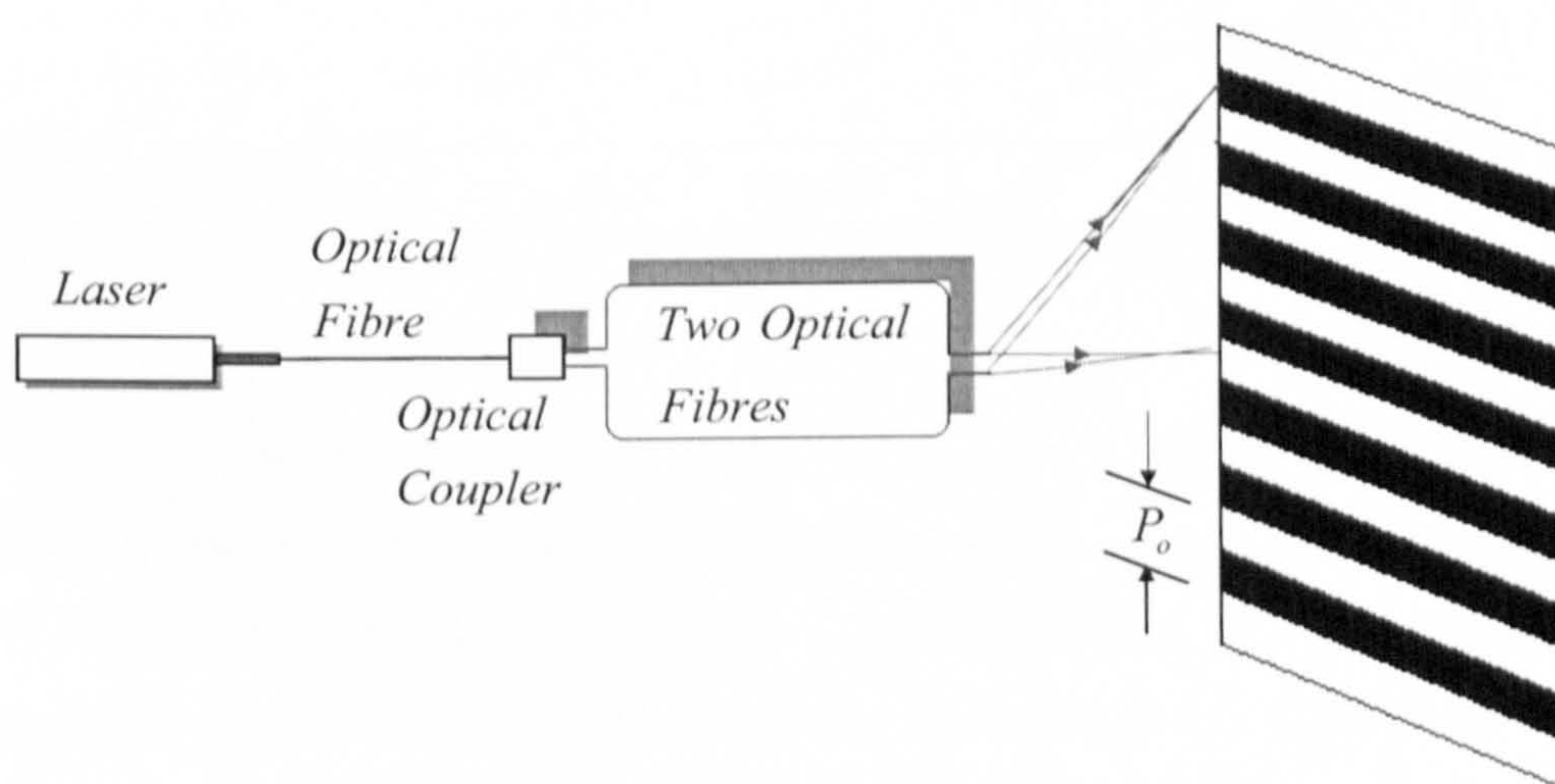
Fringes can be produced by using laser interferometers or by using digital projectors. In this project, laser interferometers were used to generate fringe patterns, as shown in Figure 1.1. As shown in the figure, two coherent laser beams projected from a twin fibre interferometer are used to illuminate an object. These beams will interfere with each other to produce a fringe pattern which is modulated by the object's surface. This pattern encodes the height profile of the object's surface. This pattern is captured by a camera to be analysed using a computer by means of fringe analysis algorithms.





**Figure 1.1: Fringe projection system.**

The twin-fibre optical system is shown in Figure 1.2. As shown in the figure, the generated laser beam will propagate into an optic fibre cable until it reaches an optical coupler. This optical coupler will divide the laser beam into two beams. These two beams will propagate into two different fibres and then illuminate the object as shown in the figure. The twin-fibre interferometer is mainly used throughout this research to project fringes on objects.



**Figure 1.2: A twin-fibre interferometer.**



One of the twin optical fibres can be moved in slight steps by a computer-controlled mechanism. A translation of the optical fibre along a line connecting the two fibre ends varies the spacing between fringes ( $P_o$ ).

In the setup shown in Figure 1.1, a fringe pattern is produced by interfering two coherent laser beams with each other. When the two laser waves are superposed, the resultant intensity at any point depends on whether these waves reinforce or cancel each other. If the two waves have the same phase, *i.e.*, the phase difference is zero, the intensity will be maximised, and if the two waves are  $180^\circ$  out of phase, then the intensity will be minimised, as shown in Figure 1.3. Repeating occurrences of the maximum and the minimum phase differences will produce a repeating pattern of maximum and minimum intensity forming what is called a **fringe pattern** as shown in Figure 1.3.

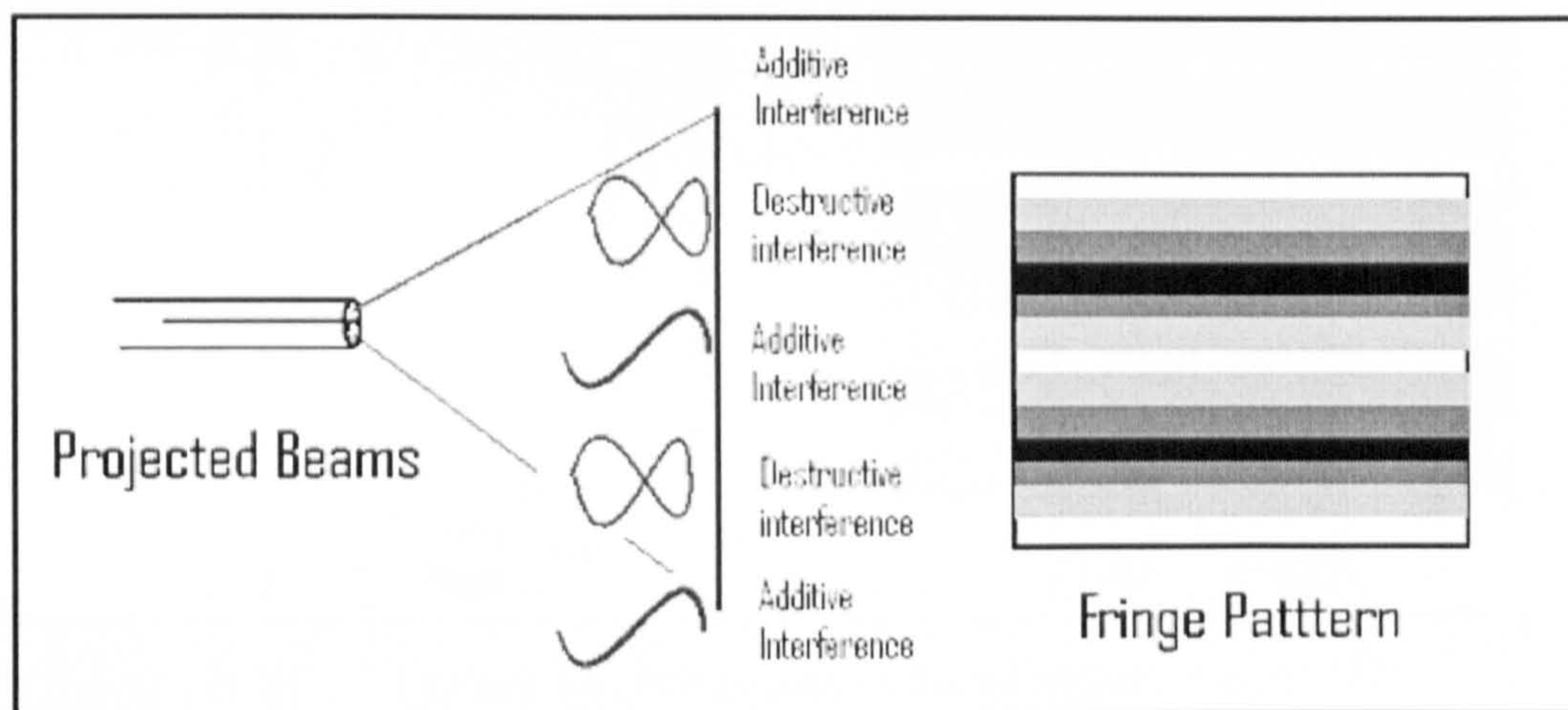


Figure 1.3: Fringe pattern for a reference plane.

However, the previous fringe pattern shown in Figure 1.3 will be modified in the presence of an object, when viewed obliquely. The Fringe pattern will be modulated according to the object's 3D height and the angle  $\theta$  between the illumination and viewing axes as shown in Figure 1.4. Accordingly, the height profile of the object under examination is encoded as a function of the spatial phase of the fringe pattern ( $\varphi$ ) that has been projected onto the object's surface. In other words, the object's height will



modulate the intensity distribution of the fringe pattern as expressed in the following equation (Robinson and Reid, 1993):

$$g(x, y) = a(x, y) + b(x, y) \cos(2\pi f_o x + \varphi(x, y)) \quad (1.1)$$

Where  $a(x, y)$  represents the background illumination,  $b(x, y)$  is the amplitude modulation of fringes,  $f_o$  is the spatial carrier frequency,  $\varphi(x, y)$  is the phase modulation of fringes (the required phase distribution) and  $x$  &  $y$  the sample indices for the  $x$  and  $y$  axes respectively.

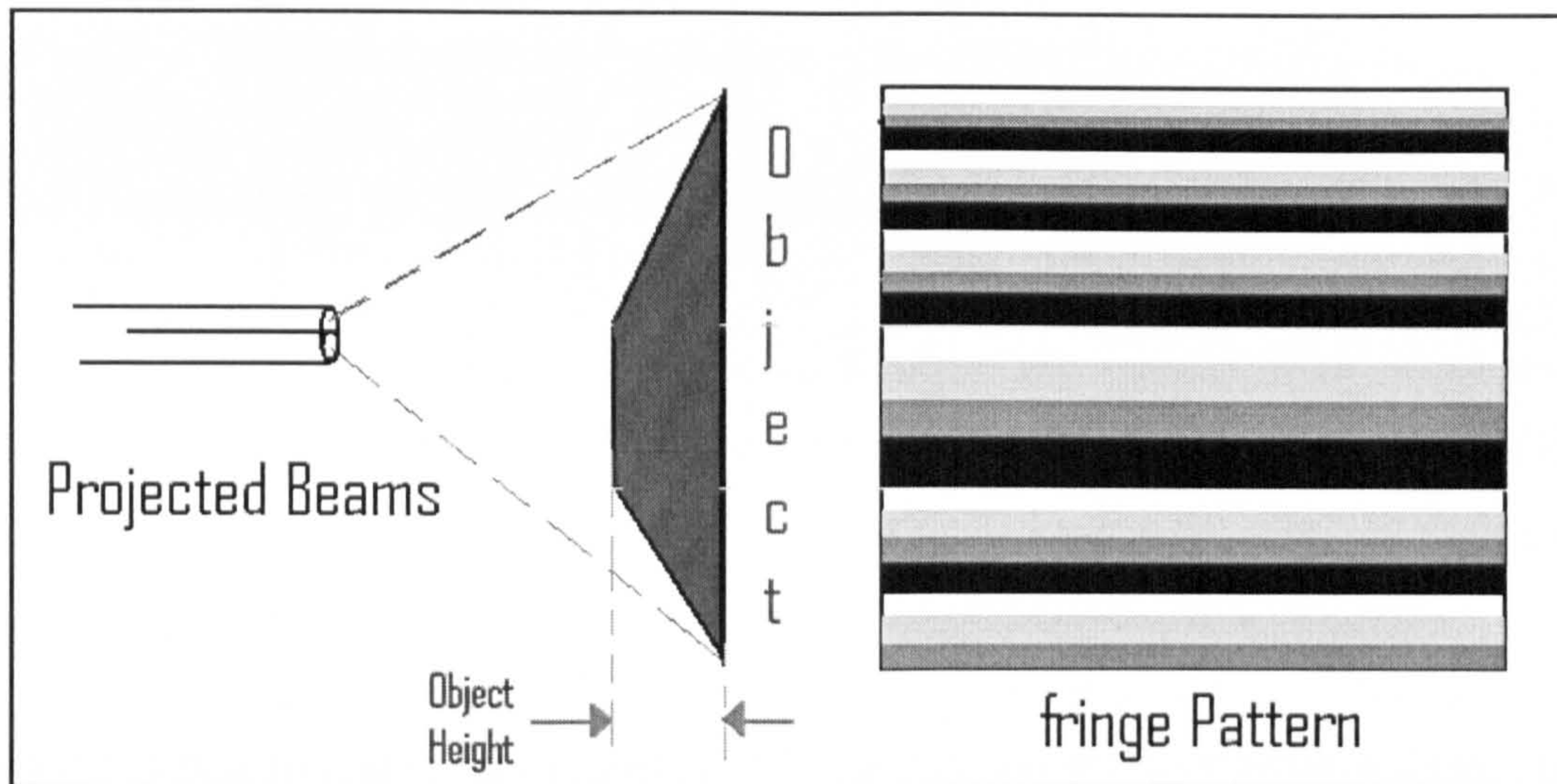


Figure 1.4: Fringe pattern for an object.

The task now is to extract the phase  $\varphi$  from the intensity distribution of the fringe pattern, given in Equation (1.1), using a fringe analysis technique. Many fringe analysis techniques have been proposed to demodulate fringe patterns. These techniques are reviewed in chapter two of this thesis. Unfortunately, these techniques produce what is called **wrapped-phase**  $\psi$  instead of the required phase  $\varphi$ . Consequently, **phase unwrapping algorithms** are required to recover the true phase  $\varphi$  from the wrapped-phase  $\psi$ . Phase unwrapping algorithms are discussed in chapter three of this thesis. Finally, the calculated phase difference  $\varphi$  has to be converted into the appropriate



height using a reliable phase to height model (Rajoub *et al.*, 2005). The steps of fringe patterns analysis can be summarized as in Figure 1.5.

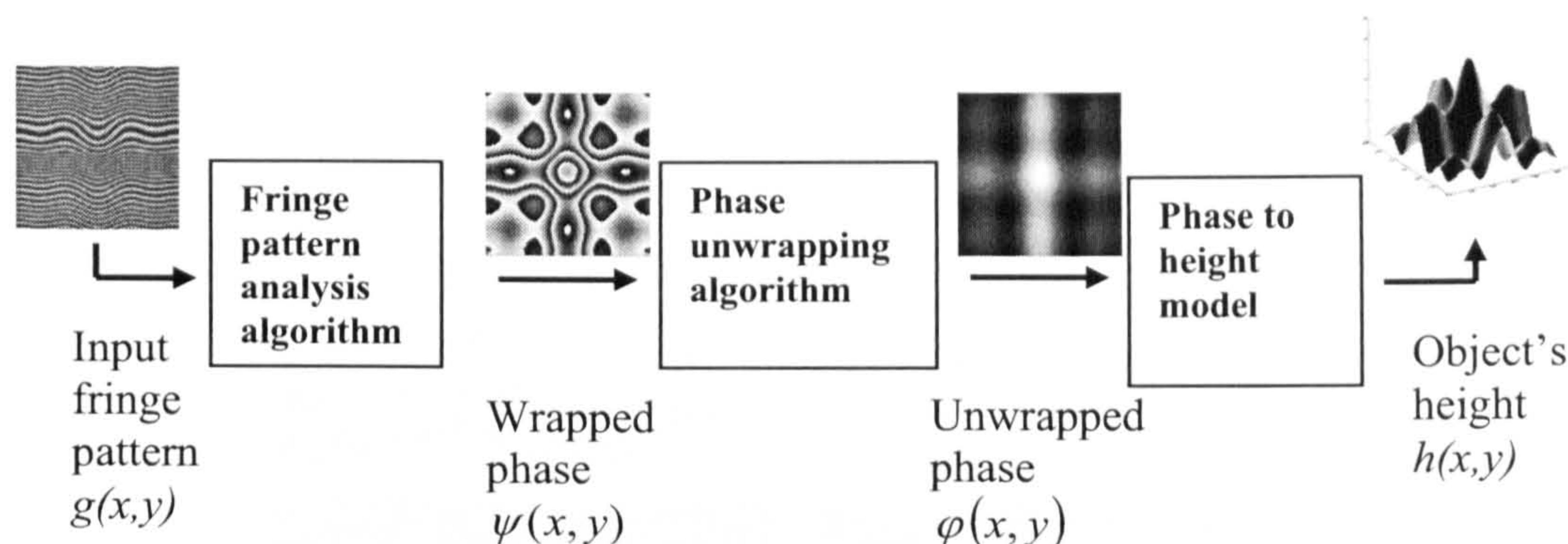


Figure 1.5: Fringe pattern analysis for an object.

### 1.3 Phase Unwrapping and its Application

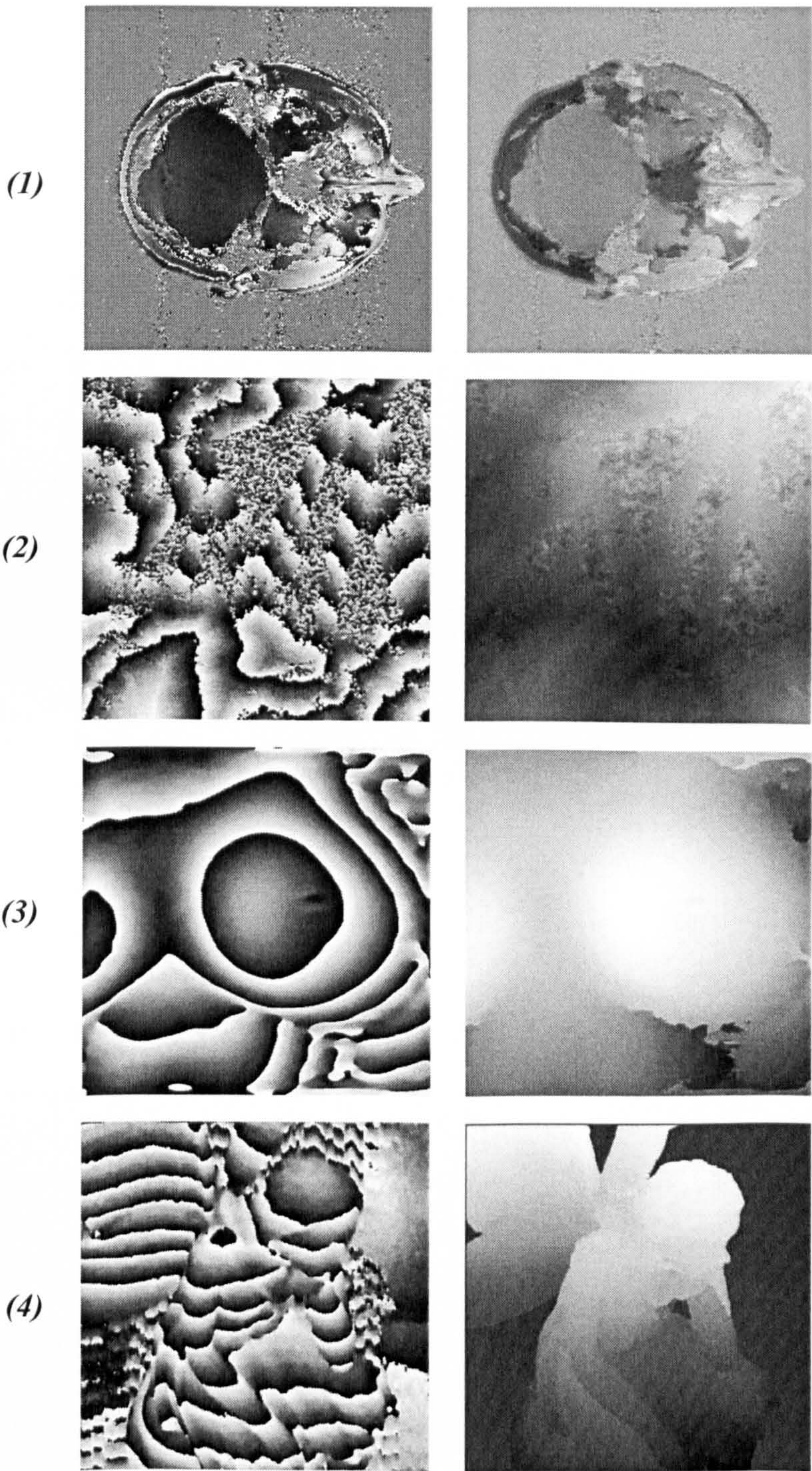
Phase unwrapping is a technique used on wrapped phase images to remove the  $2\pi$  discontinuities embedded within the phase map. It detects a  $2\pi$  phase jump and adds or subtracts an integer offset of  $2\pi$  to successive pixels following that phase jump based on a threshold mechanism, thus, retrieving the continuous form of the phase map.

Phase unwrapping has many applications in many imaging technologies such as: optical and interferometric imaging, magnetic resonance imaging (MRI), synthetic aperture radar (SAR), synthetic aperture sonar, adaptive optics, seismic processing and aperture synthesis radio astronomy (Ghiglia and Pritt, 1998). In many of the applications mentioned the extracted phase relates to physical quantities, for example surface height in interferometry, wave-front distortion in adaptive optics, the degree of magnetic field inhomogeneity in the water/fat separation problem of magnetic resonance imaging or in the relationship between object phase and its associated bi-spectrum phase in astronomical imaging.

Techniques such as these, where the required information is encoded in phase form all require the use of a phase unwrapping algorithm. Figure 1.6 shows some wrapped-phase



and unwrapped-phase images that have been produced from a range of different applications.



**Figure 1.6: Importance of phase unwrapping in many applications. The wrapped and unwrapped phase maps for: (Row 1) a human head from MRI image (Ghiglia and Pritt, 1998), (Row 2) part of the earth obtained from SAR image (Ghiglia and Pritt, 1998), (Row 3) a human breast undergoes radiotherapy treatment and (Row 4) artistic statue of a fairy. (Column a) wrapped phase maps and (Column b) corresponding unwrapped phase maps for those wrapped maps shown in column a.**



## 1.4 Research Objectives

The aim of this project is to investigate the technical issues and potential advantages in implementing three-dimensional Fourier fringe analysis and three-dimensional phase unwrapping, with particular respect to the analysis of dynamic objects.

The objectives of this research can be summarised as follows.

- Developing and implementing a three-dimensional Fourier fringe analysis technique. This step includes the development of three-dimensional windows for the sequence of fringe patterns, median filtering and filtering in the frequency domain.
- Developing and implementing a robust and fast three-dimensional phase unwrapping algorithm. Also, the developed algorithm will be compared to existing algorithms.
- Testing both developed algorithms on real dynamic objects.

## 1.5 Contributions

The contributions of this research can be summarized as following:

1. Development and implementation of a three-dimensional Fourier fringe analysis system.
2. Design and development of a three-dimensional phase unwrapping algorithm that searches for the optimal unwrapping path in the three-dimensional phase volume.
3. Design and development of a robust three-dimensional phase unwrapping algorithm, which avoids singularity loops in the phase volume following the optimal unwrapping path.
4. Extension of the existing two-dimensional quality-maps into three dimensions.
5. Investigation into the effect of quality maps on determining the optimal unwrapping path.



6. Comparison of different two-dimensional and three-dimensional phase unwrapping algorithms.
7. Evaluation of the performance of the three-dimensional Fourier fringe analysis and three-dimensional phase unwrapping algorithms on real patients.

## 1.6 Thesis Structure

The thesis is divided into six chapters as follows:

### **Chapter 1: Introduction**

This introductory chapter outlines non-contact measurement and its applications. Details of the scope of this research are also given. The chapter also describes the structure of the thesis.

### **Chapter 2: Fringe Pattern Analysis Techniques**

This chapter gives a review of a number of techniques used to demodulate fringe patterns, such as Fourier fringe analysis, phase stepping, direct phase detection, and wavelet fringe analysis.

### **Chapter 3: A Review of Two-Dimensional Phase Unwrapping Algorithms**

In this chapter, the concept of phase unwrapping is explained and a detailed review of the existing two-dimensional phase unwrapping algorithms is given.

### **Chapter 4: Three-Dimensional Fourier fringe analysis**

The implementation of a fully three-dimensional form of the Fourier fringe analysis technique is the main subject in this chapter. Also, this chapter shows the results of applying three-dimensional Fourier fringe analysis on simulated as well as real dynamic objects.

### **Chapter 5: Three-Dimensional Phase Unwrapping Algorithms**

In this chapter, a review of existing three-dimensional phase unwrapping algorithms is presented. Moreover, this chapter proposes two novel three-dimensional phase unwrapping algorithms that utilise quality maps to find an optimal unwrapping path.



The proposed algorithms are applied here to the unwrapping of both computer simulated and real phase volumes. Furthermore, this chapter compares the proposed techniques with other existing three-dimensional phase unwrapping algorithms.

## **Chapter 6: Conclusions and Future Work**

This is the final chapter in the thesis, where conclusions are drawn from the research described in this thesis. Also, a list of possible future work is outlined in this chapter.

## References:

- Abid, A.; Gdeisat, M.; Burton, D. and Lalor, M. (2006) A Comparison between Wavelet Fringe Analysis Algorithms. *Photon 06*. Manchester UK.
- Bone, D. J.; Bachor, H. A. and Sandeman, R. J. (1986) Fringe-Pattern Analysis Using a 2-D Fourier Transform. *Applied Optics*, 25, 1653-60.
- Burton, D. R. and Lalor, M. J. (1989a) Managing Some of the Problems of Fourier Fringe Analysis. Proc. SPIE Vol. 1163, p. 149, San Diego, CA, USA.
- Burton, D. R. and Lalor, M. J. (1989b) The Precision Measurement of Engineering Form by Computer Analysis of Optically Generated Contours. Proc. SPIE Vol. 1010, p. 17. Hamburg, West Germany.
- Burton, D. R. and Lalor, M. J. (1994) Multichannel Fourier Fringe Analysis as an Aid to Automatic Phase Unwrapping. *Applied Optics*, 33, 2939-48.
- 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, 245-57.
- Chan, P. H.; Bryanston-Cross, P. J. and Parker, S. C. (1995) Spatial Phase Stepping Method of Fringe-Pattern Analysis. *Optics and Lasers in Engineering*, 23, 343-54.
- Chen, S. Y. and Li, Y. F. (2003) Self-Recalibration of a Colour-Encoded Light System for Automated Three-Dimensional Measurements. *Measurement Science & Technology*, 14, 33-40.
- Dursun, A.; Ozder, S. and Ecevit, F. N. (2004) Continuous Wavelet Transform Analysis of Projected Fringe Patterns. *Measurement Science and Technology*, 15, 1768-1772.
- Engelsman, M.; Remeijer, P.; van-Herk, M.; Mijnheer, B. and Damen, E. (2003) The Theoretical Benefit of Beam Fringe Compensation and Field Size Reduction for Iso-Normal Tissue Complication Probability Dose Escalation in Radiotherapy of Lung Cancer. *Medical Physics*, 30, 1086-95.
- Gdeisat, M. A.; Burton, D. R. and Lalor, M. J. (2006) Spatial Carrier Fringe Pattern Demodulation by Use of a Two-Dimensional Continuous Wavelet Transform. *Applied Optics*, 45, 8722-8732.
- Ghiglia, D. C. and Pritt, M. D. (1998) *Two-Dimensional Phase Unwrapping: Theory, Algorithms and Software*, A Willey-Interscience Publication.

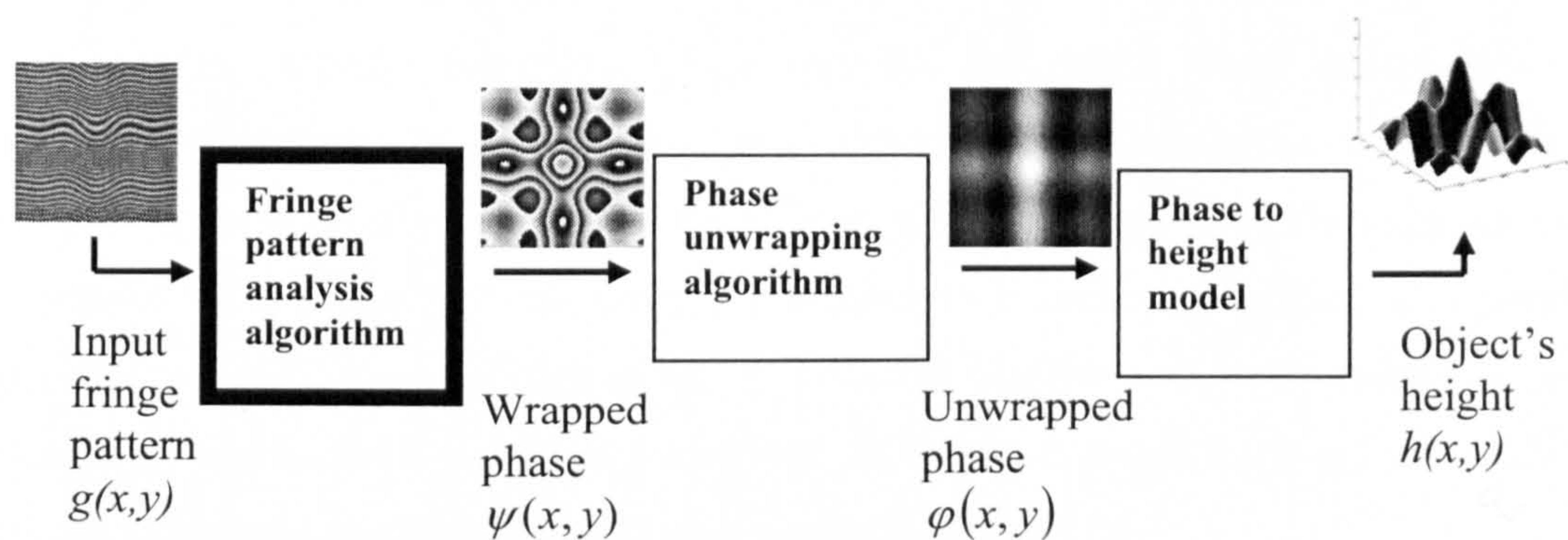
- Gorecki, C. (1992) Interferogram Analysis Using a Fourier Transform Method for Automatic 3d Surface Measurement. *Pure and Applied Optics*, 1, 103-110.
- Haist, T. and Tiziani, H. J. (2002) Color-Coded Object-Adapted Fringe Projection for Two- and Three-Dimensional Quality Control. *Technisches Messen tm*, 69, 367-73.
- Ichioaka, Y. and Inuiya, M. (1972) Direct Phase Detection System. *Applied Optics*, vol. 11, pp. 1507-1514.
- Kujawska, M. and Sitnik, R. (2000) Quality Assessment of Reverse Engineering Process Based on Full-Field True-3d Optical Measurements. *Proceedings of the SPIE The International Society for Optical Engineering*, 4076, 201-9.
- Leopold, J.; Guthier, H. and Leopold, R. (2003) New Developments in Fast 3d-Surface Quality Control. *Measurement*, 33, 179-87.
- Lilley, F. (1999) An Optical 3-D Body Surface Measurement System to Improve Radiotherapy Treatment of Cancer. *Faculty of General Engineering, James Parsons Building*. Liverpool, Liverpool John Moores University.
- Lilley, F.; Lalor, M. J. and Burton, D. R. (2000) Robust Fringe Analysis System for Human Body Shape Measurement. *Optical Engineering*, 39, 187-95.
- Moore, C.; Lilley, F.; Sauret, V.; Lalor, M. and Burton, D. (2003) Opto-Electronic Sensing of Body Surface Topology Changes During Radiotherapy for Rectal Cancer. *International Journal of Radiation Oncology Biology Physics*, 56, 248-258.
- Quan, C. G.; Tay, C. J.; Chen, L. J. and Fu, Y. (2003) Spatial-Fringe-Modulation-Based Quality Map for Phase Unwrapping. *Applied Optics*, 42, 7060-7065.
- Rajoub, B. A.; Burton, D. R. and Lalor, M. J. (2005) A New Phase-to-Height Model for Measuring Object Shape Using Collimated Projections of Structured Light. *Jornal of Optics A: Pure and Applied Optics*, 7, S368-S375.
- Rastogi, P. K. (1997) *Optical Measurement Techniques and Applications*, ARTECH HOUSE, INC.
- Robinson, D. W. and Reid, G. T. (1993) *Interferogram Analysis: Digital Fringe Pattern Measurement Techniques*, Institute of Physics Publishing.
- Smith, L. N. and Smith, M. L. (2003) Stereo Vision Technology for Object Measurement. *Proceedings of the SPIE The International Society for Optical Engineering*, 5011, 307-12.



- Su, X. and Chen, W. (2001) Fourier Transform Profilometry: A Review. *Optics and Lasers in Engineering*, 35, 263-284.
- Su, X.; Chen, W.; Qc, Z. and Chao, Y. (2001) Dynamic 3-D Shape Measurement Method Based on Ftp. *optics and Lasers in Engineering*, 36, 46-64.
- Takeda, M.; Ina, H. and Kobayashi, S. (1982) Fourier-Transform Method of Fringe-Pattern Analysis for Computer-Based Topography and Interferometry. *Journal of the Optical Society of America*, 72, 156-160.
- Tao-Xian and Xianyu-Su (2001) Area Modulation Grating for Sinusoidal Structure Illumination on Phase-Measuring Profilometry. *Applied Optics*, 40, 1201-6.
- Woisetschlager, J.; Sheffer, D. B.; Loughry, C. W.; Somasundaram, K.; Chawla, S. K. and Wesolowski, P. J. (1994) Phase-Shifting Holographic-Interferometry for Breast-Cancer Detection. *Applied Optics*, 33, 5011-5015.

Chapter Two

Fringe Pattern Analysis Techniques



---

## Chapter Two

### Fringe Pattern Analysis Techniques

---

#### 2.1 Introduction

Many techniques have been proposed for the analysis of fringe patterns. These techniques vary in accuracy, the number of frames required and processing time. The aim of any fringe pattern analysis algorithm is to obtain the phase information encoded into the fringe pattern. This phase may be wrapped between  $(-\pi, \pi)$  and needs to be unwrapped, as will be shown in the next chapter.

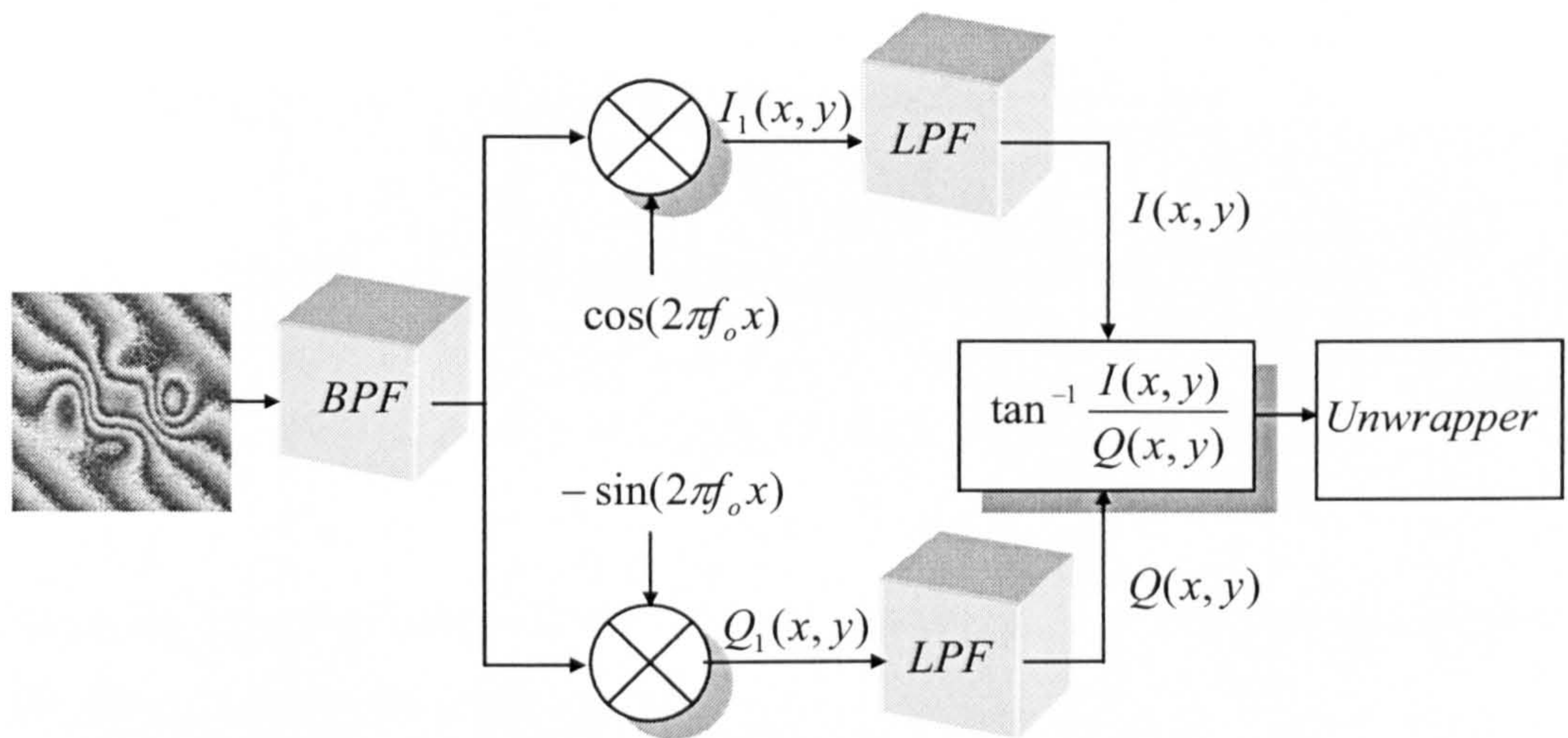
Fringe pattern analysis algorithms can be classified into two categories: spatial and temporal techniques. Spatial algorithms calculate the phase of a pixel in a fringe pattern depending on its neighbouring pixels. Examples of a spatial technique are; Fourier fringe analysis, wavelet transform fringe analysis and direct phase demodulation. Spatial techniques require at least one fringe pattern to calculate phase components. Conversely, temporal algorithms require at least three images to calculate the phase components of a fringe pattern. Temporal techniques calculate the phase of a pixel depending on the values of that pixel in different images and independent of its surrounding pixels. An example of a temporal algorithm is phase stepping. All major previous techniques are explained and reviewed in this chapter.

#### 2.2 Direct Phase Demodulation

Direct phase detection (DPD) adopts phase detection methods from communications theory that are used to demodulate FM radio signals and applies it to analyse fringe patterns (Ichioka and Inuiya, 1972).



A block diagram of a direct phase detector is shown in Figure 2.1 (Hobson *et al.*, 1997). As shown in the figure, the fringe pattern is applied to a Band Pass Filter (BPF) in order to remove the background illumination and the high frequency noise. Then the filtered fringe pattern is split into two parts and each part is applied to a multiplier as shown in Figure 2.1. The output of the multiplier is applied to a Low Pass Filter (LPF) in order to attenuate the high frequency components that result from the multiplication process. An arctangent function is used to extract the desired phase information from the outputs of the low pass filters as indicated in the diagram. The resultant phase map is returned wrapped between  $(-\pi, \pi)$  and hence an unwrapping algorithm is required in order to remove the  $2\pi$  ambiguities.



**Figure 2.1: A Diagram of direct phase detection**

As stated in the previous chapter, the intensity profile of any fringe pattern is given by:

$$g(x, y) = a(x, y) + b(x, y) \cos(2\pi f_o x + \phi(x, y)) \quad (2.1)$$

Where  $a(x, y)$  represents the background illumination,  $b(x, y)$  the amplitude modulation of fringes,  $f_o$  the spatial carrier frequency,  $\phi(x, y)$  the phase modulation of fringes (the required phase distribution) and  $x$  &  $y$  the sample indices for the  $x$  and  $y$  axes respectively.



Applying this fringe pattern to the BPF will remove the low frequency component  $a(x, y)$ . The filtered image is given by:

$$g_{BPF}(x, y) = b(x, y) \cos(2\pi f_o x + \varphi(x, y)) \quad (2.2)$$

This filtered image will be applied to the multipliers as shown in Figure 2.1 and the output images are:

$$I_1(x, y) = b(x, y) \cos(2\pi f_o x + \varphi(x, y)) * \cos(2\pi f_o x) \quad (2.3)$$

$$Q_1(x, y) = -b(x, y) \cos(2\pi f_o x + \varphi(x, y)) * \sin(2\pi f_o x) \quad (2.4)$$

Applying trigonometric identities in the above equation will yield:

$$I_1(x, y) = \frac{b(x, y)}{2} [\cos(4\pi f_o x + \varphi(x, y)) + \cos(\varphi(x, y))] \quad (2.5)$$

$$Q_1(x, y) = \frac{-b(x, y)}{2} [\sin(4\pi f_o x + \varphi(x, y)) - \sin(\varphi(x, y))] \quad (2.6)$$

Applying these two images to the LPF will remove the high frequency components and the filtered images are given by:

$$I(x, y) = \frac{b(x, y)}{2} \cos(\varphi(x, y)) \quad (2.7)$$

$$Q(x, y) = \frac{b(x, y)}{2} \sin(\varphi(x, y)) \quad (2.8)$$

The phase distribution can be calculated by:

$$\psi(x, y) = \tan^{-1} \left( \frac{I(x, y)}{Q(x, y)} \right) = \varphi(x, y) \mod \pi \quad (2.9)$$

This phase distribution is wrapped between  $(-\pi, \pi)$ , and needs to be unwrapped for future processing.



This direct phase demodulation method is applied to analyse a simulated fringe pattern shown in Figure 2.2. This fringe pattern has been produce by using a computer simulated object shown in Figure 2.3. This simulated fringe pattern can be described by:

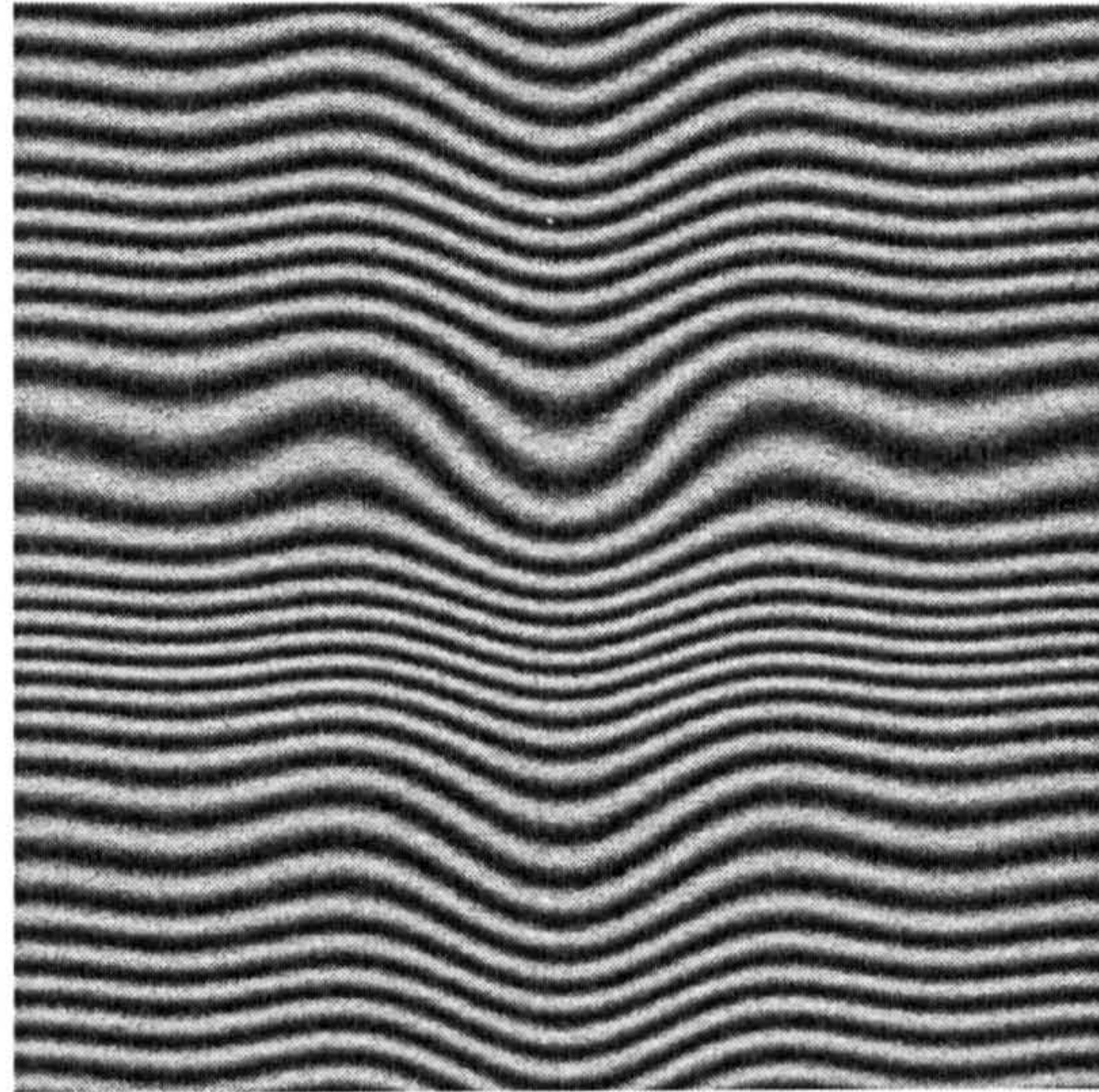
$$g_{sim}(x, y) = noise_1(x, y) + [1 + noise_2(x, y)] \cos \left( 2\pi f_o x + \left[ 10 * \left( \frac{\sin(x)}{x} + \frac{\sin(y)}{y} \right) \right] \right) \quad (2.10)$$

Where,

$noise_1(x, y)$  is a Gaussian noise with zero mean and standard deviation of 0.25.

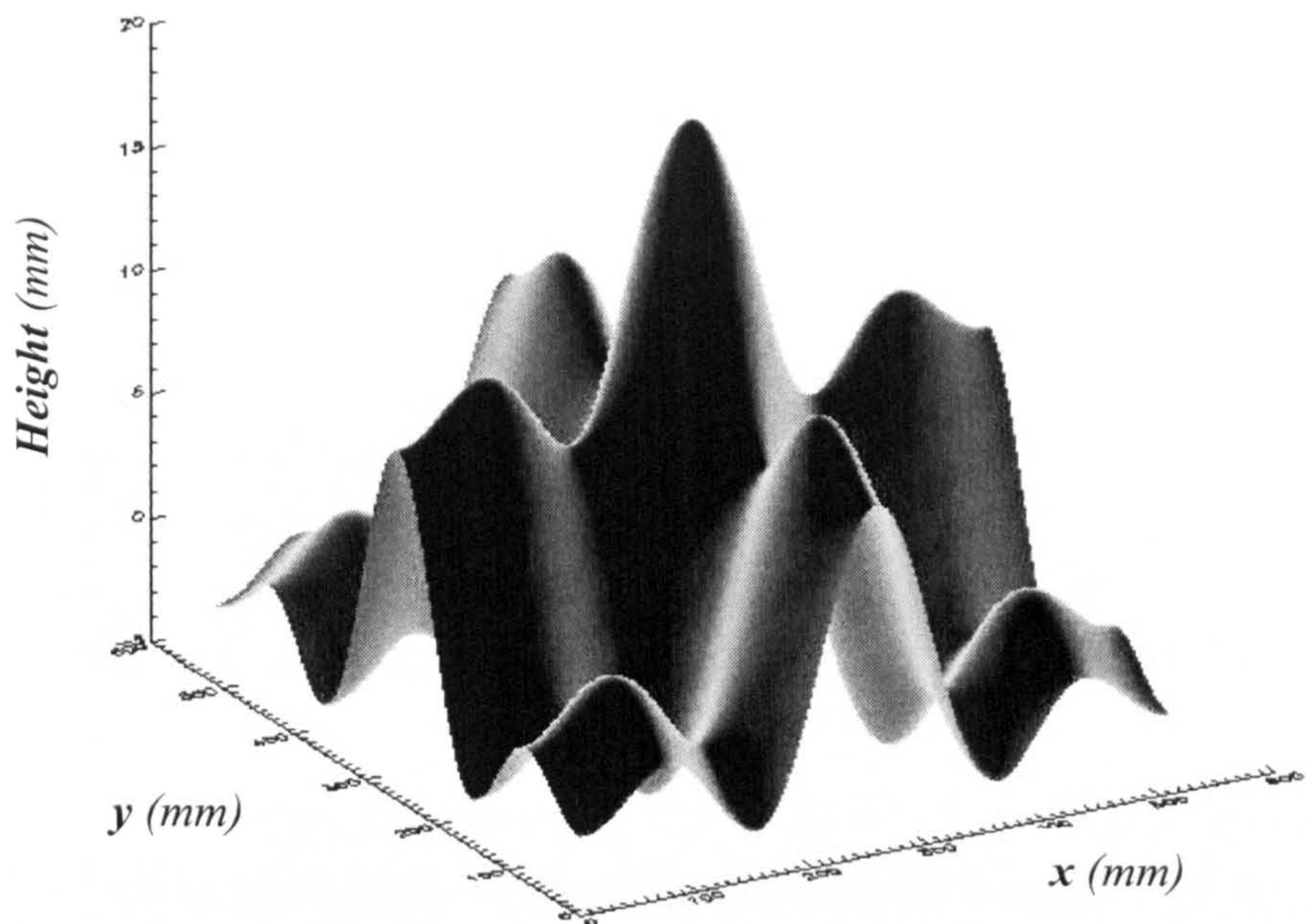
$noise_2(x, y)$  is a Gaussian noise with zero mean and standard deviation of 0.05.

Note that these values of noise are chosen arbitrary.



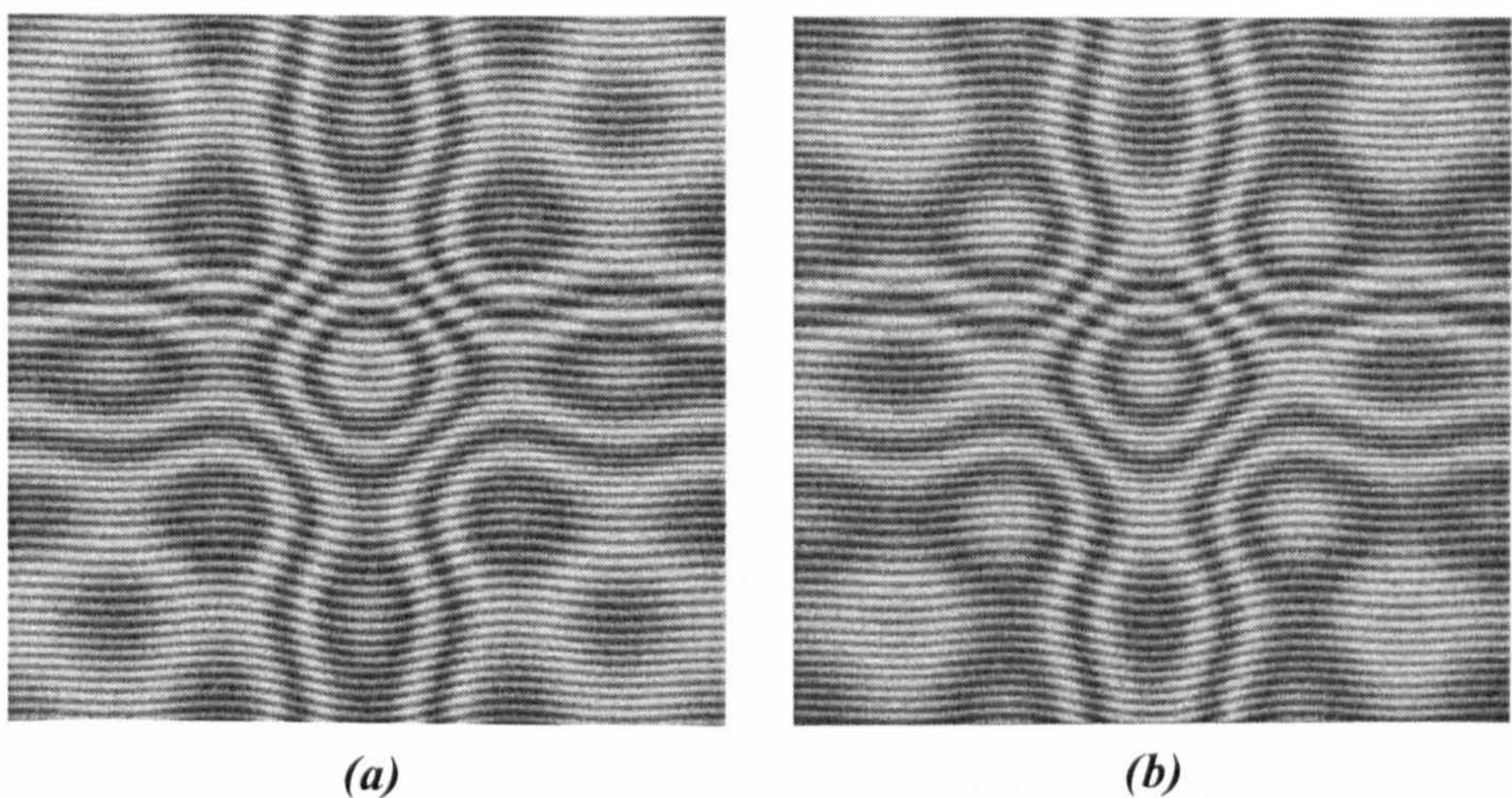
**Figure 2.2: A Computer simulated fringe pattern**





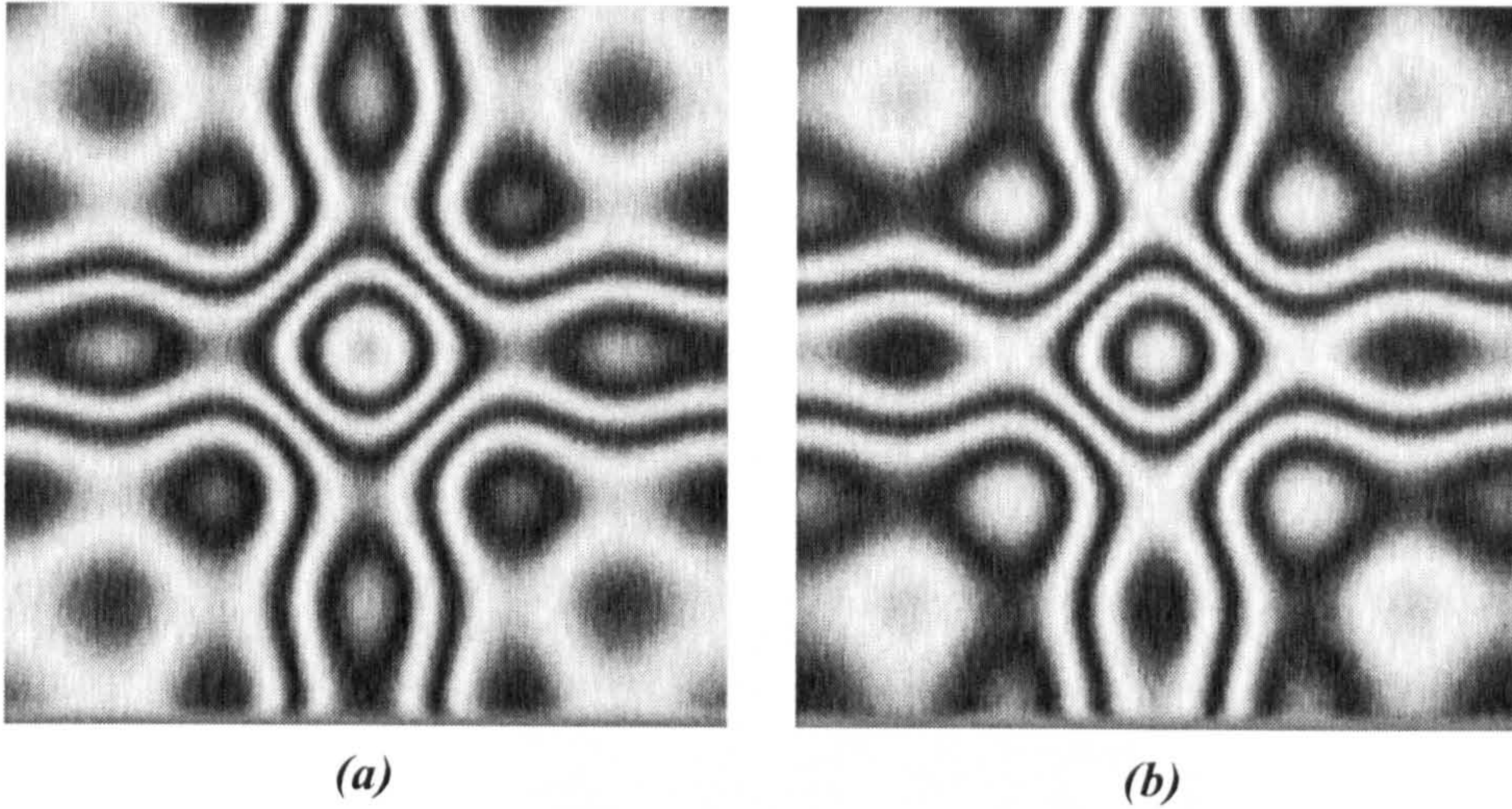
**Figure 2.3: A Computer simulated object used to produce the fringe pattern shown in Figure 2.2**

This computer simulated fringe pattern is applied to the BPF then to the multipliers as described earlier. Figure 2.4 shows the output of each multiplier denoted as  $I_1(x,y)$  and  $Q_1(x,y)$  respectively in Figure 2.1. The images shown in Figure 2.4 are filtered using LPFs and the filtered images are shown in Figure 2.5. The wrapped phase image is shown in Figure 2.6.



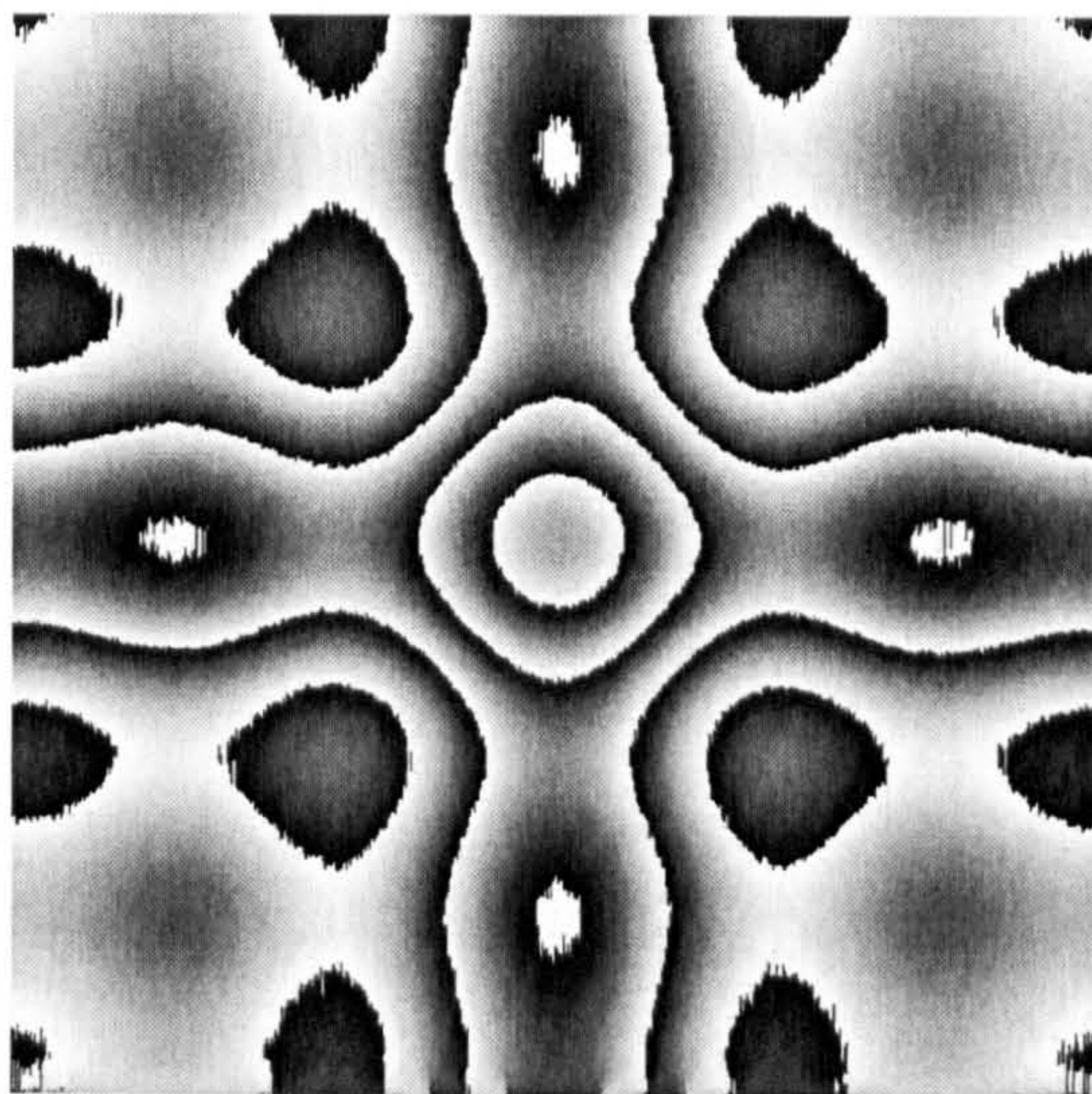
**Figure 2.4: The output of the multipliers shown in Figure 2.1 (a) Image  $I_1(x,y)$  yielded from the upper multiplier; (b) Image  $Q_1(x,y)$  yielded from the lower multiplier.**





**Figure 2.5:** The output of the LPFs shown in Figure 2.1 (a) Image  $I(x,y)$  yielded from the upper LPF,;(b) Image  $Q(x,y)$  yielded from the lower LPF

The wrapped phase distribution shown in Figure 2.6 needs to be unwrapped in order to be usable by any future processes, *i.e.*, to be used in calculating the object's height. Figure 2.7 shows a three-dimensional view of the unwrapped phase distribution yielded by applying the two-dimensional phase unwrapping algorithm proposed by Herraiez *et.al.* (Herraiez *et al.*, 2002). A detailed review of two-dimensional phase unwrapping algorithms will be covered in the next chapter.



**Figure 2.6:** The wrapped phase distribution which resulted from an analysis of the simulated fringe pattern shown in Figure 2.2, using the direct phase detection method.



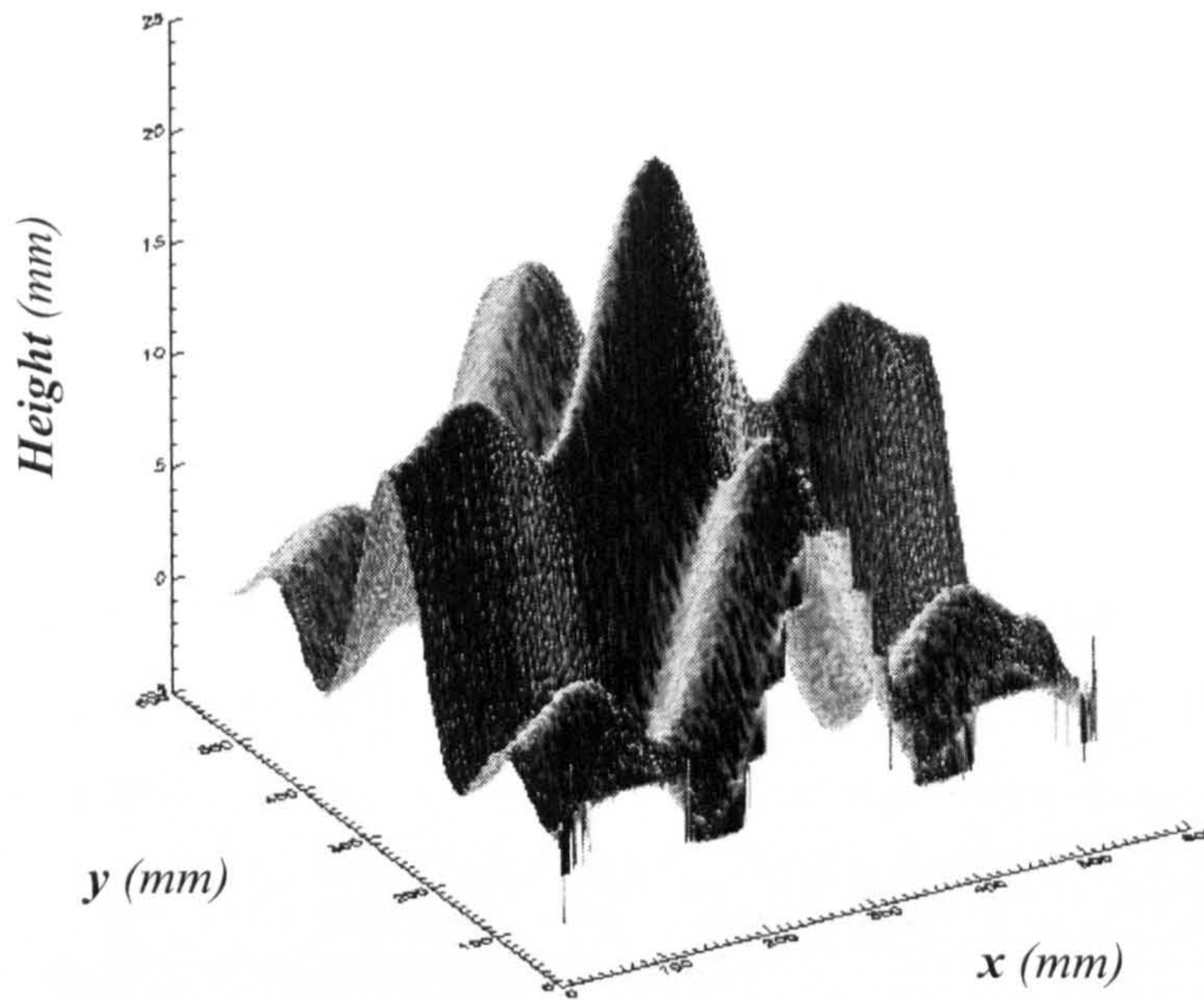


Figure 2.7: A three-dimensional view of the unwrapped phase which was yielded by unwrapping the wrapped phase map shown in Figure 2.5.

### 2.3 Phase Stepping

Phase stepping is a very robust technique for the analysis of fringe patterns. It produces a wrapped phase distribution by using at least three source fringe patterns. In comparison to single frame measurement techniques, such as Fourier fringe analysis, phase stepping is computationally relatively simple.

Phase stepping methods have been used since 1974 (Bruning and al., 1974) and have been developed by many researchers (Hariharan *et al.*, 1987; Wyant and Creath, 1992; Creath, 1993; Creath and Schmit, 1994; Hibino *et al.*, 1995). Many different variants of the phase stepping algorithms have been proposed, such as the three-frame, Carré, four-frame, five-frame and the "2+1" techniques (Creath, 1993).



As explained above, the intensity of a fringe pattern can be expressed as:

$$g(x, y) = a(x, y) + b(x, y) \cos(2\pi f_o x + \phi(x, y))$$

Three unknowns exist in this equation:  $a(x, y)$ ,  $b(x, y)$  and  $\phi(x, y)$ . The phase  $\phi(x, y)$  is the information of interest and the other two terms  $a(x, y)$  and  $b(x, y)$  need to be eliminated. To determine the required phase information, at least three independent equations are required.

A common phase stepping technique is the four-frame technique, which uses four fringe patterns with a deliberately inserted, accurately-known phase shift of  $\pi/2$  radians between them. The intensity in the four fringe patterns can be expressed as:

$$\begin{aligned} g_1(x, y) &= a(x, y) + b(x, y) \cos(2\pi f_o x + \phi(x, y)) \\ g_2(x, y) &= a(x, y) + b(x, y) \cos(2\pi f_o x + \phi(x, y) + \frac{\pi}{2}) \\ g_3(x, y) &= a(x, y) + b(x, y) \cos(2\pi f_o x + \phi(x, y) + \pi) \\ g_4(x, y) &= a(x, y) + b(x, y) \cos(2\pi f_o x + \phi(x, y) + \frac{3\pi}{2}) \end{aligned} \quad (2.11)$$

Subtracting  $g_3(x, y)$  from  $g_1(x, y)$  yields:

$$g_1(x, y) - g_3(x, y) = 2b(x, y) \cos(2\pi f_o x + \phi(x, y)) \quad (2.12)$$

Also subtracting  $g_2(x, y)$  from  $g_4(x, y)$  gives:

$$g_4(x, y) - g_2(x, y) = 2b(x, y) \sin(2\pi f_o x + \phi(x, y)) \quad (2.13)$$

Dividing the last two equations and taking the arctangent of the result gives the desired phase information

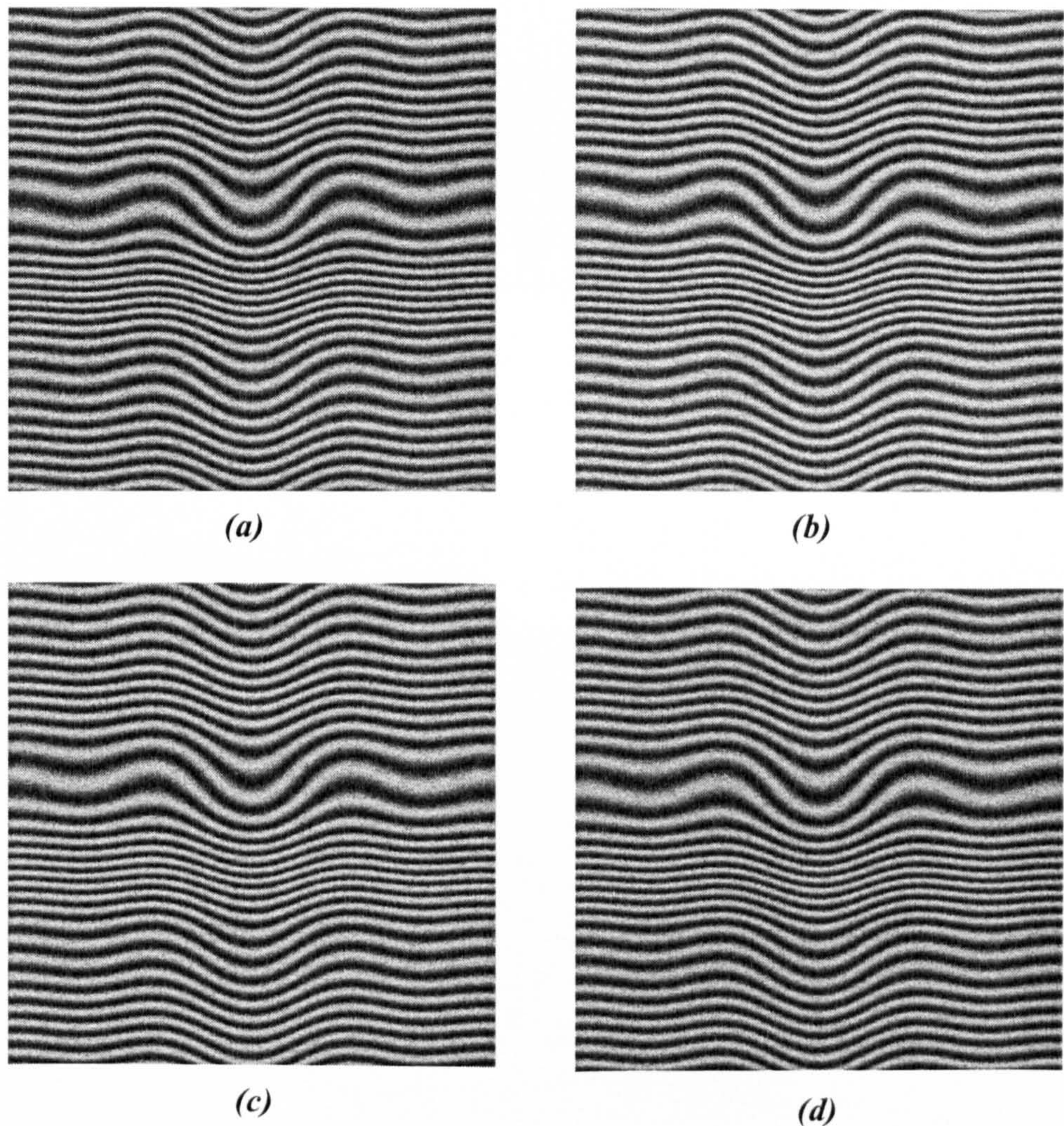
$$\psi(x, y) = \tan^{-1} \left( \frac{g_4(x, y) - g_2(x, y)}{g_1(x, y) - g_3(x, y)} \right) = [2\pi f_o x + \phi(x, y)] \bmod \pi \quad (2.14)$$



The resultant phase map is wrapped and an unwrapping algorithm is required in order to remove the  $2\pi$  steps.

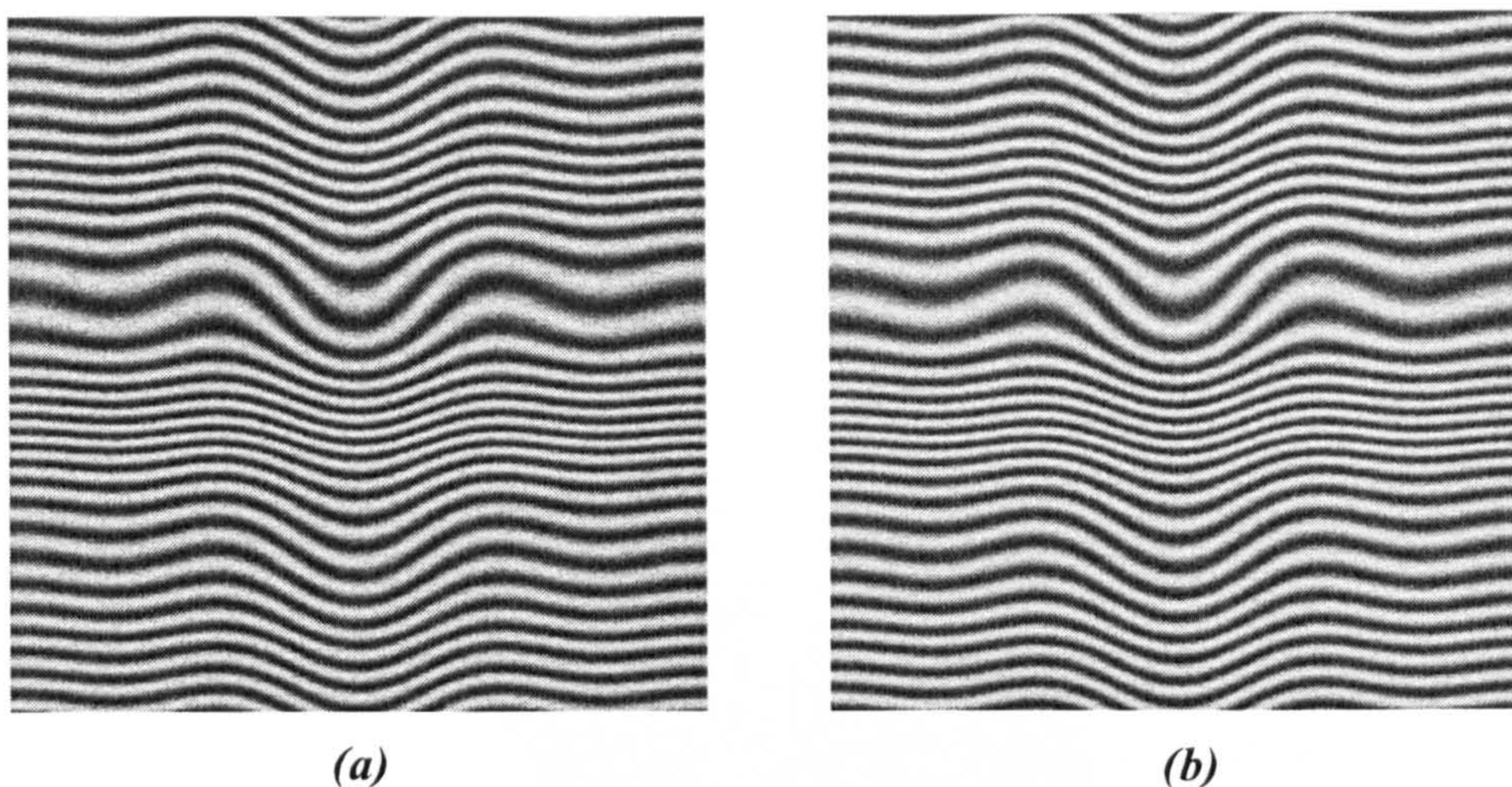
The four-frame phase stepping algorithm is implemented on a computer-generated fringe pattern shown in Figure 2.2. Figure 2.8 shows four different fringe patterns with  $0, \pi/2, \pi$  and  $3\pi/2$  radian phase shifts in (a), (b), (c) and (d) respectively.

Figure 2.9 shows the results of the subtraction operation expressed in Equations 2.12 and 2.13. As shown in the figure, the background noise  $a(x,y)$  is completely suppressed in this stage due to the fact that similar noise distributions exist in each frame and are cancelled by the subtractions.



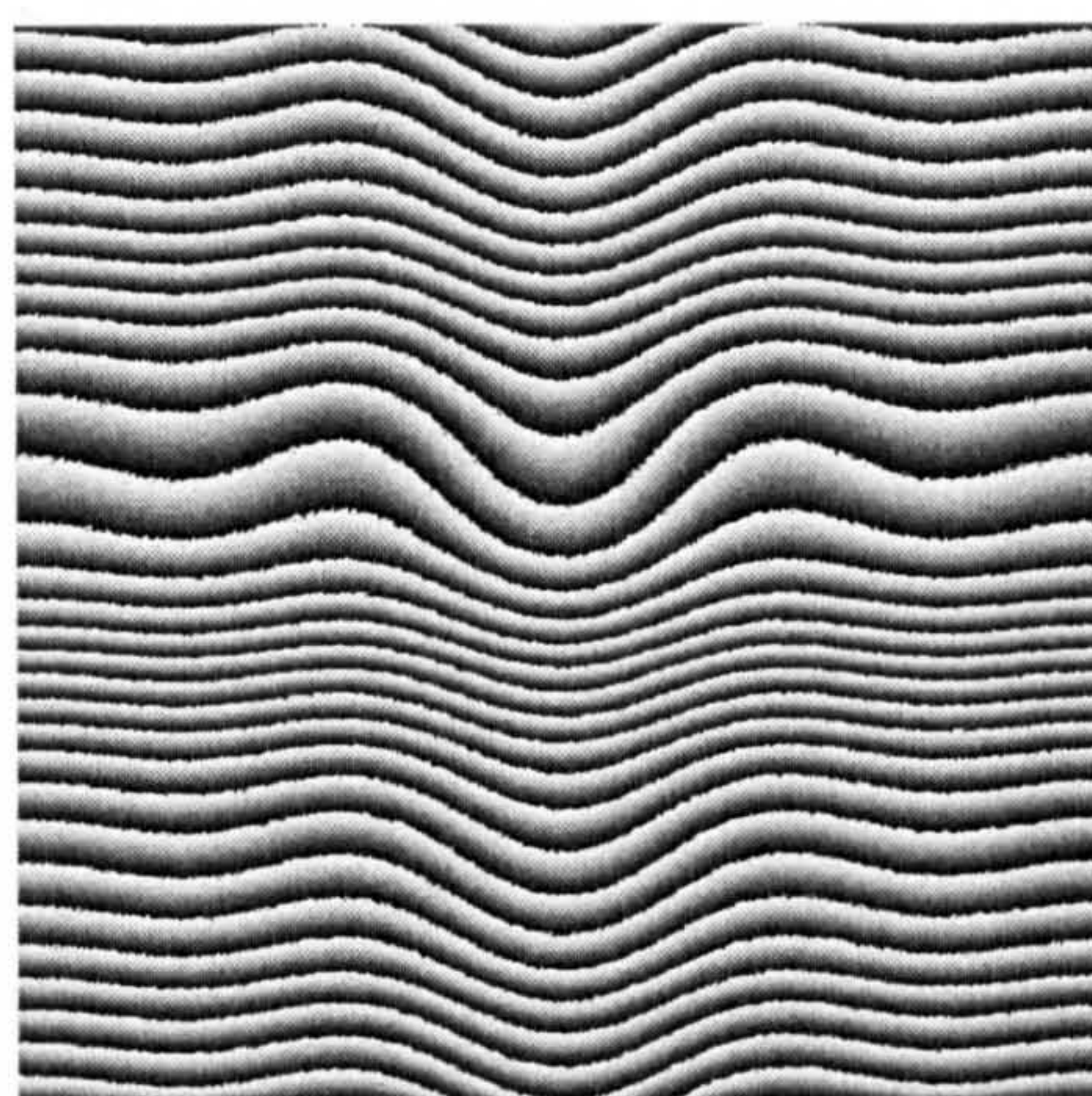
**Figure 2.8: Phase stepping images showing the four steps of the simulated fringe pattern shown in Figure 2.2. The fringes patterns with (a) 0, (b)  $\pi/2$ , (c)  $\pi$  and (d)  $3\pi/2$  phase shifts**





**Figure 2.9:** The results of subtracted images. (a) Result of subtracting the frame with  $\pi$  phase shift from the 0 phase shifted frame, (b) result of subtracting the frame with  $\pi/2$  phase shift from the  $3\pi/2$  phase shifted frame.

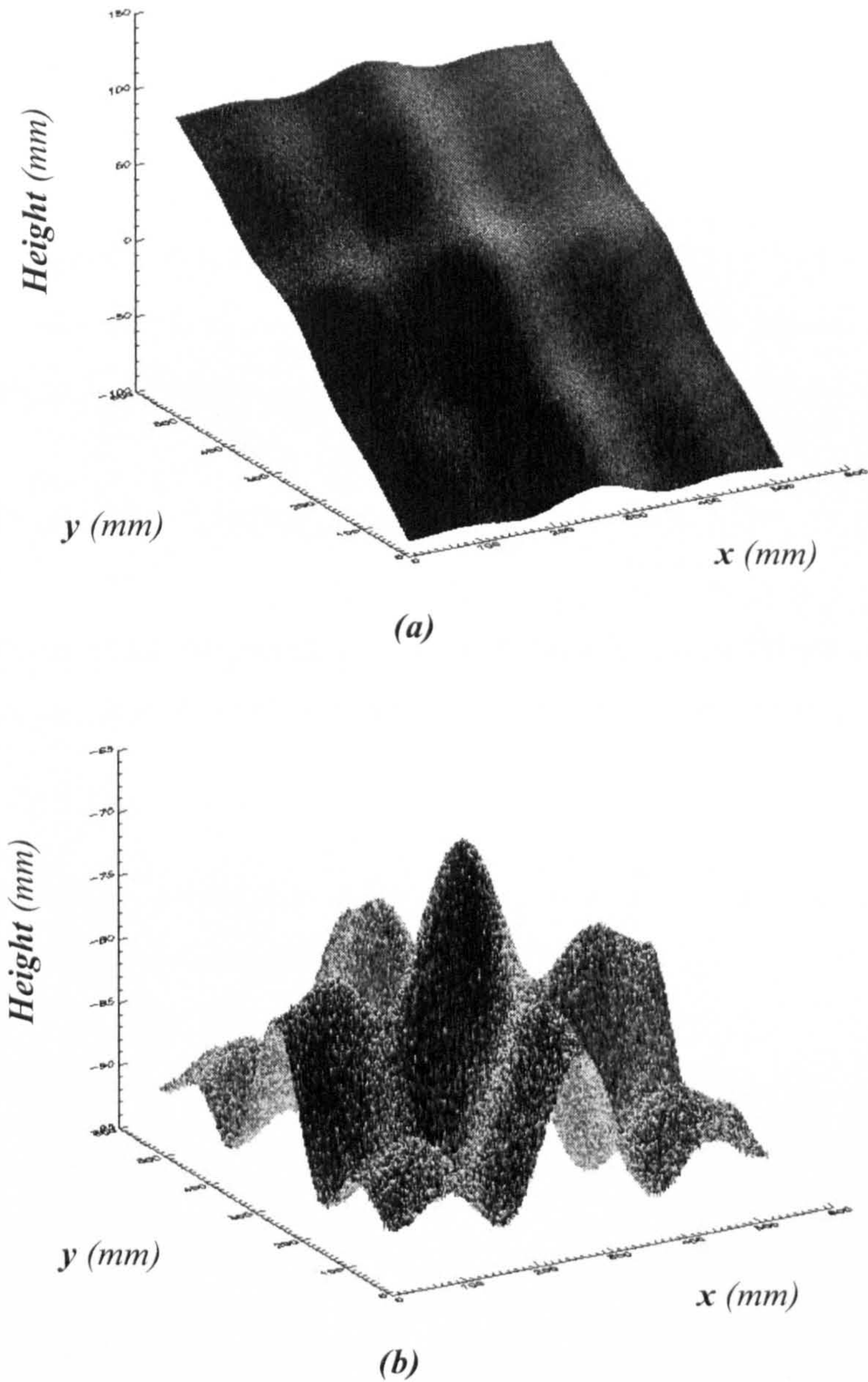
The wrapped-phase map resulted from applying the four-frame phase stepping algorithm is shown in Figure 2.10.



**Figure 2.10:** The wrapped-phase map resulting from the use of the four-frame phase stepping algorithm on a computer generated object.



Figure 2.11 shows a 3-D surface height of the unwrapped phase map. Figure 2.11(a) shows the unwrapped phase map without removing the carrier frequency, whereas Figure 2.11(b) shows the unwrapped and tilt-removed phase map (*i.e.* with the carrier frequency removed).



**Figure 2.11: Three-dimensional view of the unwrapped phase resulting from unwrapping the wrapped phase shown in Figure 2.9. (a) Without tilt removal & (b) with tilt removal**

## 2.4 Fourier Transform Fringe Analysis.

Fourier Transform Fringe Analysis (FTFA), also known as Fourier Fringe Analysis (FFA) or Fourier transform Profilometry (FTP), is a very popular technique that is well known to people working in the field of non-contact measurement. Since Takeda proposed his technique for using the Fourier transform in analysing a fringe pattern in 1982 (Takeda *et al.*, 1982), many applications have adopted this technique in order to measure 3D surface shape. A lot of further research has been carried out to improve the performance and quality of this technique.

As mentioned earlier, fringe analysis techniques have to extract the phase signal  $\varphi(x, y)$  from the intensity distribution of the source fringe pattern. An intensity profile  $g(x, y)$  in a fringe pattern may be given by the following equation:

$$g(x, y) = a(x, y) + b(x, y) \cos(2\pi f_0 x + \varphi(x, y))$$

In 1982, Takeda *et al.* suggested the use of a one-dimensional Fourier transform as a powerful tool to extract the phase information encoded in a fringe pattern (Takeda *et al.*, 1982).

Takeda expressed the cosine term in Equation (2.1) as a summation of two exponents as shown in the following equation:

$$g(x, y) = a(x, y) + c(x, y)e^{i2\pi f_0 x} + c^*(x, y)e^{-i2\pi f_0 x} \quad (2.15)$$

where,

$$c(x, y) = \frac{b(x, y)e^{i\varphi(x, y)}}{2} \quad (2.16)$$

$$c^*(x, y) = \frac{b(x, y)e^{-i\varphi(x, y)}}{2} \quad (2.17)$$



Taking the one-dimensional Fourier Transform we will have:

$$G(f_x, y) = A(f_x, y) + C(f_x - f_0, y) + C^*(f_x + f_0, y) \quad (2.18)$$

where,  $A(f_x, y)$  describes the spectrum of the background illumination,  $C(f_x - f_0, y)$  and  $C^*(f_x + f_0, y)$  are the spectra of the deformed fringes. By selecting either the  $C$  or  $C^*$  term using the window  $H$  as shown in Figure 2.12 below and shifting it toward the origin, as shown in Figure 2.13, we will isolate the term  $C(f_x, y)$  which contains the desired information.

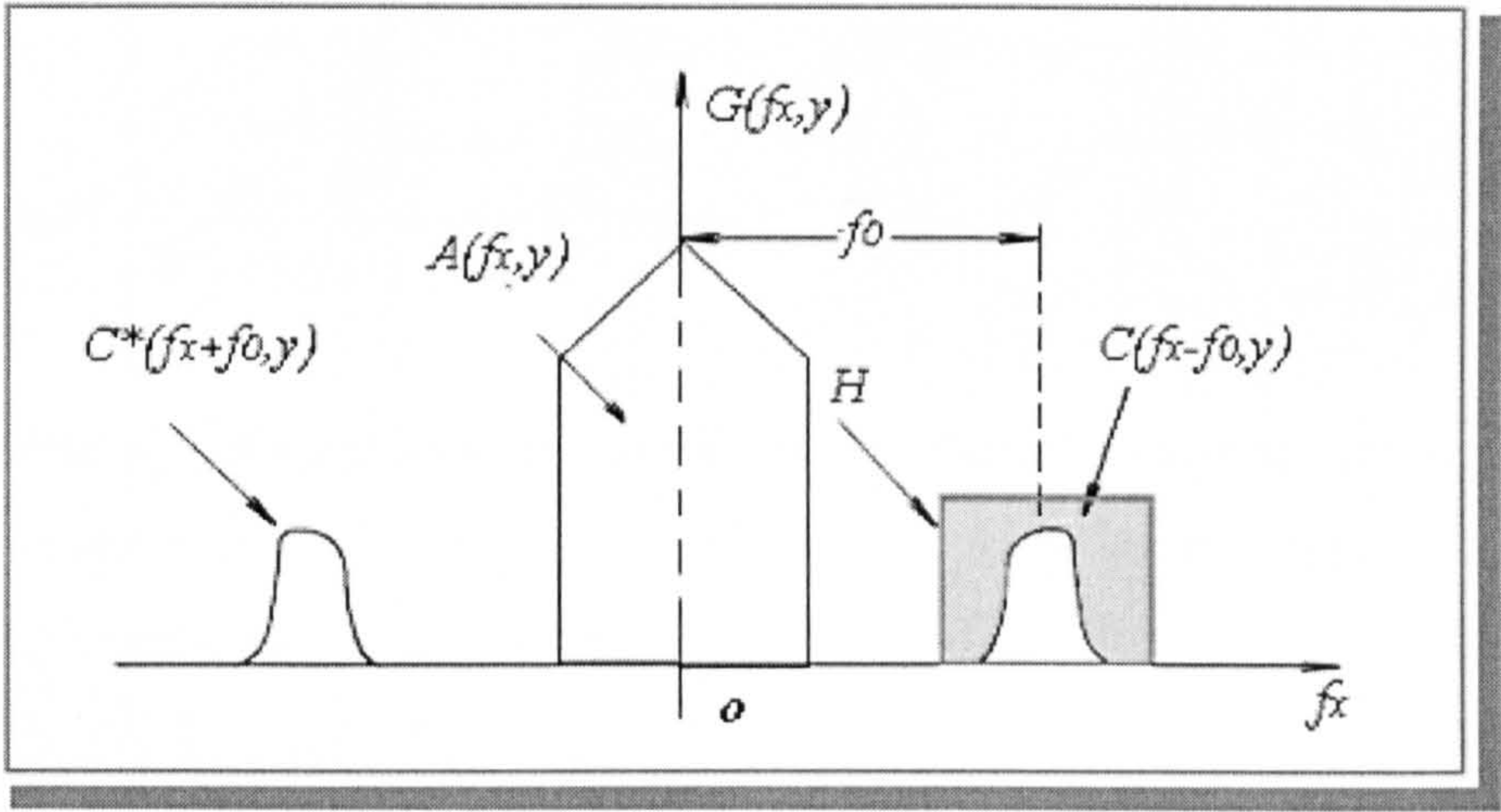


Figure 2.12: The frequency spectrum of fringe pattern.

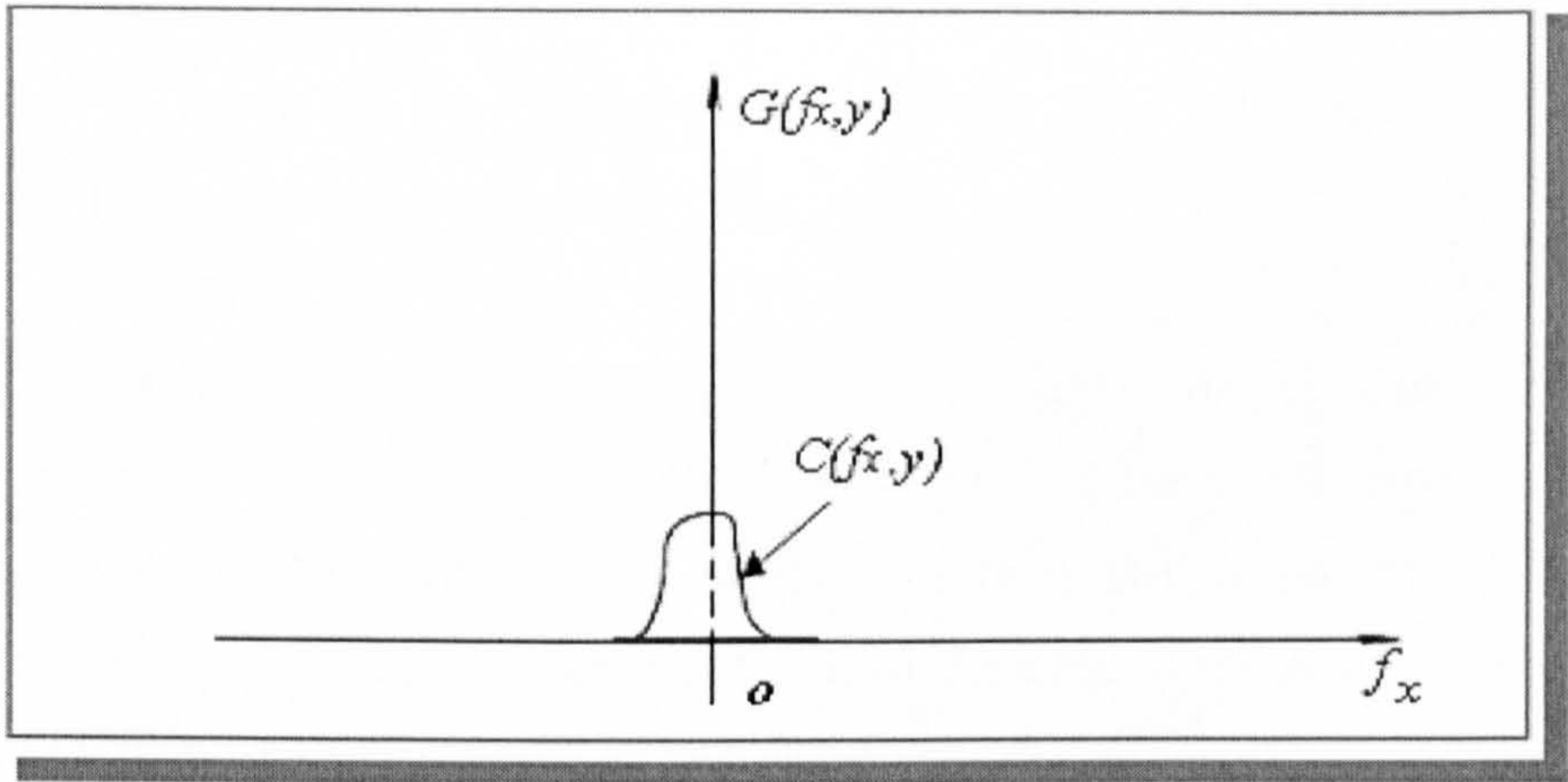


Figure 2.13: The desired frequency component



By transforming the term  $C(f_x, y)$  back into its original spatial domain using the one-dimensional Inverse Fourier Transform (1D-IFT), we will get  $c(x, y)$ .

The real and imaginary parts of  $c(x, y)$  are given by:

$$\Re\{c(x, y)\} = b(x, y) \cos(\varphi(x, y)) \quad (2.19)$$

$$\Im\{c(x, y)\} = b(x, y) \sin(\varphi(x, y)) \quad (2.20)$$

And the phase information is given by:

$$\psi(x, y) = \tan^{-1} \frac{\Im\{c(x, y)\}}{\Re\{c(x, y)\}} = \varphi(x, y) \mod \pi \quad (2.21)$$

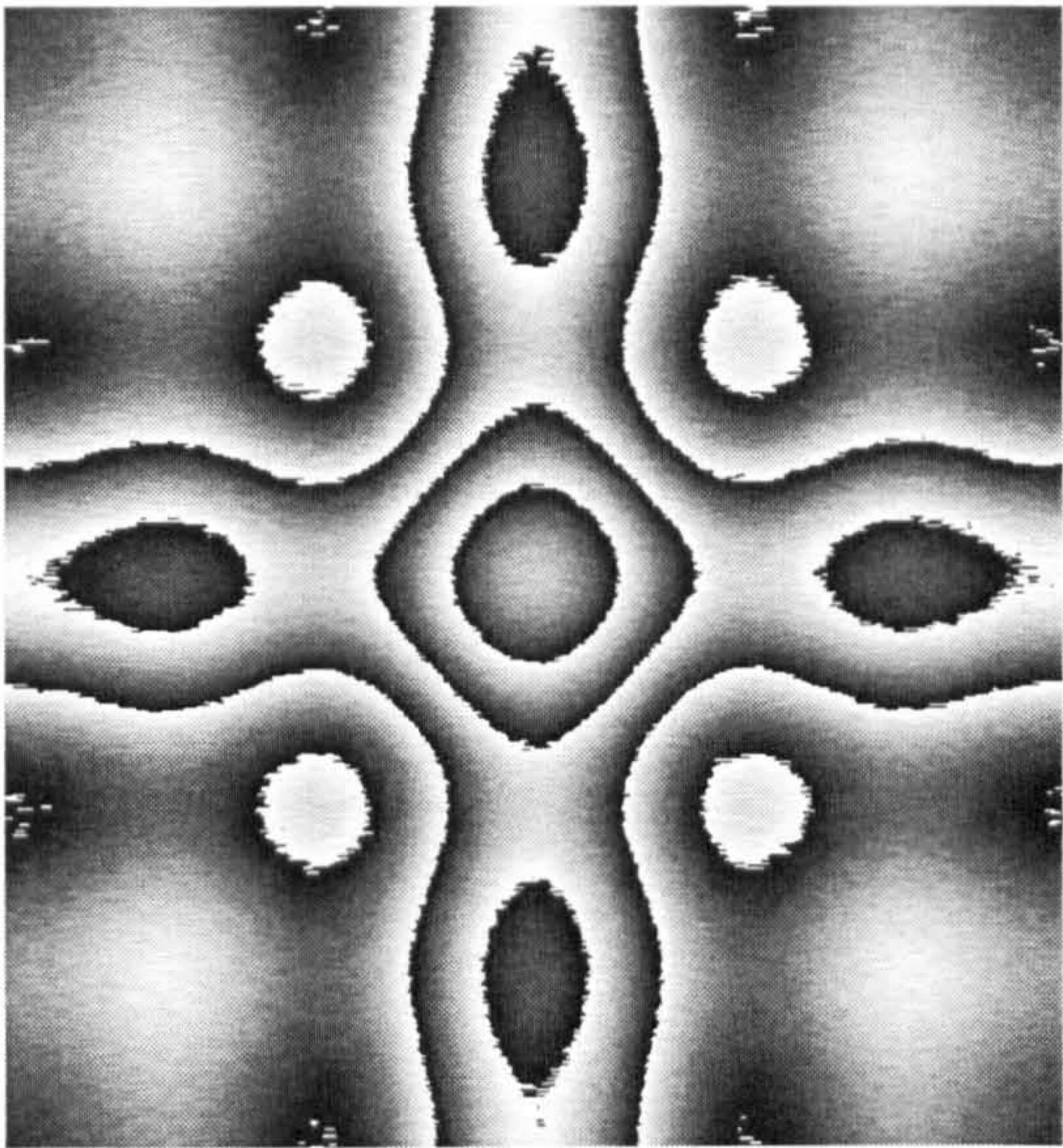
Because of the use of a trigonometric arctangent function, the extracted phase will be in the range  $-\pi$  and  $\pi$ , and will need to be unwrapped using phase unwrapping algorithms, which will be explained in the next chapter.

One-dimensional FFA is now applied to analyse the simulated fringe pattern shown in Figure 2.2, on a row by row basis. Where each row is analysed as described earlier. Figure 2.14(a) shows the resultant wrapped-phase map, whereas the unwrapped-phase surface is shown in Figure 2.14(b). Apparently from the figure, the 1D-FFA technique is less robust in analysing the example fringe pattern when compared to the phase-stepping and direct phase detection methods.

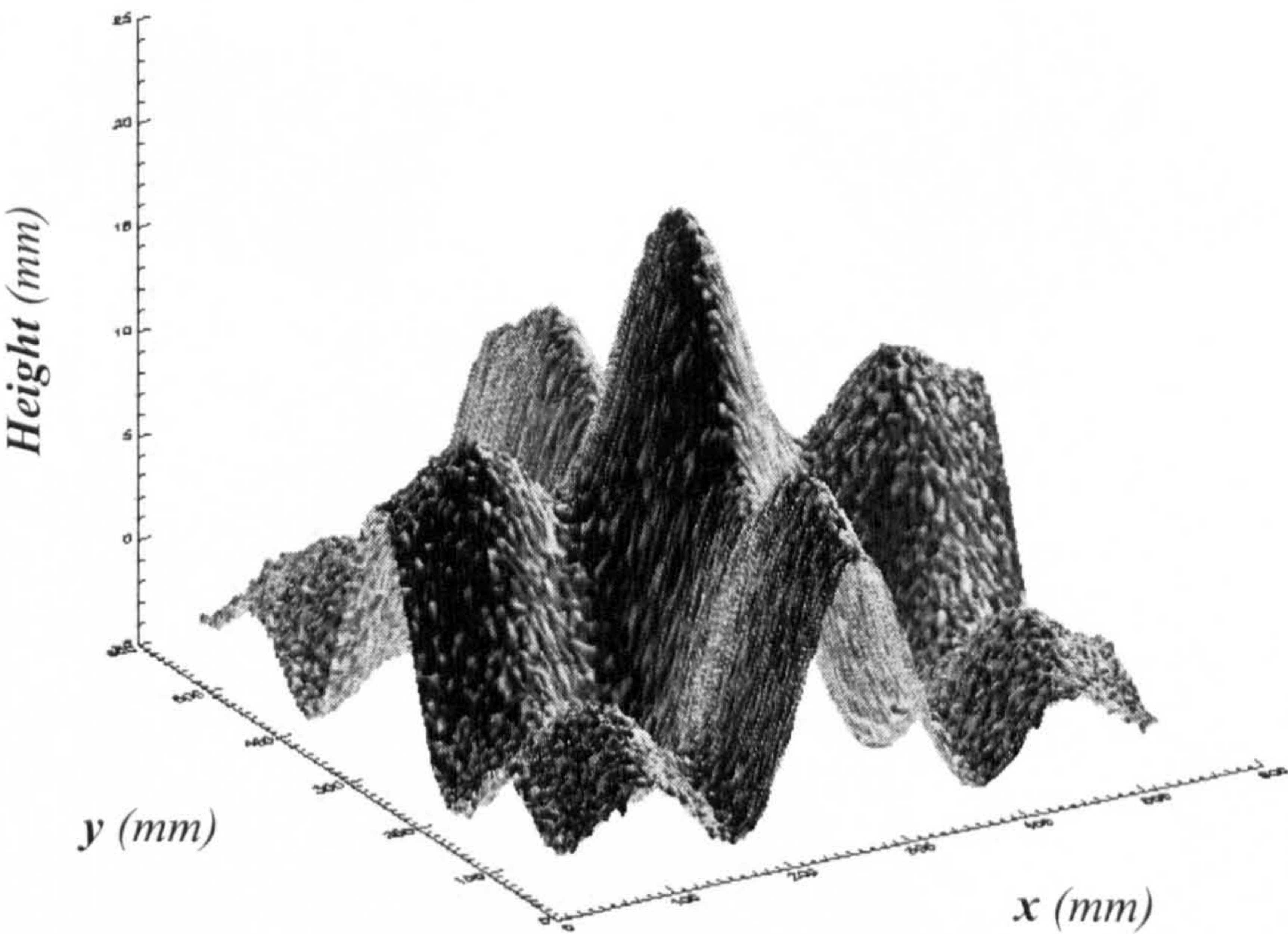
Since 1982, FFA has witnessed many improvements. Bone *et al.* introduced one major improvement in 1986 (Bone *et al.*, 1986). Bone proposed the use of two-dimensional Fourier transform to analyse the fringe pattern as a two-dimensional image. The two-dimensional transform permits better separation of the desired information components from the unwanted components. Using two dimensional FFA also increased the quality



of the results, with a corresponding slight increase in the processing time (Bone *et al.*, 1986).



(a)



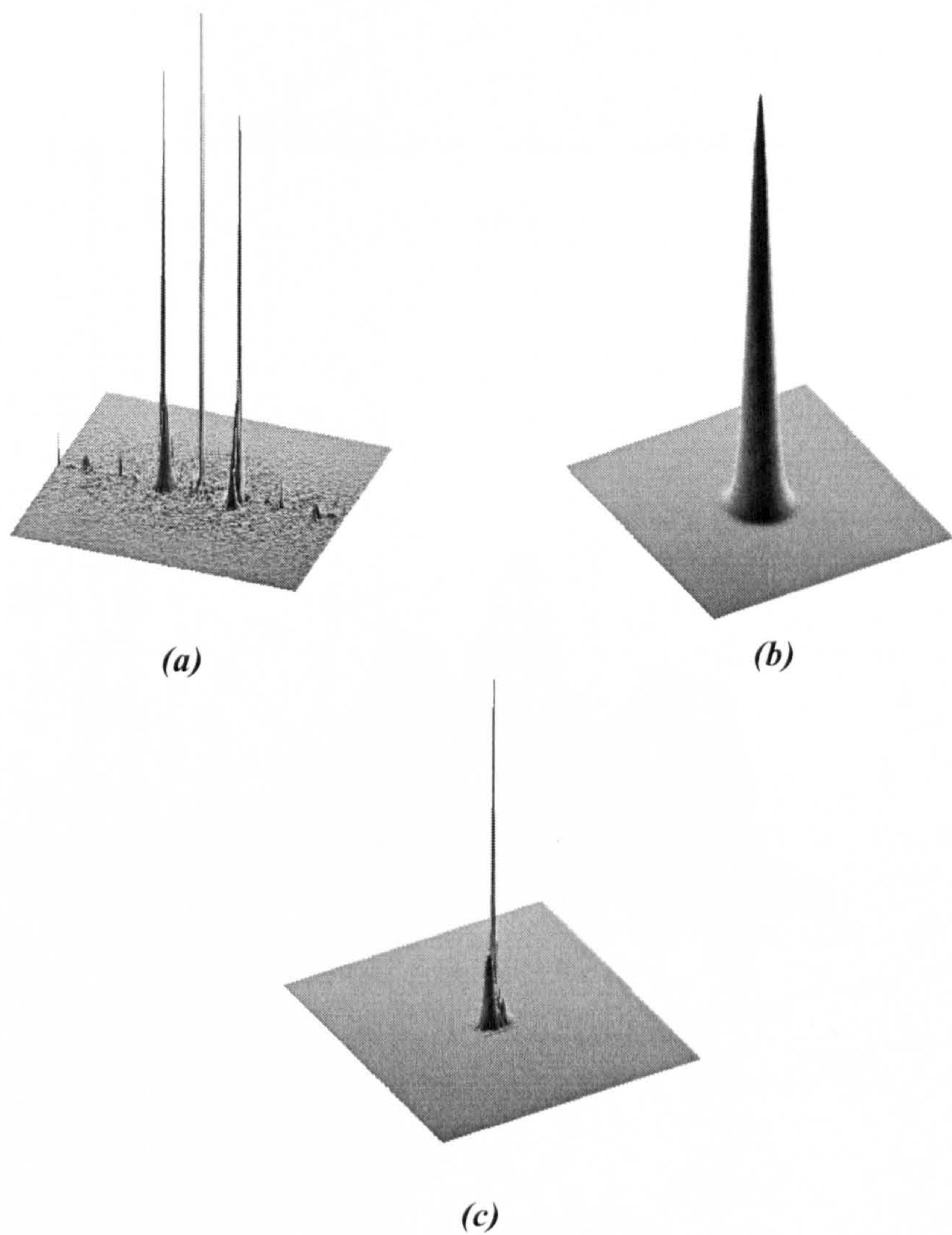
(b)

**Figure 2.14: (a) The wrapped-phase map resulting from using 1D-FFA. (b) A 3D representation of the unwrapped-phase resulting from using one-dimensional FFA.**

If the 2D-FFA technique is now applied to analyse the simulated fringe pattern shown in Figure 2.2, the results are as shown in Figure 2.15. This figure shows the individual

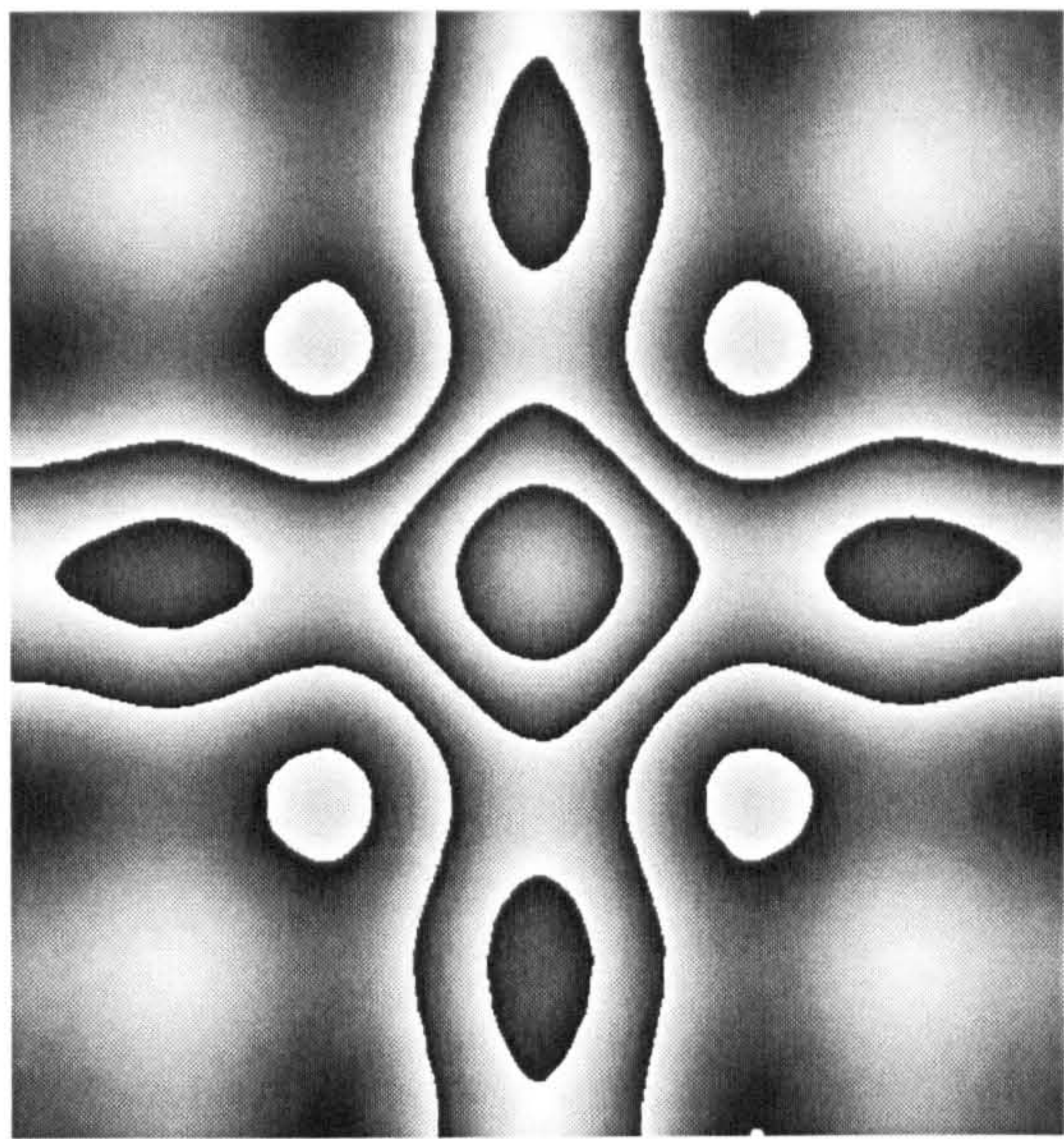


stages of the two-dimensional FFA method. Figure 2.15(a) shows the frequency spectrum of the fringe pattern. The desired component is selected using a two-dimensional exponential filter, which is shown in Figure 2.15(b), and the resulting filtered component is shown in Figure 2.15(c). Figure 2.16(a) shows the resultant wrapped-phase map in 2D form, whereas the unwrapped-phase surface is shown in Figure 2.16(b) as a 3D isometric plot. Comparing Figures 2.14(b) and 2.16(b) shows the great benefit in accuracy and data quality that is obtained from moving from 1D-FFA to 2D-FFA. It is this type of improvement that is a prime motivation towards investigating the potential benefits that could be obtained by taking into account a further third dimension in such analysis techniques.

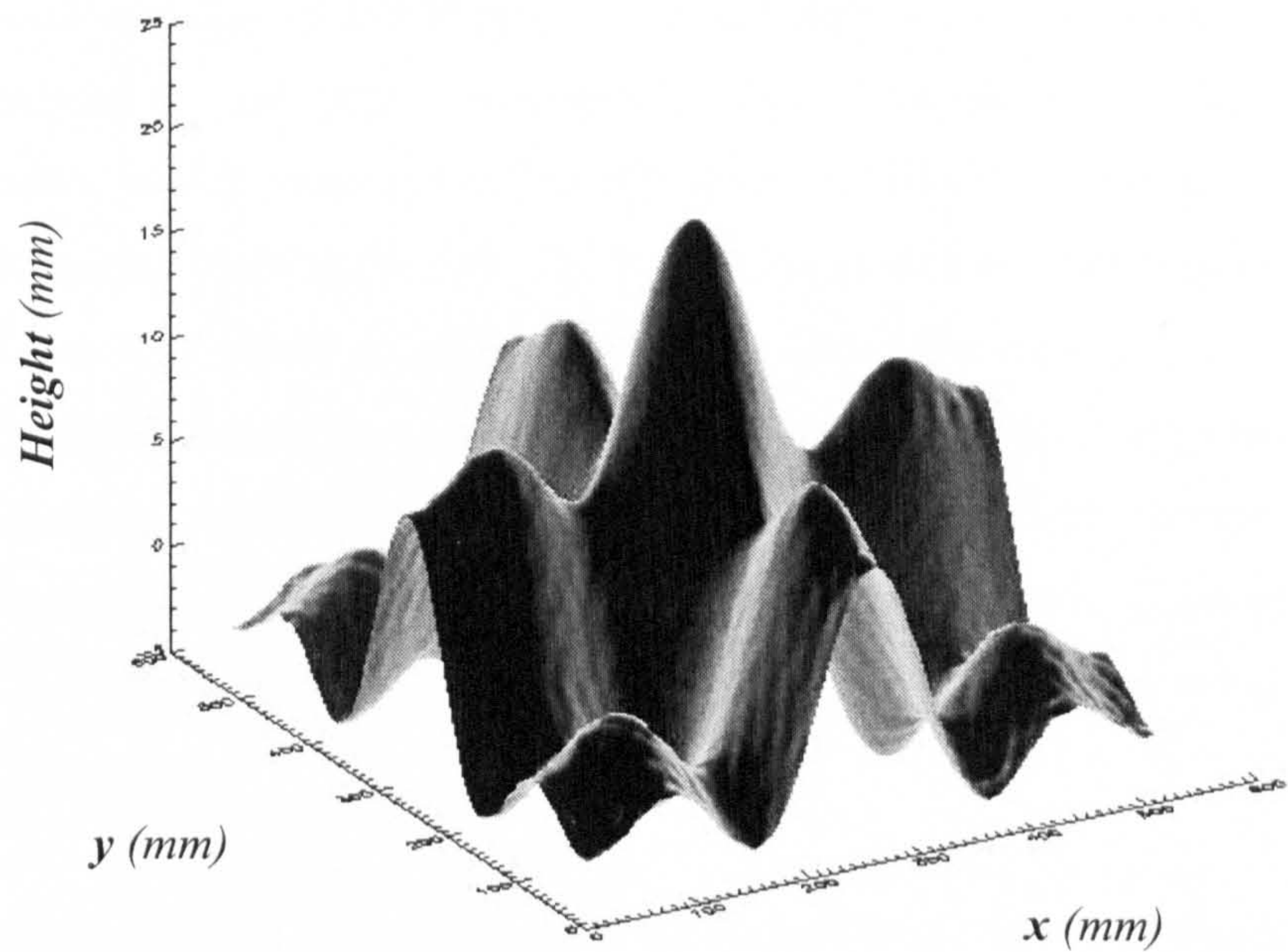


**Figure 2.15: The use of two-dimensional FFA. (a) the frequency spectrum of the fringe-pattern shown in Figure 2.13(a), (b) 2-D filter to select the desired frequency component. (c) The selected frequency component.**





(a)



(b)

Figure 2.16: (a) The wrapped-phase map that resulted from using 2D-FFA. (b) A 3D representation of the unwrapped-phase that resulted from using 2D-FFA.



After the two-dimensional Fourier fringe analysis algorithm was first published, it was subject to intensive study in order to improve its performance. Many researchers proposed an elimination of the zero frequency component, in order to make the frequency selection much easier and to extend the measurable slope of the height variation. For example; Li *et al.* in 1990 (Li *et al.*, 1993) proposed a method to eliminate the zero frequency component by projecting two fringe patterns onto the object with a phase shift of  $\pi$  radians between them, and hence subtracting the patterns from each other effectively eliminates the zero frequency component.

The Fourier fringe analysis algorithm suffers from some problems however, such as leakage and difficulty in resolving sharp-edges. Frequency leakage can be minimized by multiplying the fringe pattern by a smoothing window, such as a Hamming window. Fiona Berryman *et al.* in 2004, studied the effect of windowing in the Fourier transform, and showed the performance of applying different types of windows (Berryman *et al.*, 2004). A challenging problem is to distinguish between a discontinuity produced by a real sharp edge on the object and that produced by a phase discontinuity. David Burton *et al.* (Burton and Lalor, 1994) proposed a technique termed as multi-channel Fourier fringe analysis to aid phase unwrapping algorithms in the presence of surface discontinuities, which projected different patterns at different angles and frequencies to resolve the surface discontinuities. The basic idea of this algorithm is that the surface discontinuities will appear on all fringe patterns. David Burton *et al.* in 1995 (Burton *et al.*, 1995) show the effect of removing the carrier frequency by carrier frequency shifting in order to minimize the complexity of the wrapped phase map and to avoid phase discontinuities in some particular applications. Recently Chen *et al.* in 2005 (Chen *et al.*, 2005) proposed an elimination of the zero spectrum using a windowed Fourier transform.

Nowadays, the concern is increasingly towards measuring dynamic objects. Xianyu Su *et al.* in 2001 (Su *et al.*, 2001) proposed the use of two-dimensional FFA for shape measurement of dynamic objects. In this method a CCD camera rapidly grabs a sequence of images. These sequences are analysed frame by frame using 2D-FFA to produce a wrapped phase volume which can be unwrapped using 2D or 3D phase unwrapping algorithms, as will be shown later. Xianyu Su *et al.* in 2001 (Su and Chen, 2001) presented a thorough review about Fourier transform fringe pattern analysis.



## 2.5 Wavelet Transform Fringe Analysis.

Recently, there has been much interest in phase demodulation of fringe patterns using wavelet transforms. Wavelet transform fringe analysis is a frequency-domain fringe analysis technique, in which the demodulation of the phase is carried out by transferring the fringe pattern into a different domain. It is well known in digital signal processing theory that the wavelet transform technique is more suitable for the analysis of non-stationary signals rather than stationary signals. A stationary signal is a signal whose frequency contents does not change in time, whereas a non-stationary signal is a signal whose frequency contents do change in time (Malat, 1999). Fringe patterns tend to resemble non-stationary signals. This has motivated researchers to investigate wavelet transform techniques for the application of fringe pattern demodulation.

Many researches have proposed one-dimensional continuous wavelet transform (1D-CWT) techniques to demodulate fringe patterns. These algorithms extract the phase of fringe patterns and can be classified into two different approaches: phase estimation (Zhong and Weng, 2004) and frequency estimation (Sandoz, 1997; Cherbuliez *et al.*, 1999; Dursun *et al.*, 2004).

In the phase estimation method, the fringe pattern is applied to the 1D-CWT algorithm on a row by row basis and the resultant wavelet transform is a two-dimensional complex array. To extract the phase, this complex array is decomposed into two arrays, one for the absolute values and the other for the phase values. Then, from the absolute value array, the maximum value of each column in that array has to be extracted using one of several available ridge extraction algorithms (Delprat *et al.*, 1992; Carmona *et al.*, 1997; Liu *et al.*, 2004). After extracting the ridge, the wrapped phase of the row under processing is represented by the corresponding phase values of that ridge. Then the wrapped phase has to be unwrapped, after which it can be used to calculate the actual height values. A review of phase unwrapping algorithms will be given in the next chapter. This process is repeated until all lines in the fringe pattern are analysed.

The second approach for retrieving the phase of a fringe pattern is to estimate the instantaneous frequencies in the fringe pattern. In the frequency estimation method, the

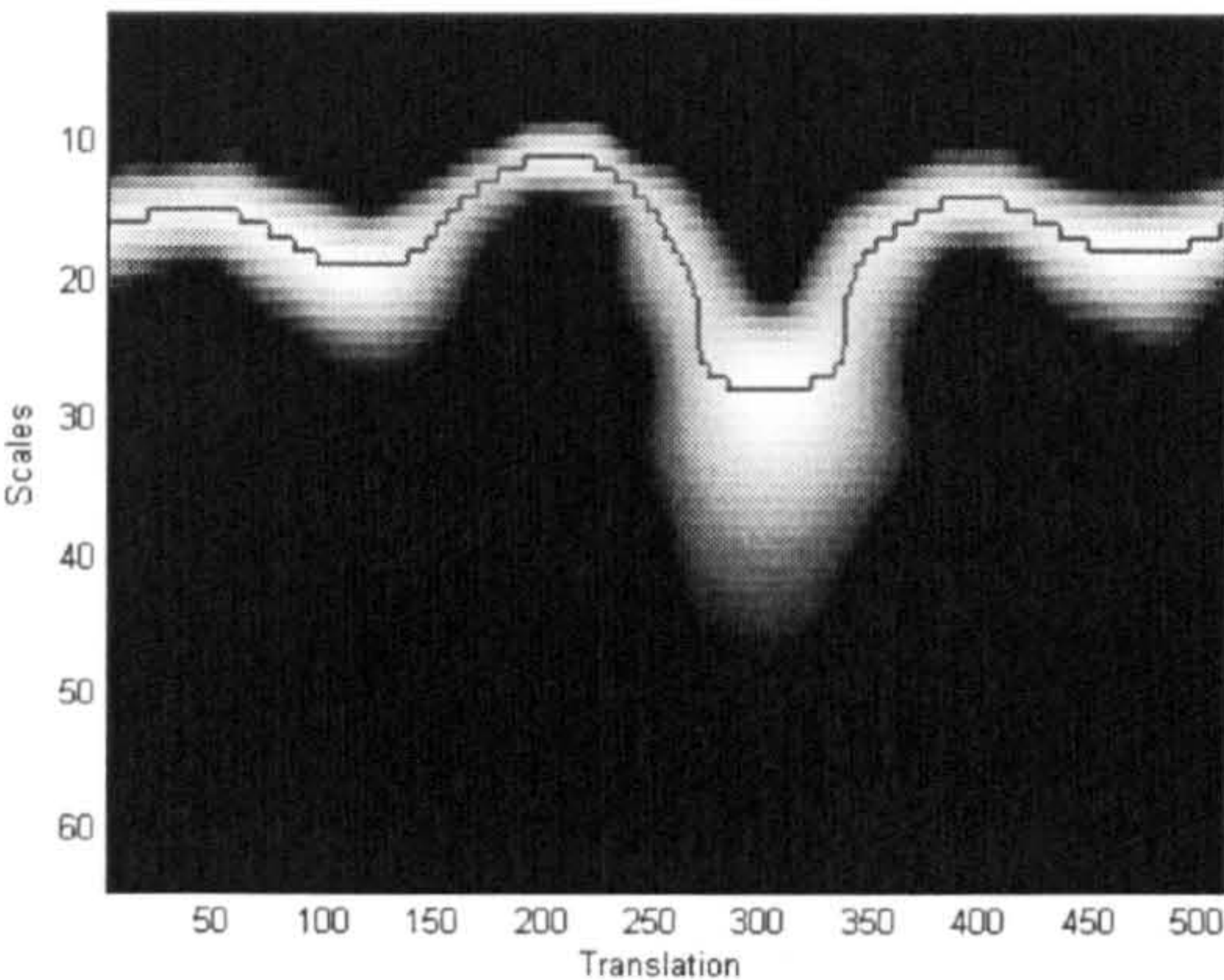


ridge of maximum values has to be extracted in a similar manner to that used in the phase estimation method. After identifying the ridge, the scale values are extracted from the positions where this ridge is located. The unwrapped phase is then calculated by integrating the scale values. In this method phase unwrapping is not required, as the integration of the scale values will directly produce the unwrapped phase. Gdeisat *et al.* show that the frequency estimation method is not reliable and it fails to demodulate real fringe patterns (Gdeisat *et al.*, 2006).

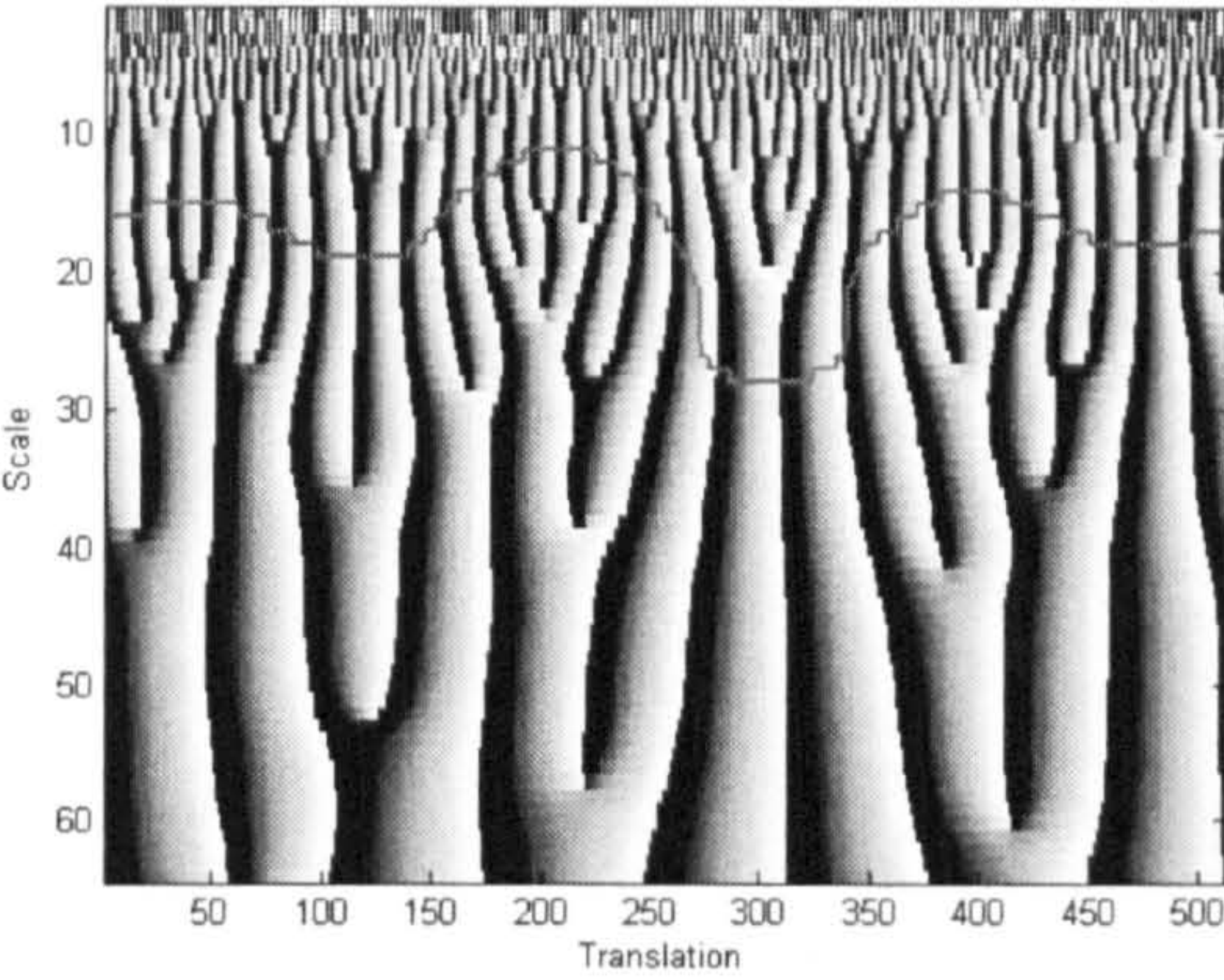
Figure 2.17 shows an illustration of the use of the phase estimation method to extract the phase from the simulated fringe pattern that was originally shown in Figure 2.2. The implementation of the wavelet fringe analysis algorithm has been carried out by author's colleague working on wavelet fringe analysis area in Liverpool John Moores University. Figure 2.17(a) shows the absolute value array resulting from applying a row of this fringe pattern to a 1D-CWT. As shown in Figure 2.17(a), the ridge of maximum values is extracted and is represented in this figure by a solid line. Figure 2.17(b) shows the phase value array for the same row. The phase values correspond to the ridge extracted in Figure 2.17(a) and represent the wrapped phase for that particular row. Figure 2.17(c) shows the complete wrapped phase map resulting from analysing all rows and a 3D view of the unwrapped phase is shown in Figure 2.17(d).

Recently, many researchers have proposed the use of the 2D-CWT to analyse fringe patterns (Kadooka *et al.*, 2003; Gdeisat *et al.*, 2006). Using the 2D-CWT is more robust for demodulation of fringe patterns and has better noise immunity than the 1D-CWT method (Gdeisat *et al.*, 2006).

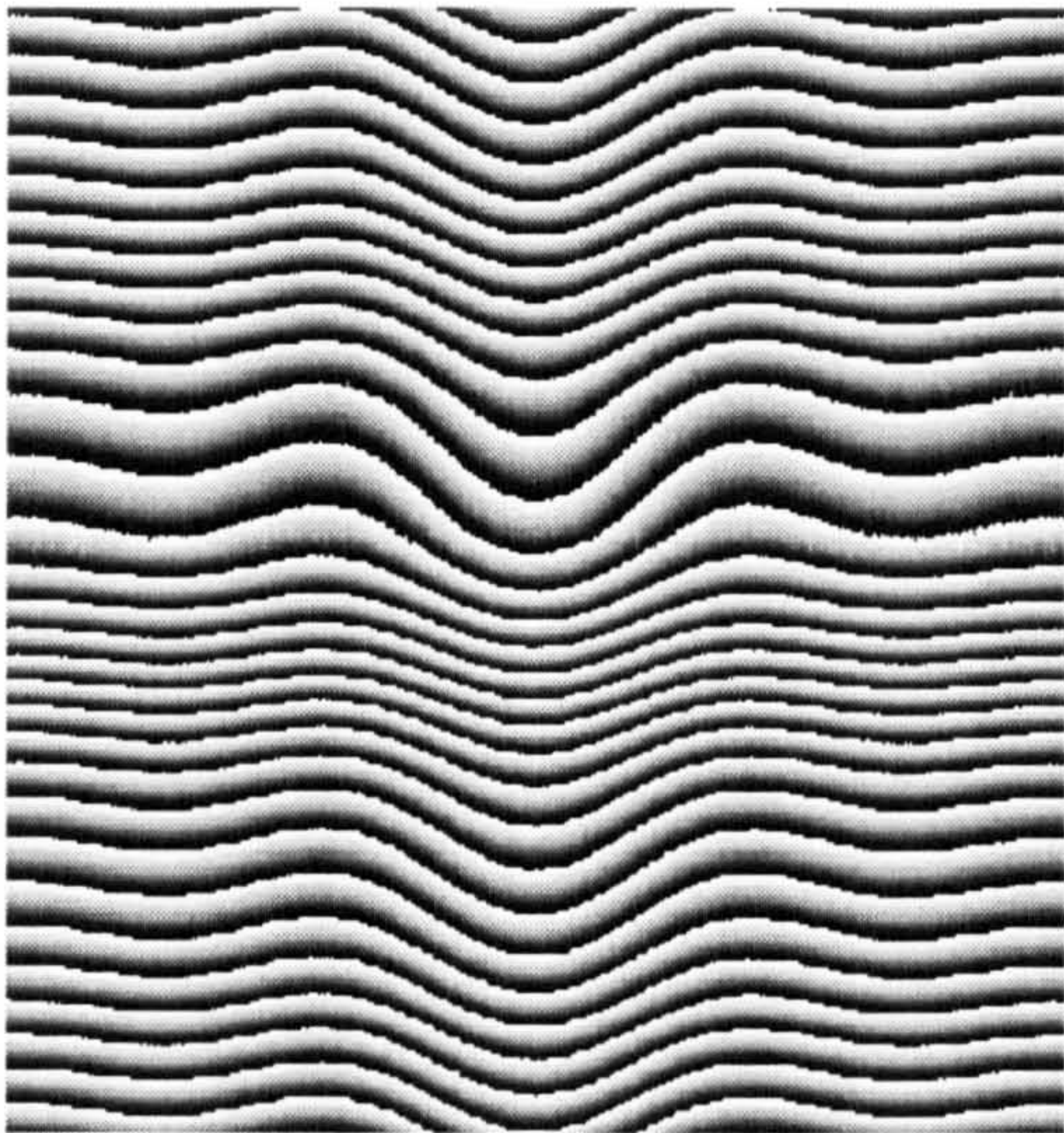




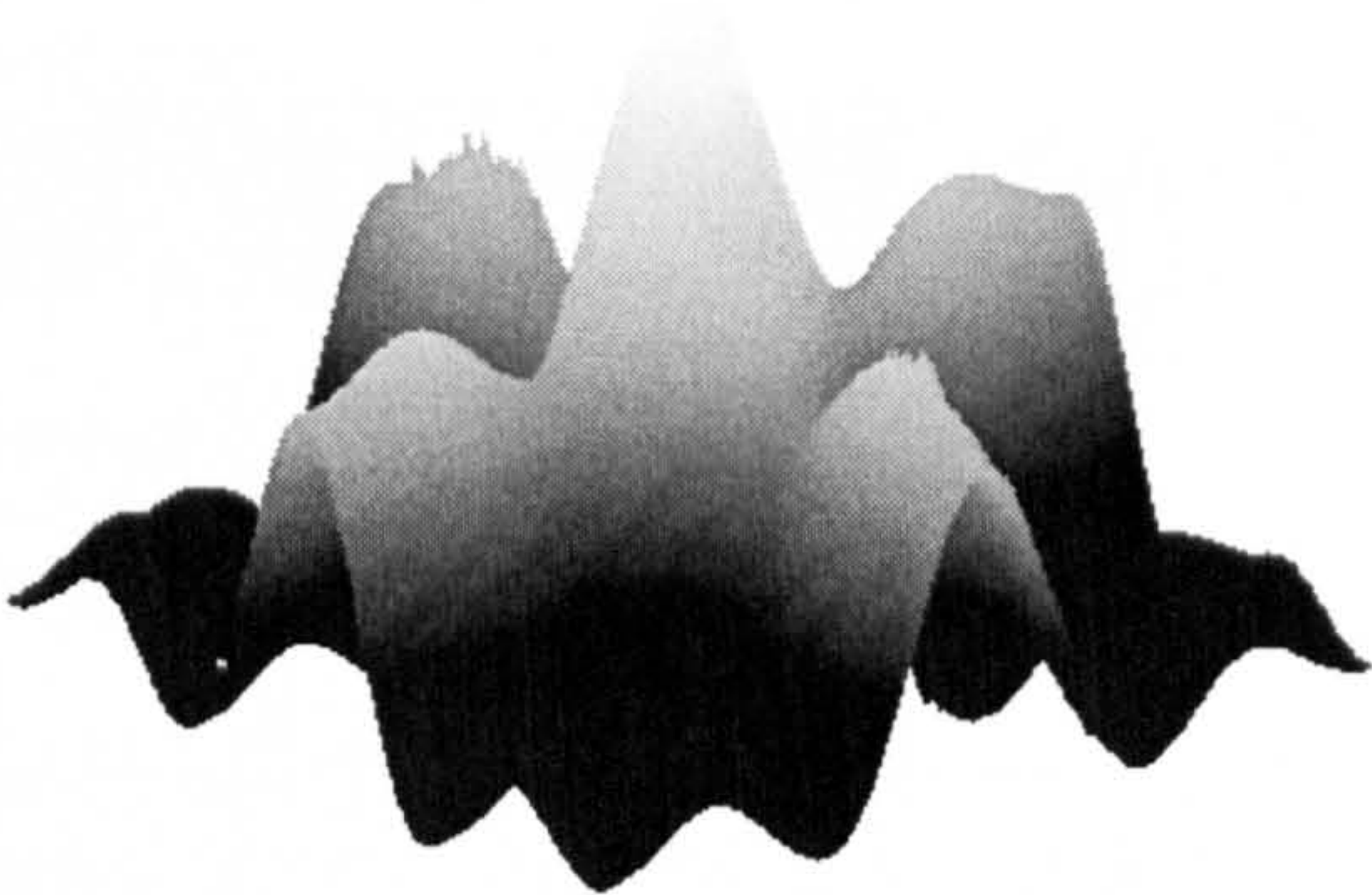
(a)



(b)



(c)



(d)

**Figure 2.17: Wavelet transform fringe analysis. (a) the extracted ridge from the modulus array resulting from applying one row of fringe pattern shown in Figure 2.2 to the 1D-CWT, (b) the corresponding ridge in the phase array, (c) the wrapped phase map and (d) the unwrapped phase map**

(Results obtained from Mr. Abdulbasit Abid, Liverpool John Moores University)



2.6 Summary

There are many algorithms that can be used to analyse fringe patterns. A number of these algorithms have been reviewed in this chapter. Phase stepping techniques require low computational complexity. The main drawback of the phase stepping technique is the requirement for more than one frame to be captured (three at least) for the analysis of an object. This requirement makes phase stepping an inappropriate choice for dynamically moving objects. FFA and WTFA offer good results and only require a single frame to be captured in order to produce a result. On the other hand, FFA and WTFA are very complicated algorithms. Generally speaking, there is no ideal fringe analysis technique that can withstand the very wide variety of different practical requirements. Every technique has advantages and drawbacks.

Table 2.1 shows a comparison between the four algorithms discussed in this chapter.

Table 2.1. Comparison of fringe analysis techniques

	DPD <sup>1</sup>	Phase stepping	FFA <sup>2</sup>	WFA <sup>3</sup>
Analysis domain	Spatial domain	Spatial domain	Frequency domain	Spatial-frequency domain
Computational complexity	Medium	Low	High	Very High
Noise Robustness	Medium	Medium	High	High
Number of frames required	One frame	Three frames at least	One frame	One frame
Possibility for analysing dynamic objects	Possible	Not possible	Possible	Possible
Coping with discontinuous and high detailed objects	Not good	Very good	Not good	Not good
Ease of automation	Low	High	Low	Medium
<sup>1</sup> DPD: Direct Phase Detection. <sup>2</sup> FFA: Fourier Fringe Analysis. <sup>3</sup> WFA: Wavelet Fringe Analysis.				



## References:

- Berryman, F.; Pynsent, P. and Cubillo, J. (2004) The Effect of Windowing in Fourier Transform Profilometry Applied to Noisy Images. *Optics and Lasers in Engineering*, 41, 815-825.
- Bone, D. J.; Bachor, H. A. and Sandeman, R. J. (1986) Fringe-Pattern Analysis Using a 2-D Fourier Transform. *Applied Optics*, 25, 1653-60.
- Bruning, J. H. and al., e. (1974) Digital Wavefront Measuring Interferometer for Testing Optical Surfaces and Lenses. *Applied Optics*, 13, 2693-2703.
- Burton, D. R. and Lalor, M. J. (1994) Multichannel Fourier Fringe Analysis as an Aid to Automatic Phase Unwrapping. *Applied Optics*, 33, 2939-48.
- 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, 245-57.
- Carmona, R. A.; Hwang, W. L. and Torresani, B. (1997) Characterization of Signals by the Ridges of Their Wavelet Transforms. *IEEE Transactions on Signal Processing*, 45, 2586-2590.
- Chen, W.; Su, X.; Cao, Y.; Qc, Z. and Xiang, L. (2005) Method of Eliminating Zero Spectrum in Fourier Transform Profilometry. *Optics and Lasers in Engineering*, 43, 1267-1276.
- Cherbuliez, M.; Jacquot, P. and Colonna de Lega, X. (1999) Wavelet Processing of Interferometric Signals and Fringe Patterns. Denver, CO, USA, SPIE-Int. Soc. Opt. Eng.
- Creath, K. (1993) *Interferogram Analysis: Digital Fringe Pattern Measurement Techniques*, Institute of Physics Publishing, Bristol and Philadelphia.
- Creath, K. and Schmit, J. (1994) Errors in Spatial Phase-Stepping Techniques. *SPIE, New Techniques and Analysis in Optical Measurements*, 2340, 170-176.
- Delprat, N.; Esudie, B.; Gillemanni, P.; Kronland-Martinet, R.; Tchamitchian, P. and Torresani, B. (1992) Asymptotic Wavelet and Gabor Analysis: Extraction of Instantaneous Frequency. *IEEE Transactions on Information Theory*, 38, 644-664.
- Dursun, A.; Ozder, S. and Ecevit, F. N. (2004) Continuous Wavelet Transform Analysis of Projected Fringe Patterns. *Measurement Science and Technology*, 15, 1768-1772.



- Gdeisat, M. A.; Burton, D. R. and Lalor, M. J. (2006) Spatial Carrier Fringe Pattern Demodulation by Use of a Two-Dimensional Continuous Wavelet Transform. *Applied Optics*, 45, 8722-8732.
- Hariharan, P.; Oreb, B. F. and Eiju, T. (1987) Digital Phase-Shifting Interferometry: A Simple Error-Compensating Phase Calculation Algorithm. *Applied Optics*, 26, 2504-2505.
- Herraez, M. A.; Burton, D. R.; Lalor, M. J. and Gdeisat, M. A. (2002) Fast Two-Dimensional Phase-Unwrapping Algorithm Based on Sorting by Reliability Following a Noncontinuous Path. *Applied Optics*, 41, 7437-44.
- Hibino, K.; Oreb, B. F. and Farrant, D. I. (1995) Phase Shifting for Non-Sinusoidal Waveforms With Phase-Shift Errors. *Journal of Optical Society of America*, 12, 761-767.
- Hobson, C. A.; Atkinson, J. T. and Lilley, F. (1997) The Application of Digital Filtering to Phase Recovery When Surface Contouring Using Fringe Projection Techniques. *Optics and Lasers in Engineering*, 27, 355-68.
- Ichioaka, Y. and Inuiya, M. (1972) Direct Phase Detection System. *Applied Optics*, vol. 11, pp. 1507-1514.
- Kadooka, K.; Kunoo, K.; Uda, N.; Ono, K. and Nagayasu, T. (2003) Strain Analysis for Moire Interferometry Using the Two-Dimensional Continuous Wavelet Transform. *Proceedings of the Society for Experimental Mechanics, Inc*, 50, 45-51.
- Li, J.; Su, X. R. and Guo, L. R. (1993) Improved Fourier Transform Profilometry for the Automatic Measurement of Three-Dimensional Object Shapes. *Optical Engineering*, 29, 1439-1444.
- Liu, H.; Cartwright, A. N. and Basaran, C. (2004) Moire Interferogram Phase Extraction: A Ridge Detection Algorithm for Continuous Wavelet Transforms. *Applied Optics*, 43, 850-857.
- Malat, S. (1999) *A Wavelet Tour of Signal Processing*, Academic press, Second Edition.
- Sandoz, P. (1997) Wavelet Transform as a Processing Tool in White-Light Interferometry. *Optics Letters*, 22, 1065-7.
- Su, X. and Chen, W. (2001) Fourier Transform Profilometry: A Review. *Optics and Lasers in Engineering*, 35, 263-284.



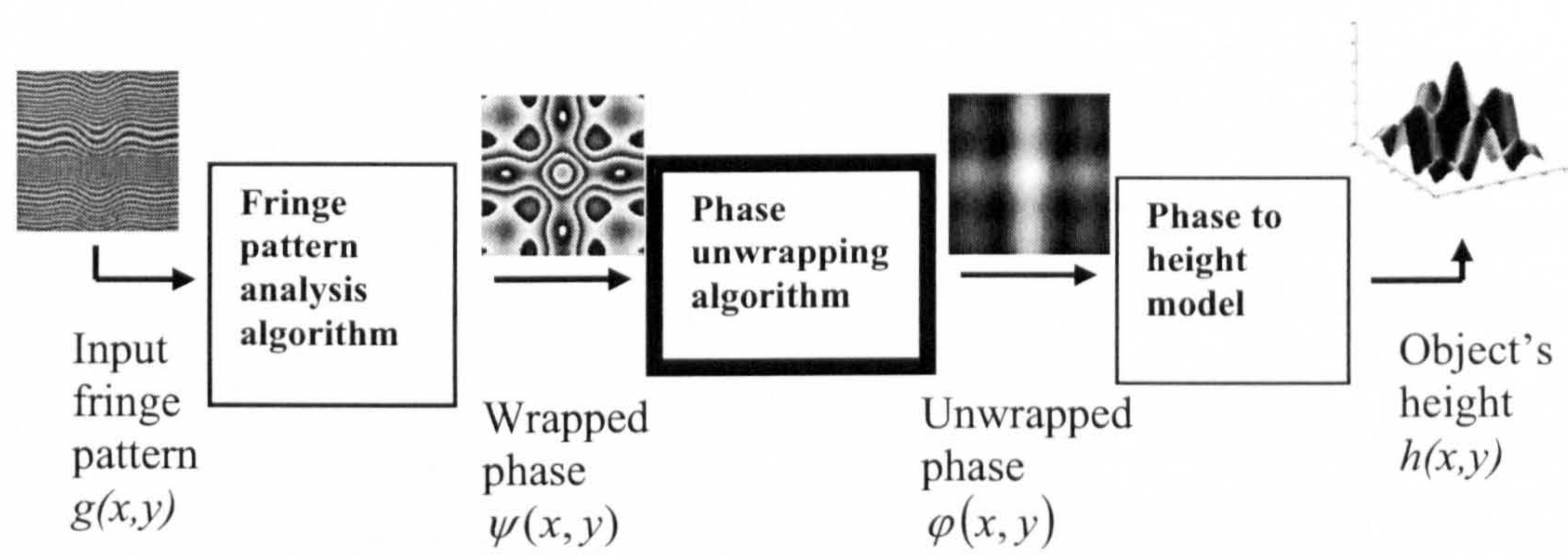
- Su, X.; Chen, W.; Qc, Z. and Chao, Y. (2001) Dynamic 3-D Shape Measurment Method Based on Ftp. *optics and Lasers in Engineering*, 36, 46-64.
- Takeda, M.; Ina, H. and Kobayashi, S. (1982) Fourier-Transform Method of Fringe-Pattern Analysis for Computer-Based Topography and Inteferometry. *Journal of the Optical Society of America*, 72, 156-160.
- Wyant, C. J. and Creath, K. (1992) Advances in Interferometric Optical Profiling. *International Journal of Machine Tools Manufacturing*, 32, 5-10.
- Zhong, J. and Weng, J. (2004) Spatial Carrier-Fringe Pattern Analysis by Means of Wavelet Transform: Wavelet Transform Profilometry. *Applied Optics*, 43, 4993-4998.



*Chapter Three*

*A Review of 2D Phase Unwrapping*

*Algorithms*





## Chapter Three

### A Review of Two-Dimensional Phase Unwrapping Algorithms

---

#### 3.1 Introduction

Many fringe analysis techniques such as; Fourier transform fringe analysis, phase stepping, wavelet transform fringe analysis... *etc*, end up with the use of the arctangent function, as shown in the previous chapter. Due to the use of this trigonometric function, the extracted phase is typically wrapped into the interval  $(-\pi, \pi)$  and discontinuities of values of  $2\pi$  appear and thus need to be unwrapped in order to provide the required continuous phase information.

Phase unwrapping can be defined to be “the process by which the absolute value of the phase angle of a continuous function that extends over a range of more than  $2\pi$  is recovered. This absolute value is lost when the phase term is wrapped upon itself with a repeat distance of  $2\pi$  due to the fundamental sinusoidal nature of the wave functions used in measurement of physical properties” (Robinson and Reid, 1993). Or simply it can be also be defined as “the process of solving the ambiguity problem caused by the fact that the absolute phase is wrapped into the interval  $(-\pi, \pi)$ , recovering the continuous phase information from the wrapped phase” (Gens, 2003).

In the case where there are no disturbances in the phase data, the unwrapped phase can be easily obtained by integrating the phase gradients over the whole data set and this may be performed in a manner that is completely independent of the actual integration path. But, there are several sources of disturbances that cause this simple unwrapping approach to fail in the vast majority of cases. For example: under-sampling, noise, object discontinuities or fringe breaks, holes or missing phase information, ...*etc* (Huntley, 1989).



During the last three decades, the field of two-dimensional phase unwrapping has been intensively studied and hundreds of journal papers have been published. Many techniques were proposed to solve the phase unwrapping problem. These algorithms vary in accuracy and computational requirements. In most cases, more accurate results require more complex computations. These techniques can be roughly classified into two major categories; local and global phase unwrapping algorithms. A complete review of the two-dimensional phase unwrapping problem has been presented by Ghiglia and Pritt (Ghiglia and Pritt, 1998).

Local phase-unwrapping algorithms, also called path-following algorithms, unwrap the phase map by integrating the phase gradients over a certain path that connects all pixels in the wrapped phase map. Unwrapped phase depends on the choice of the integration path, *i.e.* if the unwrapping algorithm follows two different paths from one pixel to another, it may produce two different answers for the unwrapped phase.

Global phase-unwrapping algorithms formulate the unwrapping process in terms of minimization of a global function to estimate the phase gradient; they do not rely on an integration path to perform unwrapping. All the algorithms in this class are known to be robust but computationally intensive. A comparison between local and global phase unwrapping algorithms was carried out by Fornaro *et al.* (Fornaro *et al.*, 1997a; Fornaro *et al.*, 1997b).

In this chapter, the major local and global phase unwrapping algorithms will be reviewed. In section 3.2, local phase unwrapping algorithms are discussed and explained. Global algorithms are discussed in section 3.3.

### 3.2 Local phase unwrapping algorithms

Local phase unwrapping algorithms find the unwrapped phase values by integrating the phase along a certain path. They follow an integration path that covers the whole wrapped phase map and this is why local phase unwrapping algorithms are also called path-following algorithms (Ghiglia and Pritt, 1998). In the local phase unwrapping methods the unwrapped phase is defined by:



$$\varphi(b) = \varphi(a) + \sum_{i=a}^{b-1} \Delta(\varphi(i)) \quad (3.1)$$

where;

$\varphi(b)$  the unwrapped phase at an arbitrary point  $b$

$\varphi(a)$  the unwrapped phase at the initial point  $a$

$\Delta$  is the difference operator where

$$\Delta\{\varphi(n)\} = \varphi(n+1) - \varphi(n) \quad (3.2)$$

Equation 3.1 states that the unwrapped phase value at pixel  $b$  is equal to the unwrapped phase value at the pixel  $a$  plus the summation (integration) of the unwrapped phase differences among a certain path from  $a$  to  $b$ . Actually, Equation 3.1 is a general equation that can be applied to any continuous function, which states that any function can be expressed as a summation of its difference. On that basis, Equation 3.1 cannot be used to determine the unwrapped phase solution and needs to be modified.

The unwrapped phase values and the wrapped phase values can be related with each other by: (Itoh, 1982)

$$\psi(n) = \varphi(n) + 2\pi \cdot k(n) \quad (3.3)$$

$$-\pi < \psi(n) \leq \pi$$

$$\varphi(n) = \psi(n) + 2\pi \cdot v(n) \quad (3.4)$$

$$-\infty < \varphi(n) < \infty$$

where;

$\psi(n)$  is the wrapped phase values.

$\varphi(n)$  is the unwrapped phase values.



$k(n)$  is the function containing the integers that must be added to the unwrapped phase  $\varphi$  to be wrapped between  $(-\pi, \pi)$ .

$n$  is an integer.

$v(n)$  is the function containing a set of integers that must be added to the wrapped phase  $\psi$  to be unwrapped.

Noting that:

$$v(n) = -k(n) \quad (3.5)$$

Define the wrapping operation  $w$  to be the operation which converts the unwrapped phase  $\varphi(n)$  to the wrapped phase, or mathematically:

$$w\{\varphi(n)\} = \arctan \left[ \frac{\sin(\varphi(n))}{\cos(\varphi(n))} \right] \quad (3.6)$$

or;

$$w\{\varphi(n)\} = \psi(n) = \varphi(n) + 2\pi \cdot k(n) \quad (3.7)$$

The difference in the wrapped phase is given by:

$$\Delta\{\psi(n)\} = \psi(n+1) - \psi(n) \quad (3.8)$$

Substitute the value of the wrapped phase in Equation 3.3 into Equation 3.8 yield:

$$\Delta\{\psi(n)\} = \Delta\varphi(n) + 2\pi * \Delta k(n) \quad (3.9)$$

Applying the wrapping operator on the wrapped phase difference  $\Delta\{\psi(n)\}$  we will get:



$$\begin{aligned}
w\{\Delta(\psi(n))\} &= \arctan\left[\frac{\sin(\Delta(\psi(n)))}{\cos(\Delta(\psi(n)))}\right] \\
&= \arctan\left[\frac{\sin(\Delta\varphi(n) + 2\pi\Delta k(n))}{\cos(\Delta\varphi(n) + 2\pi\Delta k(n))}\right] \\
&= \arctan\left[\frac{\sin(\Delta\varphi(n))}{\cos(\Delta\varphi(n))}\right]
\end{aligned} \tag{3.10}$$

so:

$$w\{\Delta(\psi(n))\} = w\{\Delta\varphi(n)\} \tag{3.11}$$

When there is no phase aliasing:

$$\text{if } |\Delta\varphi(n)| < \pi, \text{ then } w[\Delta\varphi(n)] = \Delta\varphi(n) \tag{3.12}$$

so:

$$w\{\Delta(\psi(n))\} = \Delta\varphi(n) \tag{3.13}$$

Substitute equation 3.7 into equation 3.1 we will get:

$$\varphi(b) = \varphi(a) + \sum_{i=a}^{b-1} w\{\Delta(\psi(i))\} \tag{3.14}$$

Where;

$\varphi(b)$  is the unwrapped phase at the point  $b$ .

$\varphi(a)$  is the unwrapped phase at an arbitrary starting point  $a$ , noting that at the initial point the unwrapped phase is assumed to be equal to the wrapped phase.

Equation 3.14 states that in the local phase unwrapping algorithms the unwrapping is “an integration of wrapped phase differences along the unwrapping path”.

In reality, unwrapping the wrapped phase using Equation 3.14 will propagate errors through the unwrapped phase map. Errors from unreliable regions propagate to reliable regions and cause a complete failure in the unwrapped-phase map. So, completely



relying on equation 3.14 may cause the propagation of errors throughout the whole unwrapped phase.

Many local phase unwrapping algorithms have been proposed to prevent error propagation in the unwrapped phase. These algorithms attempt to isolate unreliable regions in the phase map and minimize their effects on the unwrapped result. Local phase unwrapping methods can be classified into two main types. The first type defines the quality of each pixel in the phase map in order to unwrap the highest quality pixels first and the lowest quality pixels last. These algorithms are called **quality-guided phase unwrapping** algorithms. The second type is known as **residue-balancing** methods. Residue-balancing methods attempt to prevent error propagation by identifying residues (the source of noise in the wrapped phase). These residues must be balanced and isolated by using barriers (**branch-cuts**). Once these branch cuts are identified the unwrapping path should not pass through them. This prevents error propagation. Both types of algorithm will be explained in detail in the following subsections.

### 3.2.1 Quality-Guided Phase Unwrapping Algorithms.

Quality-guided algorithms depend on a quality measure to guide the unwrapping path. The main idea of these algorithms is to unwrap the highest quality pixels first and the lowest quality pixels last to prevent error propagation (Herraez *et al.*, 1996; Xu and Cumming, 1996). The success or failure of these algorithms depends strongly upon the production of a good quality map. The quality map can be defined as an array of values that define the quality or goodness of each pixel of the given phase map. Several two-dimensional quality-guided algorithms have been proposed over the last few decades. Bone (Bone, 1991) was the first who adopted a quality measure to guide the unwrapping process. He calculated the second-order partial derivative of each phase data point and used it to determine its quality. The matrix of quality values is called a quality map. Only those pixels whose qualities exceed a certain threshold are unwrapped, others are not processed until they exceed the threshold value, which is reduced with each subsequent iteration.



Subsequently, many quality-guided, path following algorithms were proposed (Quiroga and Bernabeu, 1994; Herraiez *et al.*, 1996; Xu and Cumming, 1996; Herraiez *et al.*, 2002b; Herraiez *et al.*, 2005). They design different unwrapping paths that rely completely upon the quality map to guide the integration path. They essentially unwrap the highest quality pixels first and leave the lowest quality pixels until last

Herraiez *et al.* were the first to introduce the concept of using the quality of edges to guide the unwrapping path. An edge can be defined as the connection between adjacent pixels, as shown in Figure 3.1. Herraiez *et al.* proposed a discrete unwrapping path based on using the qualities of the edges (Herraiez *et al.*, 2002a). This algorithm is very robust and fast and has been used in constructing a robust fringe pattern analysis system for human body shape measurement (Lilley *et al.*, 2000).

A drawback of these algorithms is that there is no guarantee that the path will not encircle an unbalanced residue and introduce a spurious discontinuity in the unwrapped solution.

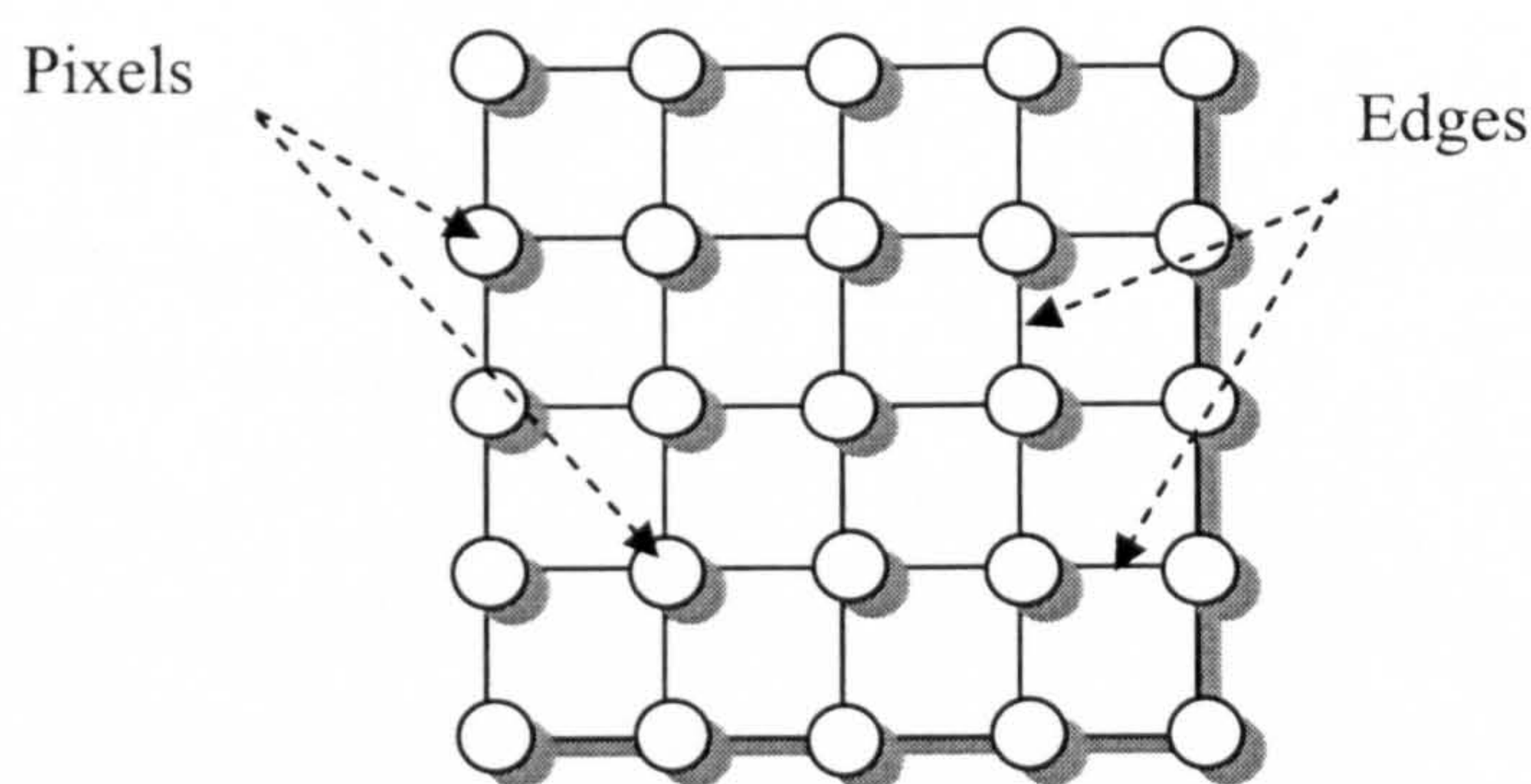


Figure 3.1: Illustration of pixels and edges.

### 3.2.2 Two-Dimensional Quality Maps

A very important feature of any quality-guided phase unwrapping algorithm is the utilisation of a reliable quality map. A good quality map determines the success or failure of the phase unwrapping algorithms. Quality maps provide additional information about the wrapped phase map. This additional information provides a measure of the quality of a particular pixel within the wrapped phase map.



Quality maps can be calculated using different methods (Ghiglia and Pritt, 1998). The most commonly used methods for calculating quality-maps in the phase unwrapping field are the second difference, phase derivative variance, maximum phase gradients and pseudo-correlation techniques.

### 3.2.2.1 Two-Dimensional Pseudo-Correlation Quality Map

The pseudo-correlation method attempts to mimic a correlation quality map which only exists for certain applications, such as synthetic aperture radar. It estimates the quality map from the phase data itself, without the need for an extra set of data.

The pseudo-correlation quality map defines the goodness of each pixel using the Equation:

$$q(m,n) = \frac{\sqrt{(\sum \cos(\psi(i,j)))^2 + (\sum \sin(\psi(i,j)))^2}}{k^2} \quad (3.15)$$

where

$\psi(i,j)$  is the wrapped phase value of the pixel  $(m,n)$ ,

$i,j$  represent the neighbours' indices for the pixel  $(m,n)$  in a  $k \times k$  window.

$k$  is the window size which should be an odd number.

### 3.2.2.2 Two-Dimensional Phase Derivative Variance Quality Map

The phase derivative variance method measures the statistical local variance of the wrapped phase derivatives. It is probably the most reliable quality map that can be extracted from the wrapped phase itself (Ghiglia and Pritt, 1998). Actually, phase derivative variance indicates the badness rather than the goodness of the phase data.

The calculation of the phase derivative variance for a pixel  $(m,n)$  using a  $k \times k$  window is defined by the equation:

$$PV(m,n) = \frac{1}{k^2} * \left( \sqrt{\sum (\hat{\Delta}^x \psi(i,j) - \overline{\hat{\Delta}^x \psi(i,j)})^2} + \sqrt{\sum (\hat{\Delta}^y \psi(i,j) - \overline{\hat{\Delta}^y \psi(i,j)})^2} \right) \quad (3.16)$$



where;

$\hat{\Delta}^x\psi(i, j)$  and  $\hat{\Delta}^y\psi(i, j)$  are the wrapped phase gradients in the  $x$  and  $y$  directions respectively.

$\overline{\hat{\Delta}^x\psi(i, j)}$  and  $\overline{\hat{\Delta}^y\psi(i, j)}$  are the mean of the values in a  $k \times k$  cube in  $\Delta^x$  and  $\Delta^y$  respectively.

$i, j$  are the neighbours' indices for the pixel  $(m, n)$  in a  $k \times k$  window.

$\hat{\Delta}^x\psi(i, j)$  and  $\hat{\Delta}^y\psi(i, j)$  are defined by :

$$\begin{aligned}\hat{\Delta}^x\psi(i, j) &= w\{\psi(i+1, j) - \psi(i, j)\} \\ \hat{\Delta}^y\psi(i, j) &= w\{\psi(i, j+1) - \psi(i, j)\}\end{aligned}\tag{3.17}$$

Where  $w$  defines a wrapping operator given previously in Equation 3.6.

Finally, the quality of the pixel is defined to be the reciprocal of the variance as described by the equation;

$$q(m, n) = \frac{1}{PV(m, n)}\tag{3.18}$$

### 3.2.2.3 Two-Dimensional Maximum Phase Gradient Quality Map

The maximum phase gradient method measures the magnitude of the largest phase gradient *i.e.*, partial derivative or wrapped phase difference in a  $k \times k$  window. Similarly to the case for the phase derivative variance quality map, the maximum gradient indicates the badness rather than the goodness of the phase data.

$$MG(m, n) = \max \left\{ \begin{array}{l} \max \left\{ \left| \hat{\Delta}^x\psi(i, j) \right| \right\} \\ \max \left\{ \left| \hat{\Delta}^y\psi(i, j) \right| \right\} \end{array} \right\}\tag{3.19}$$

### 3.2.2.4 Two-Dimensional Second Difference Quality Map

Another quality map extraction method is the second difference quality map (Herraez *et al.*, 2002a). In our experience, the second difference quality map is more robust than the



others and it has been found by the author to be the best quality map amongst the others described here. This quality map also measures the badness of each pixel in a  $k \times k$  window, using Equation. (3.20):

$$SD(m, n) = \sqrt{H^2(m, n) + V^2(m, n) + \sum_{n=1}^2 D_n^2(m, n)} \quad (3.20)$$

Where;

$$\begin{aligned} H(m, n) &= w\{\psi(m-1, n) - \psi(m, n)\} - w\{\psi(m, n) - \psi(m+1, n)\} \\ V(m, n) &= w\{\psi(m, n-1) - \psi(m, n)\} - w\{\psi(m, n) - \psi(m, n+1)\} \\ D_1(m, n) &= w\{\psi(m-1, n-1) - \psi(m, n)\} - w\{\psi(m, n) - \psi(m+1, n+1)\} \\ D_2(m, n) &= w\{\psi(m+1, n-1) - \psi(m, n)\} - w\{\psi(m, n) - \psi(m-1, n+1)\} \end{aligned} \quad (3.21)$$

$H(m, n)$  and  $V(m, n)$  are the horizontal and vertical second differences respectively.  $D_n(m, n)$  is the  $n^{\text{th}}$  diagonal second difference. The second difference measures the badness of each pixel, so the quality of each pixel is the reciprocal of the second difference value.

### 3.2.3 Residue-Balancing Phase Unwrapping Algorithms.

The second approach in local phase unwrapping methods is known collectively as residue-balancing techniques. These algorithms aim to produce a path-independent wrapped phase map. Path-dependency occurs because of the existence of inconsistent points called residues.

Residue-balancing algorithms search for residues in a wrapped-phase map and attempt to balance positive and negative residues by placing cut lines between them. The reason for these cut lines is to create an unwrapping barrier and prevent the unwrapping path penetrating them.

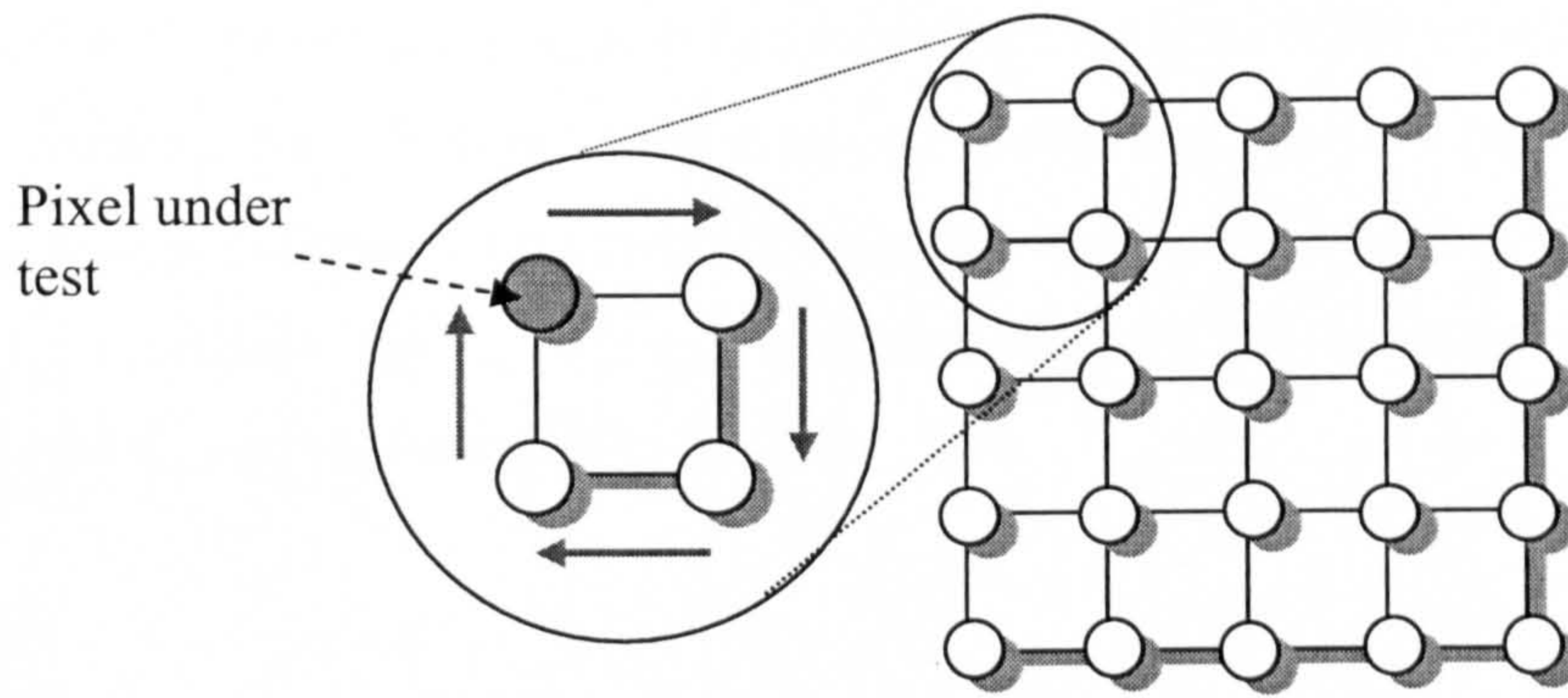
The residue is identified for each pixel in the phase map by calculating the wrapped gradients in a  $2 \times 2$  closed loop, as shown in Figure 3.2, by the following equation:



$$r = \left. \begin{aligned} &\Re\left[\frac{\psi_{i,j} - \psi_{i+1,j}}{2\pi}\right] + \Re\left[\frac{\psi_{i+1,j} - \psi_{i+1,j+1}}{2\pi}\right] + \Re\left[\frac{\psi_{i+1,j+1} - \psi_{i,j+1}}{2\pi}\right] \\ &+ \Re\left[\frac{\psi_{i,j+1} - \psi_{i,j}}{2\pi}\right] \end{aligned} \right\} \quad (3.22)$$

where, the operator  $\Re [ ]$  rounds its argument to the nearest integer.

The result of Equation 3.22 above can only take three possible values; +1, -1 or zero. On that basis, a pixel under test is considered to be a positive residue if the value of  $r$  is +1, and it is considered to be a negative residue if the value is -1. On the other hand, the pixel is not a residue if the value of  $r$  is zero.



**Figure 3.2: Identifying residues in a  $2 \times 2$  closed path.**

After identifying all residues in the wrapped phase map, these residues have to be balanced by the means of branch cuts. Figure 3.3 shows a branch cut that balances two pairs of opposite-sign residues.

Branch-cuts act as barriers to prevent the unwrapping path going thorough them. If these branch cuts are avoided during the unwrapping process, no errors propagate and the unwrapping path is considered to be path independent. On the other hand, if these branch cuts are penetrated during the unwrapping, then errors propagate throughout the whole phase map, and in this case the unwrapping path is considered to be path dependent.



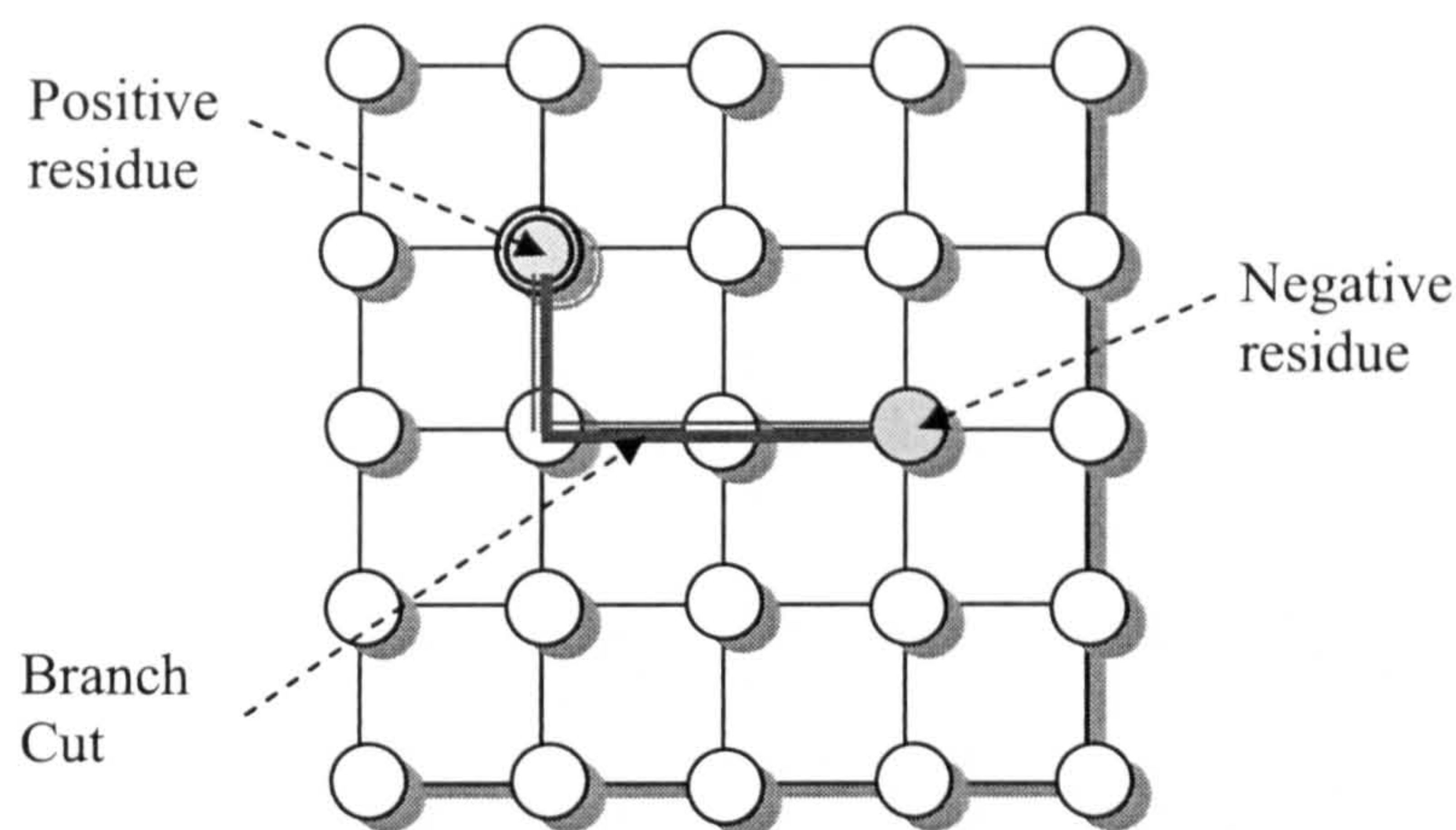
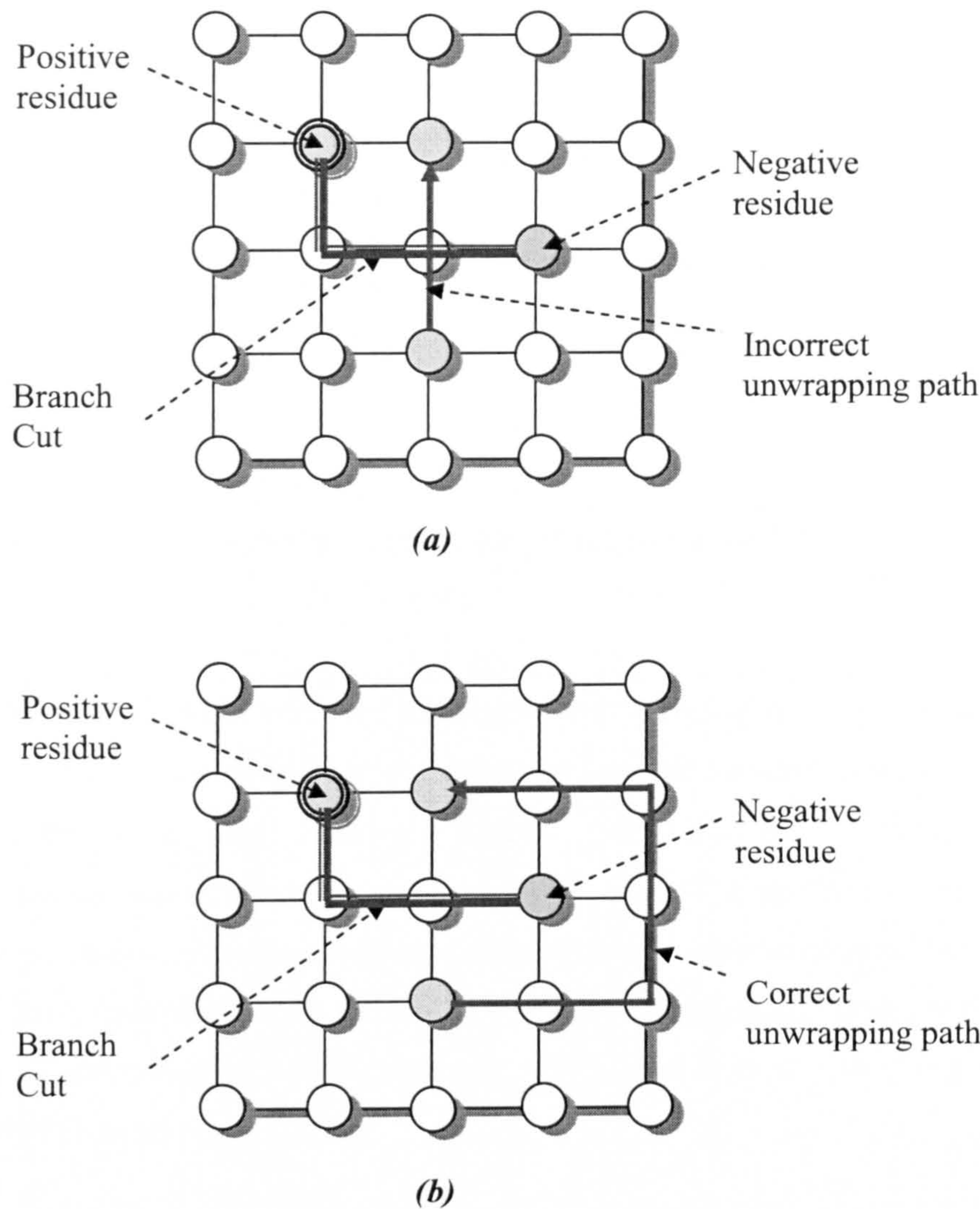


Figure 3.3: Illustration of branch cuts in the wrapped phase map.

Figure 3.4 illustrates the principle of unwrapping around the branch cuts. In Figure 3.4(a) the unwrapping path has penetrated the branch cut which has been placed so as to balance two residues in the wrapped phase map. These errors propagate and create  $2\pi$  discontinuities in the unwrapped phase map. On the other hand, Figure 3.4(b) shows the correct unwrapping path that avoids the branch cut, consequently, error propagation in the unwrapped solution is avoided.

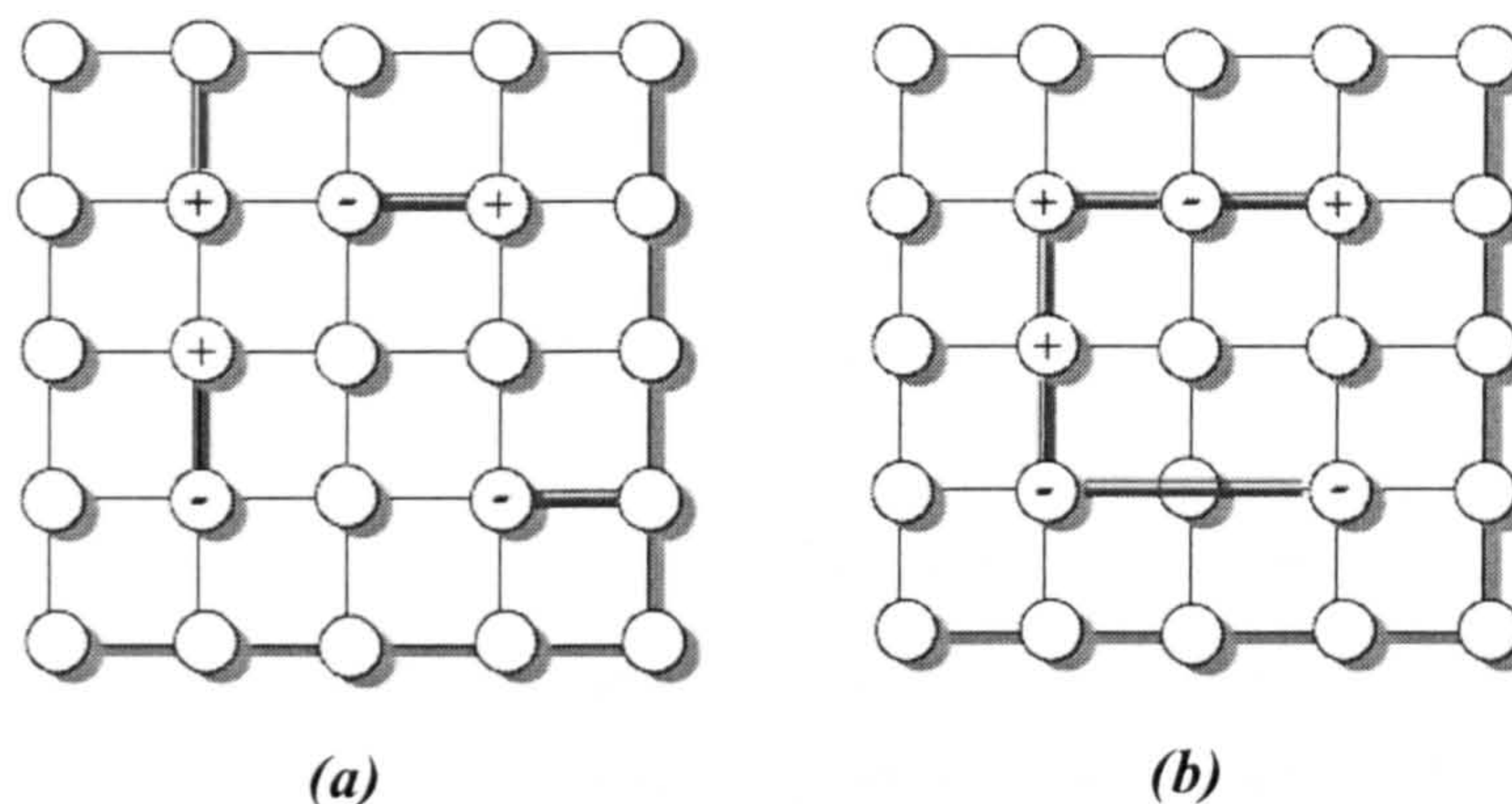




**Figure 3.4: Unwrapping path with the existence of the branch cuts, (a) incorrect unwrapping path and (b) correct path.**

Several algorithms have been proposed to place branch cuts between residues. These methods can be divided into two types: dipole branch cutting and tree branch cutting. In the dipole branch cutting techniques, the branch cut is only placed either between two residues of opposite sign, or between a residue and a border, as shown in Figure 3.5(a). Whereas, in the tree branch cutting methods, the branch cuts form trees that connect a group of residues. The net charge of any group in a tree has to be zero in order to be balanced. Any unbalanced group is connected to the border. An example of the tree branch cutting technique is shown in Figure 3.5(b).





**Figure 3.5: branch cutting techniques, (a) dipole branch cutting and (b) tree branch cutting**

Huntley was the first person to introduce the dipole branch cutting technique (Huntley, 1989). He used the nearest neighbour algorithm to find the optimal branch cutting distribution in the phase map. Huntley's method was improved by using more sophisticated search strategies such as: improved nearest neighbour, simulated annealing, minimum-cost matching, stable marriages, reverse simulated annealing and genetic algorithms. These algorithms attempt to find the corresponding dipoles with the minimum total connection length (Buckland *et al.*, 1995; Cusack *et al.*, 1995; Gutmann and Weber, 1999; Karout *et al.*, 2007a).

Recently, Karout *et al.* proposed a new technique of dipole branch cutting based on what they called a residue vector. They show that the optimal branch cut positioning has to follow the residue vector in order to obtain an accurate result (Karout *et al.*, 2007b).

The tree branch-cut technique was first introduced by Goldstein *et al.* (Goldstein *et al.*, 1988). In this technique, the pixels are utilized to create the branch cuts, such that any pixel can be marked as a part of branch cut that connects two different residues, as shown in Figure 3.5(b). The major drawback of this algorithm is that the branch cuts may be placed on good pixels, which may cause error propagation. To overcome this problem Flynn introduced the mask-cut algorithm, in which the branch-cut placement relies upon a quality map. In his method, the branch cuts are placed on bad quality pixels to connect residues (Flynn, 1996). Ghiglia and Pritt suggested removing the



dipoles as a pre-processing step in order to enhance the Goldstein algorithm (Ghiglia and Pritt, 1998). The dipoles are removed by connecting them using the nearest neighbour procedure proposed by Huntley (Huntley, 1989).

### 3.2.4 Other Local phase unwrapping algorithms.

A different robust local phase unwrapping algorithm is the minimum discontinuity algorithm. Flynn looked at phase unwrapping from a different point of view; his algorithm finds a solution that actually minimizes the discontinuities (Flynn, 1997). The algorithm utilizes a tree-growing approach that traces paths of discontinuities in the phase, detects the paths that form loops, and adds multiples of  $2\pi$  to the phase values enclosed by the loops in order to minimize the discontinuities. It performs this process iteratively until no more loops are detected. The process is guaranteed to converge to a minimum discontinuity solution.

A unique feature of Flynn's algorithm is that it works with or without a quality map. If a quality map is supplied, then the discontinuities are weighted by the quality values, so that the discontinuities in the low-quality regions do not contribute appreciably to the overall measure of discontinuity. As a result the discontinuities in the resulting unwrapped phase map tend to be confined to the lowest-quality regions.

## 3.3 Global phase unwrapping algorithms

In the previous section, it was stated that local phase unwrapping algorithms follow a certain unwrapping path in order to unwrap the phase. They begin at a grid point and integrate the wrapped phase differences over that path, which ultimately covers the entire phase map. Local phase unwrapping algorithms explicitly (residue-balancing algorithms) or implicitly (quality-guided algorithms) generate branch cuts and define the unwrapping path around these cuts in order to minimize error propagation.

On the other hand, Global phase unwrapping algorithms take a completely different approach to phase unwrapping. These algorithms formulate the phase unwrapping problem in a generalised minimum-norm sense. For that reason, many researchers refer



to the global phase unwrapping algorithms as minimum-norm algorithms (Ghiglia and Pritt, 1998).

Global phase unwrapping algorithms attempt to find the unwrapped phase by minimising a global error function as shown in Equation 3.23:

$$\varepsilon^p = \|solution - problem\|^p \quad (3.23)$$

As shown earlier in this chapter, for a case where there is no undersampling, the gradient of the unwrapped phase is equal to the wrapped gradient of the wrapped phase.

$$\Delta\varphi(x, y) = w\{\Delta\psi((x, y))\} \quad (3.24)$$

Global phase unwrapping algorithms seek the unwrapped phase whose local gradients in the  $x$  and  $y$  direction match, as closely as possible, the wrapped gradients of the wrapped phase, as described in Equation 3.25:

$$\varepsilon^p = \sum_{i=0}^{M-2} \sum_{j=0}^{N-1} |\Delta^x \varphi(i, j) - \hat{\Delta}^x \psi(i, j)|^p + \sum_{i=0}^{M-1} \sum_{j=0}^{N-2} |\Delta^y \varphi(i, j) - \hat{\Delta}^y \psi(i, j)|^p \quad (3.25)$$

where;

$\Delta^x \varphi(i, j)$  and  $\Delta^y \varphi(i, j)$  are the unwrapped phase gradients in the  $x$  and  $y$  directions respectively, and they are given by:

$$\Delta^x \varphi(i, j) = \varphi(i+1, j) - \varphi(i, j) \quad (3.26)$$

$$\Delta^y \varphi(i, j) = \varphi(i, j+1) - \varphi(i, j) \quad (3.27)$$

$\hat{\Delta}^x \psi(i, j)$  and  $\hat{\Delta}^y \psi(i, j)$  are the wrapped values of the wrapped phase gradients in the  $x$  and  $y$  directions respectively, and they are given by:



$$\hat{\Delta}^x \psi(i, j) = w\{\psi(i+1, j) - \psi(i, j)\} \quad (3.28)$$

$$\hat{\Delta}^y \psi(i, j) = w\{\psi(i, j+1) - \psi(i, j)\} \quad (3.29)$$

The wrapping operator  $w\{\}$  is defined in Equation 3.6 above.

Global phase unwrapping algorithms can be divided into three different categories: unweighted least-squares, weighted least-squares and  $L^p$ -norm methods. All of these types are discussed in the following subsections.

### 3.3.1 Unweighted Least-squares

The unweighted least squares methods, which was first introduced by Hunt, attempts to minimise the difference between the unwrapped phase gradients and the wrapped values of the wrapped phase gradient, with the minimisation performed in a least-squares manner (Hunt, 1979), *i.e.*,  $p=2$  in Equation 3.25. In other words, the unweighted least squares method attempts to find the unwrapped phase solution that will minimise the following equation:

$$\varepsilon^2 = \sum_{i=0}^{M-2} \sum_{j=0}^{N-1} |\Delta^x \varphi(i, j) - \hat{\Delta}^x \psi(i, j)|^2 + \sum_{i=0}^{M-1} \sum_{j=0}^{N-2} |\Delta^y \varphi(i, j) - \hat{\Delta}^y \psi(i, j)|^2 \quad (3.30)$$

Ghiglia *et al.* have simplified the above equation, and they showed that the above equation can be rewritten as: (Ghiglia and Pritt, 1998)

$$(\varphi(i+1, j) - 2\varphi(i, j) + \varphi(i-1, j)) + (\varphi(i, j+1) - 2\varphi(i, j) + \varphi(i, j-1)) = \rho(i, j) \quad (3.31)$$

where;

$$\rho(i, j) = (\hat{\Delta}^x \psi(i, j) - \hat{\Delta}^x \psi(i-1, j)) + (\hat{\Delta}^y \psi(i, j) - \hat{\Delta}^y \psi(i, j-1)) \quad (3.32)$$



Equation 3.32 represents a discrete version of Poisson's equation and can be written in matrix notation by:

$$\mathbf{P} \vec{\varphi} = \vec{\rho} \quad (3.33)$$

where  $\mathbf{P}$  represents the discrete Laplacian operation defined by the left-hand side of Equation 3.31,  $\vec{\varphi}$  and  $\vec{\rho}$  are one-dimensional vectors containing the values of the two-dimensional arrays  $\varphi(x, y)$  and  $\rho(x, y)$  respectively.

Unweighted least squared algorithms attempt to find the unwrapped phase solution by solving Poisson's equation. *Ghiglia et al.* have proposed the use of Gauss-Seidel relaxation (Press *et al.*, 1992) to find a solution for the unwrapped phase (Ghiglia and Pritt, 1998). In this method, initial values of zeros are given to the unwrapped phase solution. Then, the unwrapped phase solution is updated iteratively using Equation 3.34 until the difference or the error reaches a certain tolerance.

$$\varphi(i, j) = \frac{(\varphi(i+1, j) + \varphi(i-1, j) + \varphi(i, j+1) + \varphi(i, j-1)) - \rho(i, j)}{4} \quad (3.34)$$

Ghiglia *et al.* have shown that using Gauss-Seidel relaxations is not practical due to its extremely slow convergence (Ghiglia and Pritt, 1998). On the other hand, they have proposed the use of multigrid methods (Briggs, 1987; Press *et al.*, 1992) to speed up the convergence and achieve less error in a shorter time (Ghiglia and Pritt, 1998).

Another method of solving the unweighted least square phase unwrapping problem was also introduced by Ghiglia and Pritt (Ghiglia and Pritt, 1998), which was based on the use of the Fourier transform.



### 3.3.2 Weighted Least-squares

The unweighted least squares methods attempt to calculate the unwrapped phase solution that minimises Equation 3.25. These methods assign equal weights to all pixels in the wrapped phase map. If the wrapped phase contains residues and some corrupted areas, the whole solution will be affected and errors propagate through the entire solution space.

Weighted least squares algorithms are designed to overcome these problems. Each pixel in the wrapped phase map is assigned a certain weight. Noisy pixels and residues are assigned low weights to reduce their effect on the unwrapped solution. Weighted least squares methods attempt to find the unwrapped phase solution that minimises the following formula:

$$\varepsilon^2 = \sum_{i=0}^{M-2} \sum_{j=0}^{N-1} U(i, j) \times |\Delta^x \varphi(i, j) - \hat{\Delta}^x \psi(i, j)|^2 + \sum_{i=0}^{M-1} \sum_{j=0}^{N-2} V(i, j) \times |\Delta^y \varphi(i, j) - \hat{\Delta}^y \psi(i, j)|^2 \quad (3.35)$$

where;  $U(i, j)$  and  $V(i, j)$  are the gradient weights assigned for x-gradient and y-gradient respectively. These gradient weights are defined by:

$$U(i, j) = \min(q_{i+1,j}^2, q_{i,j}^2) \quad (3.36)$$

$$V(i, j) = \min(q_{i,j+1}^2, q_{i,j}^2) \quad (3.37)$$

where,  $q_{i,j}$  is the quality of the pixel  $(i, j)$  calculated using one of the quality maps discussed earlier in this chapter.

Equation 3.35 can be rewritten as: (Ghiglia and Pritt, 1998)



$$\begin{aligned}
&U(i, j)\Delta^x\varphi(i, j) - U(i-1, j)\Delta^x\varphi(i-1, j) + \\
&V(i, j)\Delta^y\varphi(i, j) - V(i, j-1)\Delta^y\varphi(i, j-1) = \sigma(i, j)
\end{aligned} \tag{3.38}$$

where;

$$\begin{aligned}
\sigma(i, j) = &\left( U(i, j)\hat{\Delta}^x\psi(i, j) - U(i-1, j)\hat{\Delta}^x\psi(i-1, j) \right) \\
&+ \left( V(i, j)\hat{\Delta}^y\psi(i, j) - V(i, j-1)\hat{\Delta}^y\psi(i, j-1) \right)
\end{aligned} \tag{3.39}$$

Using matrix notation, Equation 3.38 can be rewritten as:

$$\mathbf{Z}\vec{\varphi} = \vec{\sigma} \tag{3.40}$$

where  $\mathbf{Z}$  represents the discrete weighted Laplacian operation defined in the left-hand side of Equation 3.38 and  $\vec{\sigma}$  are one-dimensional vectors containing the values calculated from Equation 3.39.

Unlike the unweighted least squares methods, the weighted least squares method cannot be solved directly by means of a Fourier transform, therefore they must be solved iteratively.

The Gauss-Seidel relaxation used to solve the unweighted case can also be applied to the weighted least squares case, as proposed by Ghiglia and Pritt (Ghiglia and Pritt, 1998). The unwrapped phase solution can be calculated iteratively using the equation:

$$\varphi(i, j) = \frac{\left( U(i, j)\varphi(i+1, j) + U(i-1, j)\varphi(i-1, j) \right) + \left( V(i, j)\varphi(i, j+1) + V(i, j-1)\varphi(i, j-1) \right) - \sigma(i, j)}{v(i, j)} \tag{3.41}$$

where  $v(i, j)$  is defined by:

$$v(i, j) = U(i, j) + U(i-1, j) + V(i, j) + V(i, j-1) \tag{3.42}$$



Because of the slow convergence of this algorithm, multigrid methods are used to speed up the convergence. (Ghiglia and Pritt, 1998)

Moreover, Ghiglia *et al.* have proposed two more methods to find the unwrapped phase solution. These methods are the Picard iteration method and the preconditioned conjugate gradient (PCG) method. The Picard method is simple and easy to implement, but the convergence is not guaranteed. In contrast, the PCG method has excellent convergence properties. (Ghiglia and Pritt, 1998)

### 3.3.3 L<sup>p</sup>-Norm Method

A more advanced method, developed by Ghiglia and Romero, is the minimum L<sup>p</sup>-norm method. This method finds the unwrapped phase solution that minimizes Equation 3.25 for an arbitrary value of  $p$ , unlike the least squares method where  $p=2$  (Ghiglia and Romero, 1996).

In the general case, Equation 3.25 can be simplified as: (Ghiglia and Pritt, 1998)

$$\begin{aligned} \dot{U}(i, j)\Delta^x\varphi(i, j) - \dot{U}(i-1, j)\Delta^x\varphi(i-1, j) + \\ \dot{V}(i, j)\Delta^y\varphi(i, j) - \dot{V}(i, j-1)\Delta^y\varphi(i, j-1) = \dot{\sigma}(i, j) \end{aligned} \quad (3.43)$$

where;  $\dot{U}(i, j)$  and  $\dot{V}(i, j)$  are the data dependent weights assigned for  $x$ -gradient and  $y$ -gradient respectively and they are defined by:

$$\dot{U}(i, j) = \left| \Delta^x\varphi(i, j) - \hat{\Delta}^x\psi(i, j) \right|^{p-2} \quad (3.44)$$

$$\dot{V}(i, j) = \left| \Delta^y\varphi(i, j) - \hat{\Delta}^y\psi(i, j) \right|^{p-2} \quad (3.45)$$

and  $\dot{\sigma}(i, j)$  is defined by:



$$\begin{aligned} \dot{\sigma}(i, j) = & \left( \dot{U}(i, j) \hat{\Delta}^x \psi(i, j) - \dot{U}(i-1, j) \hat{\Delta}^x \psi(i-1, j) \right) \\ & + \left( \dot{V}(i, j) \hat{\Delta}^y \psi(i, j) - \dot{V}(i, j-1) \hat{\Delta}^y \psi(i, j-1) \right) \end{aligned} \quad (3.46)$$

Clearly from the equation, the  $L^p$ -norm case is very similar to that of the case of weighted least squares. The only difference between the two cases is that of the definition of the weighted factors. In the weighted least squares method, the weighted factors are defined independently from a quality map, whereas in the  $L^p$ -norm case these factors are extracted from the wrapped phase data itself. So, solving the  $L^p$ -norm method can be carried out using the same algorithms used for the weighted least squares case (Ghiglia and Pritt, 1998)

### 3.3.4 Other global integration methods

Fornaro *et al.* have proposed a different global phase unwrapping algorithm. In their method, the phase unwrapping problem is introduced using Green's formulation (Fornaro *et al.*, 1995; Fornaro *et al.*, 1996). Many researchers have proposed different phase unwrapping algorithms that estimate the parameters of a polynomial function that best matches the unwrapped solution by using regression (Slocumb and Kitchen, 1994; Schwarz, 2002; Karout *et al.*, 2006).



### 3.4 Summary

This chapter presents a review of a number of two-dimensional phase unwrapping techniques that have been proposed in the past. Two main categories of phase unwrapping algorithms were discussed; local and global methods. Local phase unwrapping techniques attempt to place branch cuts, explicitly or implicitly, as barriers to prevent error propagation. On the other hand, global algorithms do not search for residues, nor do they place branch cuts to prevent error propagation. Instead, they attempt to find an unwrapped solution by estimating the unwrapped gradients, depending upon a minimisation approach. Both algorithms have their merits and demerits. In general terms, there is no unwrapper that is significantly better than any of the others and also there is no generic unwrapper that can overcome all phase unwrapping problems.

This chapter has also outlined a number of two-dimensional quality maps that aid phase unwrapping algorithms. Quality maps can be used by local or global phase unwrapping algorithms. The use of quality maps in designing a phase unwrapping algorithm will improve the performance of that algorithm.



**References:**

- Bone, D. J. (1991) Fourier Fringe Analysis: The Two-Dimensional Phase Unwrapping Problem. *Applied Optics*, 30, 3627-32.
- Briggs, W. L. (1987) *A Multigrid Tutorial*, SIAM Philadelphia.
- Buckland, J. R.; Huntley, J. M. and Turner, S. R. E. (1995) Unwrapping Noisy Phase Maps by Use of a Minimum-Cost-Matching Algorithm. *Applied Optics*, 34, 5100-8.
- Cusack, R.; Huntley, J. M. and Goldrein, H. T. (1995) Improved Noise-Immune Phase-Unwrapping Algorithm. *Applied Optics*, 34, 781-9.
- Flynn, T. J. (1996) Consistent 2-D Phase Unwrapping Guided by a Quality Map. Vol: 4, pp: 2057-2059. Lincoln, NE, USA, IEEE.
- Flynn, T. J. (1997) Two-Dimensional Phase Unwrapping with Minimum Weighted Discontinuity. *Journal of the Optical Society of America A (Optics, Image Science and Vision)*, 14, 2692-701.
- Fornaro, G.; Franceschetti, G. and Lanari, R. (1995) Two-Dimensional Phase Unwrapping Using Green's Formulation. Paris, Fr, SPIE, Bellingham, WA, USA.
- Fornaro, G.; Franceschetti, G. and Lanari, R. (1996) Interferometric Sar Phase Unwrapping Using Green's Formulation. *IEEE Transactions on Geoscience and Remote Sensing*, 34, 720-7.
- Fornaro, G.; Franceschetti, G.; Lanari, R.; Sansosti, E. and Tesauro, M. (1997a) Global and Local Phase-Unwrapping Techniques: A Comparison. *Journal of the Optical Society of America A (Optics, Image Science and Vision)*, 14, 2702-8.
- Fornaro, G.; Franceschetti, G.; Lanari, R.; Sansosti, E. and Tesauro, M. (1997b) How Global and Local Phase Unwrapping Techniques Are Connected. Singapore, IEEE.
- Gens, R. (2003) Two-Dimensional Phase Unwrapping for Radar Interferometry: Developments and New Challenges. *International Journal of Remote Sensing*, 24, 703-710.
- Ghiglia, D. C. and Romero, L. A. (1996) Minimum  $L^p$ -Norm Two-Dimensional Phase Unwrapping. *Journal of the Optical Society of America A (Optics, Image Science and Vision)*, 13, 1999-2013.



- Ghiglia, D. C. and Pritt, M. D. (1998) *Two-Dimensional Phase Unwrapping: Theory, Algorithms and Software*, A Willey-Interscience Publication.
- Goldstein, R. M.; Zebker, H. A. and Werner, C. L. (1988) Satellite Radar Interferometry: Two-Dimensional Phase Unwrapping. 4 ed. Springfield, VA, USA.
- Gutmann, B. and Weber, H. (1999) Phase Unwrapping with the Branch-Cut Method: Clustering of Discontinuity Sources and Reverse Simulated Annealing. *Applied Optics*, 38, 5577-93.
- Herraez, M. A.; Burton, D. R.; Lalor, M. J. and Clegg, D. B. (1996) Robust, Simple, and Fast Algorithm for Phase Unwrapping. *Applied Optics*, 35, 5847-52.
- Herraez, M. A.; Burton, D. R.; Lalor, M. J. and Gdeisat, M. A. (2002a) Fast Two-Dimensional Phase-Unwrapping Algorithm Based on Sorting by Reliability Following a Noncontinuous Path. *Applied Optics*, 41, 7437-44.
- Herraez, M. A.; Gdeisat, M. A.; Burton, D. R. and Lalor, M. J. (2002b) Robust, Fast, and Effective Two-Dimensional Automatic Phase Unwrapping Algorithm Based on Image Decomposition. *Applied Optics*, 41, 7445-55.
- Herraez, M. A.; Boticario, J. G.; Lalor, M. J. and Burton, D. R. (2005) Agglomerative Clustering-Based Approach for Two-Dimensional Phase Unwrapping. *Applied Optics*, 44, 1129-40.
- Hunt, B. R. (1979) Matrix Formulation of the Reconstruction of Hase Values from Hase Differences. *Journal of the Optical Society of America*, 5, 416-425.
- Huntley, J. M. (1989) Noise-Immune Phase Unwrapping Algorithm. *Applied Optics*, 28, 3268-70.
- Itoh, K. (1982) Analysis of the Phase Unwrapping Algorithm. *Applied Optics*, 21, 2470.
- Karout, S.; Gdeisat, M.; Burton, D. and Lalor, M. (2006) Hybrid Genetic Algorithm Using a Parametric Method to Solve the Two-Dimensional Phase Unwrapping Problem. *Photon 06*. Manchester UK.
- Karout, S. A.; Gdeisat, M. A.; Burton, D. and Lalor, M. J. (2007a) Two-Dimensional Phase Unwrapping Using a Hybrid Genetic Algorithm. *Applied Optics*, 46, 730-743.
- Karout, S. A.; Gdeisat, M. A.; Burton, D. R. and Lalor, M. J. (2007b) Residue Vector a New Approach to Branch Cut Placement in Phase Unwrapping - the Theoretical Study. *Applied Optics*, 46.



- Lilley, F.; Lalor, M. J. and Burton, D. R. (2000) Robust Fringe Analysis System for Human Body Shape Measurement. *Optical Engineering*, 39, 187-95.
- Press, W. H.; Flannery, B. P.; Teukolsky, S. A. and Vetterling, W. T. (1992) *Numerical Recipes: The Art of Scientific Computing*, Cambridge, University of Cambridge.
- Quiroga, J. A. and Bernabeu, E. (1994) Phase-Unwrapping Algorithm for Noisy Phase-Map Processing. *Applied Optics*, 33, 6725-31.
- Robinson, D. W. and Reid, G. T. (1993) *Interferogram Analysis: Digital Fringe Pattern Measurement Techniques*, Institute of Physics Publishing.
- Schwarz, O. (2002) Improved Hybrid Phase Unwrapping Algorithm in Speckle Interferometry. Seattle, WA, USA, SPIE-Int. Soc. Opt. Eng.
- Slocumb, B. J. and Kitchen, J. (1994) A Polynomial Phase Parameter Estimation-Phase Unwrapping Algorithm. Adelaide, SA, Australia, IEEE.
- Xu, W. and Cumming, I. (1996) Region Growing Algorithm for Insar Phase Unwrapping. Lincoln, NE, USA, IEEE, Piscataway, NJ, USA.



## *Chapter Four*

# *Three-Dimensional Fourier Fringe Analysis*



## Chapter Four

### Three-Dimensional Fourier Fringe analysis

---

#### 4.1 Introduction

Fringe pattern analysis methods, discussed earlier in chapter 2, can be used to measure the surface shape of both static and dynamic objects. To measure the surface of static objects, temporal or spatial fringe pattern analysis techniques are used. The temporal techniques require at least three source fringe patterns with a known phase shift between them. On the other hand spatial techniques require only a single fringe pattern.

The measurement of the surface of dynamic objects can be accomplished by primarily using spatial fringe pattern analysis techniques rather than the temporal fringe analysis techniques. Perhaps the most widely used spatial fringe pattern analysis technique is that of two-dimensional Fourier fringe analysis (2D-FFA).

The history of the Fourier Fringe analysis algorithm started in 1982 when Takeda *et al.* suggested the use of the Fourier transform to analyse fringe patterns (Takeda *et al.*, 1982). Their paper had a major impact upon researchers but little direct influence on industry. Takeda's technique used one-dimensional signal processing algorithms to analyse fringe patterns, which are basically two-dimensional images. This is the main reason that this method did not find its way into practical applications.

The obvious development of the Takeda's algorithm is to extend it to two-dimensions to make it more suitable to analyse fringe patterns. Many researchers have worked in improving the Takeda's 1D-FFA algorithm, extending it into two dimensions, and adapting it, successfully, to specific applications. To name a few; Bone (Bone *et al.*, 1986), Burton and Lalor (Burton and Lalor, 1989), Li *et al.* (Li *et al.*, 1990), and Gorecki (Gorecki, 1992).



The two-dimensional Fourier fringe analysis technique has been used for the measurement of both static objects and also for the measurement of dynamic objects whose surface shape changes with time (Su *et al.*, 2001). But this technique is not very suitable for the measurement of dynamic objects. In the case of static objects, one fringe pattern is analysed in order to measure the object's 3D surface. On the other hand, measuring the surface of a dynamic object requires a time-sequence of fringe patterns and each fringe pattern is analysed individually. This state-of-the-art technique has a limitation characterised by it inherently being a two-dimensional image processing technique. Consequently, the use of this technique to analyse a sequence of fringe patterns for a dynamic object does not utilise any potential relationship between consecutive fringe patterns.

Fourier fringe analysis algorithms can be divided into two stages: phase extraction and phase unwrapping. The first stage extracts the phase of a fringe pattern by using a Fourier transform and carrying out filtering in the frequency domain. The phase produced by the first stage is wrapped and limited to the  $-\pi$  and  $\pi$  range. The wrapped phase contains  $2\pi$  jumps, which should be removed using a phase unwrapping algorithm. This is the second stage in an FFA algorithm.

In this chapter, the state-of-art two-dimensional Fourier fringe analysis technique is extended into three dimensions. The proposed three-dimensional Fourier fringe analysis algorithm will be used to measure the surface of dynamic objects. This chapter will discuss the wrapped phase extraction in 3D, whereas the three-dimensional phase unwrapping will be discussed in the next chapter. The following section explains the theory of the three-dimensional Fourier fringe analysis. Section 4.3 explains three-dimensional filtering which is used to select the required frequency. Computer simulated and experimental results are discussed in sections 4.4 and 4.5.



## 4.2 Theoretical Background

In the case of three-dimensional Fourier fringe analysis the intensity of each voxel is given by:

$$g(x, y, t) = a(x, y, t) + b(x, y, t) \cos[2\pi(\alpha x + \beta y + \gamma z) + \varphi(x, y, t)] \quad (4.1)$$

where  $a(x, y, t)$  represents the background illumination,  $b(x, y, t)$  the amplitude modulation of fringes,  $\alpha$ ,  $\beta$  and  $\gamma$  the spatial carrier frequencies in the  $x$ ,  $y$  and  $z$  dimensions respectively,  $\varphi(x, y, t)$  the phase modulation of the fringes (the required phase distribution) and  $x$ ,  $y$  and  $z$  the sample indices for the  $x$ ,  $y$  and  $z$  axes respectively.

Similar to the one-dimensional case, Equation 4.1 can be rewritten as:

$$g(x, y, z) = a(x, y, z) + c(x, y, z) e^{i2\pi(\alpha x + \beta y + \gamma z)} + c^*(x, y, z) e^{-i2\pi(\alpha x + \beta y + \gamma z)} \quad (4.2)$$

where;

$$c(x, y, z) = \frac{b(x, y, z) \cdot e^{i\varphi(x, y, z)}}{2} \quad (4.3)$$

and

$$c^*(x, y, z) = \frac{b(x, y, z) \cdot e^{-i\varphi(x, y, z)}}{2} \quad (4.4)$$

Applying the three-dimensional Fourier transform we will have:

$$G(f_x, f_y, f_z) = A(f_x, f_y, f_z) + C(f_x - \alpha, f_y - \beta, f_z - \gamma) + C^*(f_x + \alpha, f_y + \beta, f_z + \gamma) \quad (4.5)$$



where,  $A(f_x, f_y, f_t)$  is the spectrum of the background illumination,  $C(f_x - \alpha, f_y - \beta, f_t - \gamma)$  and  $C^*(f_x + \alpha, f_y + \beta, f_t + \gamma)$  are the spectra of the deformed fringes. The frequency spectrum of  $G(f_x, f_y, f_t)$  is shown in Figure 4.1.

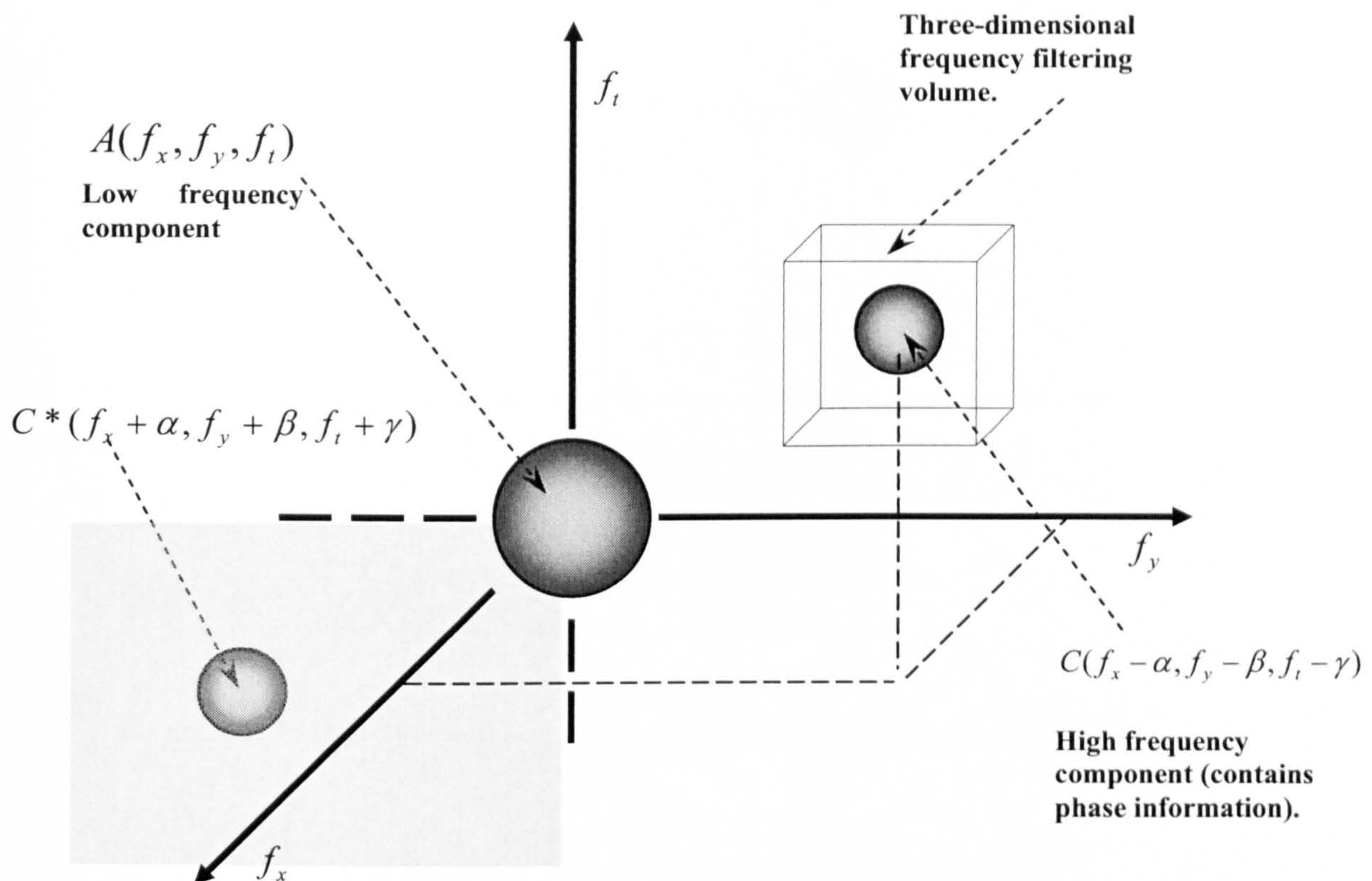


Figure 4.1: Frequency spectrum for a three-dimensional fringe patterns volume.

As shown in Figure 4.1 the phase information is encoded into the conjugate high frequency components  $C(f_x - \alpha, f_y - \beta, f_t - \gamma)$  and  $C^*(f_x + \alpha, f_y + \beta, f_t + \gamma)$ . So in order to retrieve the phase information one of these components has to be selected using a three-dimensional filtering volume as shown in Figure 4.1. Figure 4.2 shows the required component which is the output of the filtering operation. As shown in the figure the filtering operation should assure that it selects the required component and filters out all the other components. Improper filtering will definitely lead to the return of incorrect phase information. Then the required component has to be shifted back to the origin; this shift procedure will remove the carrier frequencies from the extracted phase. Figure 4.3 shows the shifted component and the removal of the carrier frequencies.



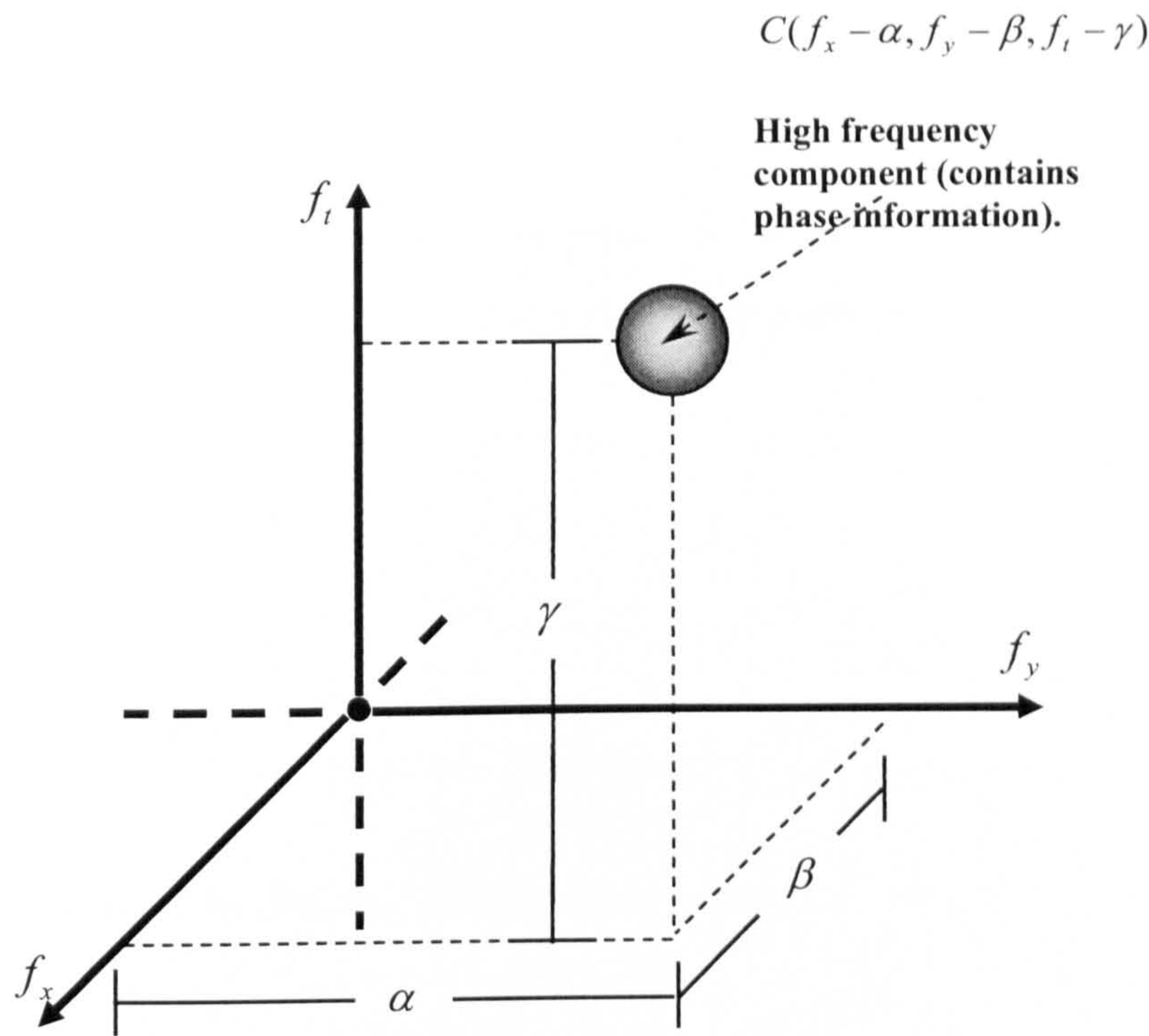


Figure 4.2: Required frequency component selected by a 3D filtering volume

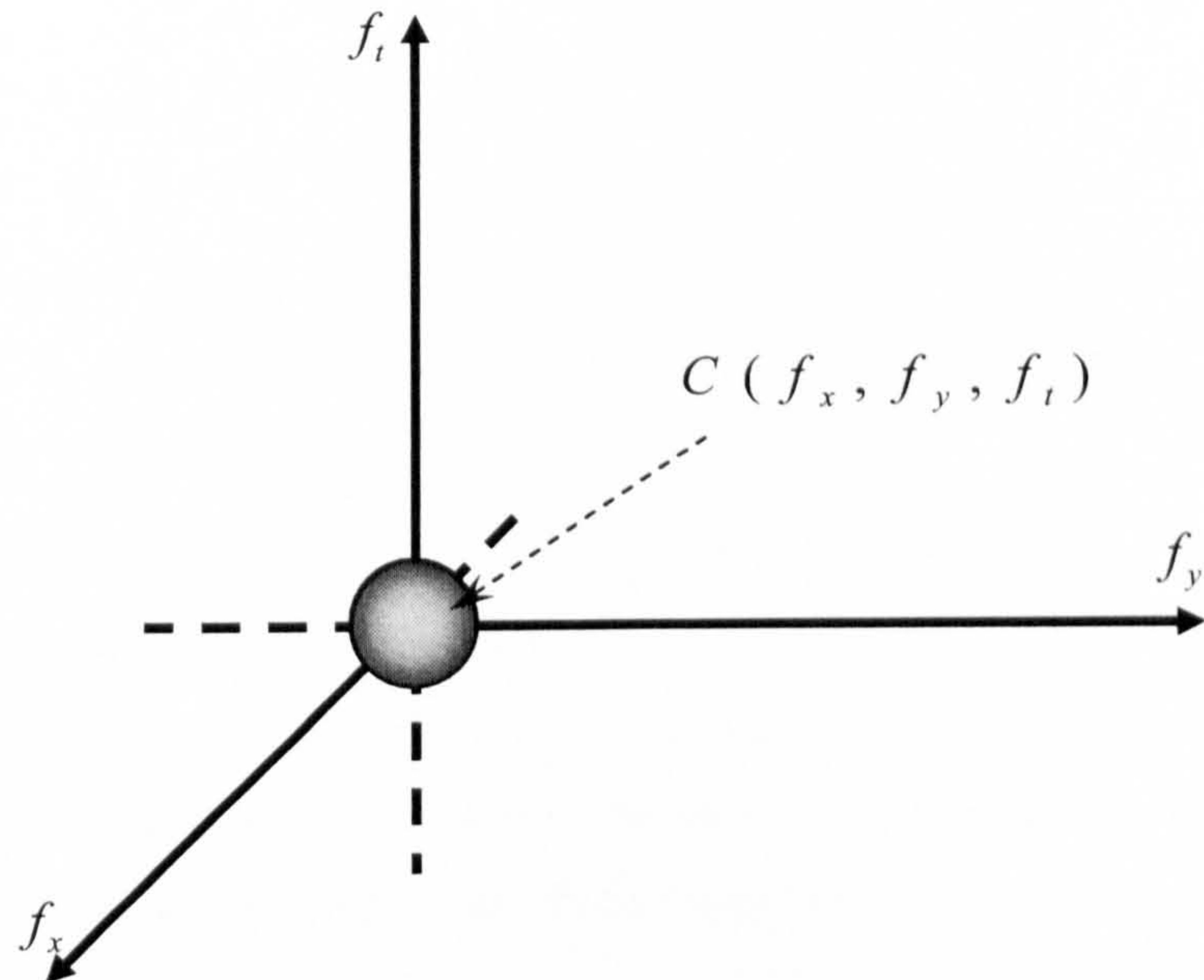


Figure 4.3: Removing the carrier frequencies by shifting the required component toward the origin



By transforming the term  $C(f_x, f_y, f_t)$  back into its original spatial domain using the three-dimensional inverse Fourier transform, we will return to the spatial domain  $c(x, y, t)$ .

As described earlier in chapter 2, extracting the phase information from the term  $c(x, y, t)$  can be carried out by extracting the real and imaginary parts, where;

$$\Re\{c(x, y, t)\} = \frac{1}{2}b(x, y, t)\cos(\varphi(x, y, t)) \quad (4.6)$$

and

$$\Im\{c(x, y, t)\} = \frac{1}{2}b(x, y, t)\sin(\varphi(x, y, t)) \quad (4.7)$$

The phase then can be extracted by dividing the imaginary part by the real part:

$$\psi(x, y, t) = \tan^{-1} \frac{\Im\{c(x, y, t)\}}{\Re\{c(x, y, t)\}} = \varphi(x, y, t) \mod \pi \quad (4.8)$$

Alternatively, phase can be extracted using the following equation:

$$\psi(x, y, t) = \Im\{\ln[c(x, y, t)]\} = \varphi(x, y, t) \mod \pi \quad (4.9)$$

In both cases, the extracted phase volume is wrapped between  $-\pi$  and  $\pi$ , and it needs to be unwrapped using a three-dimensional phase unwrapping algorithm, as will be demonstrated in the next chapter.

### 4.3 Filtering in the Frequency Domain

Filtering in the frequency domain is considered to be a critical operation in Fourier fringe analysis techniques. The quality of the final result relies upon the design of a high-quality filter.



In the literature, many researchers proposed different types and shapes of filters to perform the filtering in the two-dimensional FFA case (Stephenson, 1994; Arevalillo Herraiz *et al.*, 1999; Lilley *et al.*, 2000). Lilley *et al.* proposed a two-dimensional Butterworth filter in order select the required frequency component . They built a very robust two-dimensional Fourier fringe analysis system using the two-dimensional Butterworth filter for human body shape measurement (Lilley *et al.*, 2000). Indeed the Butterworth filter produces very accurate results when used in human body shape measurement.

In this system a three-dimensional Butterworth filter is adopted, in order to select the required component and reduce the problems of filter ringing and frequency leakage that can occur when cubic filters are employed.

A three-dimensional Butterworth filter is defined by:

$$H(f_x, f_y, f_z) = \frac{1}{\sqrt{\left(1 + \left(\frac{f_x}{f_{xc}}\right)^{2n} + \left(\frac{f_y}{f_{yc}}\right)^{2n} + \left(\frac{f_z}{f_{zc}}\right)^{2n}\right)}} \quad (4.10)$$

where;

$f_x$ ,  $f_y$  and  $f_z$  are three dimensional frequency axes that correspond to the  $x$ ,  $y$  and  $z$  axes in the time domain.

$f_{xc}$ ,  $f_{yc}$  and  $f_{zc}$  are the cut-off frequencies in the  $f_x$ ,  $f_y$  and  $f_z$  directions respectively.

$n$  is the order of the filter.

Choosing appropriate cut-off frequencies is very critical. In this system, initial cut-off frequencies are determined to be the boundaries of the object in  $f_x$ ,  $f_y$  and  $f_z$  directions. Then, these values are tuned based on try and error procedure to obtain the best result.



## 4.4 Computer Simulation Results

The proposed three-dimensional Fourier fringe analysis technique has been evaluated using many simulation of computer-generated dynamic fringe patterns. In this section two simulated objects are considered. The first object is a simple growing hemisphere whose radius is increasing with time, *i.e.*, with the frame number (see Appendix A.1). The sphere grows from a minimum radius to a maximum radius during 100 video frames. Each frame consists of  $256 \times 256$  pixels. The simulated fringe pattern volume is created for this growing sphere using the following equation.

$$g_i(x, y, t) = [Obj_i(x, y, t) + noise_1(x, y, t)] + [1 + noise_2(x, y, t)] \cos(2\pi(15x + 10y + 5z) + Obj_i(x, y, t)) \quad (4.11)$$

Where;

$g_i(x, y, t)$  is the fringe pattern volume for the growing hemisphere object.

$noise_1(x, y, t)$  is a Gaussian noise term with zero mean and standard deviation of 0.5.

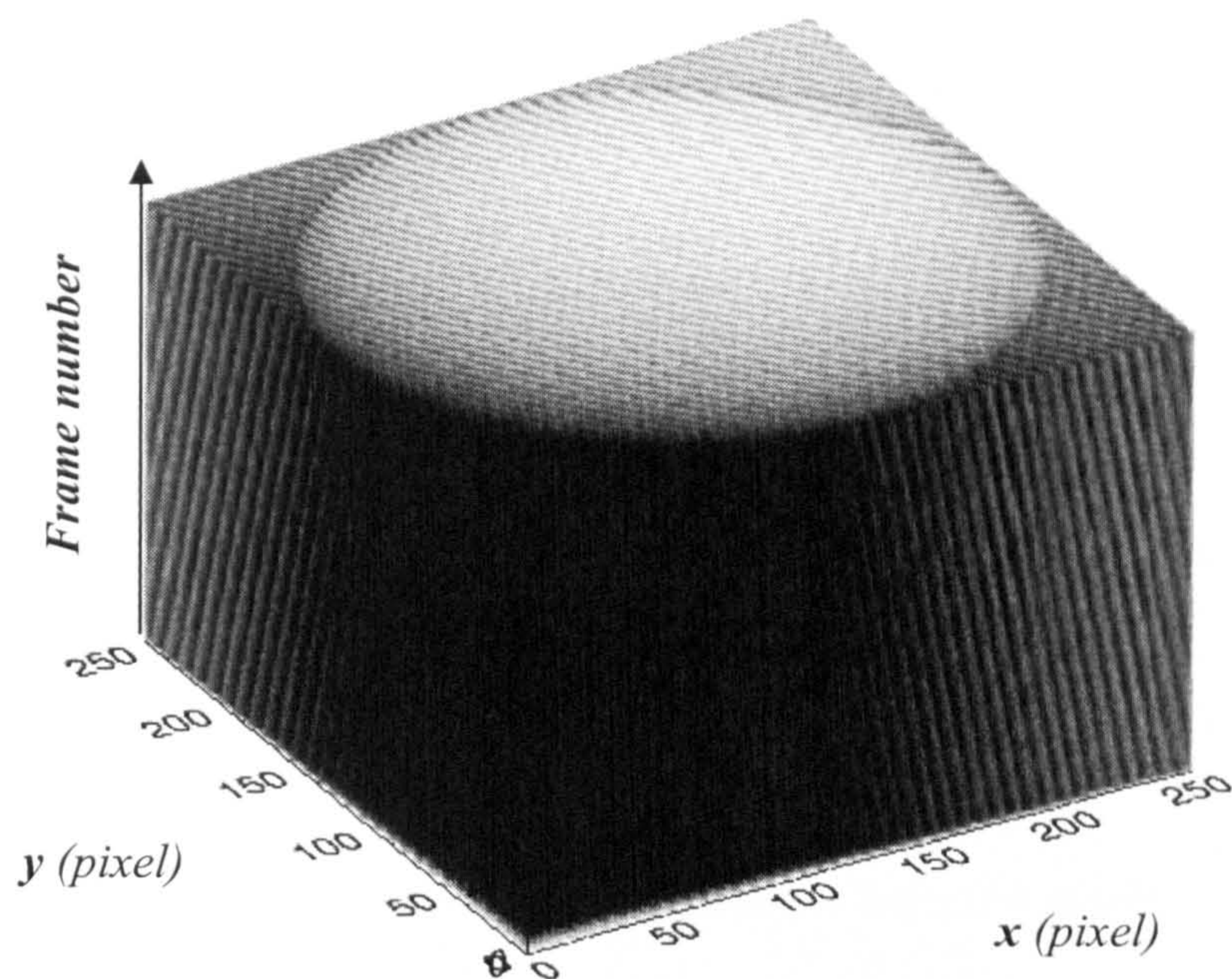
$noise_2(x, y, t)$  is a Gaussian noise term with zero mean and standard deviation of 0.1.

$Obj_i(x, y, t)$  is mathematical modelling of the simulated spherical object as defined in appendix A.1.

Note that the noise values are chosen arbitrary.

Figure 4.4 shows the fringe pattern volume of the sphere. This fringe pattern volume has to be processed by the means of the three-dimensional Fourier fringe analysis technique. Similarly to the two-dimensional FFA technique, this volume has to be smoothed first using a Hanning volume which minimizes the leakage effect caused by using a discrete version of the Fourier transform.





**Figure 4.4: The fringe patterns volume for the spherical object**

After applying the Hanning volume, the fringe pattern volume is transformed into the Fourier domain using a three-dimensional Fourier transform. The frequency spectrum of this volume is shown in Figure 4.5. Then, a three-dimensional filtering volume is applied to filter out all the unwanted components and to keep the desired component that contains the phase information. Figure 4.6 shows the results of the filtering operation. These results were obtained using an ideal cubic filter that has a value of one inside the filter passband volume and all values zeroed elsewhere. The selected frequency component should then be shifted back to the origin to remove the carrier frequencies in the  $x$ ,  $y$  and  $z$  directions, as explained previously in this chapter.



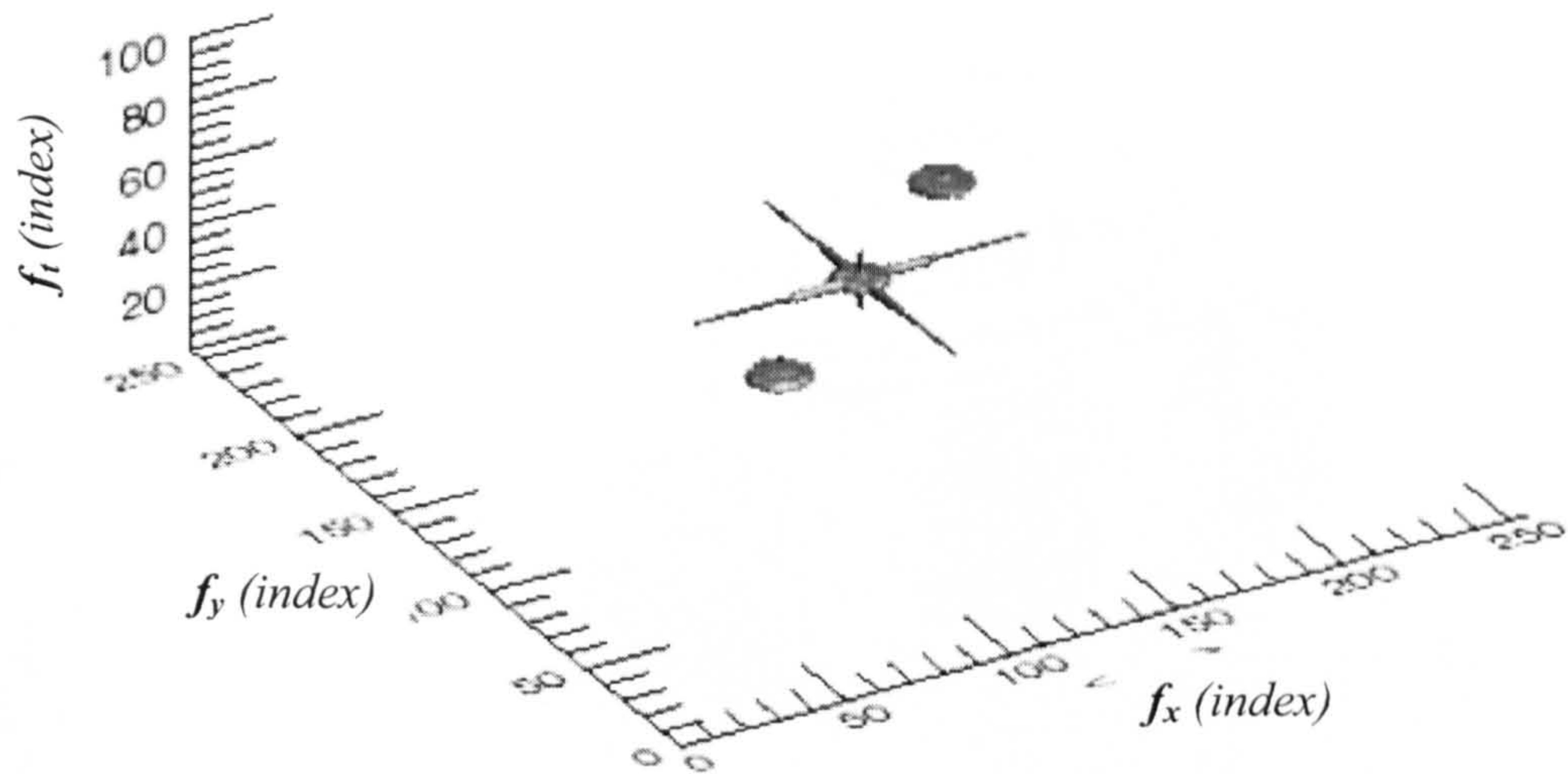


Figure 4.5: Frequency spectrum of the spherical object

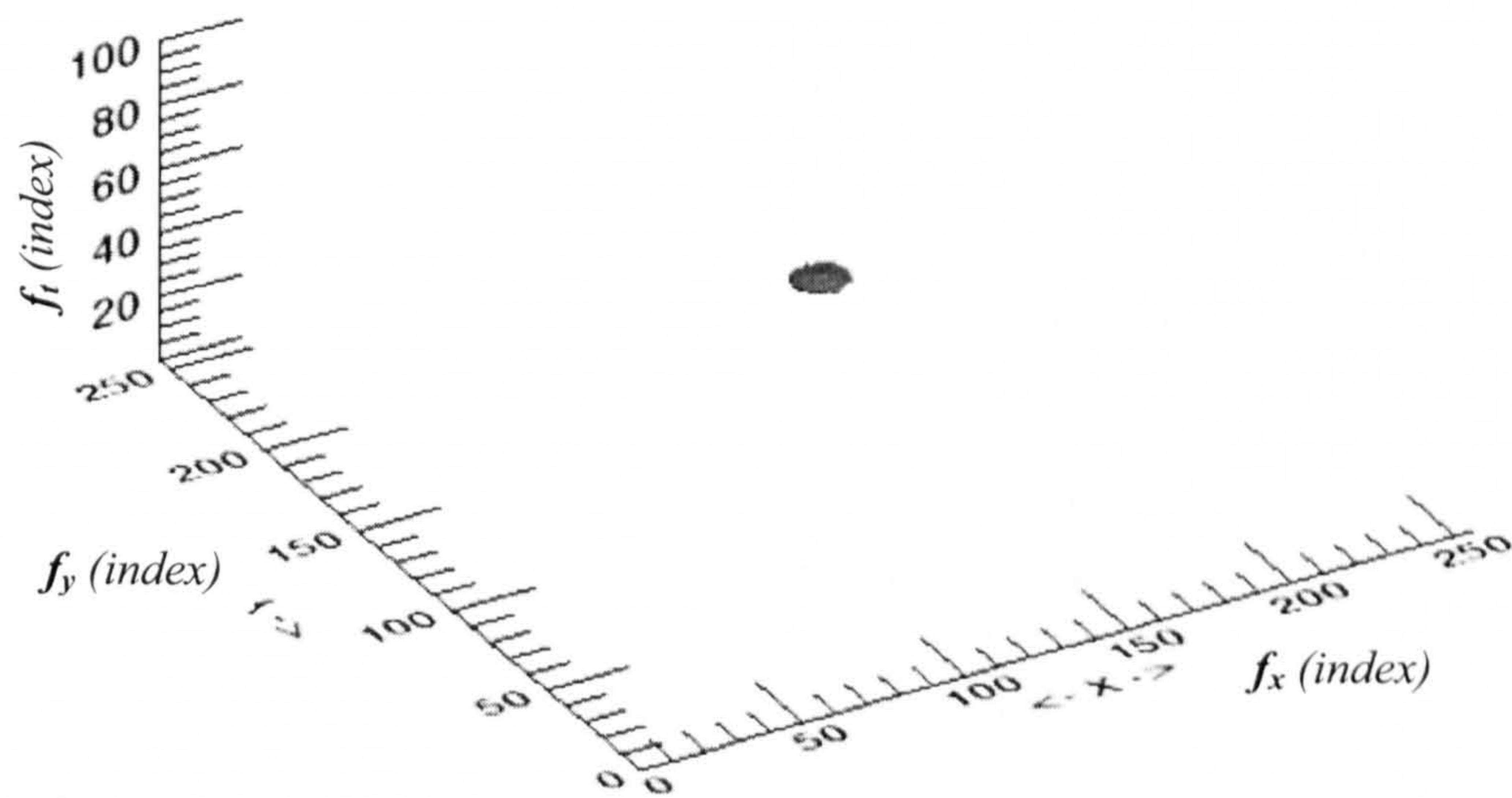
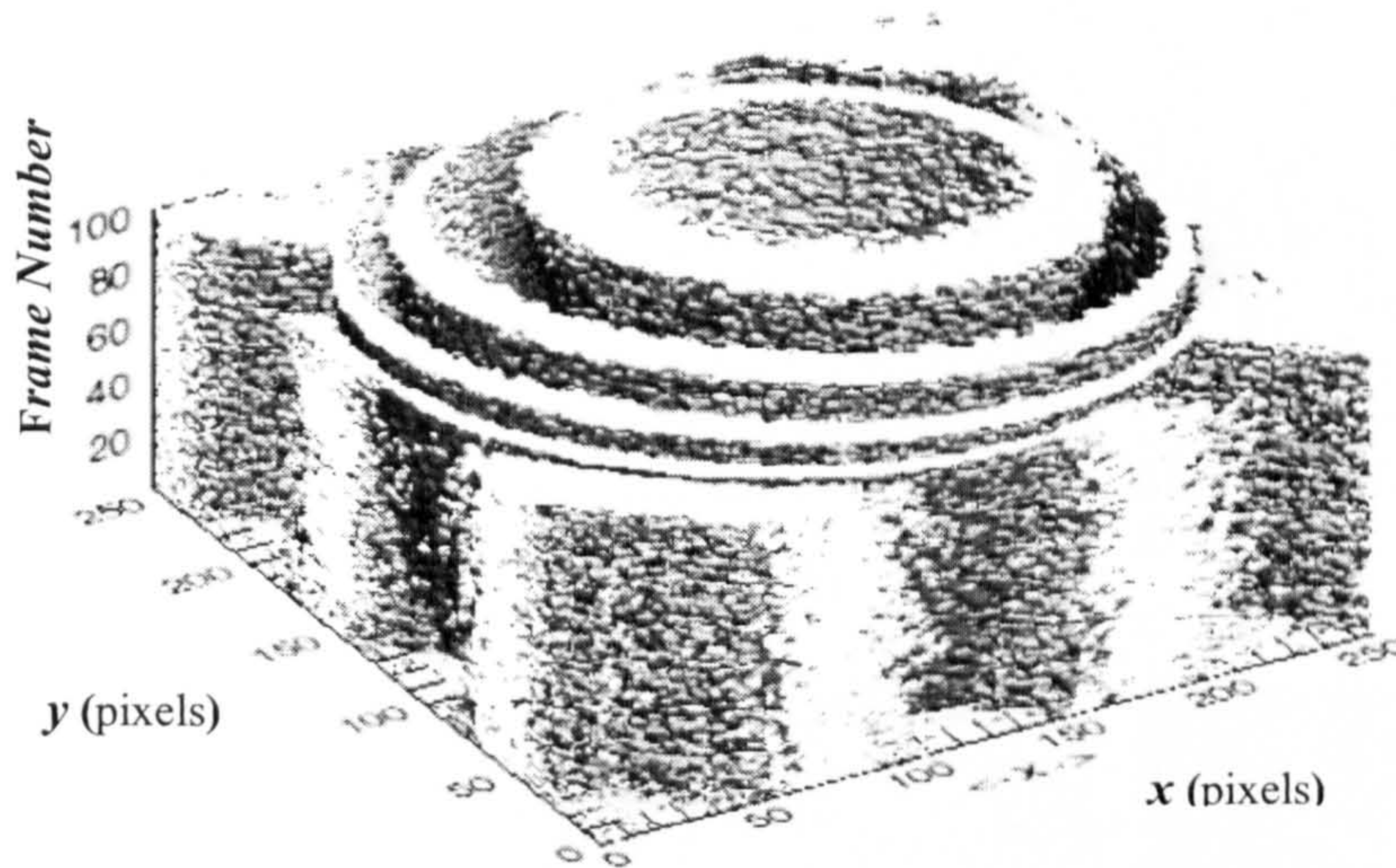


Figure 4.6: The filtered and shifted spectrum of the spherical object

After filtering out all unwanted components, the wrapped phase volume is calculated using Equation 4.8. Figure 4.7 shows the wrapped phase volume for this spherical object.



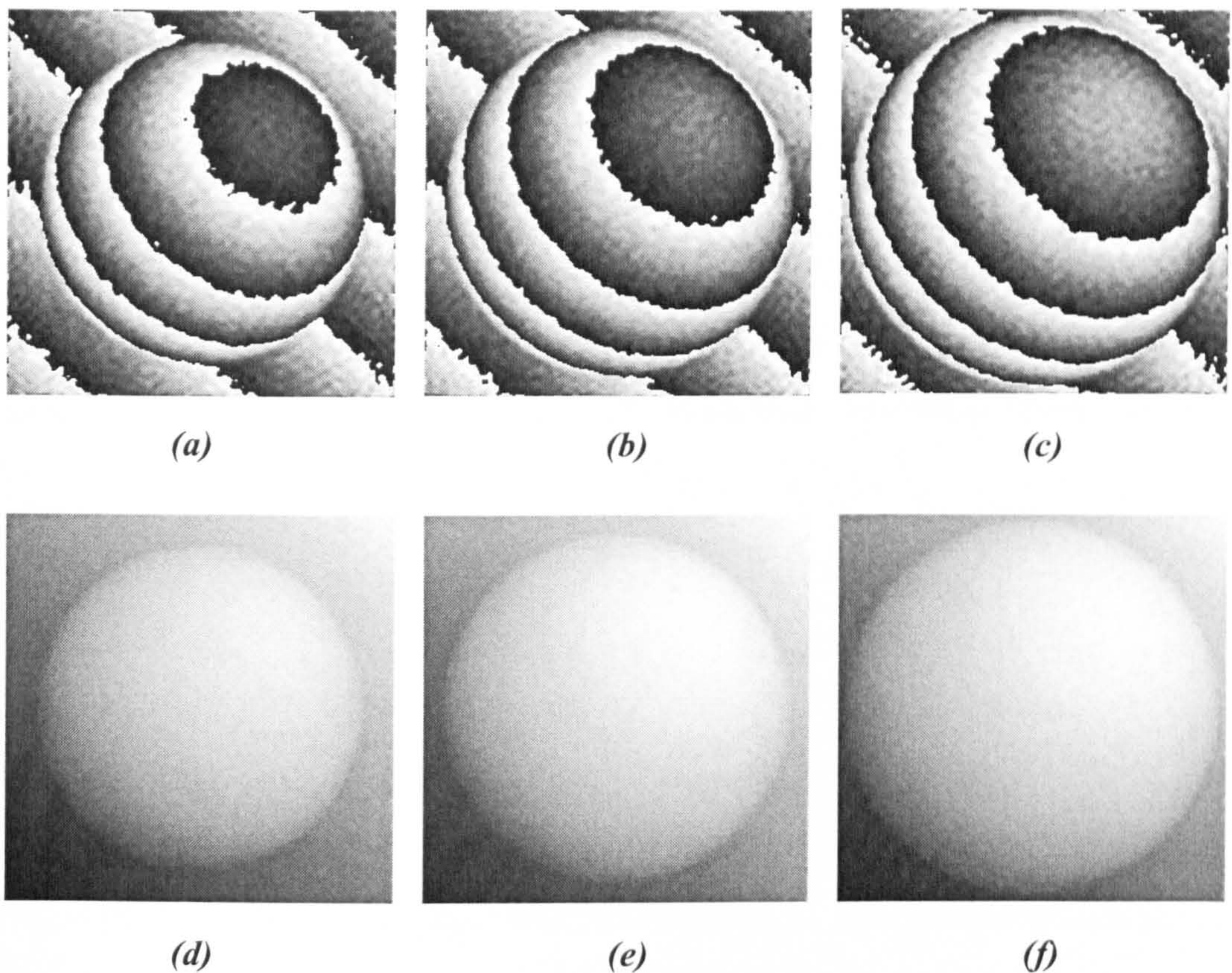


**Figure 4.7: The wrapped-phase volume of the spherical object**

This wrapped volume is unwrapped using a three-dimensional phase unwrapping algorithm. The author himself designed two robust three-dimensional phase unwrapping algorithms, called the best path algorithm (Abdul-Rahman *et al.*, 2005; Abdul-Rahman *et al.*, 2006) and the best path avoiding singularity loops algorithms. Both algorithms are indeed robust and reliable, as will be demonstrated in the next chapter.

The three-dimensional best path phase unwrapping algorithm is applied to unwrap the wrapped phase volume shown in Figure 4.7 above. Figure 4.8 shows the wrapped and the unwrapped phase maps for the 10<sup>th</sup>, 50<sup>th</sup> and 90<sup>th</sup> frames. As shown in this figure, the radius of the sphere is increasing with time, *i.e.* with the frame number, which means that the three-dimensional Fourier transform fringe analysis technique is able to track the movement of this dynamic object.

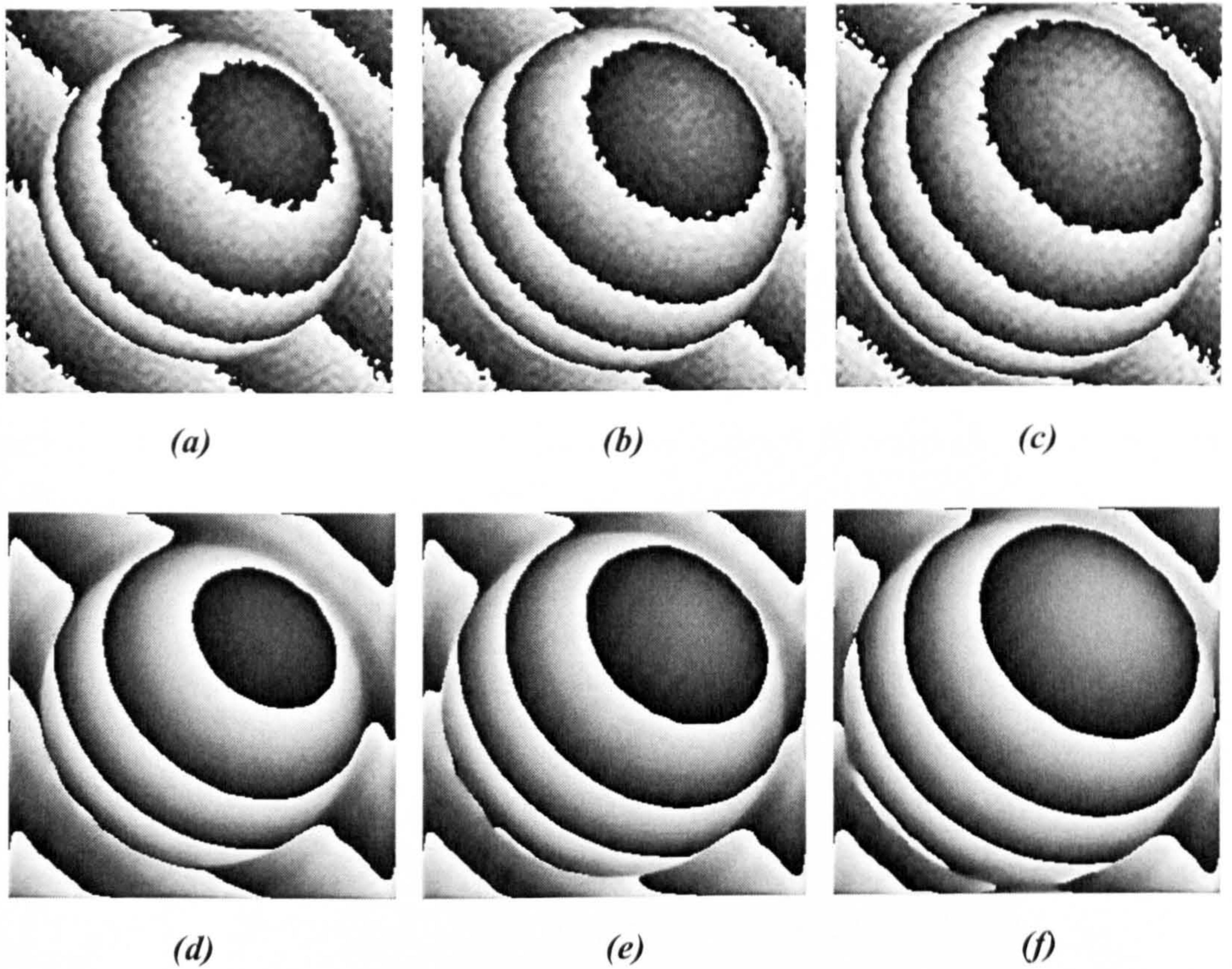




**Figure 4.8: The three-dimensional Fourier transform fringe analysis results for frames 10, 50 and 90 of the spherical object. (a), (b) and (c) The wrapped phase maps. (d), (e) and (f) The unwrapped phase maps using the three-dimensional best-path phase unwrapping algorithm.**

As mentioned earlier in this chapter, filtering in the frequency domain is one of the most important steps in any Fourier fringe analysis system. Figure 4.9 shows the effect of this operation on the quality of the wrapped phase volume. The first row in this figure represents three wrapped phase maps for frame numbers 10, 50 and 90 respectively. These wrapped phase maps were obtained by the application of an ideal cubic filter on the frequency domain data. The second line shows the resultant wrapped phase maps for the same frames when applying a Butterworth filter. The use of the Butterworth filter has apparently improved the wrapped-phase quality and reduces the ringing effects as can be seen from the figure.





**Figure 4.9: the effect of frequency filtering on the wrapped phase. The wrapped phase maps resulted from applying: (a), (b) and (c) Ideal cubic filter; (d), (e) and (f) Butterworth filter**

The second computer generated object is a complicated moving object whose surface shape is changing with time, *i.e.*, with the frame number (see Appendix A.2). Each frame of this object consists of  $256 \times 256$  pixels, whereas, the movement of this object is carried out during 100 successive video frames. A simulated fringe pattern volume is created for this dynamic object using the equation:

$$g_{ii}(x, y, t) = [Obj_{ii}(x, y, t) + noise_1(x, y, t)] + [1 + noise_2(x, y, t)] \cos(2\pi(15x + 10y + 5z) + Obj_{ii}(x, y, t)) \quad (4.12)$$

where;

$g_{ii}(x, y, t)$  is the fringe pattern volume for the second object.



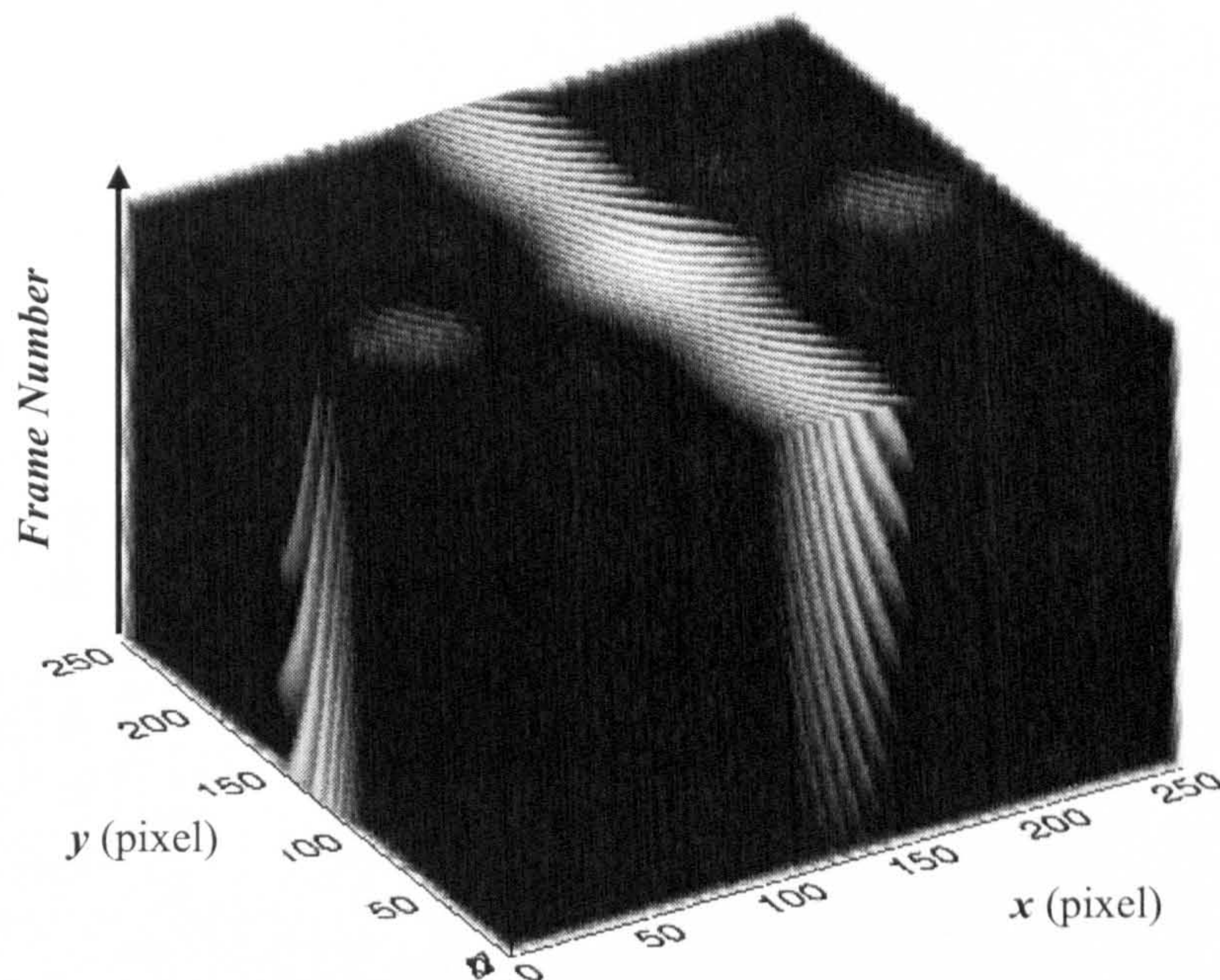
$noise_1(x, y, t)$  is a Gaussian noise term with zero mean and standard deviation of 0.5.

$noise_2(x, y, t)$  is a Gaussian noise term with zero mean and standard deviation of 0.1.

$Obj_{ii}(x, y, t)$  is mathematical modelling for the second simulated object as defined in appendix A.2.

Note that the values of noise are chosen arbitrary.

The three-dimensional Fourier fringe analysis algorithm is used to demodulate the fringe pattern sequence for this object. The steps of the proposed three-dimensional Fourier transform fringe analysis are shown in Figures 4.10 and 4.11 and the wrapped-phase volume is shown in Figure 4.12.



**Figure 4.10:** The fringe patterns volume for the second simulated object



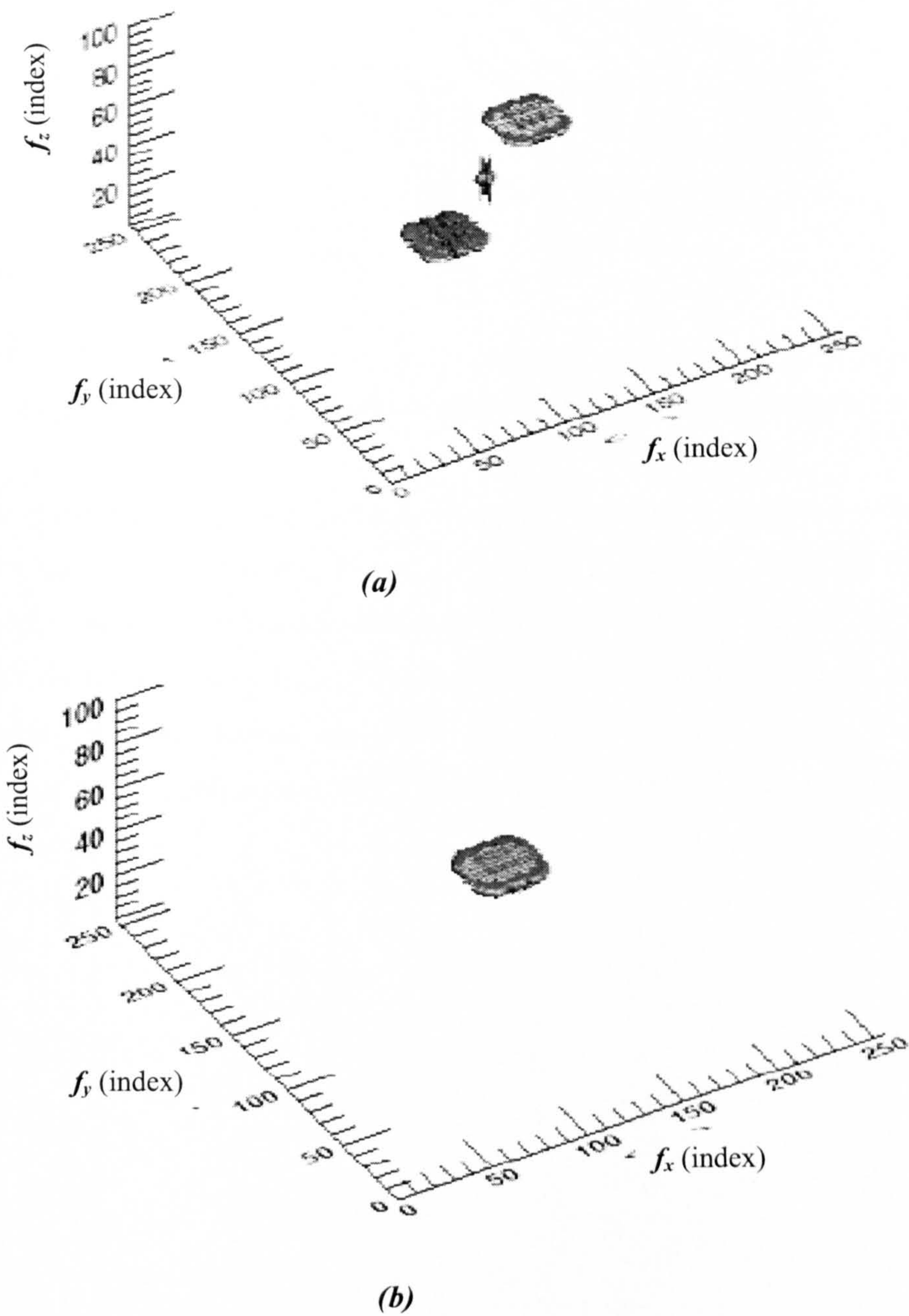
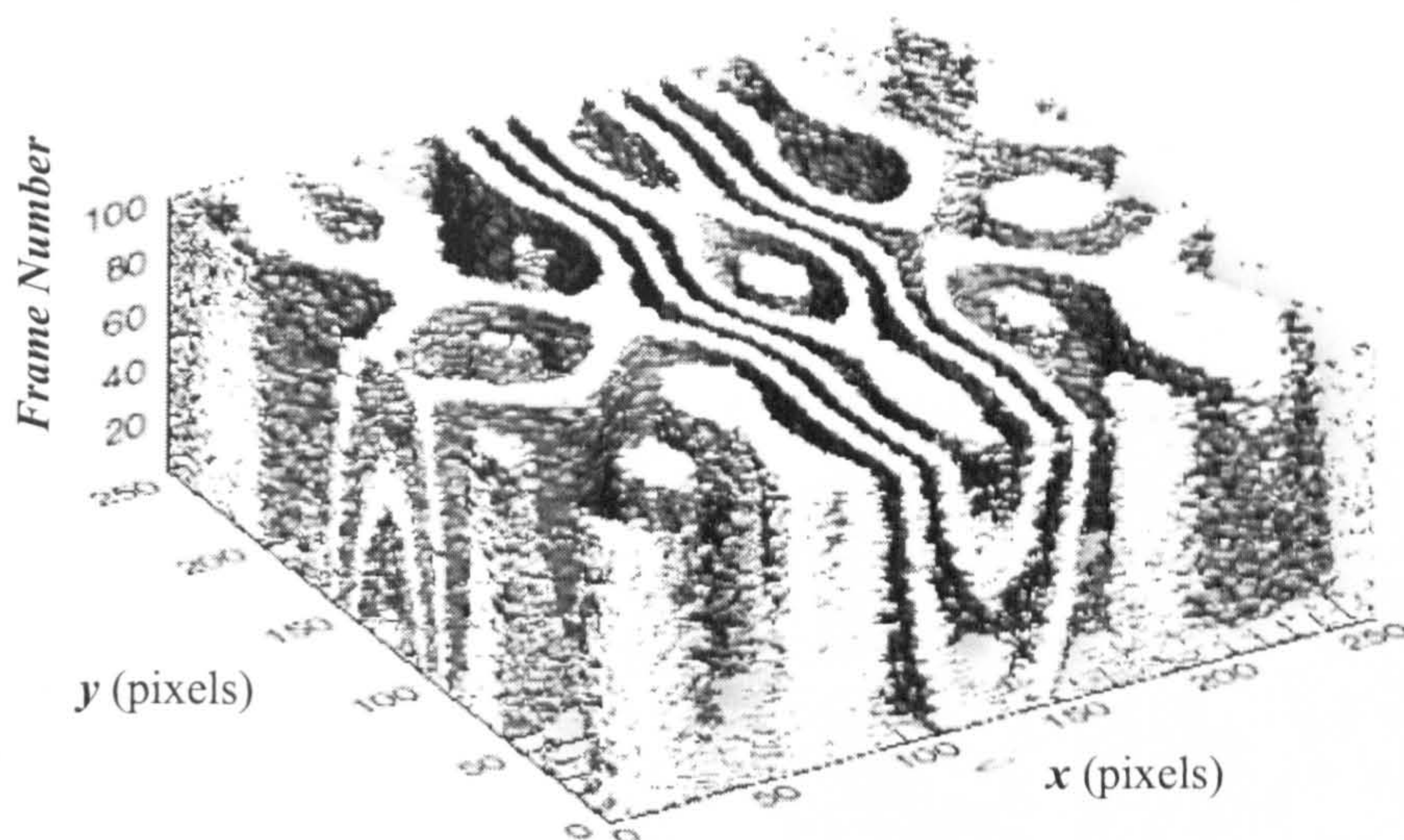


Figure 4.11: (a) Frequency spectrum of the second simulated object, (b) The required frequency component of the second simulated object

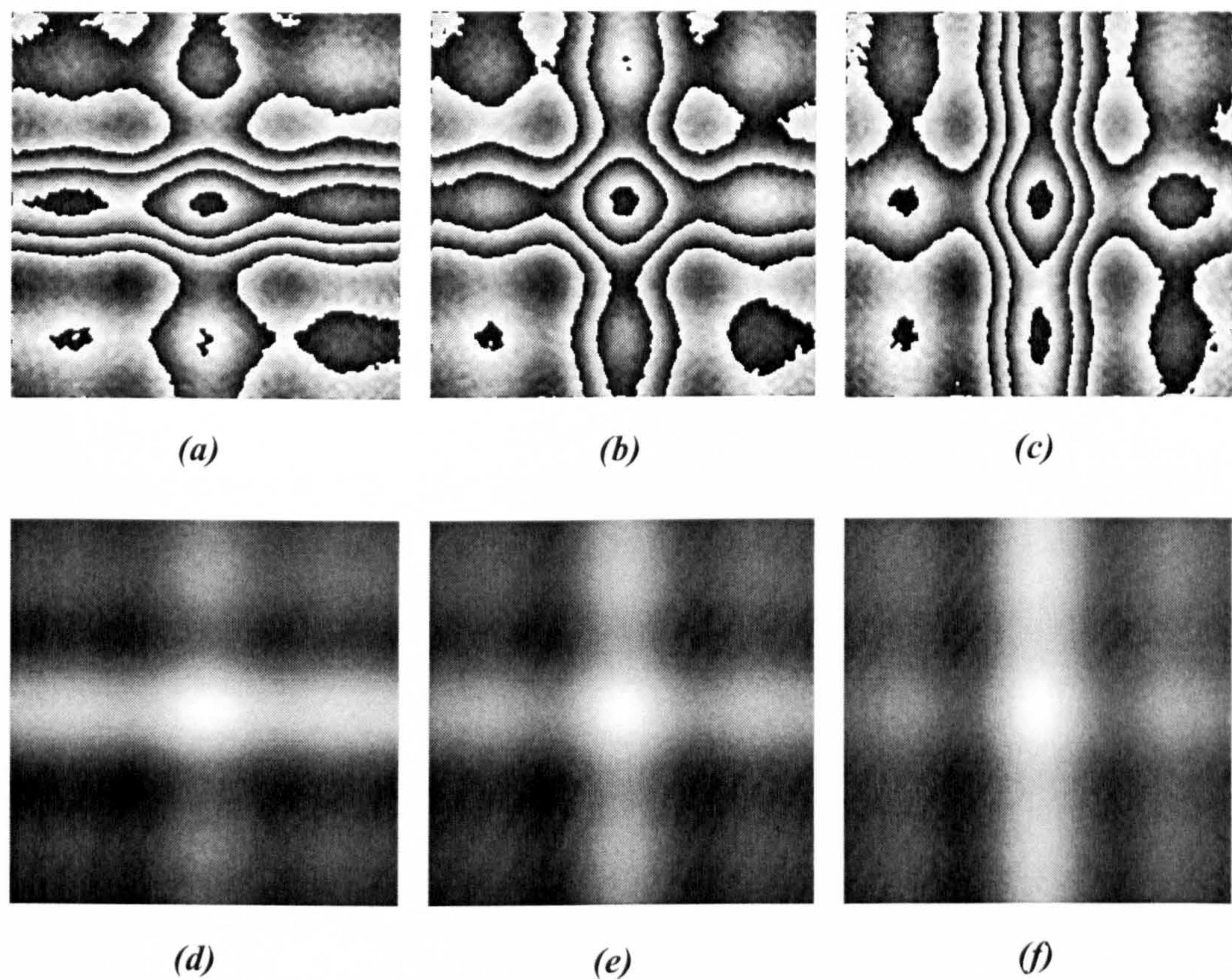




**Figure 4.12: The wrapped-phase volume of the second simulated object**

The three-dimensional best path phase unwrapping algorithm is also used to unwrap the wrapped phase volume shown in Figure 4.12 above. Figure 4.13 shows the unwrapped phase results for arbitrary frames. The wrapped phase maps for the 10<sup>th</sup>, 50<sup>th</sup> and 90<sup>th</sup> frames are shown in Figures 4.13(a), (b) and (c) respectively. Meanwhile the unwrapped phase maps for those frames are shown in Figures (d), (e) and (f). Figure 4.13 demonstrates that the proposed 3D-FFA technique is capable of tracking the movement of this dynamic object.





**Figure 4.13:** The three-dimensional Fourier transform fringe analysis results for frames 10, 50 and 90 of the second simulated object. The wrapped phase maps are shown in (a), (b) and (c). The unwrapped phase maps using the three-dimensional best-path phase unwrapping algorithm are shown in (d), (e) and (f)



## 4.5 Experimental Results

The proposed three-dimensional Fourier transform fringe analysis algorithm has also been evaluated experimentally in that it has been used to demodulate several real three-dimensional fringe pattern volumes. A video sequence that consists of the moving modulated fringe patterns was captured over a certain measurement period for an object. This video sequence was subsequently analysed using the three-dimensional Fourier fringe analysis algorithm. Analysing a fringe pattern volume involves first smoothing by a Hanning volume, then transforming to the frequency domain, filtering out all unwanted frequency components in the frequency domain and extracting the wrapped phase volume. This wrapped phase volume is then unwrapped using a novel best path three-dimensional phase unwrapping algorithm proposed by the author (Abdul-Rahman *et al.*, 2005; Abdul-Rahman *et al.*, 2006) which will be explained in the next chapter.

The practical application that the proposed three-dimensional Fourier transform fringe analysis algorithm is tested on is that of dynamic human body measurement in radiotherapy treatment. This is an exacting application of 3D measurement technology, which requires accurate dynamic measurements to be made in conditions that are far from ideal.

Two fringe pattern volumes have been used to test the performance of the proposed algorithm in demodulating fringe patterns for dynamic objects. The first example is taken from experiments carried out in the laboratory, simulating patient motion and the second example is from a real clinical setting. Each fringe pattern volume consists of 50 frames with  $512 \times 512$  pixels in each frame. The first fringe pattern volume has been obtained from analysing the chest/thorax region of a mannequin that has been manually raised and lowered in the mid-sagittal plane with a dorsal/ventral motion simulating respiration. The second volume is actual patient data acquired in a real medical clinical trial in a radiotherapy treatment room, exhibiting respiratory and ordinary levels of expected patient movement for a patient who is actually undergoing treatment for breast cancer.



Figures 4.13 – 4.19 show the individual steps of the three-dimensional Fourier transform fringe analysis technique in analysing mannequin's chest described above. Figure 4.14 shows the volume of fringe patterns for the mannequin's chest. This volume is smoothed by means of a Hanning volume then transformed into the Fourier domain. The frequency spectrum is shown in Figure 4.15. The three components of the frequency spectrum can be seen more clearly in Figure 4.16. A 3-D Butterworth filter is applied to select the desired frequency component. The Butterworth filter is shown in Figure 4.17 whereas Figure 4.18 shows the filtered component. Finally, Figure 4.19 shows the wrapped-phase volume.

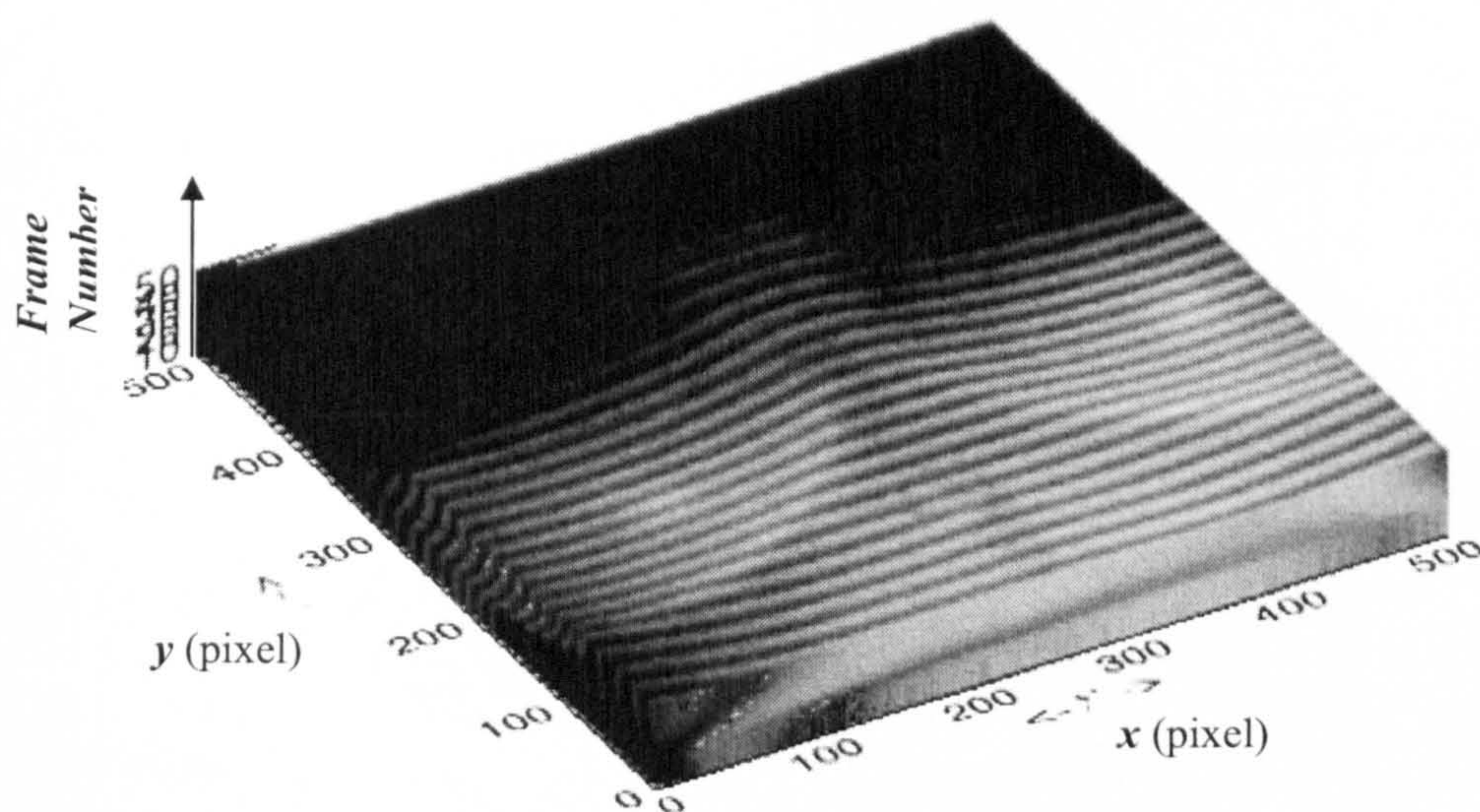


Figure 4.14: The fringe patterns volume of the mannequin's chest



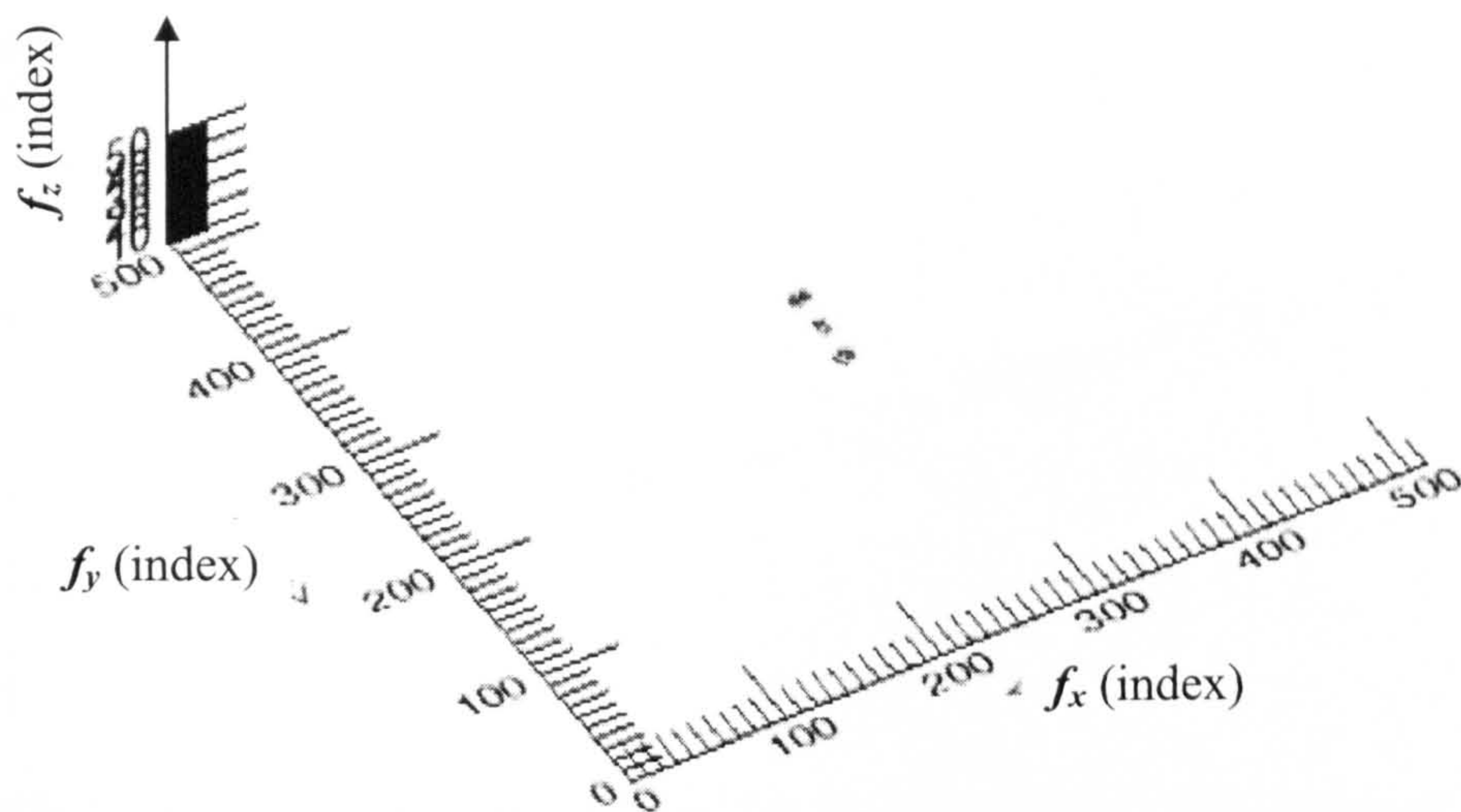


Figure 4.15: Frequency spectrum of the mannequin's chest fringe patterns volume

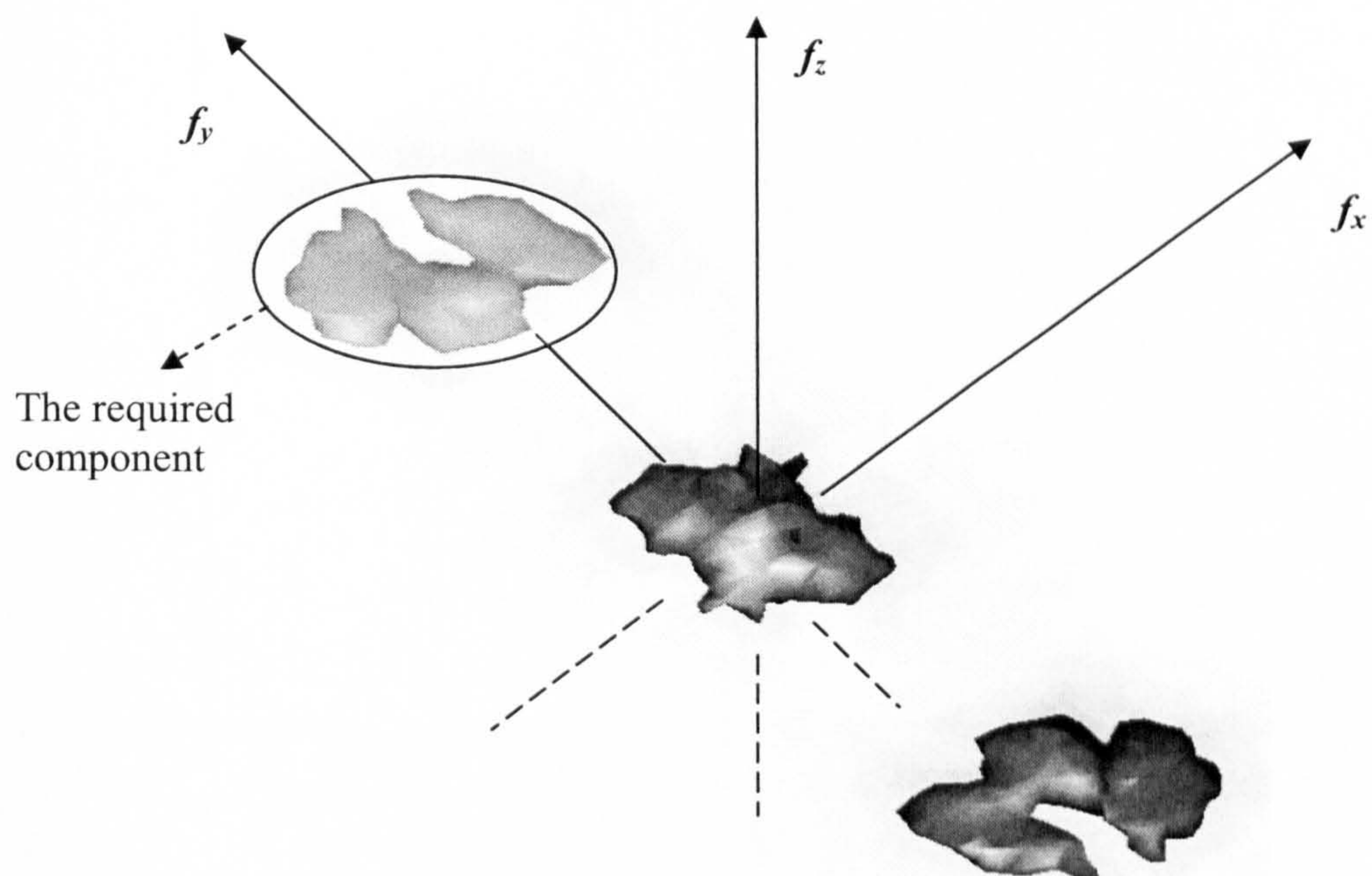


Figure 4.16: Frequency spectrum of the mannequin's chest fringe patterns volume (zoomed)



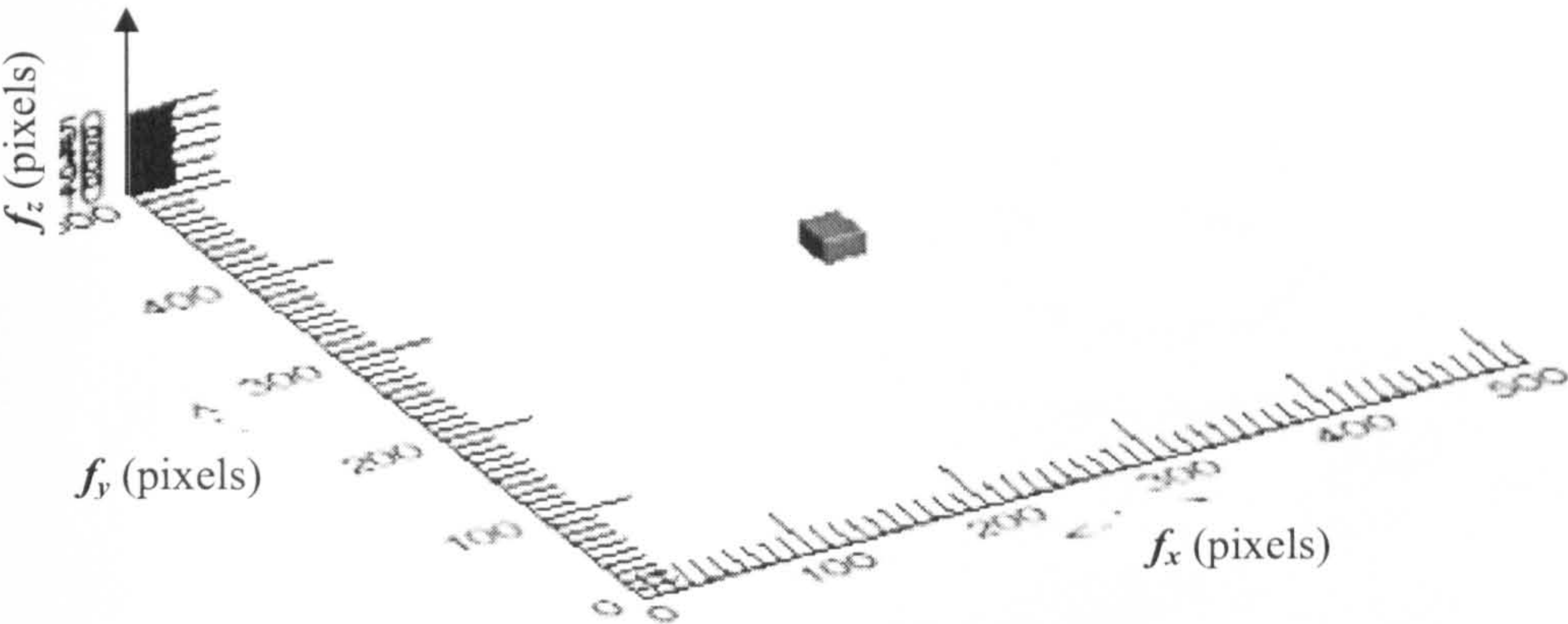


Figure 4.17: Butterworth filter designed for selecting the desired components of the mannequin’s chest.

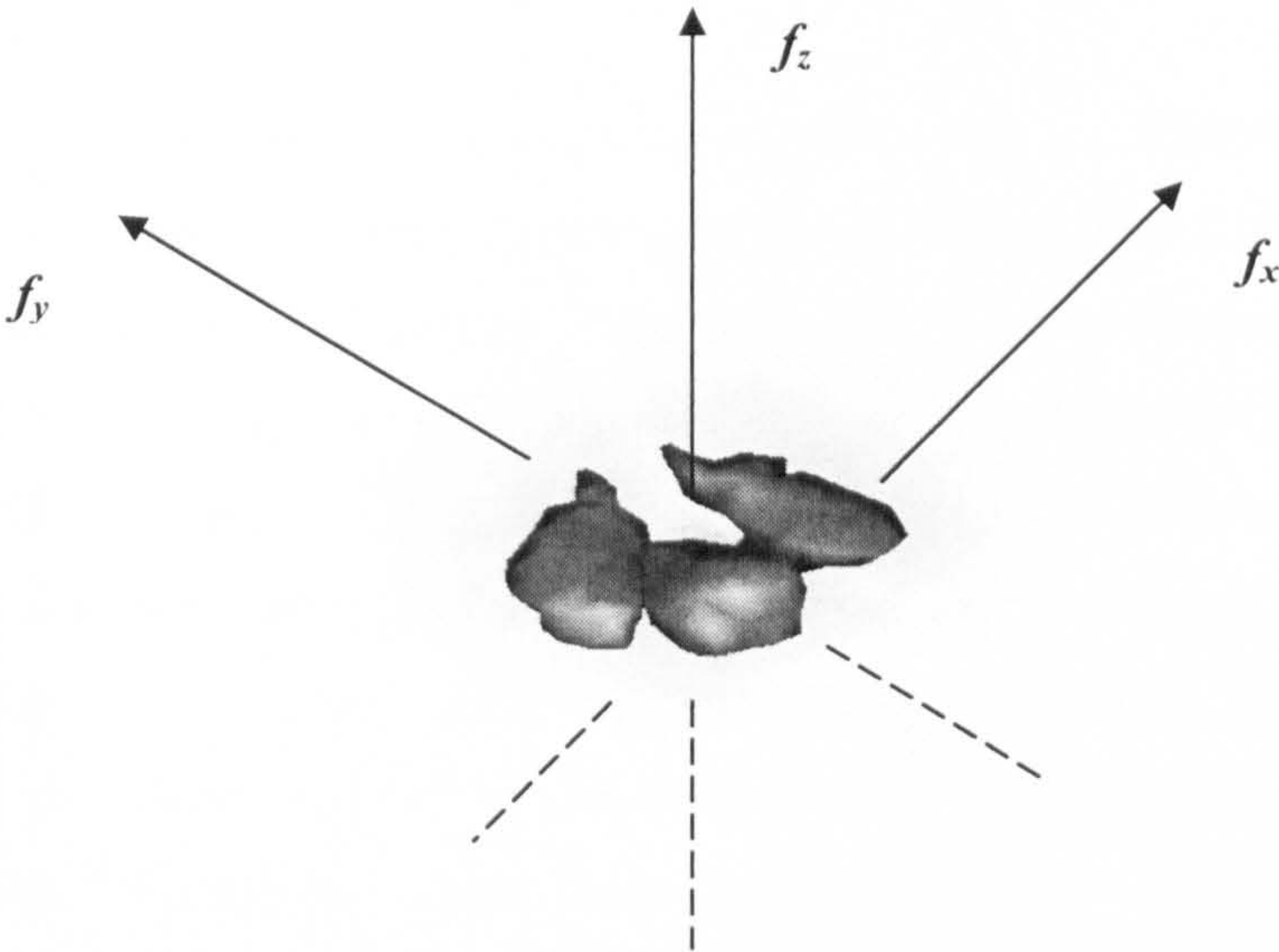
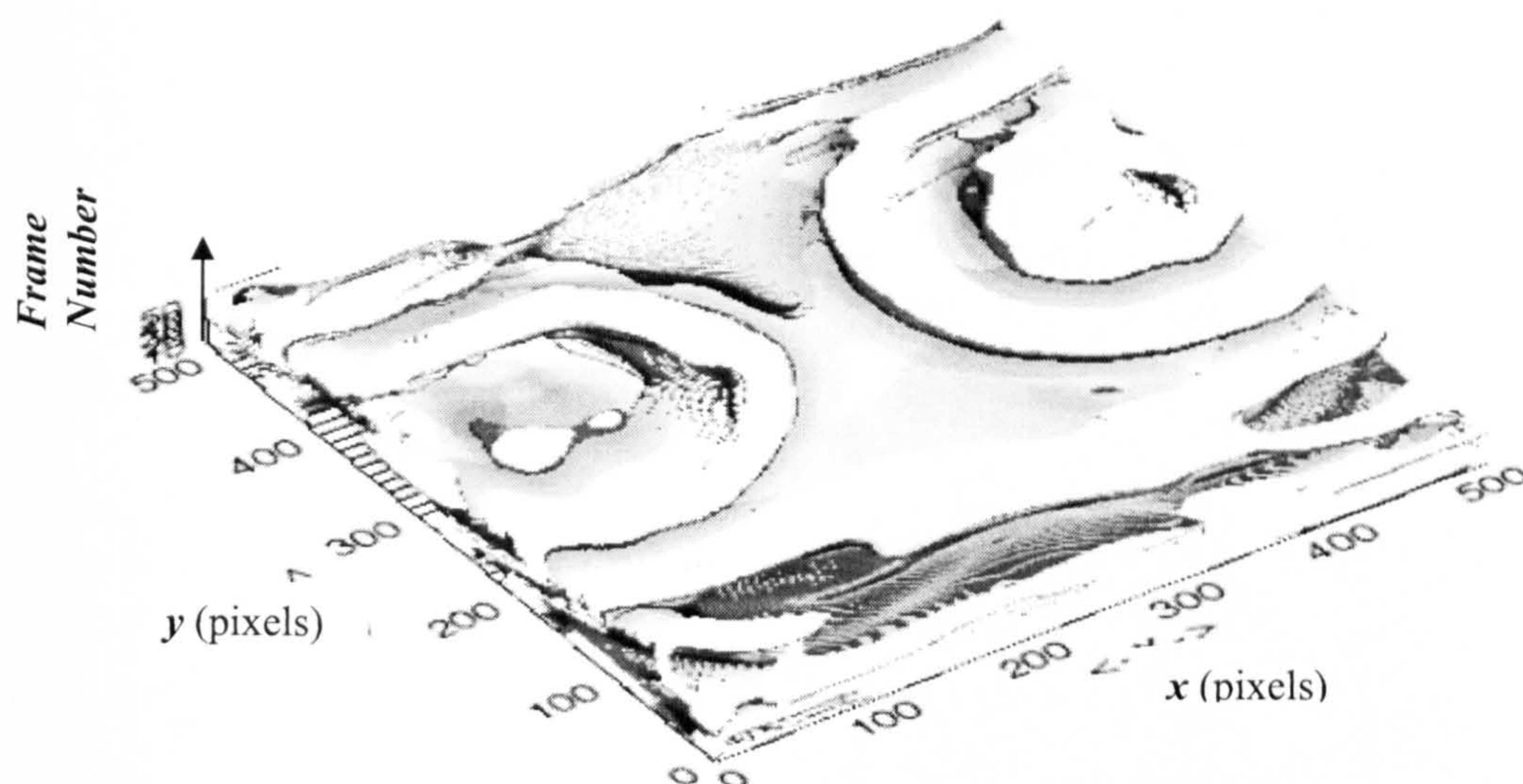


Figure 4.18: The desired frequency component in the mannequin’s chest

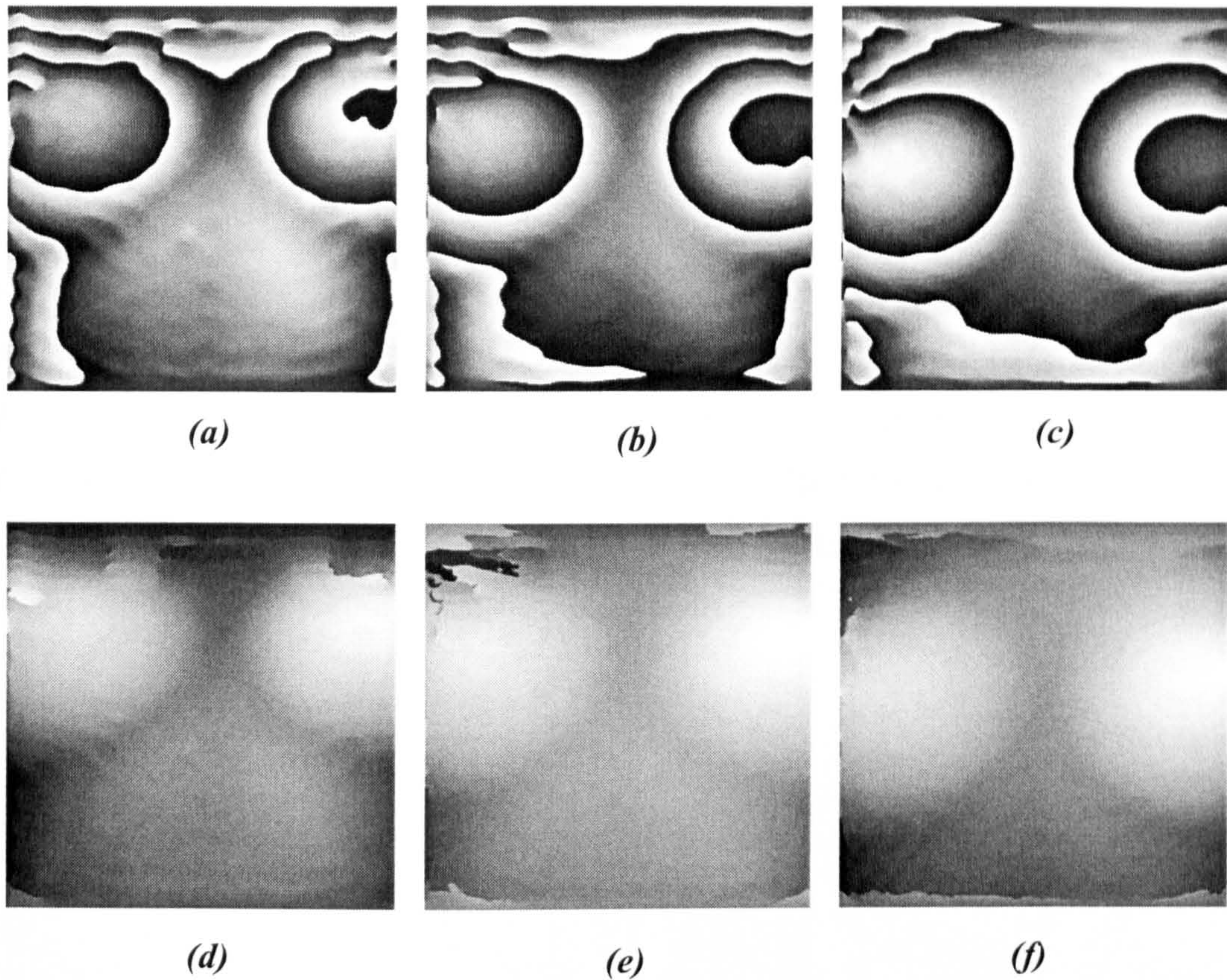




**Figure 4.19: The wrapped-phase volume of the mannequin's chest**

The previous wrapped-phase volume is unwrapped using the three-dimensional best path phase unwrapping algorithm. Figure 4.20 shows three different arbitrary frames of the wrapped-phase volume resulting from measuring the mannequin's chest. Figures (a), (b) and (c) show the wrapped-phase maps for frames number 10, 24 and 40 respectively. Figures (d), (e) and (f) show the unwrapped-phase maps resulting from the three-dimensional best-path phase unwrapping algorithm (Abdul-Rahman *et al.*, 2005; Abdul-Rahman *et al.*, 2006). The topic of three-dimensional phase unwrapping algorithms is discussed in detail in the next chapter. Clearly, from Figure 4.20, the three-dimensional Fourier transform fringe analysis technique was able to detect the movement of the object.



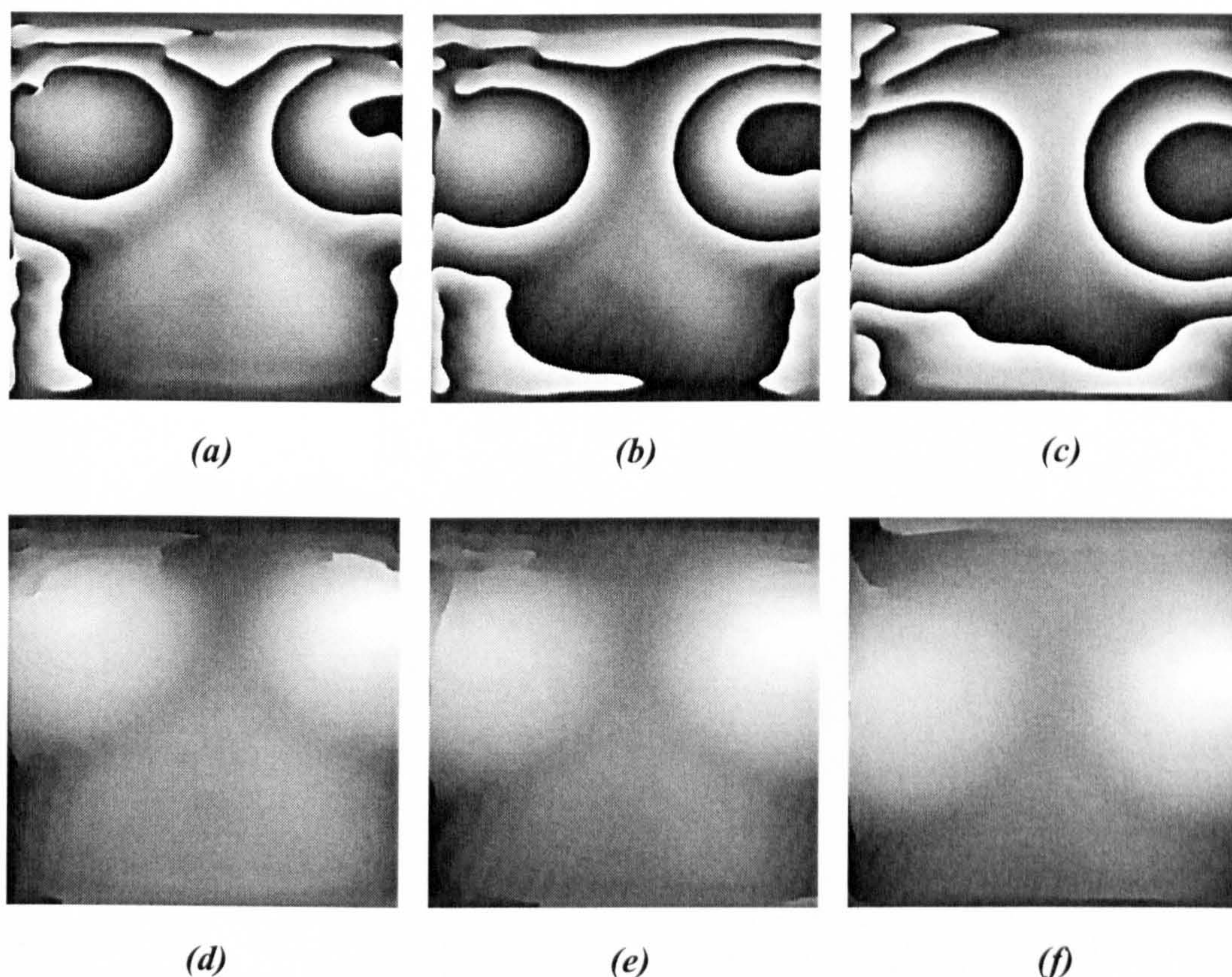


**Figure. 4.20: The three-dimensional Fourier transform fringe analysis results for frames 10, 24 and 40 of the mannequin's chest. (a), (b) and (c) The wrapped phase maps. (d), (e) and (f) The unwrapped phase maps using the three-dimensional best-path phase unwrapping algorithm.**

The performance of the three-dimensional Fourier transform fringe analysis method is compared with the state-of-art two-dimensional Fourier transform fringe analysis technique which is now used to measure the body shape of patients undertaking radiotherapy treatment in the Christie Hospital in Manchester (Lilley *et al.*, 2000). The resulting wrapped phase maps are unwrapped using a very robust two-dimensional phase unwrapping technique proposed by Herrarz *et al.* (Herraez *et al.*, 2002). Figure 4.21 shows the results of this two-dimensional Fourier transform fringe analysis method.

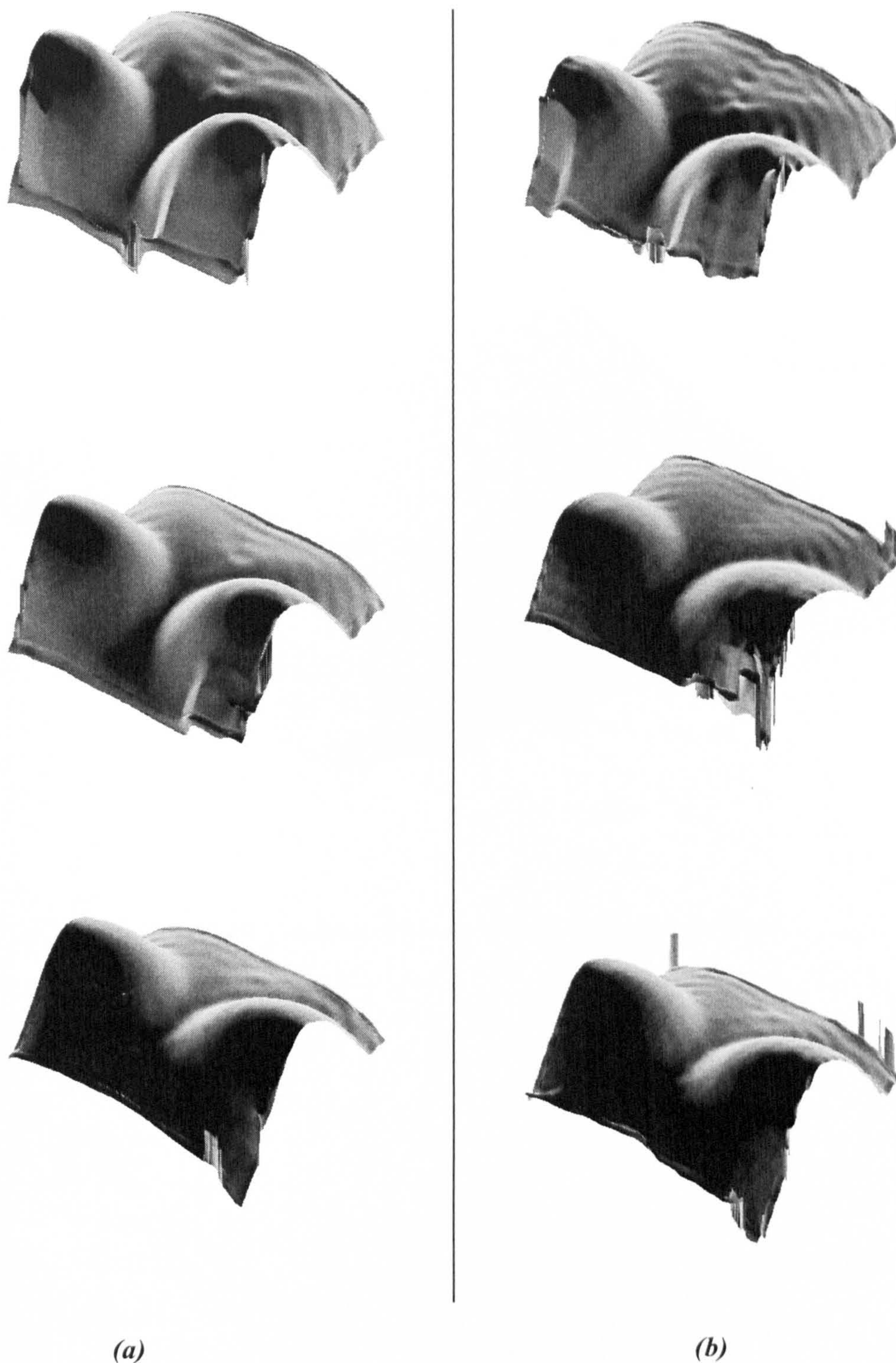


Comparing between Figures 4.20 and 4.21 shows that the current three-dimensional Fourier fringe analysis produces a good result which is almost the same as the two-dimensional Fourier fringe analysis despite the great deal of development that the two-dimensional Fourier transform fringe analysis has witnessed since 1986 (Bone *et al.*, 1986; Malcolm *et al.*, 1990; Burton and Lalor, 1994; Stephenson, 1994; Burton *et al.*, 1995; Gdeisat *et al.*, 2006). Figure 4.22 shows a three-dimensional comparison between the two-dimensional and three-dimensional Fourier transform fringe analysis algorithm.



**Figure 4.21: The two-dimensional Fourier transform fringe analysis results for frames 10, 24 and 40 of the mannequin's chest. (a), (b) and (c) The wrapped phase maps. (d), (e) and (f) The unwrapped phase maps using the two-dimensional best-path phase unwrapping algorithm.**





**Figure 4.22: Three-dimensional comparison between two-dimensional and three-dimensional Fourier transform fringe analysis for the mannequin's chest. Column (a) the results of the two-dimensional Fourier transform fringe analysis and, Column (b) the results of the three-dimensional Fourier transform fringe analysis**



The second experimental example is that of fringe patterns acquired from a real patient in a medical clinical trial in a radiotherapy treatment room as shown in Figure 4.23. The bright spot that appears in the figure is a laser spot that used to implement a laser triangulation technique in order to retrieve the absolute height. The steps of the proposed three-dimensional Fourier transform fringe analysis method are shown in Figures 4.24 – 4.27 and the wrapped-phase volume is shown in Figure 4.28.

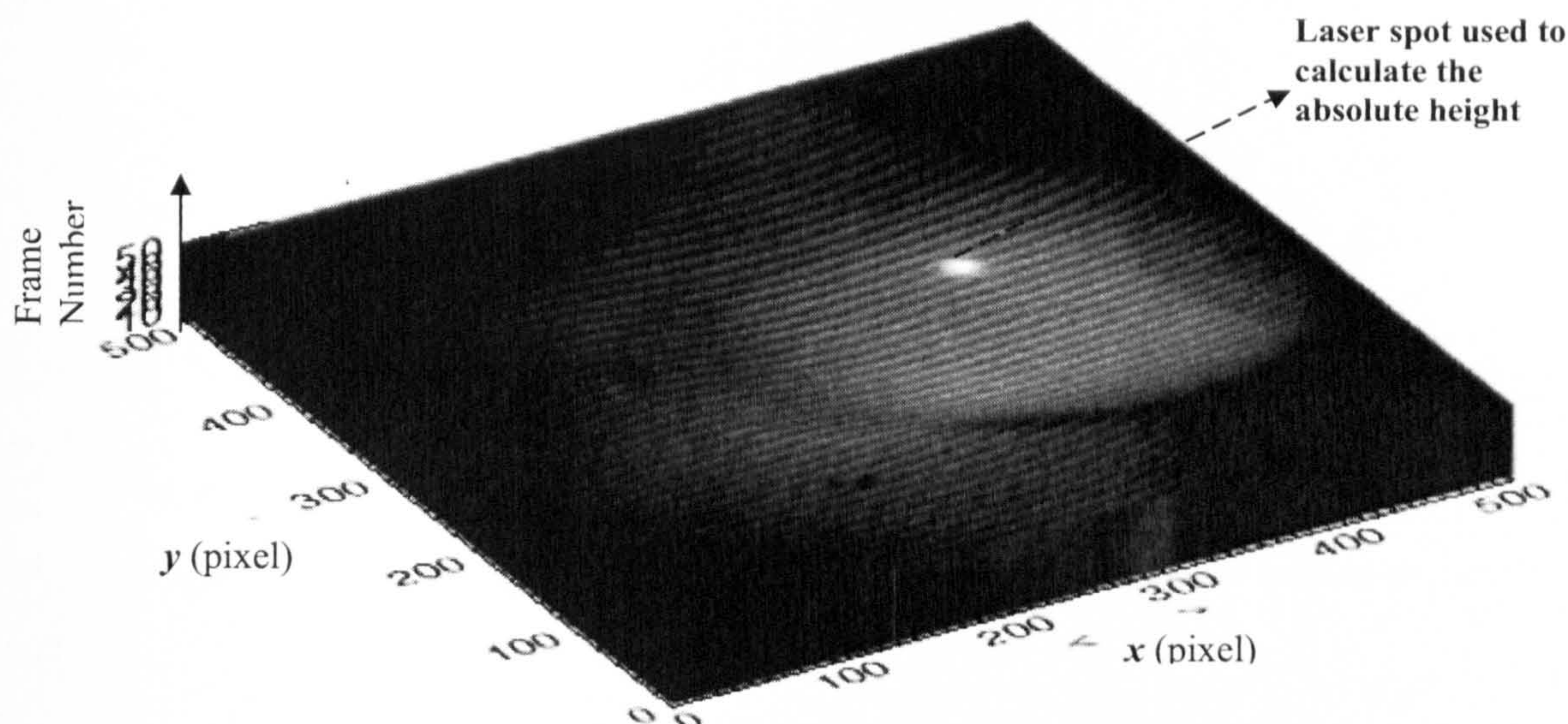


Figure 4.23: The fringe pattern volume of a patient's chest

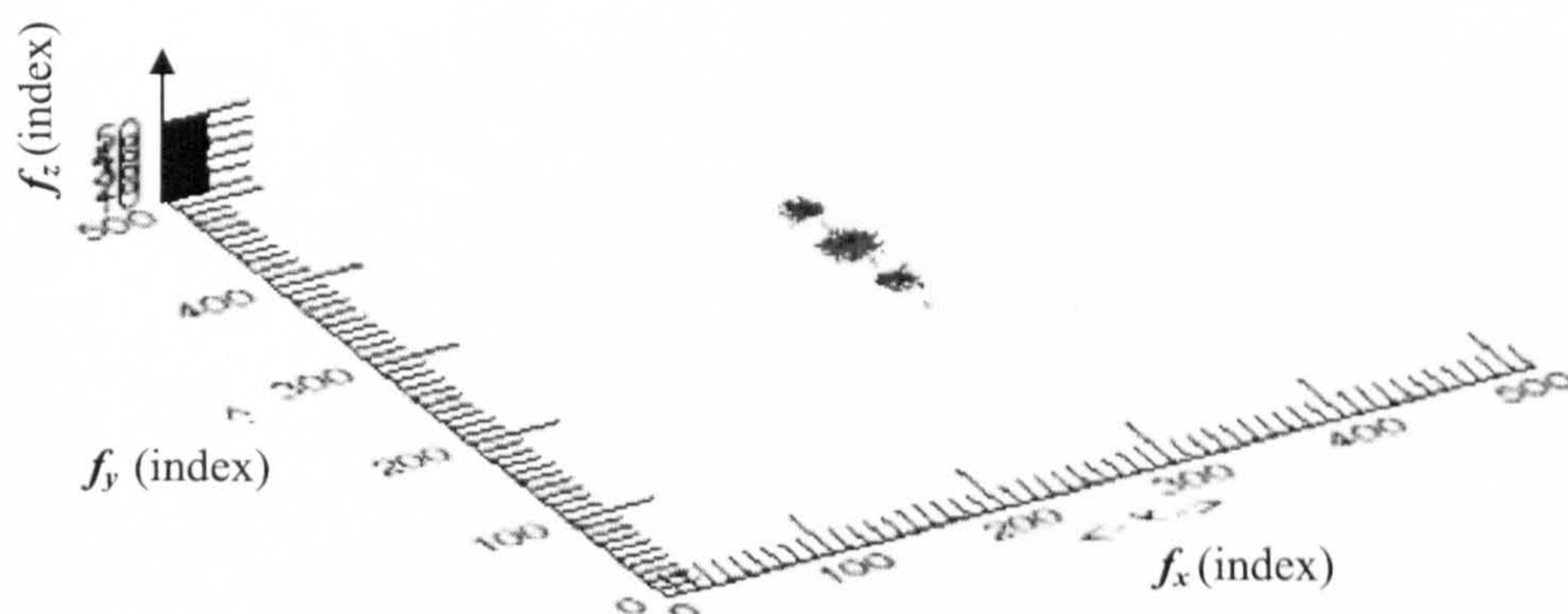


Figure 4.24: Frequency spectrum of a patient's chest fringe pattern volume



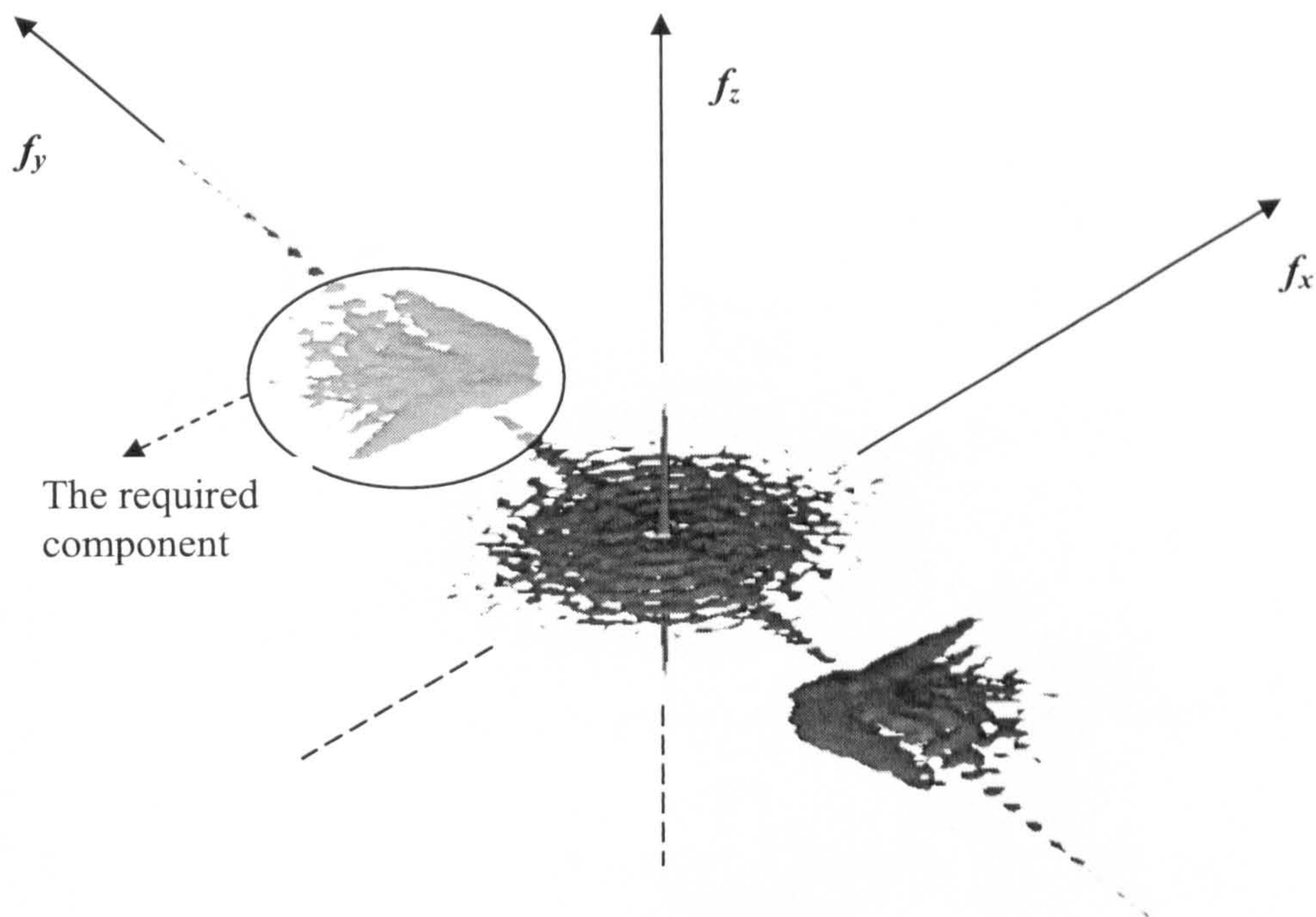


Figure 4.25: Frequency spectrum of a patient’s chest fringe pattern volume (zoomed)

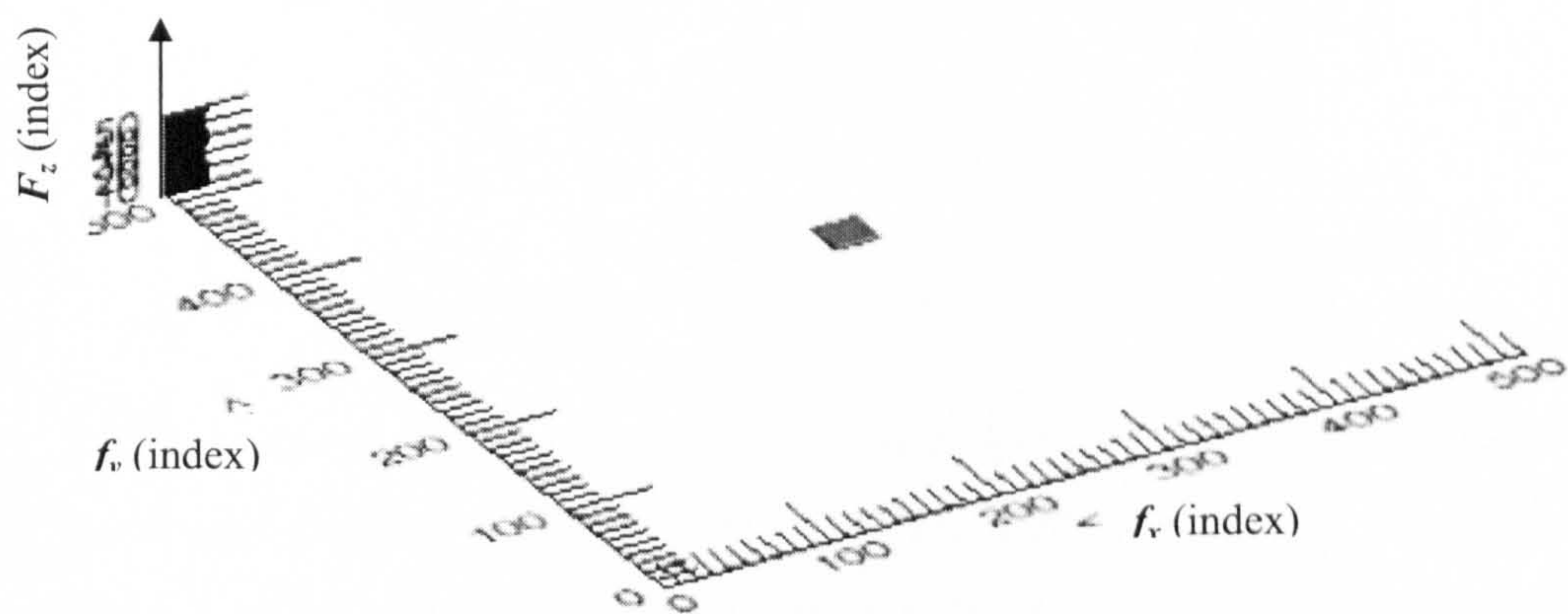


Figure 4.26: Butterworth filter designed for selecting the desired components of a patient’s chest



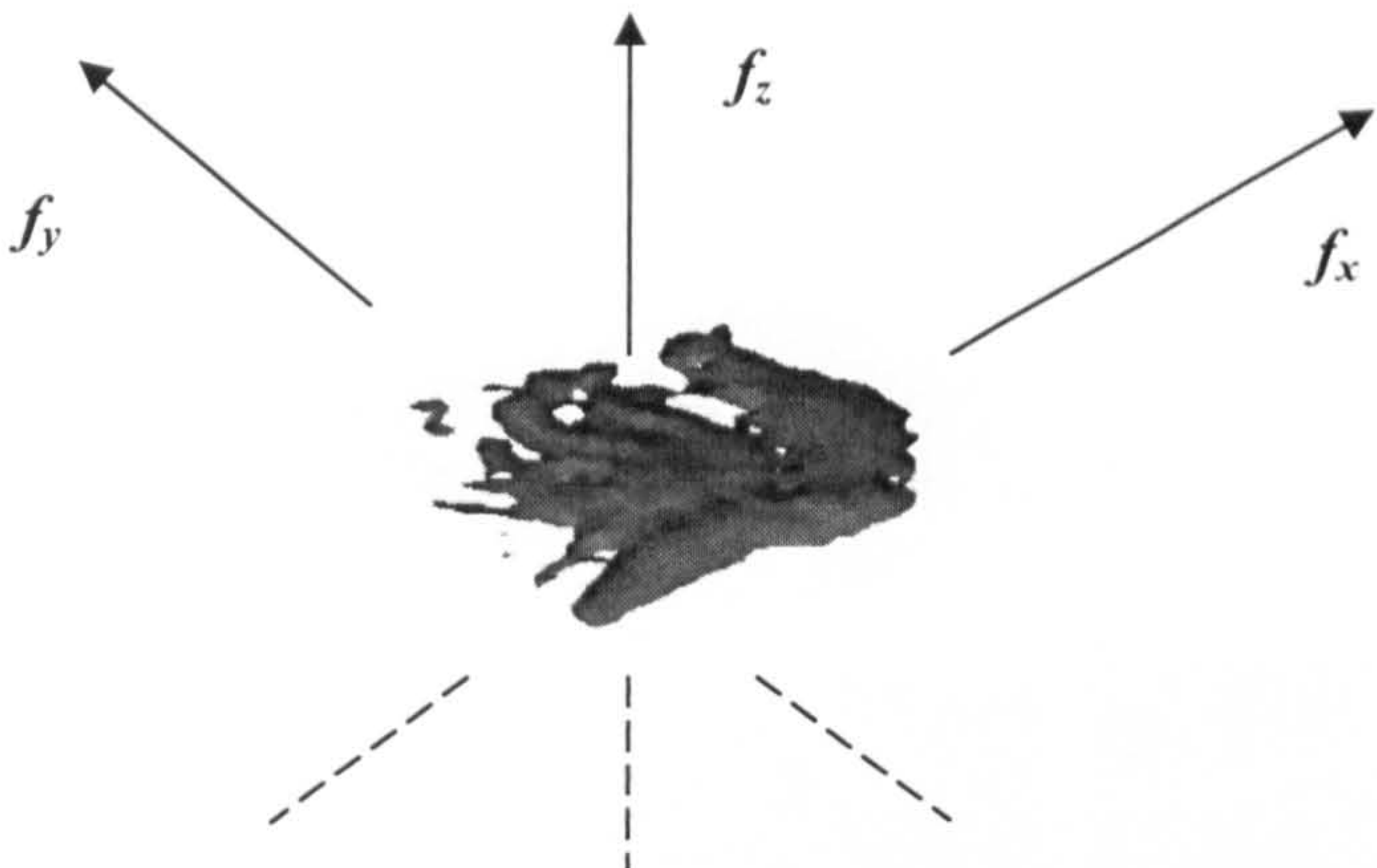


Figure 4.27: The desired frequency component in a patient’s chest

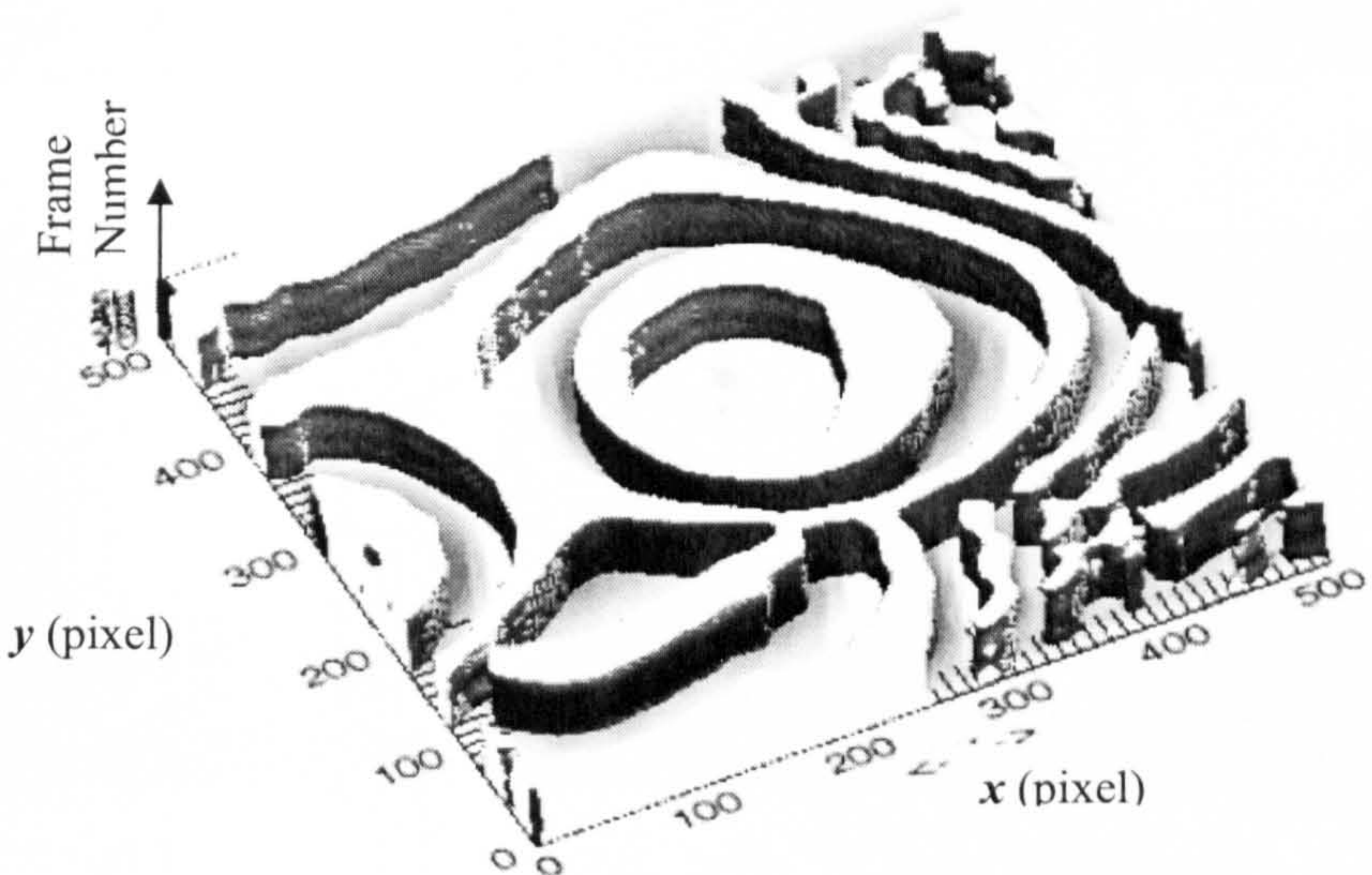
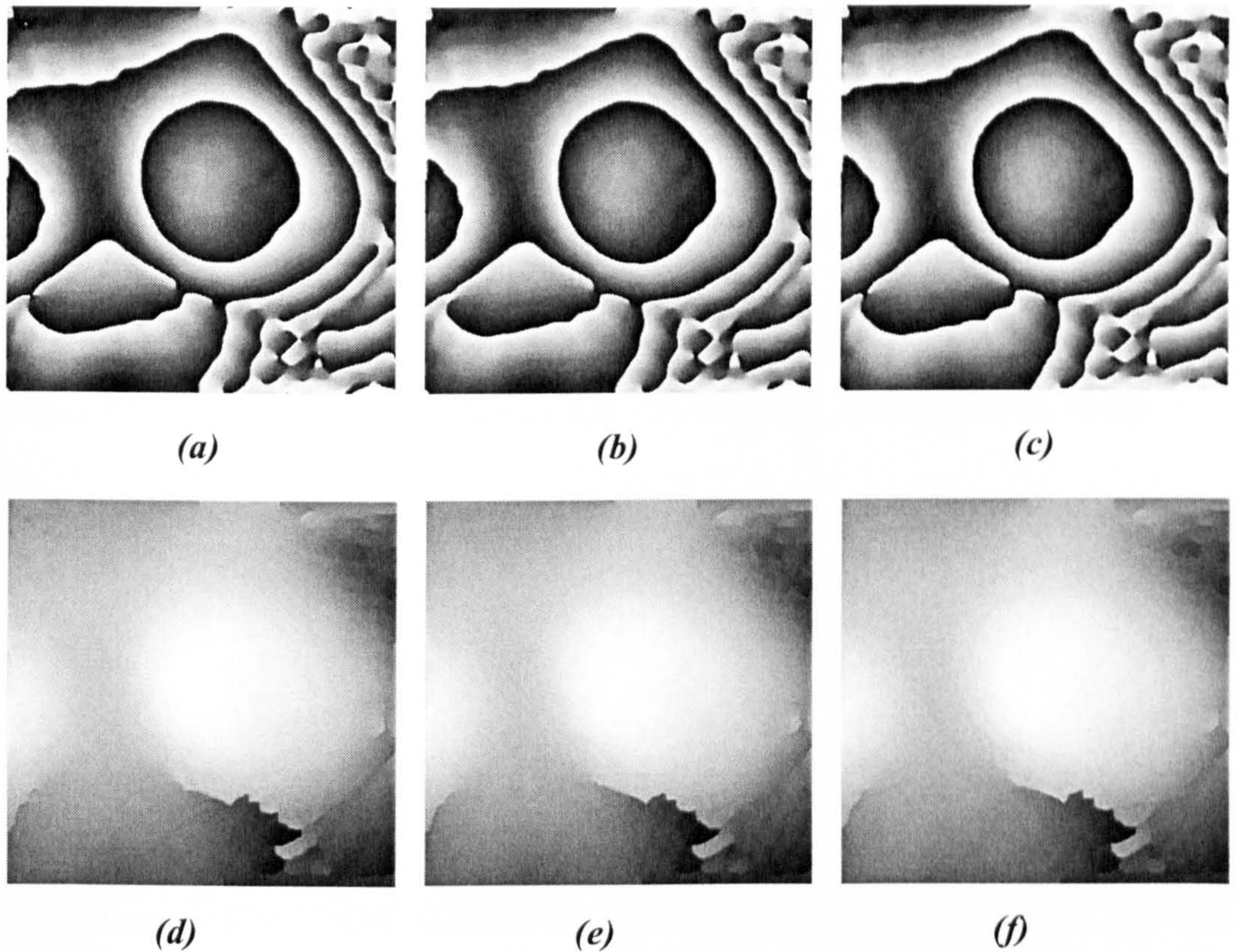


Figure 4.28: The wrapped-phase volume of a patient’s chest

Finally, the three-dimensional best-path phase unwrapping algorithm was applied to unwrap the wrapped-phase volume shown in Figure 4.28. Figure 4.29 shows three different frames of the wrapped-phase volume resulting from measuring the patient’s chest. Figures (a), (b) and (c) show the wrapped-phase maps for frames number 10, 24 and 40 respectively. Figures (d), (e) and (f) show the unwrapped-phase maps resulting from the three-dimensional best-path phase unwrapping algorithm (Abdul-Rahman *et al.*, 2005; Abdul-Rahman *et al.*, 2006).

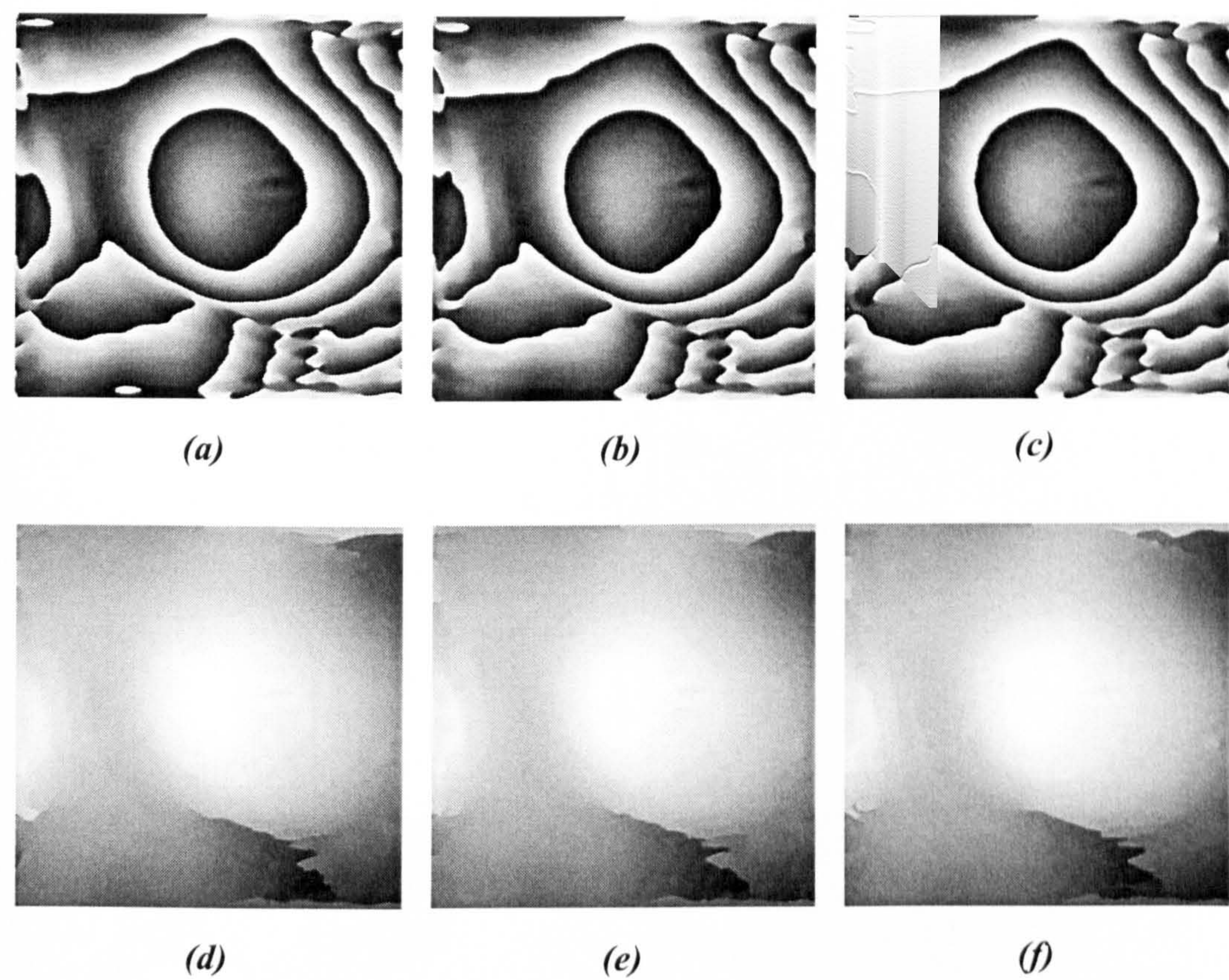


These particular frames are also processed by means of the state-of-art two-dimensional Fourier transform fringe analysis technique. Figures 4.30 shows the results of using two-dimensional Fourier transform fringe analysis whereas Figure 4.31 shows a three-dimensional comparison between the two-dimensional and three-dimensional Fourier transform fringe analysis techniques.



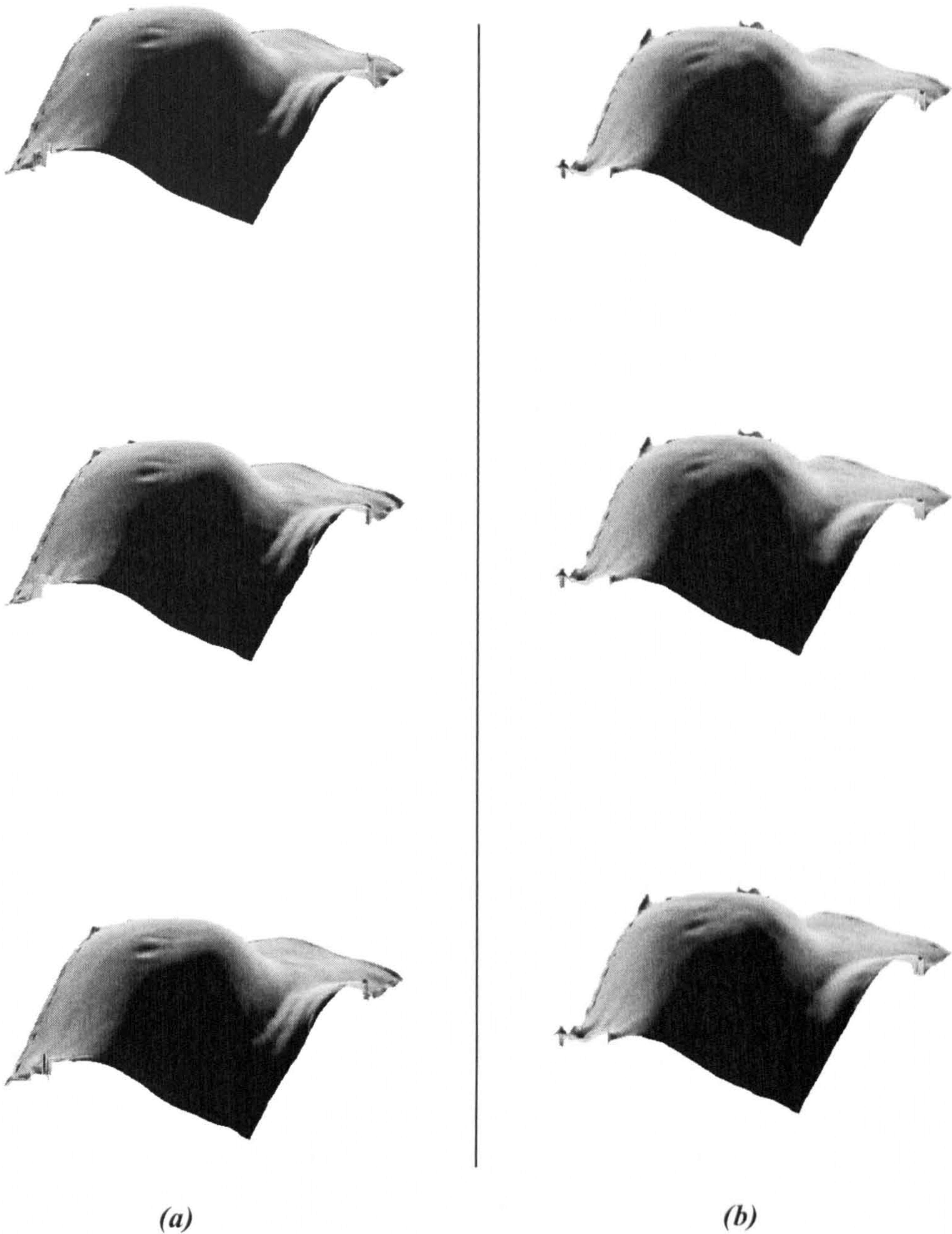
**Figure 4.29: The three-dimensional Fourier transform fringe analysis results for frames 10, 24 and 40 of a patient's chest. (a), (b) and (c) The wrapped phase maps. (d), (e) and (f) The unwrapped phase maps using the three-dimensional best-path phase unwrapping algorithm.**





**Figure 4.30: The two-dimensional Fourier transform fringe analysis results for frames 10, 24 and 40 of a patient’s chest. (a), (b) and (c) The wrapped phase maps. (d), (e) and (f) The unwrapped phase maps using the two-dimensional best-path phase unwrapping algorithm.**





**Figure 4.31: Three-dimensional comparison between two-dimensional and three-dimensional Fourier transform fringe analysis for a patient's chest. Column (a) the results of two-dimensional Fourier transform fringe analysis and, Column (b) the results of three-dimensional Fourier transform fringe analysis**



## 4.5 Summary

In this chapter, the three-dimensional Fourier fringe analysis algorithm has been proposed to demodulate fringe pattern volumes in order to measure dynamic objects. The proposed technique extends the state-of-art two-dimensional Fourier fringe analysis method into three-dimensions.

Moreover, this chapter proposes the extension of the two-dimensional Butterworth filter into three dimensions, in order to select the required frequency component. Using a Butterworth filter reduces the ringing effect on the output results.

The proposed system has been tested using both simulated and real dynamic objects. Computer simulated results show that the proposed technique can be applied to track the movement of a dynamic object at any time. Moreover, two real fringe pattern volumes for dynamic objects have also been used to evaluate the performance of the proposed algorithm. The first object was a moving dummy set up in the laboratory so as to that mimic a patient in a treatment room, this video sequence generated in the laboratory by the author, and the second object was real patient in a radiotherapy treatment room as part of a clinical trial. The video of the patient's fringe patterns was obtained from Christi Hospital in Manchester. Results show that the proposed technique is able to demodulate fringe pattern volumes successfully.

This chapter also compares the proposed 3D-FFA technique with the state-of-art 2D-FFA technique. This comparison shows that the newly proposed algorithm is already equivalent in performance to the 2D-FFA method, despite the fact that the 2D-FFA algorithm has undergone many developments and refinements since 1986.

The proposed 3D-FFA algorithm and the 2D-FFA algorithm have been implemented using the IDL 6.1 programming language. These algorithms have been executed on a PC with a Pentium 4 processor running at a 3.2 GHz clock speed. The memory of the PC is 4G-Byte RAM. The execution time of the proposed algorithm varies from volume to volume and depends on the particular phase distribution being analyzed. The execution times for simulated and real volumes discussed earlier for each algorithm are listed in Table 1.



**Table 4.1.** The execution times in minutes for the 3D-FFA and 2D-FFA for analysing the real clinical patient.

Algorithm	size	time
2D-FFA	512×512×50	1.0
3D-FFA	512×512×50	3.5

It worth mentioning that all previous fringe patterns was obtained using the fringe analysis system described previously in chapter one. This system uses a twin-fibre interferometer that operates on HeNe Laser. The camera used to capture the fringe patterns is a monochrome CCD camera which captures 25 frames in one second. More details of the specifications of the system used in all previous experiments are listed in Table 4.2.

**Table 4.2.** System specifications

Laser type	HeNe
Camera type	Monochrome CCD camera
Camera speed	25 frame/second
Field of view (cm)	50×50 (cm)
Field of view (pixels)	512×512 (pixel)
Number of fringes in the image	≈ 25 fringe/image
Fringe spacing	≈16 pixels



## References

- Abdul-Rahman, H.; Gdeisat, M.; Burton, D. and Lalor, M. (2005) Fast Three-Dimensional Phase-Unwrapping Algorithm Based on Sorting by Reliability Following a Non-Continuous Path. 1 ed. Munich, Germany, SPIE-Int. Soc. Opt. Eng. 5856 part 1, 32-40.
- Abdul-Rahman, H. S.; Gdeisat, M. A.; Burton, D. R. and Lalor, M. J. (2006) Three-Dimensional Phase Unwrapping Algorithms: A Comparison. *Photon06 conference*. Manchester, UK.
- Arevalillo Herraiez, M.; Burton, D. R. and Lalor, M. J. (1999) Accelerating Fast Fourier Transform and Filtering Operations in Fourier Fringe Analysis for Accurate Measurement of Three-Dimensional Surfaces. 2 ed. Brighton, UK, Elsevier.
- Bone, D. J.; Bachor, H. A. and Sandeman, R. J. (1986) Fringe-Pattern Analysis Using a 2-D Fourier Transform. *Applied Optics*, 25, 1653-60.
- Burton, D. R. and Lalor, M. J. (1989) Managing Some of the Problems of Fourier Fringe Analysis. Proc. SPIE Vol. 1163, p. 149. San Diego, CA, USA.
- Burton, D. R. and Lalor, M. J. (1994) Multichannel Fourier Fringe Analysis as an Aid to Automatic Phase Unwrapping. *Applied Optics*, 33, 2939-48.
- 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, 245-57.
- Gdeisat, M. A.; Burton, D. R. and Lalor, M. J. (2006) Eliminating the Zero Spectrum in Fourier Transform Profilometry Using a Two-Dimensional Continuous Wavelet Transform. *Optics Communications*, 266, 482-489.
- Gorecki, C. (1992) Interferogram Analysis Using a Fourier Transform Method for Automatic 3d Surface Measurement. *Pure and Applied Optics*, 1, 103-110.
- Herraiez, M. A.; Burton, D. R.; Lalor, M. J. and Gdeisat, M. A. (2002) Fast Two-Dimensional Phase-Unwrapping Algorithm Based on Sorting by Reliability Following a Noncontinuous Path. *Applied Optics*, 41, 7437-44.
- Li, J.; Su, X. and Gu, R. (1990) An Improved Fourier Transform Profilometry for Automatic Measurement of 3-D Object Shapes. *Optical Engineering*, 29, 1439.
- Lilley, F.; Lalor, M. and Burton, D. (2000) Robust Fringe Analysis System for Human Body Shape Measurement. *Optical Engineering*, 39, 187-195.



- Malcolm, A. A.; Burton, D. R. and Lalor, M. J. (1990) Full Field Fourier Fringe Analysis for Industrial Inspection. SPIE Proc, Vol. 1265, pp. 165-178, WA, USA.
- Stephenson, P. R. (1994) Evaluation and Solutions of Key Problems in Fourier Fringe Analysis. PhD Thesis. *General Engineering Research Institute*. Liverpool United Kingdom, Liverpool John Moores University.
- Su, X.; Chen, W.; Qc, Z. and Chao, Y. (2001) Dynamic 3-D Shape Measurement Method Based on FTP. *Optics and Lasers in Engineering*, 36, 46-64.
- Takeda, M.; Ina, H. and Kobayashi, S. (1982) Fourier-Transform Method of Fringe-Pattern Analysis for Computer-Based Topography and Interferometry. *Journal of the Optical Society of America*, 72, 156-60.



## *Chapter Five*

# *Three-Dimensional Phase Unwrapping*

## *Algorithms*



## Chapter Five

### Three-Dimensional Phase Unwrapping Algorithms

---

#### 5.1 Introduction.

Three-dimensional phase unwrapping is currently a subject of increasing interest. Many applications produce three-dimensional wrapped phase volumes, for example in the medical field with MRI imaging and in the measurement of dynamic objects in optical interferometry. These three-dimensional wrapped phase volumes could be unwrapped frame by frame using one of the two-dimensional phase unwrapping algorithms which were discussed earlier in chapter three. Unwrapping these volumes by means of two-dimensional algorithms causes a great loss of information that is encoded within the third axis.

Analysing dynamic objects using the 3D-FFA technique, which was discussed in the previous chapter, also produces three-dimensional wrapped phase volumes. These volumes need to be unwrapped by means of robust three-dimensional phase unwrapping algorithms in order to extract the unwrapped phase volumes. Subsequently the absolute heights are calculated using either an empirical or analytical phase-to-height model (Rajoub *et al.*, 2005).

In this chapter, a review of existing three-dimensional phase unwrapping algorithms is presented. Two novel three-dimensional phase unwrapping algorithms are then proposed. These two algorithms are tested upon several real dynamic objects and their performance is investigated. The results demonstrate that the proposed algorithms are very robust when compared with other three-dimensional phase unwrapping algorithms.



## 5.2 A Review of Existing Three-Dimensional Phase Unwrapping Algorithms.

Three-dimensional phase unwrapping is still a new area of research and only a few algorithms have been proposed. In 2001, Huntley proposed a three-dimensional noise immune phase unwrapping algorithm that extended the two-dimensional residue-balancing method into three dimensions (Huntley, 2001). In this method, all residues in the phase volume are identified and connected together to form singularity loops. These loops are then set as prohibited three-dimensional barrier surfaces during the unwrapping process, through which the unwrapping path must not cross. This is analogous, in a three-dimensional manner, to the use of cut lines in the two-dimensional form of phase unwrapping. Huntley claims that there is only a single solution to form these singularity loops, which means that a unique solution does exist. This is in contrast to the case for two-dimensional phase unwrapping algorithms where no unique solution exists.

Huntley shows that in some cases either two, or more, singularity loops can be intersected with each other in what he called knot points. These knot points are considered as ambiguity sources when constructing the singularity loops (Huntley, 2001). Many researchers have studied Huntley's algorithm in more detail. For example, Marklund *et al.* (Marklund *et al.*, 2005) and Salfity *et al.* (Salfity *et al.*, 2006).

Cusack *et al.* proposed another robust three-dimensional phase unwrapping algorithm that was used to unwrap MRI data (Cusack and Papadakis, 2002). This algorithm uses a quality measurement to guide the final unwrapping path. Many iterations have to be carried out in order to complete the unwrapping of the whole phase volume, as in each individual iteration only those voxels whose quality exceeds a certain threshold are unwrapped. The unwrapping of the remaining voxels is left to subsequent iterations, during which the threshold value is gradually reduced until the unwrapping process is complete.

Huntley's and Cusack's algorithms are both used to unwrap a three-dimensional wrapped-phase volumes resulted from MRI imaging. Comparing between Huntley's



and Cusack's algorithms shows that Huntley's algorithm is more robust and can be used in other applications successfully. On the hand, Cusack's algorithm does not produce good results when used in applications other than MRI. Furthermore, both algorithm unwrap wrapped phase volumes in moderate execution times. More comparison between these two algorithms is carried out during this chapter.

Jenkinson proposed another three-dimensional phase unwrapping algorithm, which tends to follow a global-error minimization approach. This technique divides the whole wrapped-phase volume into multiple regions. These regions are chosen in such a way that each region contains no phase wraps, *i.e.* the regions meet at and border the phase wraps, but each 3D region must not contain a wrap. The individual regions are treated as single units by the algorithm. A cost function that calculates the difference in the phase values at the interface of adjacent regions has to be minimized. When the cost between the two regions is at a minimum, the two regions are merged together. The process continues until only a single large region is left. This method has been designed to process two-dimensional and three-dimensional MRI data, but it can be extended to permit the unwrapping of N-dimensional data (Jenkinson, 2003).

Recently, Hasan proposed an N-dimensional phase unwrapping algorithm that utilises the Hilbert transform to detect any phase jumps (Hasan, 2006). Fang *et al.* proposed a multidimensional phase unwrapping algorithm which extends the phase unwrapping integral into higher dimensions (Fang *et al.*, 2006).

Multi-wavelength/ Temporal phase unwrapping requires a projection and acquisition of a sequence of fringe maps (Huntley and Saldner, 1993). In this phase unwrapping method, the phase at each pixel is measured as a function of time. Unwrapping is then carried out along the time axis for each pixel independently of the others. Thus boundaries and regions with poor signal-to-noise ratios do not adversely influence good data points (Huntley and Saldner, 1993).



### 5.3 Three-Dimensional Best Path (BP) Phase Unwrapping Algorithm.

In this section, the author proposes a novel three-dimensional quality-guided phase unwrapping algorithm. This algorithm relies upon a quality measure and follows a discrete unwrapping path to unwrap the highest quality voxels first and the lowest quality voxels last, in order to prevent error propagation. The proposed technique extends Arevallilo Herraiez's algorithm into three-dimensions (Herraiez *et al.*, 2002). The principle of the proposed technique has been presented at two international conferences and was compared with its two-dimensional counterpart and other three-dimensional algorithms (Abdul-Rahman *et al.*, 2005; Abdul-Rahman *et al.*, 2006). This algorithm is found to be surprisingly robust as will be demonstrated.

In a similar manner to the case for other quality-guided phase unwrapping algorithms, two main issues determine their behaviour: the choice of the quality function that will guide the unwrapping path and the design of the unwrapping path that minimizes error propagation. These two main issues are discussed and explained from the perspective of the proposed algorithm in the following subsections.

#### 5.3.1 Three-Dimensional Quality Maps

Two-dimensional quality-guided phase unwrapping algorithms use different criteria to determine the quality of each individual pixel in the phase map. The pseudo-correlation, phase derivative variance, maximum gradient and the second difference methods are the most well-known two-dimensional quality maps and they have been integrated into many phase unwrapping algorithms and produce very robust results, as was explained in chapter three.

The previous quality maps are extended here into three dimensions to suit the proposed algorithm. The method of calculation for each quality map in the three-dimensional case is explained next.

##### 5.3.1.1 Three-Dimensional Pseudo-Correlation Quality Map

In order to calculate the three-dimensional pseudo-correlation value for a voxel,  $v(m,n,l)$ , in a phase volume, using a  $3 \times 3 \times 3$  cube, the summation of cosines and sines of



the values of the voxel and its 26 neighbours (consisting of 9 in the plane below, 8 in the plane itself and 9 in the plane above) are calculated by using the equation:

$$q(m, n, l) = \frac{\sqrt{[\sum \cos(\psi(i, j, k))]^2 + [\sum \sin(\psi(i, j, k))]^2}}{27} \quad (5.1)$$

where;

$\psi(m, n, l)$ : is the wrapped phase value of the voxel  $v(m, n, l)$ .

$i, j, k$  represent the neighbours' indices of the voxel  $v(m, n, l)$  in the  $3 \times 3 \times 3$  cube.

### 5.3.1.2 Three-Dimensional Phase Derivative Variance Quality Map

The phase derivative variance value measures the statistical variance of the phase derivative. Actually, phase derivative variance indicates the badness rather than the goodness of the phase data.

The calculation of the phase derivative variance value for a voxel  $v(m, n, l)$  in the phase volume, using a  $3 \times 3 \times 3$  cube, is defined by the equation:

$$PV(m, n, l) = \frac{1}{27} * \left( \sqrt{\sum (\hat{\Delta}^x \psi(i, j, k) - \overline{\hat{\Delta}^x \psi(i, j, k)})^2} + \sqrt{\sum (\hat{\Delta}^y \psi(i, j, k) - \overline{\hat{\Delta}^y \psi(i, j, k)})^2} + \sqrt{\sum (\hat{\Delta}^z \psi(i, j, k) - \overline{\hat{\Delta}^z \psi(i, j, k)})^2} \right) \quad (5.2)$$

where;

$\hat{\Delta}^x \psi(i, j, k)$ ,  $\hat{\Delta}^y \psi(i, j, k)$  and  $\hat{\Delta}^z \psi(i, j, k)$  are the wrapped phase gradients in the  $x$ ,  $y$  and  $z$  directions respectively.

$\overline{\hat{\Delta}^x \psi(i, j, k)}$ ,  $\overline{\hat{\Delta}^y \psi(i, j, k)}$  and  $\overline{\hat{\Delta}^z \psi(i, j, k)}$  are the mean of the values in a  $3 \times 3 \times 3$  cube in  $\hat{\Delta}^x \psi(i, j, k)$ ,  $\hat{\Delta}^y \psi(i, j, k)$  and  $\hat{\Delta}^z \psi(i, j, k)$  respectively.

$i, j, k$  are the neighbours' indices of the voxel  $v(m, n, l)$  in a  $3 \times 3 \times 3$  cube.

$\hat{\Delta}^x \psi(i, j, k)$ ,  $\hat{\Delta}^y \psi(i, j, k)$  and  $\hat{\Delta}^z \psi(i, j, k)$  are defined by:



$$\begin{aligned}
\hat{\Delta}^x \psi(i, j, k) &= w\{\psi(i+1, j, k) - \psi(i, j, k)\} \\
\hat{\Delta}^y \psi(i, j, k) &= w\{\psi(i, j+1, k) - \psi(i, j, k)\} \\
\hat{\Delta}^z \psi(i, j, k) &= w\{\psi(i, j, k+1) - \psi(i, j, k)\}
\end{aligned} \tag{5.3}$$

Where  $w$  defines a wrapping operator that will wrap all values of its argument into the range  $[-\pi, \pi]$  by adding or subtracting an integer number of  $2\pi$  radians to its argument, *i.e.*  $y = w\{x\}$  is equivalent to  $y = \arctan(\sin(x)/\cos(x))$ .

Finally, the quality of the voxel is defined to be the reciprocal of the variance, as described by the equation.

$$q(m, n, l) = \frac{1}{PV(m, n, l)} \tag{5.4}$$

#### 5.3.1.3 Three-Dimensional Maximum Gradient Quality Map

The maximum phase gradient method measures the magnitude of the largest phase gradient, *i.e.* wrapped phase difference. The maximum gradient of a voxel  $v(m, n, l)$  can be calculated by finding the maximum gradient of the voxel and its 26 neighbours in the phase map in the  $x$ ,  $y$  or  $z$  directions, as described in Equation (5). Similar to the case for the phase derivative variance quality map, the maximum gradient indicates the badness rather than the goodness of the phase data, so the quality is calculated using the reciprocal of the gradient.

$$MG(m, n, l) = \max \left\{ \begin{array}{l} \max \left\{ |\hat{\Delta}^x \psi(i, j, k)| \right\} \\ \max \left\{ |\hat{\Delta}^y \psi(i, j, k)| \right\} \\ \max \left\{ |\hat{\Delta}^z \psi(i, j, k)| \right\} \end{array} \right\} \tag{5.5}$$

#### 5.3.1.4 Three-Dimensional Second Difference Quality Map

To calculate the quality of a voxel using the second difference algorithm, the second differences between the wrapped phase voxel and its neighbours are calculated using the equation:



$$SD(m, n, l) = \sqrt{H^2(m, n, l) + V^2(m, n, l) + N^2(m, n, l) + \sum_{n=1}^{10} D_n^2(m, n, l)} \quad (5.6)$$

where,

$$\begin{aligned} H(m, n, l) &= w\{\psi(m-1, n, l) - \psi(m, n, l)\} - w\{\psi(m, n, l) - \psi(m+1, n, l)\} \\ V(m, n, l) &= w\{\psi(m, n-1, l) - \psi(m, n, l)\} - w\{\psi(m, n, l) - \psi(m, n+1, l)\} \\ N(m, n, l) &= w\{\psi(m, n, l-1) - \psi(m, n, l)\} - w\{\psi(m, n, l) - \psi(m, n, l+1)\} \\ D_1(m, n, l) &= w\{\psi(m-1, n-1, l) - \psi(m, n, l)\} - w\{\psi(m, n, l) - \psi(m+1, n+1, l)\} \\ D_2(m, n, l) &= w\{\psi(m+1, n-1, l) - \psi(m, n, l)\} - w\{\psi(m, n, l) - \psi(m-1, n+1, l)\} \\ D_3(m, n, l) &= w\{\psi(m-1, n-1, l-1) - \psi(m, n, l)\} - w\{\psi(m, n, l) - \psi(m-1, n+1, l+1)\} \\ D_4(m, n, l) &= w\{\psi(m, n-1, l-1) - \psi(m, n, l)\} - w\{\psi(m, n, l) - \psi(m, n+1, l+1)\} \\ D_5(m, n, l) &= w\{\psi(m+1, n-1, l-1) - \psi(m, n, l)\} - w\{\psi(m, n, l) - \psi(m-1, n+1, l+1)\} \\ D_6(m, n, l) &= w\{\psi(m-1, n, l-1) - \psi(m, n, l)\} - w\{\psi(m, n, l) - \psi(m+1, n, l+1)\} \\ D_7(m, n, l) &= w\{\psi(m-1, n+1, l-1) - \psi(m, n, l)\} - w\{\psi(m, n, l) - \psi(m+1, n-1, l+1)\} \\ D_8(m, n, l) &= w\{\psi(m+1, n, l-1) - \psi(m, n, l)\} - w\{\psi(m, n, l) - \psi(m-1, n, l+1)\} \\ D_9(m, n, l) &= w\{\psi(m, n+1, l-1) - \psi(m, n, l)\} - w\{\psi(m, n, l) - \psi(m, n-1, l+1)\} \\ D_{10}(m, n, l) &= w\{\psi(m+1, n+1, l-1) - \psi(m, n, l)\} - w\{\psi(m, n, l) - \psi(m-1, n-1, l+1)\} \end{aligned} \quad (5.7)$$

H, V and N are the horizontal, vertical and normal second differences respectively.  $D_n$  is the  $n^{\text{th}}$  diagonal second difference. The second difference also measures the badness of each voxel, so the quality of each voxel is the reciprocal of the second difference value.

The second difference quality map can be calculated without using the diagonal differences, this reduces the computational complexity but it may reduce the robustness of the algorithm.

### 5.3.2 Unwrapping Path

Most quality-guided phase unwrapping algorithms rely on the quality of the voxels in the wrapped phase volume in order to guide the unwrapping path. The unwrapping path of the proposed algorithm relies on the quality of the edges as an intermediate stage, rather than relying on the voxels' quality.



An edge is an intersection of two voxels that are connected in the  $x$ ,  $y$  or  $z$  directions. Edges can be classified as horizontal, vertical, and normal edges, or they may alternatively be called  $x$ -typed,  $y$ -type and  $z$ -type edges, as shown in Figure 5.1. In this figure the voxels are represented by circles and the edges are represented by squares. The quality of an edge is defined as the summation of the qualities of the two voxels that the edge connects, as shown in Figure 5.1, where the numbers inside the circles represent voxels' qualities, whereas the numbers inside the squares represent edges' qualities. The proposed algorithm sets the quality value of an edge that connects a border voxel with another voxel in the phase volume to zero.

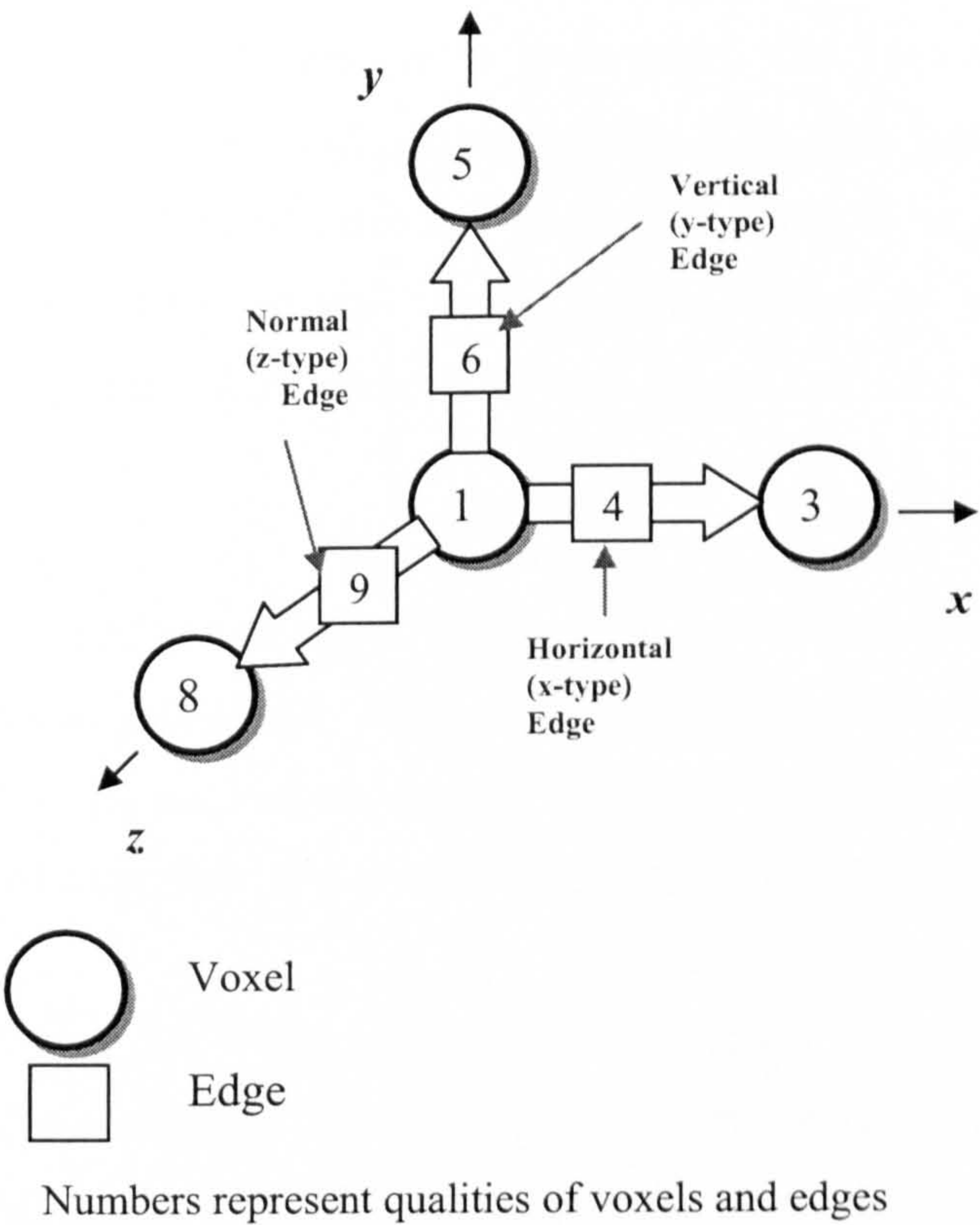


Figure 5.1: Definition of the edges and their qualities.

In the proposed algorithm the unwrapping path is determined by the qualities of the edges. The definition of the unwrapping path is relatively simple. All the edges are stored in an array and sorted by the value of their qualities. Those edges with higher qualities are resolved first, as will be explained below. The construction of the unwrapping path is similar to the discrete unwrapping path described in 2D by



Arevallilo Herraiez *et al.* (Herraiez *et al.*, 2002) but is now extended into three dimensions.

Figures 5.2(a) – 5.2(f) illustrate the principle of the proposed algorithm. The volume shown in Figure 5.2(a) is a wrapped phase volume that needs to be unwrapped. In this figure, the edges' qualities are already sorted and their order is shown in Fig 5.2(a). The integer numbers represent the order of edges in the sorted array. The edge that has the order 1 is processed first, then the edge with order 2, and so on.

Initially all voxels are considered to not belong to any group, which is represented by the white circles in Figure 5.2(a). Both voxels forming the edge with order 1 in Figure 5.2(a) are unwrapped first with respect to each other, and both voxels are joined together to form a single group, as illustrated in Fig 5.2(b). After processing this edge, the algorithm searches for the next edge to be processed which is the edge with order 2 shown in Figure 5.2(b). Because both voxels forming edge 2 do not belong to any group, they are unwrapped with respect to each other and form a new group as illustrated in Figure 5.2(c).

The voxels forming the edge 3 in Figure 5.2(c) are to be processed in this stage. As we can see, one of these voxels already belongs to a group, but the other voxel does not belong to any group. The ungrouped voxel is unwrapped with respect to the grouped voxel, and is set to be a member of that existing group, as shown in Figure 5.2(d).

Edge number 4 in Figure 5.2(d) connects two grouped voxels, so in this case the group of fewer members is unwrapped with respect to the group with more members, by adding or subtracting multiple integers of  $2\pi$  to each voxel in the group. Then both groups merge into a single group, as illustrated in Figure 5.2(e).

The algorithm continues this process of forming new groups, joining ungrouped voxels to other existing groups and merging groups together, until all of the voxels have been unwrapped and become members of a single large group. Figure 5.2(f) shows the situation after several more edges have been processed. Here edge 5 has been processed and added to the existing large group. Also two separate new groups have been formed in processing edges 6 and 7 as the unwrapping process has been carried out. In the next



stage which is not shown, these two new groups are about to be joined, which will occur when edge 8 is processed. The unwrapping process will continue in this manner until complete. Note that at each stage the unwrapping of a voxel, or a group of voxels, with respect to another group may require the addition or subtraction of multiples of  $2\pi$ .

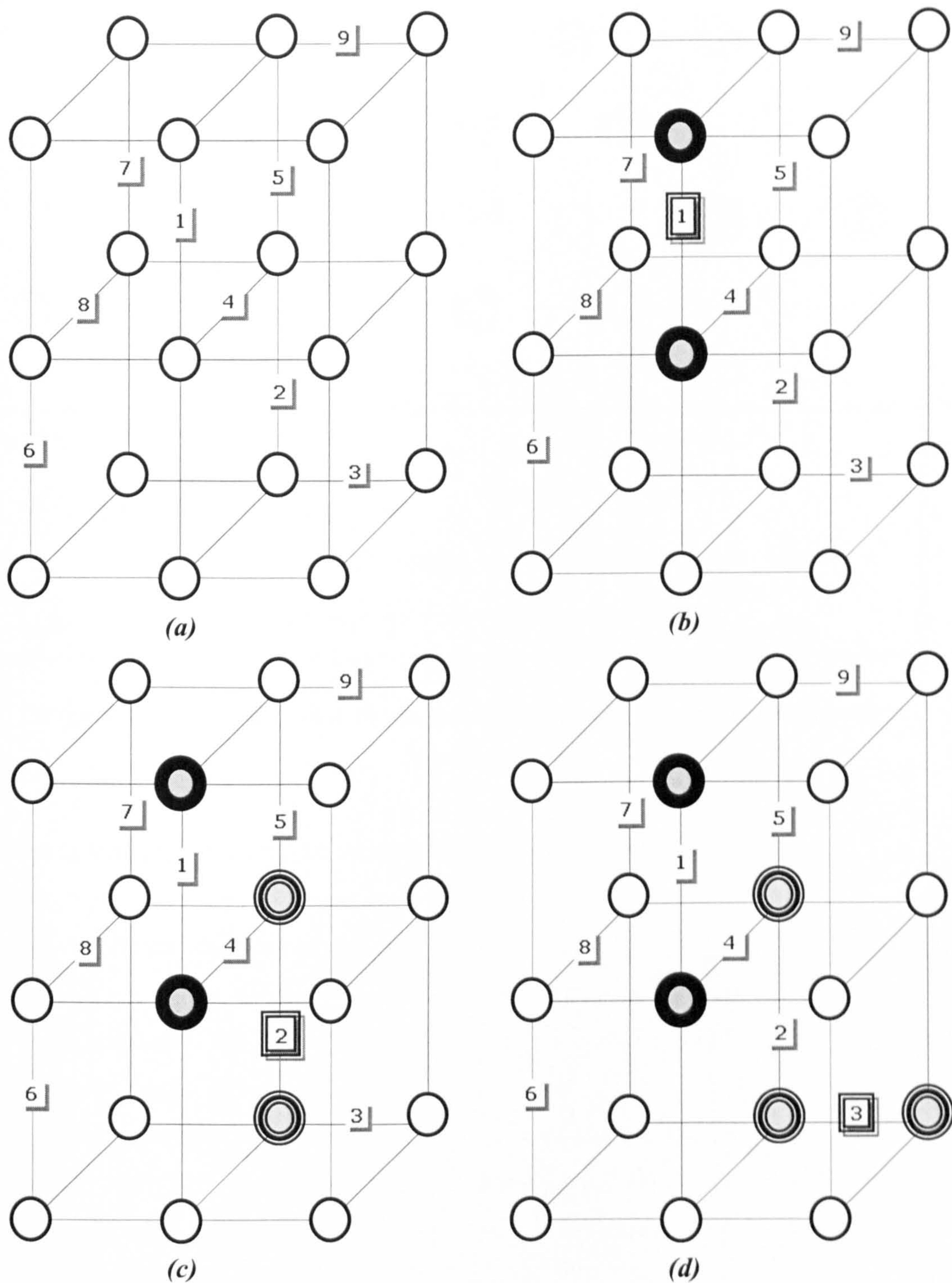


Figure 5.2: A demonstration for the unwrapping path of the proposed algorithm.



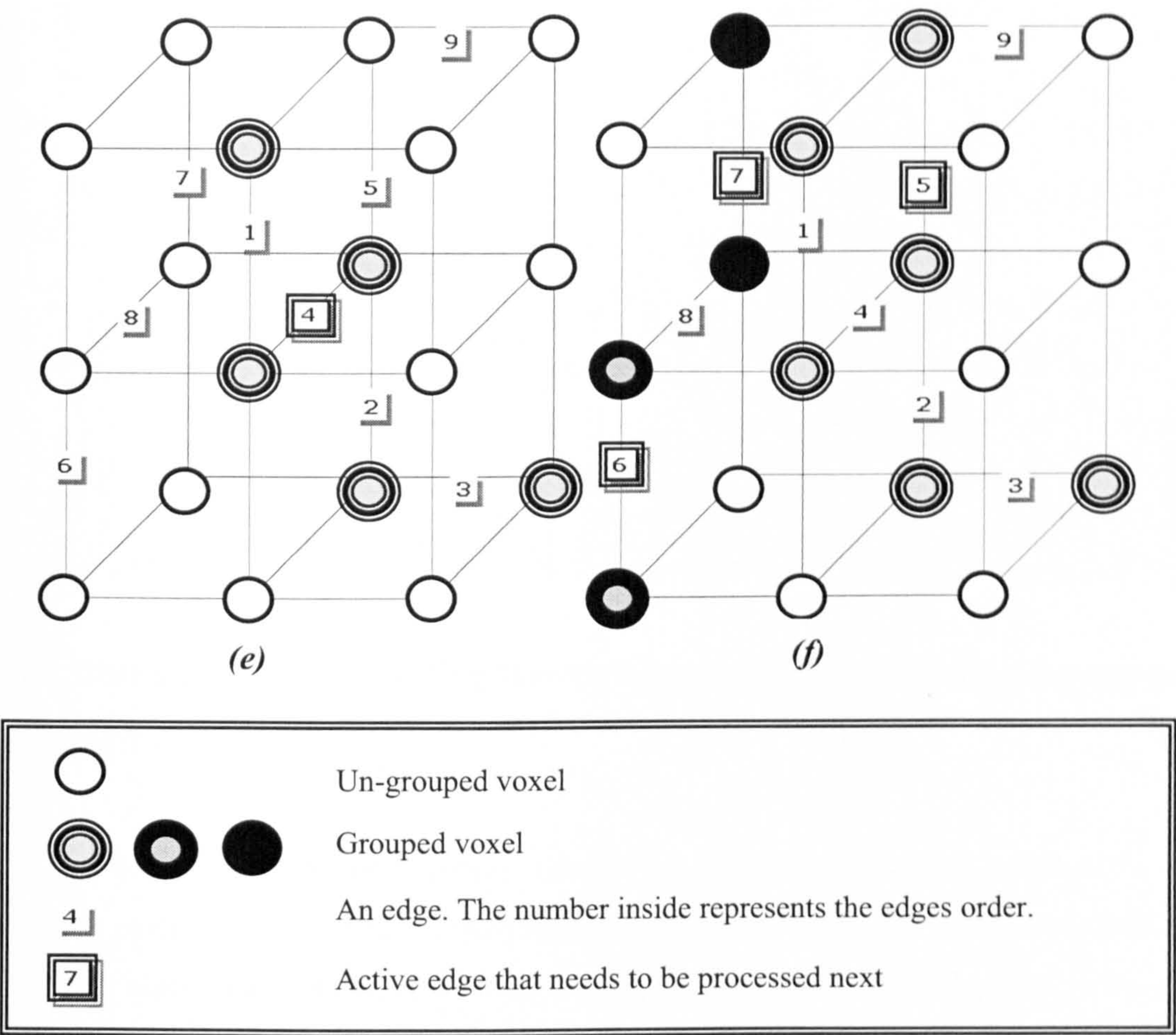


Figure 5.2: A demonstration for the unwrapping path of the proposed algorithm  
(Continued).

The proposed algorithm can be summarized in the following steps:

1. Determine the qualities of all voxels.
2. Calculate the horizontal, vertical and normal edges' qualities and set the qualities of the edges that are connected with the borders to zero so that they will be processed last.
3. Sort all of the edges according to their qualities in a descending order.
4. Unwrap voxels according to the edges' qualities; so that the voxels which form the highest quality edges are unwrapped first according to the following rules:



(a) If both voxels do not belong to any group and have not been unwrapped before; the voxels are unwrapped with respect to each other and are gathered into a single group.

(b) If one of the voxels has been processed before, and belongs to a group, but the other has not, then the voxel that has not been processed before is unwrapped with respect to the other voxels in the group, and it now joins this group.

(c) If both voxels have been processed before and both belong to different groups, then the two groups are unwrapped with respect to each other. The smaller group is unwrapped with respect to the larger group. Then the two groups are joined together to construct a single group. If the two groups are the same size, then one of these groups is arbitrarily unwrapped with respect to the other group.

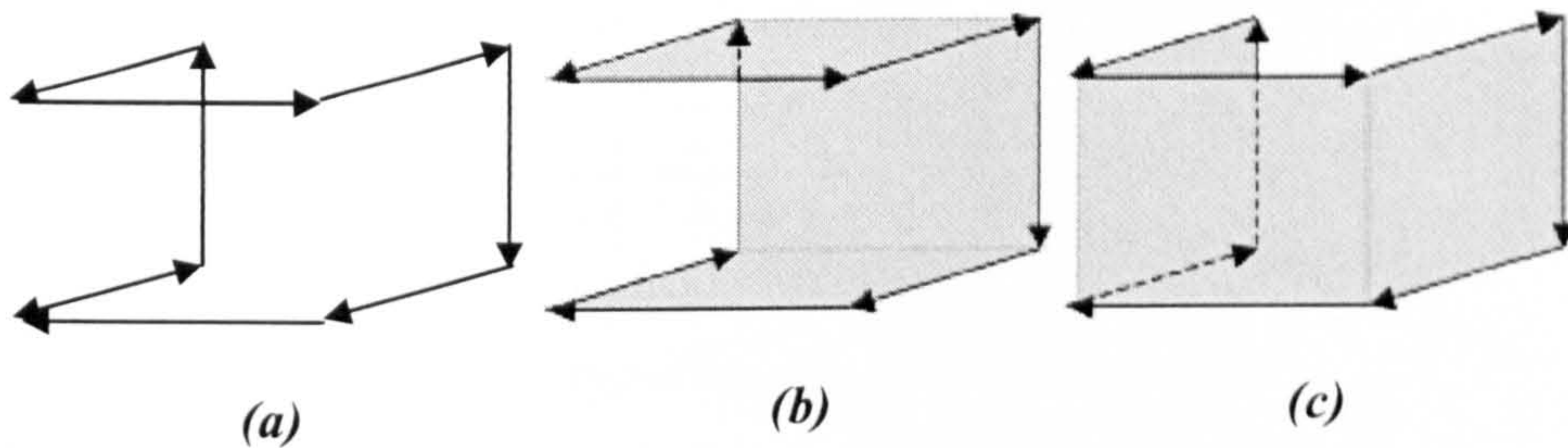
## 5.4 Best Path Avoiding Singularity Loops (BPASL) Phase Unwrapping Algorithm.

In this section, the author proposes another novel phase unwrapping algorithm. This will be referred to as the best path avoiding singularity loops (BPASL) algorithm. The BPASL algorithm is a very powerful technique for unwrapping 3D phase volumes. It combines the power of the three-dimensional noise-immune algorithm that was proposed by Huntley (Huntley, 2001) with the author's three-dimensional best path algorithm that was described in the previous section.

The three-dimensional noise-immune algorithm attempts to create a path-independent phase volume by identifying all singularity loops. Once singularity loops are identified, the unwrapping path is considered to be path-independent if these loops are not subsequently penetrated by the unwrapping path. In reality, it is almost impossible to achieve a completely path independent-phase volume, even after identifying all singularity loops. One problem that violates the principal of producing a truly path-independent volume is that of loop ambiguity. These ambiguities arise when some loops can be covered by two different surfaces, one of them is correct, whereas the other is wrong. An example of this case is the C-shaped loop ambiguity which is shown in Figure 5.3. Figure 5.3 (a) shows a singularity loop and Figures 5.3 (b) and (c) show that two different surfaces can be obtained from the same singularity loop. One of these



surfaces is correct and must not be penetrated by the unwrapping path in order to achieve a path-independent phase volume, whereas the other surface is incorrect. Obviously if the incorrect surface is chosen, then bad data regions are still left unmarked and it is still possible for the unwrapping path to pass through this erroneous phase data and propagate errors. Singularity loop ambiguities are discussed in detail by Salfity *et al.* in 2006 (Salfity *et al.*, 2006). So relying completely upon finding and avoiding singularity loops may still lead to unexpected errors.



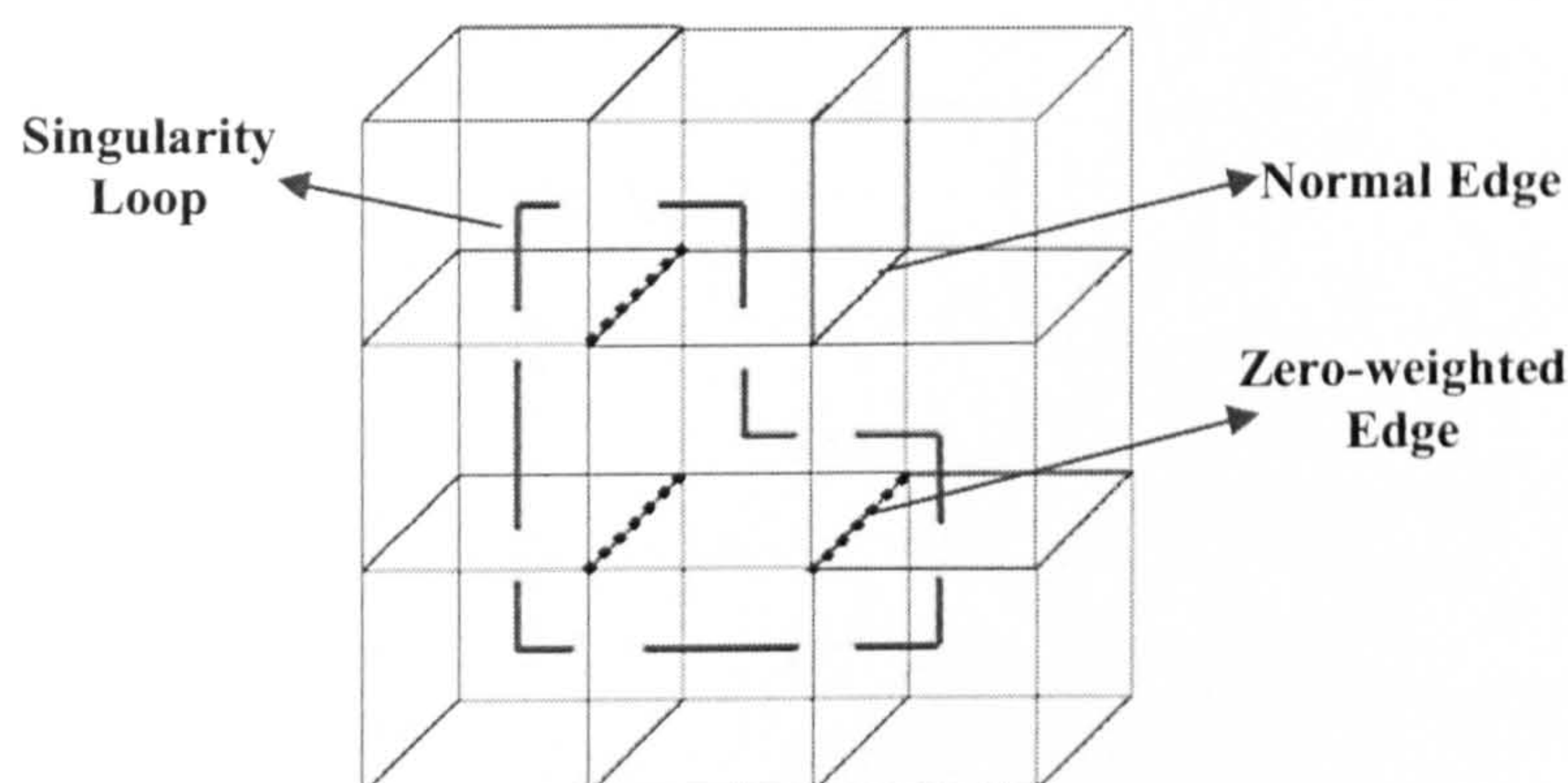
**Figure 5.3: Singularity loop ambiguity resulting from a C-shaped loop (Salfity *et al.*, 2006)**

On the other hand, the best path phase unwrapping algorithms do not identify singularity loops at all. They rely completely upon a quality measure to unwrap the phase volume. Ignoring singularity loops may cause the unwrapping path to penetrate these loops and errors may propagate in the unwrapped phase map.

Therefore the BPASL algorithm takes advantages from each of both of the previous techniques. Firstly, it identifies singularity loops and attempts to prohibit the unwrapping path from passing through them in order to prevent error propagation. It also calculates the quality of each individual voxel in order to ensure the unwrapping of the highest quality voxels first and the worst quality voxels last. The BPASL algorithm is a hybrid between the best path algorithm described earlier in this chapter (Abdul-Rahman *et al.*, 2005) and the three-dimensional noise-immune phase unwrapping algorithm first proposed by Huntley in 2001 (Huntley, 2001) and then explained in greater detail by Marklund *et al.* in 2005 (Marklund *et al.*, 2005).



The key difference between this algorithm and the best path algorithm is the introduction of the concept of zero-weighted edges. Zero-weighted edges are those edges that pass through a singularity loop, as shown in Figure 5.4.



**Figure 5.4: Definition of zero-weighted edges.**

The BPASL algorithm can be outlined in the following steps:

- 1- Identify zero-weighted edges.**
- 2- Calculate all edges' qualities and set the quality of all zero-weighted edges to zero.**
- 3- Sorting the edges according to their qualities.**
- 4- Unwrap the phase volume according to the BP rules described in the previous section.**

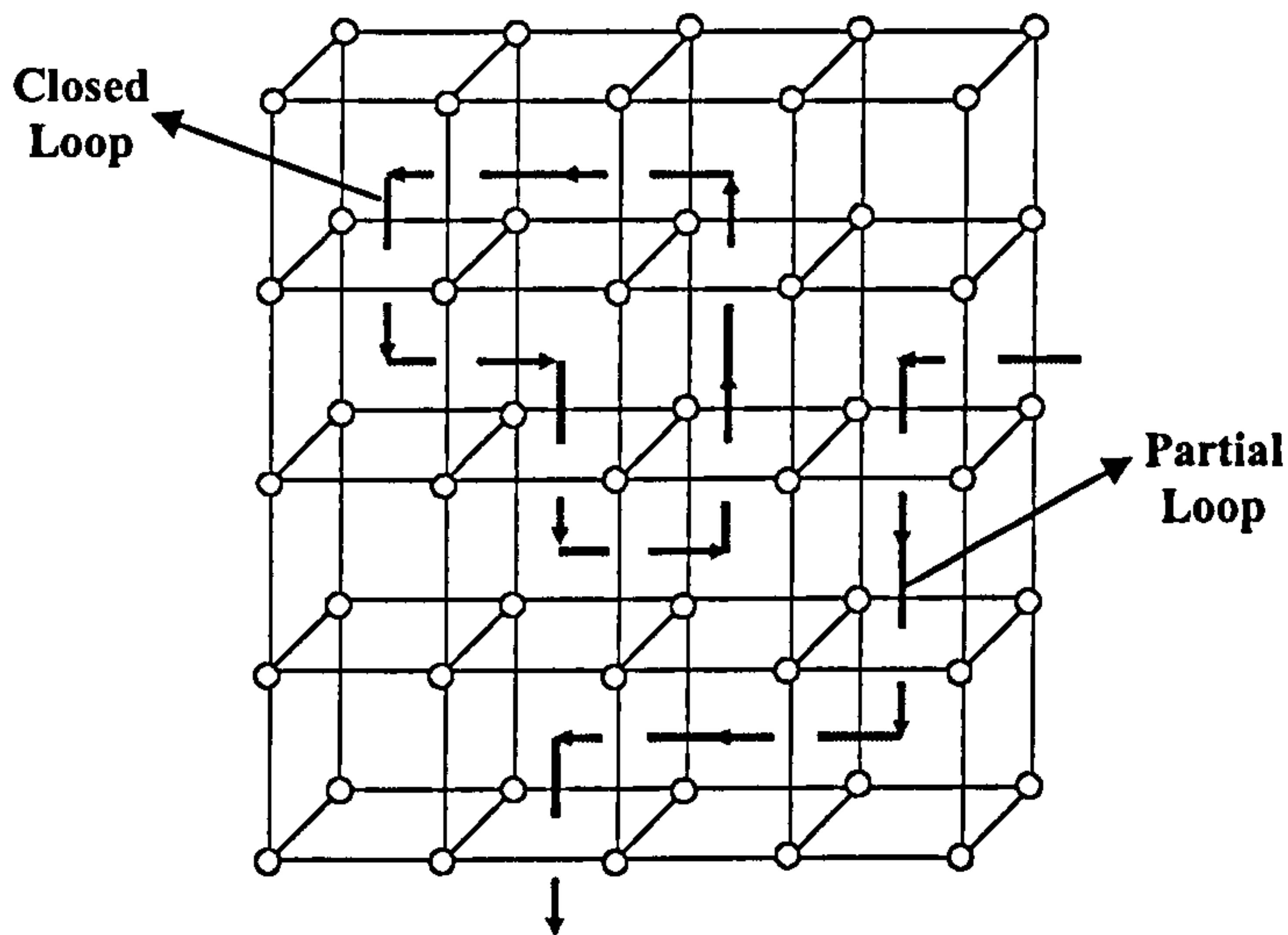
Identifying zero-weighted edges requires identifying all singularity loops in the phase volume, which is described by Marklund *et al.* (Marklund *et al.*, 2005), and will be explained in detail in the following sections.

#### **5.4.1 Identifying Singularity Loops.**

A singularity loop (SL) is a group of residues that can be connected together to form either a closed loop or a partial loop which terminates at the boundary of the phase volume as shown in the Figure 5.5. The process of identifying SLs is achieved by carrying out the following steps:



- 1- Identifying all residues located in the phase volume in the  $x$ ,  $y$  and  $z$  directions.
- 2- Sorting residues into loops by following certain rules that will be explained shortly.



**Figure 5.5: Definition of the closed loops and partial loops in the phase volume.**

#### 5.4.1.1 Identifying Residues in the Phase Volume.

Three different types of residues are introduced in the case of a three-dimensional wrapped phase volume. In the case of two-dimensional wrapped phase maps, residues are identified by calculating the phase differences in a  $2 \times 2$  loop as was described in chapter 3. Similarly, residues in 3D volumes are identified by calculating phase differences in a  $2 \times 2$  loop. Identifying  $r_x$  residues requires calculating the phase difference in a loop located in the  $yz$  plane. In the case of  $r_y$  residues, the loop must be located in the  $zx$  plane and for  $r_z$  residues the loop must be in the  $xy$  plane (this is the only type of residue found in the 2D case).



Identifying  $r_x$ ,  $r_y$  and  $r_z$  residues is carried out using the following equations:

$$r_x = \left. \begin{aligned} & \Re\left[\frac{\psi_{i,j,k} - \psi_{i,j+1,k}}{2\pi}\right] + \Re\left[\frac{\psi_{i,j+1,k} - \psi_{i,j+1,k+1}}{2\pi}\right] + \Re\left[\frac{\psi_{i,j+1,k+1} - \psi_{i,j,k+1}}{2\pi}\right] \\ & + \Re\left[\frac{\psi_{i,j,k+1} - \psi_{i,j,k}}{2\pi}\right] \end{aligned} \right\} \quad (5.8)$$

$$r_y = \left. \begin{aligned} & \Re\left[\frac{\psi_{i,j,k} - \psi_{i,j,k+1}}{2\pi}\right] + \Re\left[\frac{\psi_{i,j,k+1} - \psi_{i+1,j,k+1}}{2\pi}\right] + \Re\left[\frac{\psi_{i+1,j,k+1} - \psi_{i+1,j,k}}{2\pi}\right] \\ & + \Re\left[\frac{\psi_{i+1,j,k} - \psi_{i,j,k}}{2\pi}\right] \end{aligned} \right\} \quad (5.9)$$

$$r_z = \left. \begin{aligned} & \Re\left[\frac{\psi_{i,j,k} - \psi_{i+1,j,k}}{2\pi}\right] + \Re\left[\frac{\psi_{i+1,j,k} - \psi_{i+1,j+1,k}}{2\pi}\right] + \Re\left[\frac{\psi_{i+1,j+1,k} - \psi_{i,j+1,k}}{2\pi}\right] \\ & + \Re\left[\frac{\psi_{i,j+1,k} - \psi_{i,j,k}}{2\pi}\right] \end{aligned} \right\} \quad (5.10)$$

Where, the operator  $\Re[ ]$  rounds its argument to the nearest integer.

In these equations,  $r_x$ ,  $r_y$  and  $r_z$  can only take the values of 0 (no residue), +1 (a positive residue) or -1 (a negative residue). Figure 5.6 shows the direction of positive and negative residues of the three types. The thick arrow in the figure represents the direction of the residues and the four small arrows indicate the direction needed to be followed to identify the existence of the residues.



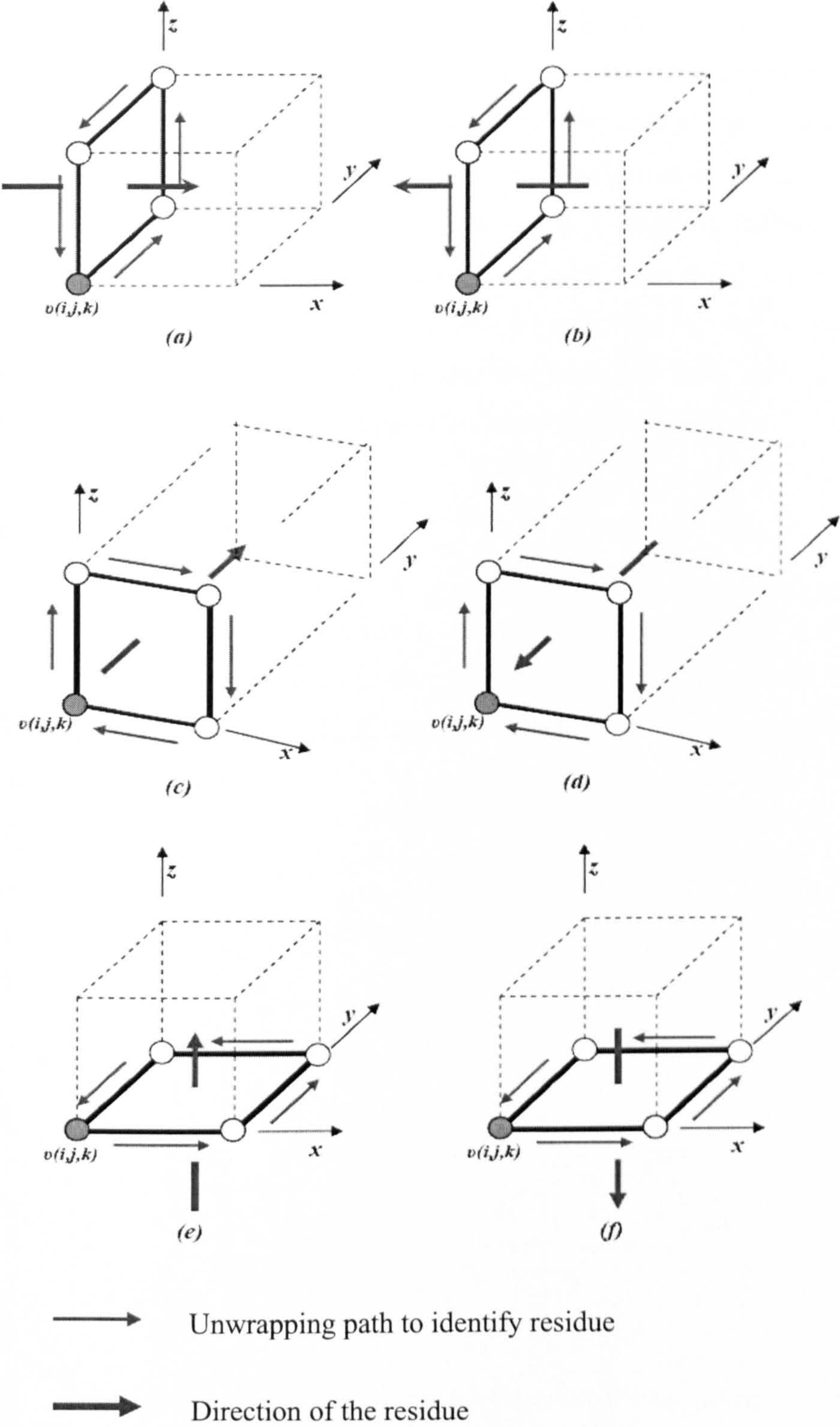


Figure 5.6: Possible residue types and directions in a three-dimensional wrapped phase volume. (a) Positive x-type residue, (b) Negative x-type residue, (c) Positive y-type residue, (d) Negative y-type residue, (e) Positive z-type residue and (f) Negative z-type residue.

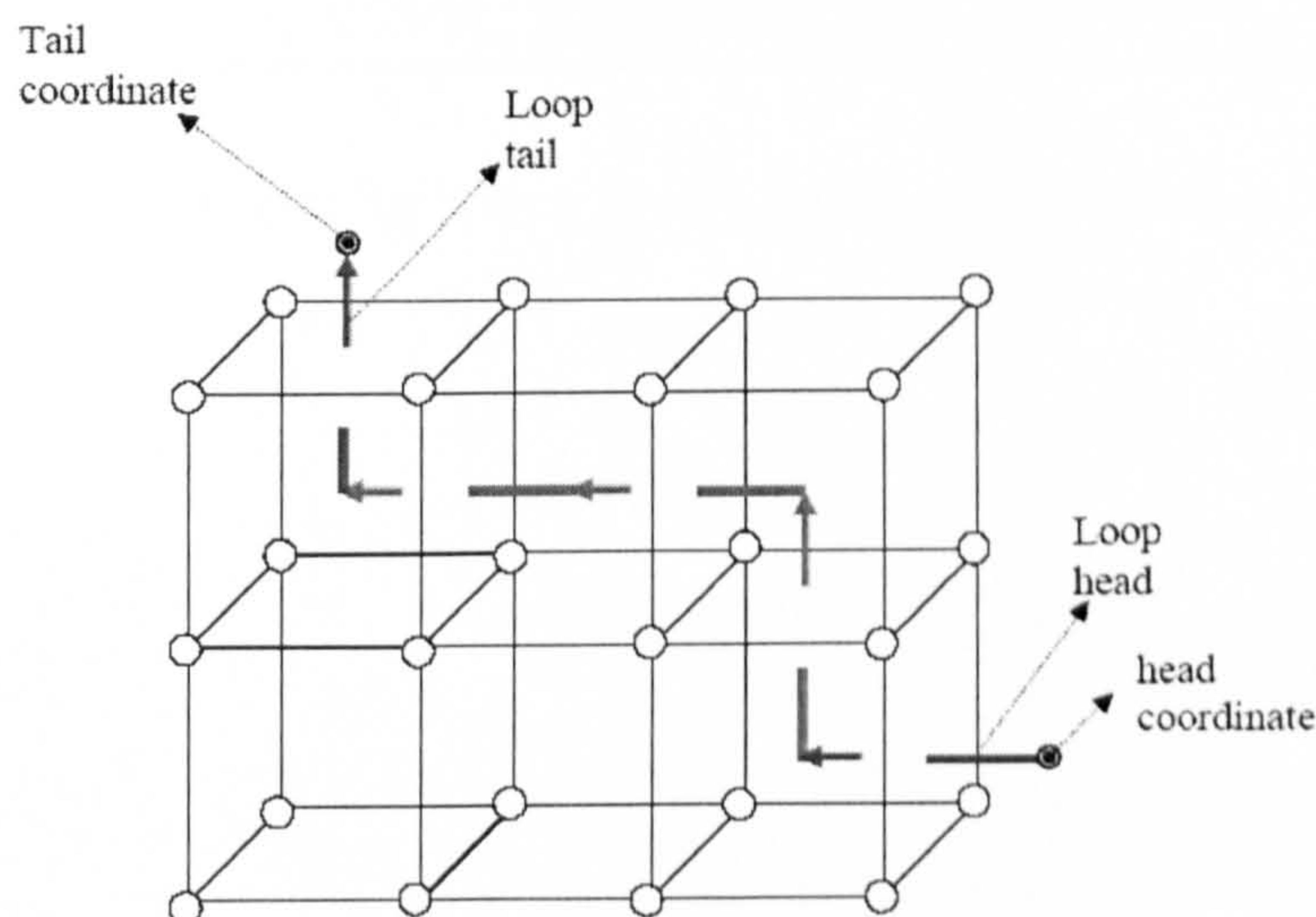


#### 5.4.1.2 Sorting Residues into Closed or Partial Loops.

Identifying singularity loops in the phase data volume depends on two facts outlined and discussed by Huntley (Huntley, 2001). The first fact is that all residues must form a closed loop in the phase volume space, or if not must terminate at the volume border. The second fact is that the number of residues entering any cube in the volume space must be equal to the number of residues leaving that cube. Depending on these facts, the process of identifying the singularity loops is a kind of searching algorithm that connects appropriate residues with each other, where any residue that enters any cube in the phase volume is connected to another residue that leaves the same cube. A detailed sorting algorithm is explained by Marklund *et al.* (Marklund *et al.*, 2005)

#### 5.4.2 Closing Partial Loops.

When a loop terminates on the boundary of the phase volume, it forms a partial, or unclosed, loop and therefore it needs to be closed before the procedure of identifying the zero-weighted edges can be applied. Let us assume that we want to close the partial loop that is shown in Figure 5.7.



**Figure 5.7: Example of a partial loop that need to be closed.**



The closing procedure can be summarized in the following steps:

- 1- Randomly, choose one end of the loop to be the head and the other end to be the tail, as shown in Figure 5.7.
- 2- Calculate the coordinates of the head and the tail which are shown in Figure 5.7.
- 3- Identify the five artificial residues that can potentially be connected to the tail residue in three dimensions. For clarity in our example, Figure 5.8(a) only shows three out of the 5 possible artificial residues that can be connected to the tail. The other two potential residue paths (those leading into and out of the plane of the page) have been left out here as they would obscure the diagram.
- 4- Calculate the coordinates of the five residues which were identified in step 3.
- 5- Calculate the distance between each artificial residue and the loop head.
- 6- Choose the artificial residue with the minimum distance to be connected to the loop. If two residues have the same distance, then one of these residues is arbitrarily connected to the loop. In our example one of the artificial residues is connected to the loop as shown in Figure 5.8(b).
- 7- Check if this artificial residue closes the loop. If yes, then the loop is closed and this procedure is ended. Otherwise continue with step 8
- 8- Mark this artificial residue as a new loop tail and go back to step 2.

This process is continued until the loop is finally closed, as shown in Figure 5.8 (c).



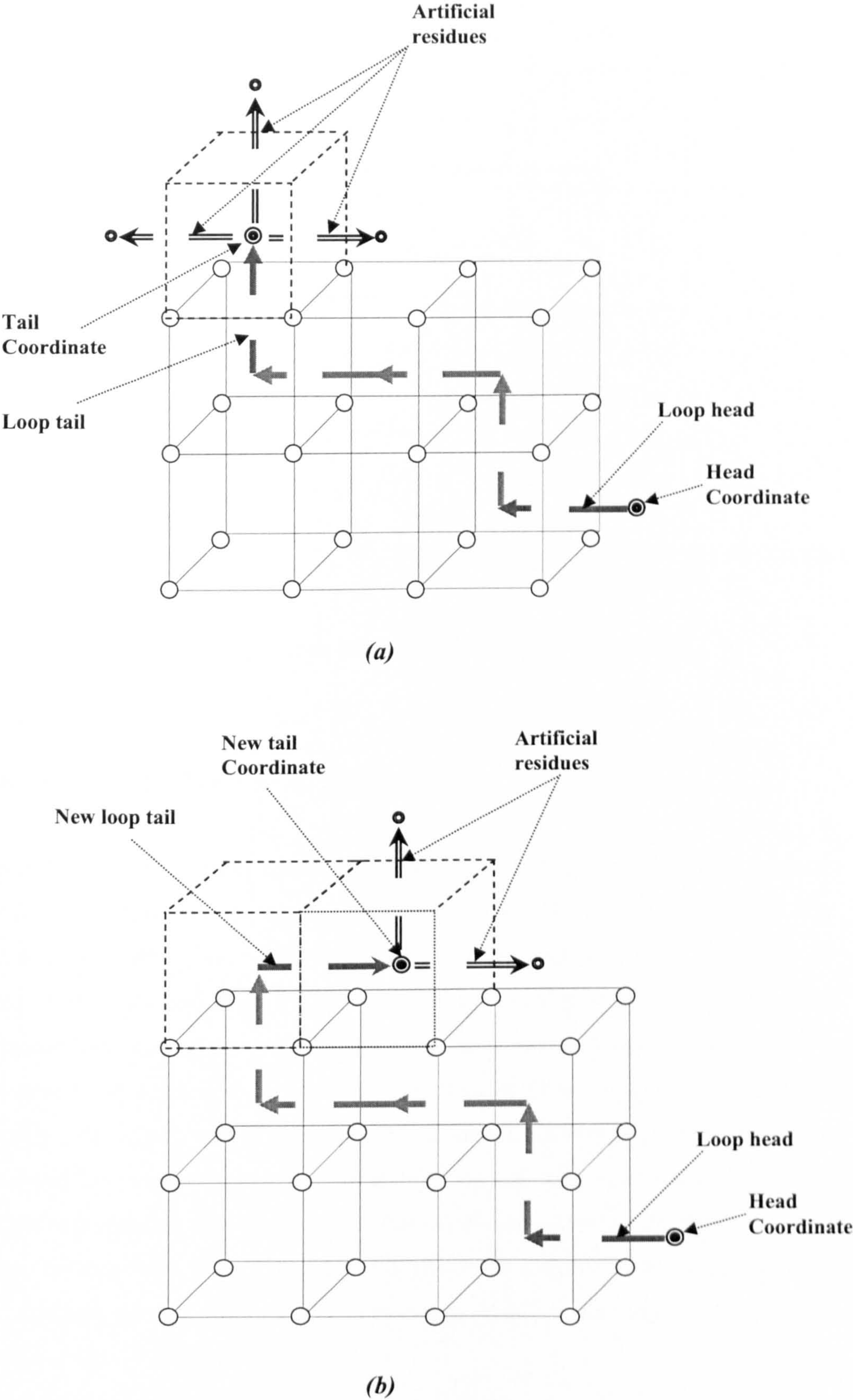
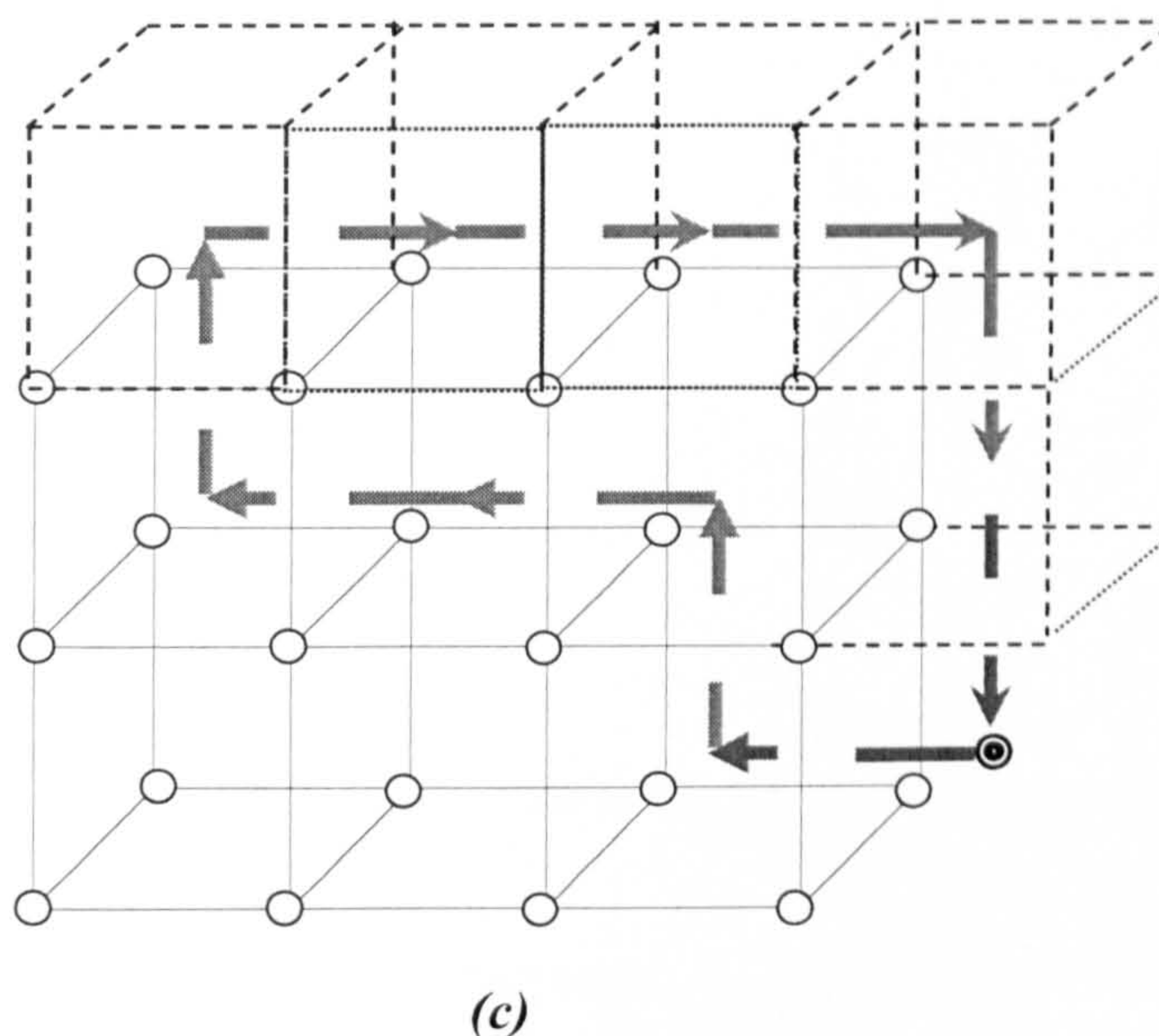


Figure 5.8: Demonstration of closing partial loops.



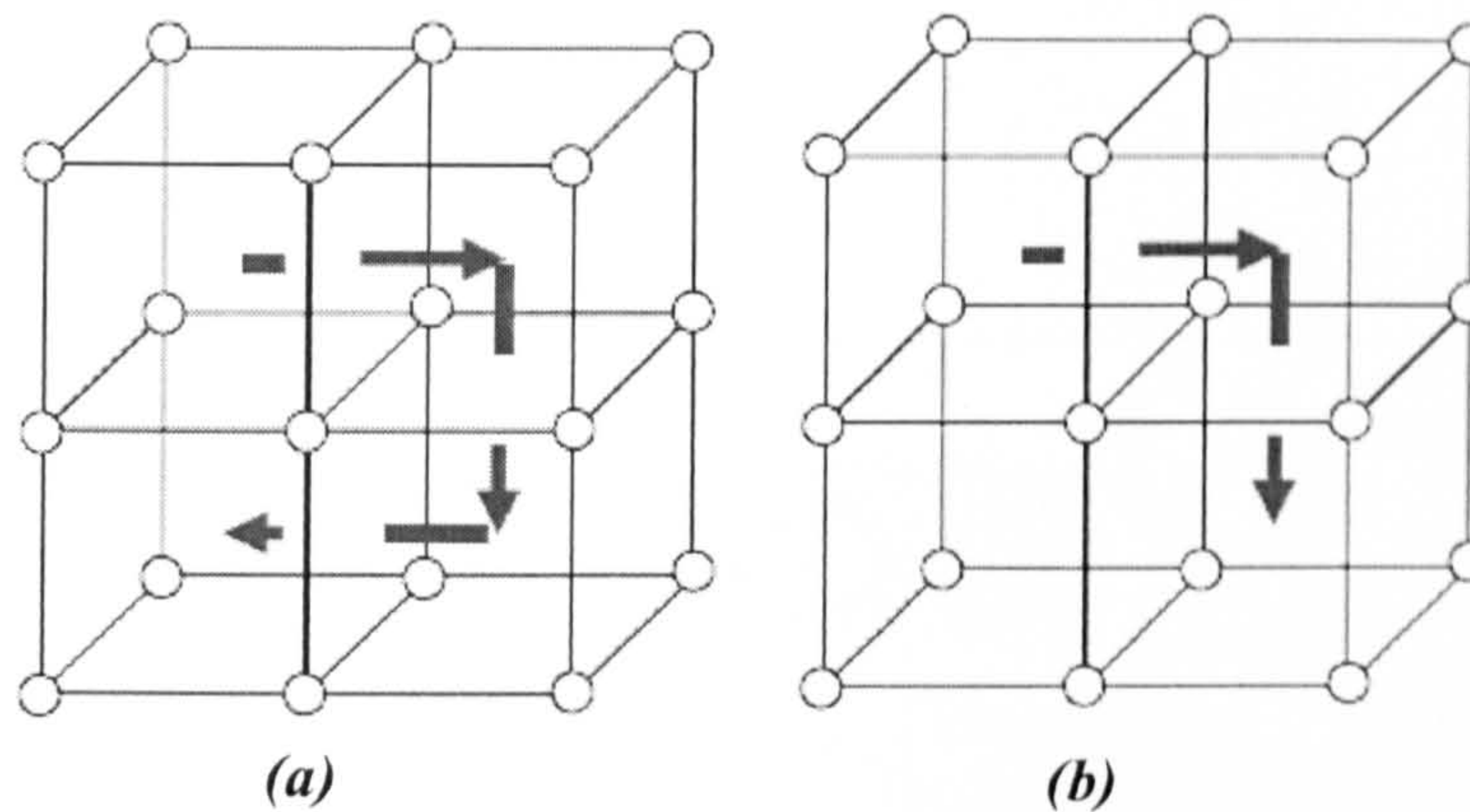


**Figure 5.8: Demonstration of closing partial loops (Continued).**

### 5.4.3 Identifying Zero-Weighted Edges.

After identifying all singularity loops and closing partial loops, the algorithm proceeds to identify zero-weighted edges, *i.e.* those edges that penetrate the singularity loops. The procedure of identifying zero-weighted edges can be carried out as described by Marklund *et al.* (Marklund *et al.*, 2005). The main idea of this procedure depends upon shrinking the loop towards its centre until it vanishes. As the loops shrink, it passes through successive edges and each time it passes through an edge, that edge will be permanently marked as a zero-weighted edge. Shrinking the loop is based upon finding U-shaped and L-shaped segments that form this loop and then by replacing them with appropriate segments, as will be explained below. U-shaped segments consist of three successive residues that form a U shape, as shown in Figure 5.9(a). Meanwhile, L-shaped segments consist of only two successive residues forming an L-shape, as illustrated in Figure 5.9(b) below.





**Figure 5.9: (a) A U-shaped segment. (b) An L-shaped segment.**

A loop must contain either U-shaped segments, L-shaped segments, or both, to be closed. So the procedure of shrinking the loop is based upon finding these U-shaped and L-shaped segments and replacing them with new artificial segments in a manner that shrinks the loop.

The procedure of processing a closed loop can be illustrated in the flow chart that is shown in Figure 5.10. As shown in the chart, a closed loop is examined to find out if it contains any U-shaped segments. If a U-shaped segment is discovered it should be replaced by the appropriate single residue value and the edge inside that U-shape is marked as a zero-weighted edge, as will be explained in detail in the following section. After replacing a U-shape by a single artificial residue value, the number of the residue inside the loop is counted. If this number is greater than three residues, the U-shape search procedure is carried out again until this number is less than or is equal to three.

In the case where no U-shape is discovered in the loop, the algorithm will instead search for the first L-shape segment that is present in the loop. If an L-shape is discovered, the algorithm has to check whether the replacement of this L-shape increases or decreases the area within that loop. If the area would be increased, then the algorithm leaves that L-shape and searches for the first ‘good L-shape’, *i.e.* one that minimizes the area of the loop. When a good L-shape is discovered, the algorithm replaces it with two new artificial residues and the edge located in between the L-shape arm is marked as a zero-weighted edge, as will be explained in the following sections. After replacing an L-



shape, the algorithm again searches for a U-shape. This process will be repeated until the number of residues in that loop is less than two.

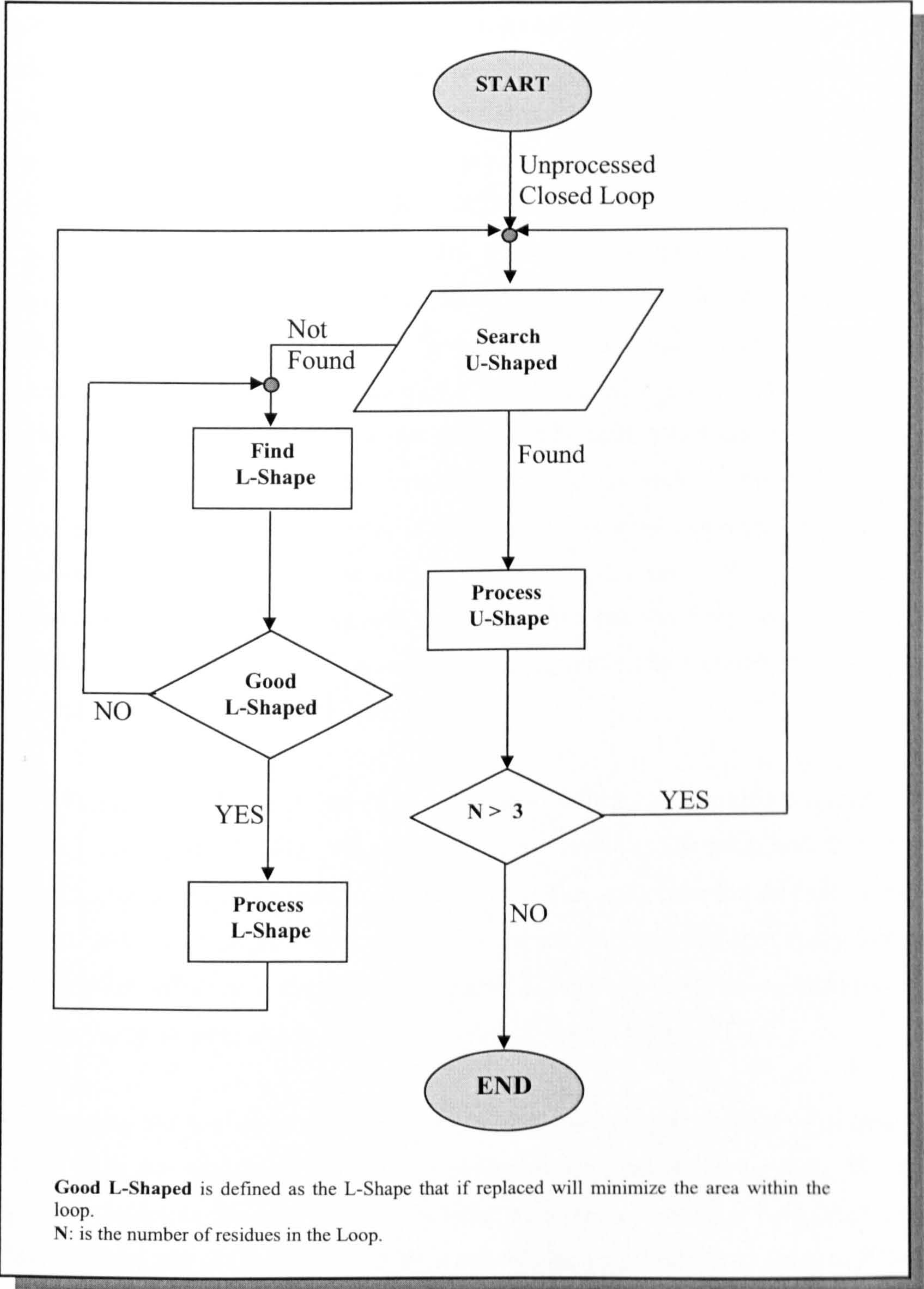


Figure 5.10: A flow chart for processing a closed loop.



As an example, suppose that the closed singularity loop shown in Figure 5.11(a) is to be processed by the procedure described previously. First the algorithm searches for any U-shaped segments located in that loop. Apparently, this closed loop does not contain any U-shaped segments, so the algorithm then searches for any L-shaped segments. Suppose that the algorithm finds the L-shaped segment that is shown in Figure 5.11(b). The algorithm will check if the replacement of this L-shaped segment will increase or decrease the area within the loop. The new possible choice of path for the L-shaped segment that is to be replaced is shown in Figure 5.11(b) as marked by dotted arrows. Clearly, this possible replacement of the L-shaped segment will increase the overall loop area, so the procedure will ignore this choice of L-shaped segment and will continue to search for another L-shaped segment that will decrease the entire area of the loop (this kind of L-shaped segment will be referred to as a 'good L-shaped' segment). As the procedure continues, it will search for new L-shaped segments. Figure 5.11(c) shows a new L-shaped segment that has subsequently been found and the potential replacement L-shaped segment is shown in Figure 5.11(c), as marked by dotted arrows. This L-shaped segment is considered to be a good L-shaped segment, because its replacement will decrease the entire area of the loop. In this case, the algorithm will replace this L-shaped segment and will mark the edge that the loop passes through during its shrinking process as a zero-weighted edge, marked as a dotted line edge in Figure 5.11(d). See the legend of Figure 5.11.

After replacing an L-shaped segment, the algorithm will again examine the loop to detect the presence of any U-shaped segments. In our case, the algorithm will find the U-shaped segment which is shown in Figure 5.11(d). The replacement of this U-shaped segment is with only a single residue that definitely will minimize the area of the loop, so the algorithm will directly replace this U-shaped segment by its replacement and will mark the appropriate edge as a zero-weighted edge, as shown Figure 5.11(e).

After replacing the U-shaped segment and marking the appropriate edge as a zero-weighted edge, the algorithm will check the number of residues in the loop. If the number of residues in the loop is less than three, then the processing is completed for this loop and the algorithm has to identify a new closed loop in order to process it. In our example, the number of residues in the loop shown in Figure 5.11(e) is 10, *i.e.* a value greater than three, so the algorithm will search for a U-shaped segment once



again. Clearly, as shown in Figure 5.11(e), the loop does not contain any U-shaped segments, so the technique will proceed by finding the next good L-shaped segment, which it does and this is illustrated by the segment marked by dotted arrows in Figure 5.11(e). The L-shaped segment will be replaced and the appropriate edge will be marked as a zero-weighted edge, as shown in Figure 5.11(f). Again, the algorithm now will search for a U-shaped segment and will replace it with its equivalent single residue value, as shown in Figures 5.11(f) and 5.11(g).

This process will be repeated many times until the whole loop is processed. Figures 5.11(g), 5.11(h), 5.11(i), 5.11(j) and 5.11(k) show the stages of shrinking the sampled loop until the whole loop is processed and all zero-weighted edges associated within the original loop are identified, as shown in Figure 5.11(k).

As explained above, the algorithm for identifying zero-weighted edges relies completely on finding U-shaped and L-shaped segments that are located inside the loop and then replacing these segments with appropriate replacements that would minimize the area within the loop. Also, it marks every individual edge that the loop passes through during this shrinking procedure as a zero-weighted edge. But several questions still remain before we can claim to have fully described the operation of the algorithm. For example, how can we identify these U-shaped and L-shaped segments? What are the rules for replacing these segments? What are the rules that will identify which edge is to be marked as a zero-weighted edge? All of these questions will be answered and explained in the following subsection.

#### 5.4.3.1 Detecting U-Shaped Segments.

Suppose that  $r_{c-1}$ ,  $r_c$  and  $r_{c+1}$  are three successive residues that exist in a closed loop, where their types and signs and indices are given by:

$$r_{c-1} \left\{ \begin{array}{l} type = t_{c-1} \\ sign = s_{c-1} \\ index = (i_{c-1}, j_{c-1}, k_{c-1}) \end{array} \right\} \Rightarrow r_c \left\{ \begin{array}{l} type = t_c \\ sign = s_c \\ index = (i_c, j_c, k_c) \end{array} \right\} \Rightarrow r_{c+1} \left\{ \begin{array}{l} type = t_{c+1} \\ sign = s_{c+1} \\ index = (i_{c+1}, j_{c+1}, k_{c+1}) \end{array} \right\}$$

(If further clarification of the definition of the residue 'index' parameter is required please refer back to Figure 5.6.)



These three residues form a U-shaped segment, if and only if, they obey the following rules:

$$\left\{ \begin{array}{l} r_{c-1}.type = r_{c+1}.type \neq r_c.type \\ r_{c-1}.sign = -r_{c+1}.sign \\ \|r_{c-1}.index - r_{c+1}.index\| = 1 \end{array} \right\} \quad (5.11)$$

If all the above conditions are true for the three residues, then these three residues form a U-shaped segment and should be replaced in a manner that will be explained shortly.



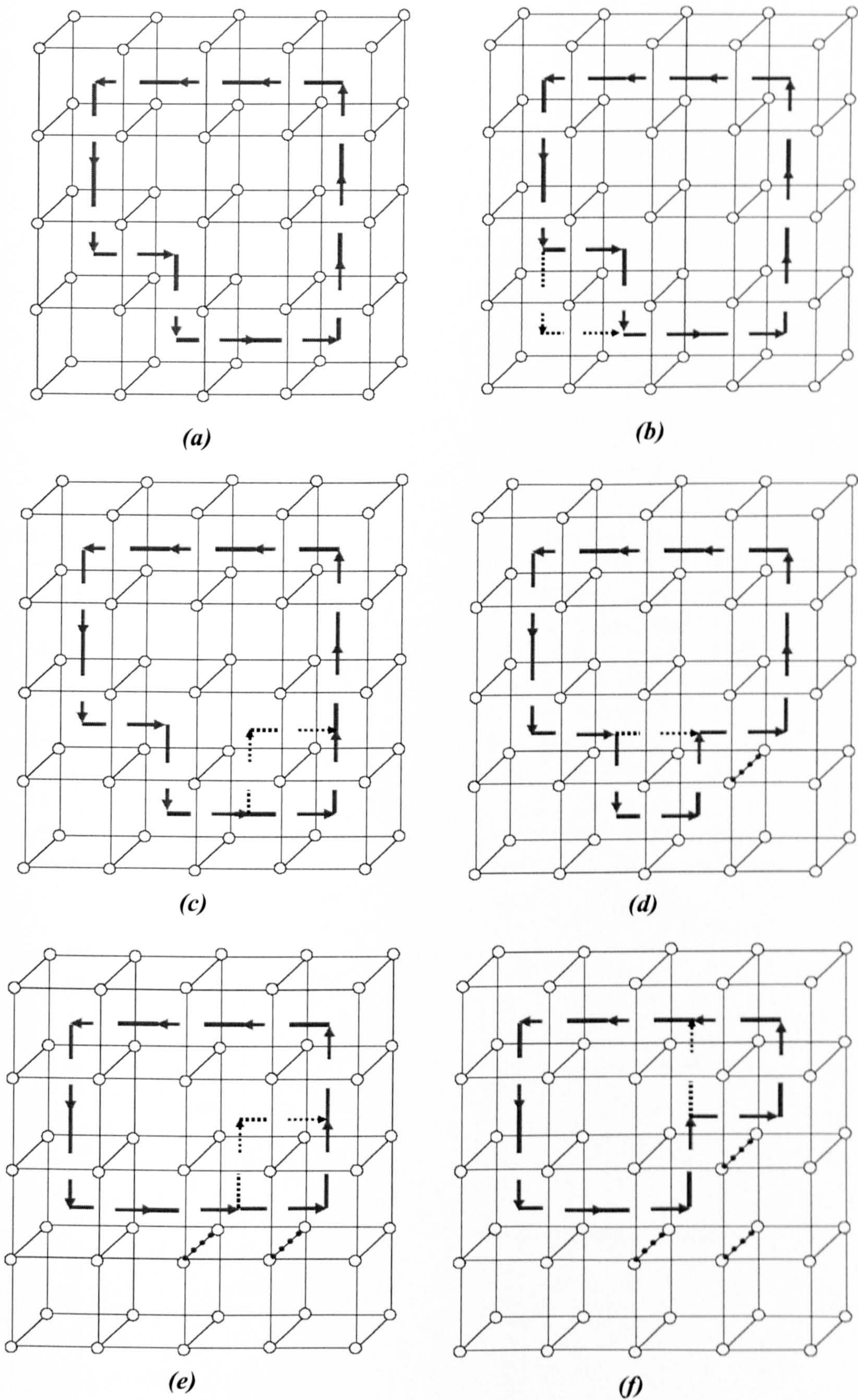


Figure 5.11: Demonstration of processing a closed loop.



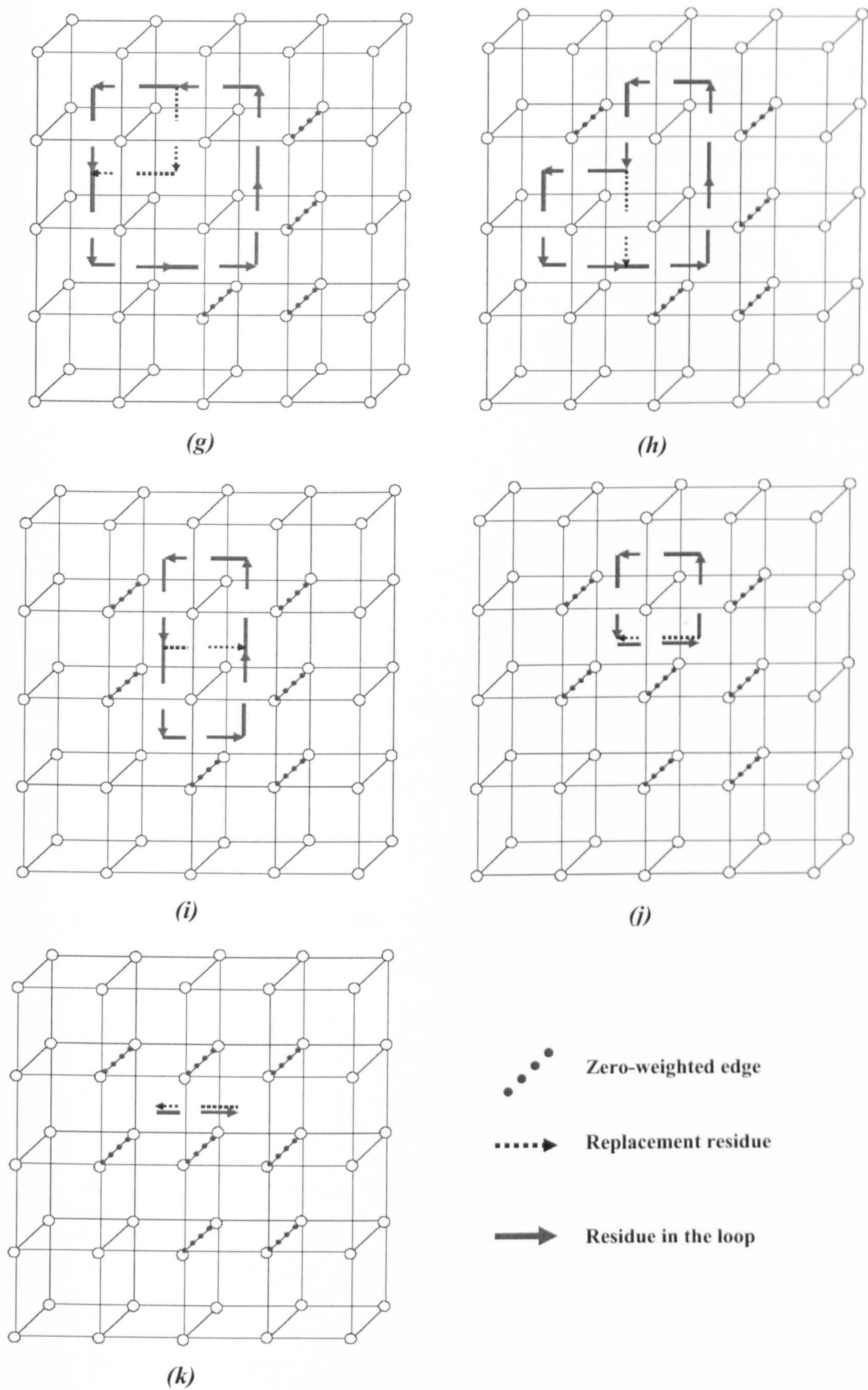


Figure 5.11: Demonstration of processing a closed loop. (Continued)



### 5.4.3.2 Detecting L-shaped Segments.

Suppose that  $r_c$  and  $r_{c+1}$  are two successive residues in a closed loop, where their types and signs and indices are given by:

$$r_c \left\{ \begin{array}{l} type = t_c \\ sign = s_c \\ index = (i_c, j_c, k_c) \end{array} \right\} \Rightarrow r_{c+1} \left\{ \begin{array}{l} type = t_{c+1} \\ sign = s_{c+1} \\ index = (i_{c+1}, j_{c+1}, k_{c+1}) \end{array} \right\}$$

These two residues form an L-shaped segment if:

$$\{r_c.type \neq r_{c+1}.type\} \quad (5.12)$$

### 5.4.3.4 Replacement of U-Shaped Segment

As illustrated in the example shown in Figure 5.11 (d), only one artificial residue is needed to replace a U-shaped segment. However, when replacing a U-shaped segment a number of questions arise, such as: what is the type, sign and index of the new residue? What is the direction of the edge that needs to be marked as a zero-weighted edge? And finally, what is the index of the zero-weighted edge?

In order to answer these questions, several examples are now presented.

- *Example 1:*

Figure 5.12 shows a U-shaped segment located in the  $zx$  plane that needs to be replaced. This U-shaped segment consists of the following residues:

$$r_{c-1} \left\{ \begin{array}{l} type = x \\ sign = +ve \\ index = (i, j, k) \end{array} \right\} \Rightarrow r_c \left\{ \begin{array}{l} type = z \\ sign = -ve \\ index = (i, j, k) \end{array} \right\} \Rightarrow r_{c+1} \left\{ \begin{array}{l} type = x \\ sign = -ve \\ index = (i, j, k-1) \end{array} \right\}$$

Note that these three residues obey the U-shaped rules defined in Equation 5.11.



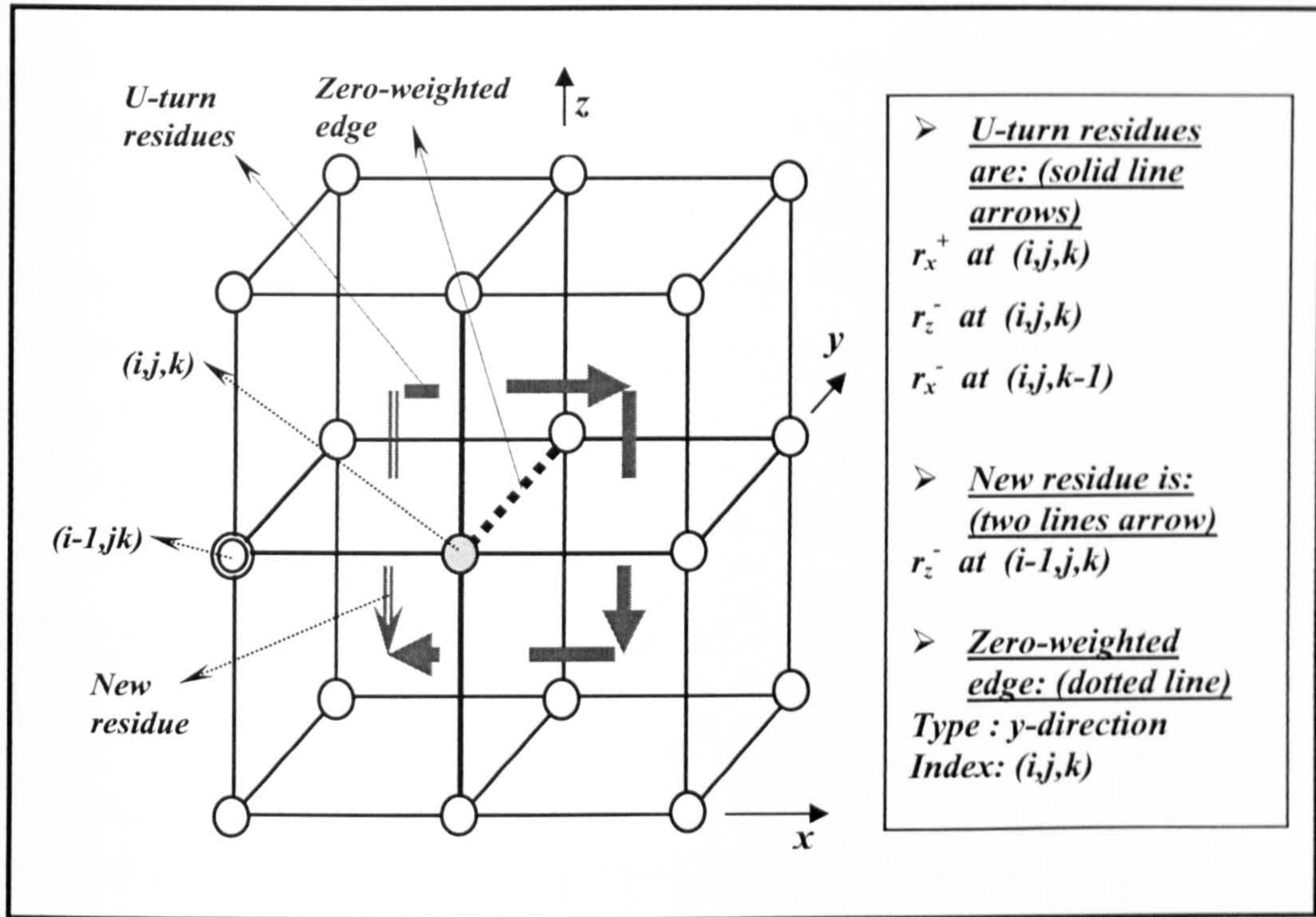


Figure 5.12: Example 1 of replacing U-shaped segments.

As shown in Figure 5.12, the new artificial residue needed to replace the U-shaped segment, that is shown as a double-line arrow in the figure is given by:

$$r_n \left\{ \begin{array}{l} type = z \\ sign = -ve \\ index = (i-1, j, k) \end{array} \right\}$$

On the other hand, the edge needed to be marked as a zero-weighted edge, represented by a dotted line in the figure, is given by:

$$zwe \left\{ \begin{array}{l} type = y \\ index = (i, j, k) \end{array} \right\}$$

This means that the edge connected between the voxels  $(i,j,k)$  and  $(i,j+1,k)$  has to be marked as a zero-weighted edge. (Note that if the type of this edge was an 'x-type', rather than the 'y-type' that is given in this example, then for this case it would be the edge that connects voxel  $(i,j,k)$  with voxel  $(i+1,j,k)$  which has to be marked as the zero-weighted edge).



From this example, we can conclude that the new residue inherits the type and sign of the middle residue  $r_c$ . Furthermore, the type of the zero-weighted edge that must be marked is that type that is missing from the U-shaped segment. In other words, by definition the U-shaped segment will contain only two of the three possible types of residue and the type of the zero weighted edge will be set to the other third possible type that is not present in the U-shaped segment.

This is expressed in the following equations:

$$r_n.type = r_c.type \quad (5.13)$$

$$r_n.sign = r_c.sign \quad (5.14)$$

$$zwe.type \neq (r_c.type \text{ or } r_{c\pm 1}.type) \quad (5.15)$$

Two more questions still need to be answered; the first is how to know the index of the new residue, and the second is how to know the index of the zero-weighted edge. These two questions are investigated further in the following examples.

- *Example 2:*

In this example we consider the U-shaped residues shown in Figure 5.13. These residues are given by:

$$r_{c-1} \left\{ \begin{array}{l} type = z \\ sign = -ve \\ index = (i, j, k+1) \end{array} \right\} \Rightarrow r_c \left\{ \begin{array}{l} type = x \\ sign = -ve \\ index = (i, j, k) \end{array} \right\} \Rightarrow r_{c+1} \left\{ \begin{array}{l} type = z \\ sign = +ve \\ index = (i-1, j, k+1) \end{array} \right\}$$

In this case the new residue and the zero-weighted edge are:

$$r_n \left\{ \begin{array}{l} type = x \\ sign = -ve \\ index = (i, j, k+1) \end{array} \right\}$$

$$zwe \left\{ \begin{array}{l} type = y \\ index = (i, j, k+1) \end{array} \right\}$$

As seen in this example, the type and sign of the new residue still inherits the type and sign of the middle residue  $r_c$ . Also, the type of the zero-weighted edge still obeys the rule explained in the first example.



By careful consideration of the index of the new residue in the first and second examples, it can be noticed that the distance between the new residue and the middle residue,  $r_c$ , is always equal to 1. Or as a formula we can write:

$$\|r_c.index - r_n.index\| = 1 \quad (5.16)$$

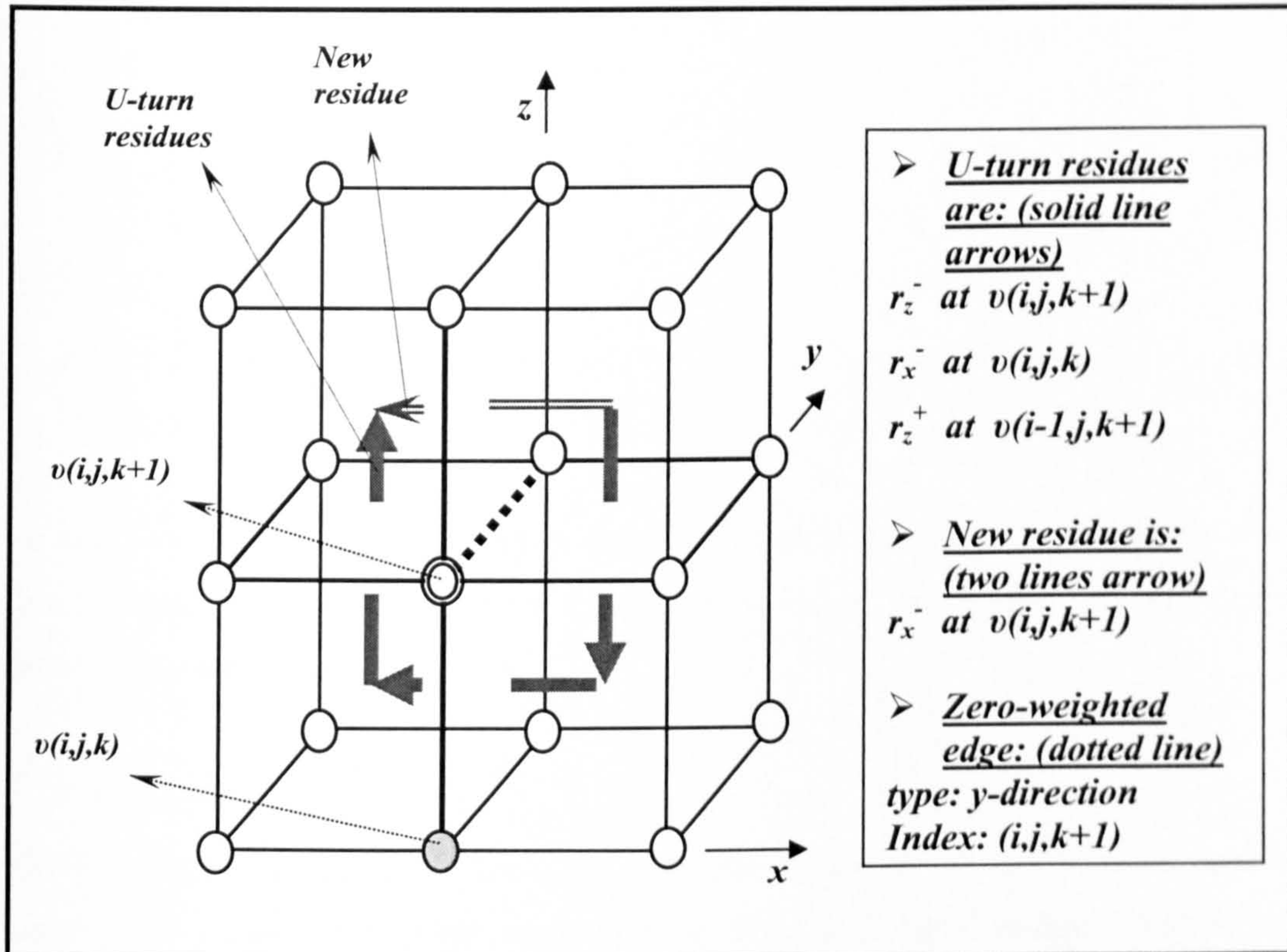


Figure 5.13: Example 2 to illustrate the process of replacing U-shaped segments.

We refer to the index of the middle residue  $r_c$  as  $(i,j,k)$  in both examples. In the first example the index of the new residue is  $(i-1,j,k)$ . We can note that the change in the index occurred here in the  $x$ -coordinate and the type of the first (or last) residue was of the  $x$ -type, (note that the first and the last residue have the same type in the U-shaped segment). Furthermore, in the current example, the new residue index is  $(i,j,k+1)$  and the type of the first (or last) residue in the U-shaped segment is of the  $z$ -type. From these two examples we can conclude:



$$r_n.index = \begin{cases} i \pm \lambda_x \\ j \pm \lambda_y \\ k \pm \lambda_z \end{cases} \quad (5.17)$$

where;

$$\lambda_x = \begin{cases} 1, & \text{if } r_{c+1}.type = x \\ 0, & \text{otherwise} \end{cases} \quad (5.18)$$

$$\lambda_y = \begin{cases} 1, & \text{if } r_{c+1}.type = y \\ 0, & \text{otherwise} \end{cases} \quad (5.19)$$

$$\lambda_z = \begin{cases} 1, & \text{if } r_{c+1}.type = z \\ 0, & \text{otherwise} \end{cases} \quad (5.20)$$

We still have to determine in which cases we have to add the lambda values and in which cases we have to subtract them. These issues are further investigated by considering more examples.

- *Example 3:*

In this example we consider a U-shaped segment located in the  $xy$  plane, as shown in Figure 5.14(a). Figure 5.14(b) shows a top view of the segment shown in Figure 5.14(a).

The U-shaped residues are:

$$r_{c-1} \left\{ \begin{array}{l} type = x \\ sign = -ve \\ index = (i+1, j-1, k) \end{array} \right\} \Rightarrow r_c \left\{ \begin{array}{l} type = y \\ sign = +ve \\ index = (i, j, k) \end{array} \right\} \Rightarrow r_{c+1} \left\{ \begin{array}{l} type = x \\ sign = +ve \\ index = (i+1, j, k) \end{array} \right\}$$

The new residue, as shown in the figure is:

$$r_n \left\{ \begin{array}{l} type = y \\ sign = +ve \\ index = (i+1, j, k) \end{array} \right\}$$



The zero-weighted edge in this example is:

$$zwe \left\{ \begin{array}{l} type = z \\ index = (i+1, j, k) \end{array} \right\}$$

In this example, we attempt to illustrate how to correctly decide whether to add 1 to the index or whether we have to subtract 1 from the index. By looking at the all examples we can conclude that the addition or subtraction is a function of the sign of the third residue in the U-shaped segment. When the sign of  $r_{c+1}$  is positive, then we have to add lambda, and whenever that sign is negative we have to subtract lambda, as is the case for all three of the above examples. So, as a general rule; the index of the new residue can be calculated by:



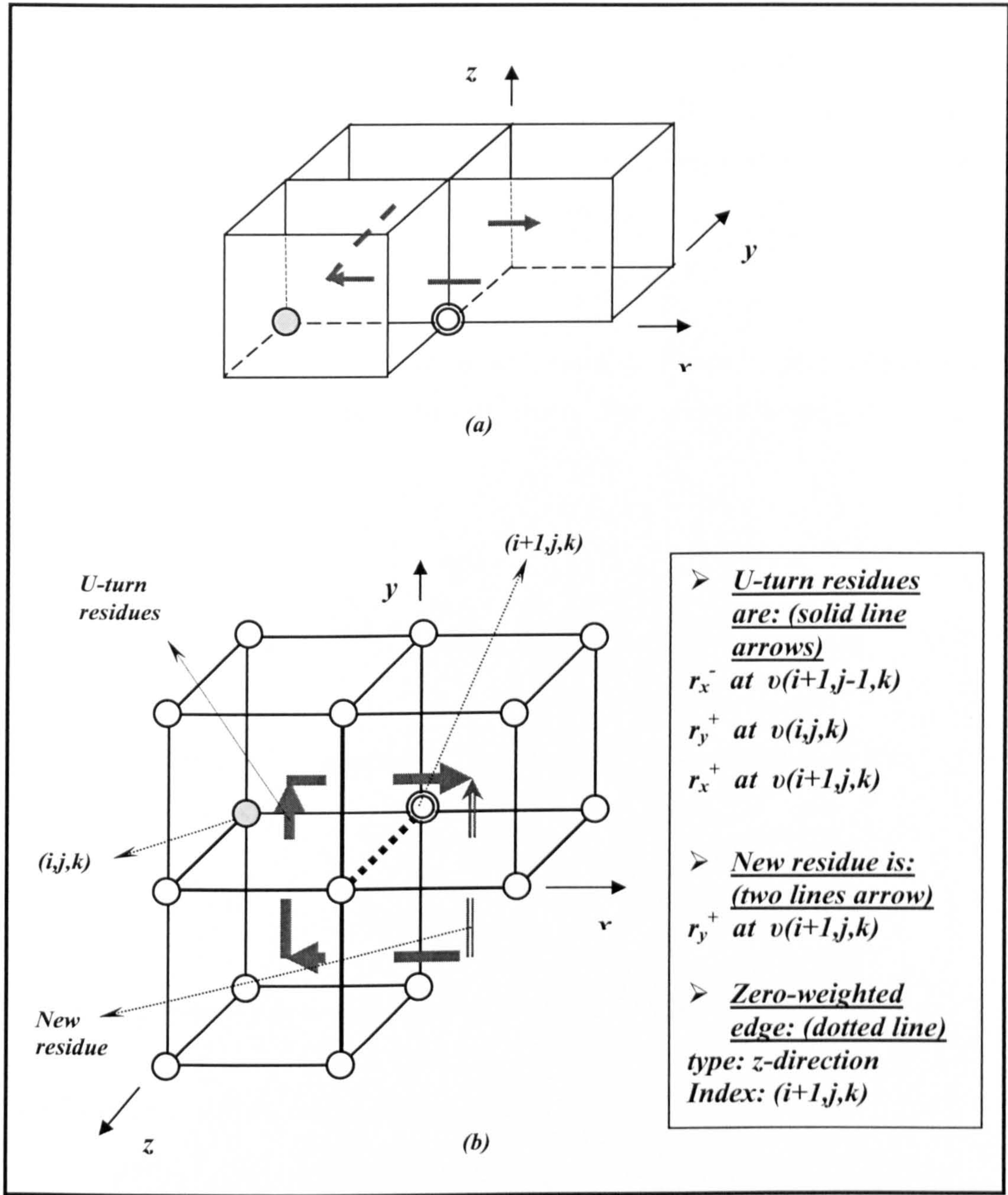


Figure 5.14: (a) Example 3 to illustrate the process of replacing U-shaped segments. (b) A top view of the figure shown in (a)

$$r_n.index = \begin{cases} i + \lambda_x \times (r_{c+1}.sign) \\ j + \lambda_y \times (r_{c+1}.sign) \\ k + \lambda_z \times (r_{c+1}.sign) \end{cases} \quad (5.21)$$

Where;

$\lambda_x, \lambda_y$  and  $\lambda_z$  are defined by equations 5.18, 5.19 and 5.20 above.



In the previous examples we concluded how to determine the type, sign and the index for the new replacement residue. Moreover, we also determined how to calculate the type of the zero-weighted edge which is to be marked. This leaves us with only one question still pending? What is the index of the zero-weighted edge? An additional example is considered in order to illustrate this issue.

• *Example 4:*

The U-shaped residues in this example are shown in Figure 5.15(a). Figure 5.15(b) shows a side-on view of Figure 5.15(a) for clarity. The residues forming this U-shaped segment are:

$$r_{c-1} \left\{ \begin{array}{l} \text{type} = z \\ \text{sign} = -ve \\ \text{index} = (i, j-1, k+1) \end{array} \right\} \Rightarrow r_c \left\{ \begin{array}{l} \text{type} = y \\ \text{sign} = +ve \\ \text{index} = (i, j, k) \end{array} \right\} \Rightarrow r_{c+1} \left\{ \begin{array}{l} \text{type} = z \\ \text{sign} = +ve \\ \text{index} = (i, j, k+1) \end{array} \right\}$$

Using the rules defined by Equations 5.13, 5.14 and 5.21, we can conclude that:

$$r_n \left\{ \begin{array}{l} \text{type} = y \\ \text{sign} = +ve \\ \text{index} = (i, j, k+1) \end{array} \right\}$$

This result matches the new residue shown in Figure 5.15(b)

From Figure 5.15(b), the zero-weighted edge is:

$$zwe \left\{ \begin{array}{l} \text{type} = x \\ \text{index} = (i, j, k+1) \end{array} \right\}$$

From all the examples given above, we can see that the index of the zero-weighted edge is always equal to the index of one of the side residues in the U-shaped segment,  $r_{c-1}$  or  $r_{c+1}$ . Furthermore, we can note that if the sign of the middle residue  $r_c$  is positive, then the index of the zero-weighted edge is equal to the index of last residue in the U-shaped segment,  $r_{c+1}$ . In contrast, if the sign of the middle residue  $r_c$  is negative, then the index



of the zero-weighted edge is the same as the index of the first residue in the U-shaped segment,  $r_{c-1}$ .

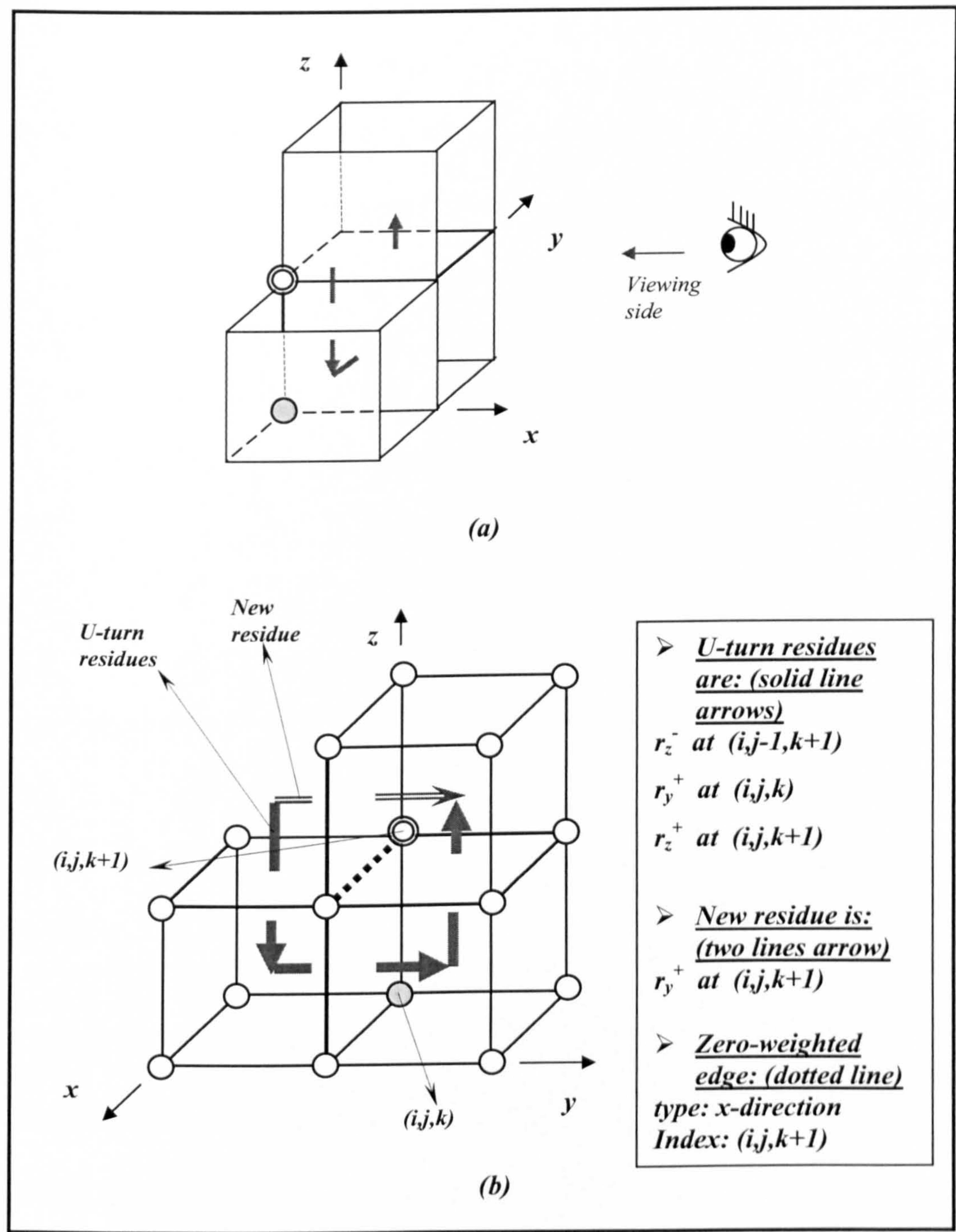


Figure 5.15: (a) Example 4 of replacing U-shaped segments. (b) A side view of the figure shown in (a)



As a result we can summarize the rules for replacing U-shaped segments as:

**Given the following U-shaped segment:**

$$r_{c-1} \Rightarrow r_c \Rightarrow r_{c+1}$$

**The new residue  $r_n$  needed to replace that segment is:**

$$r_n.type = r_c.type \quad (5.22)$$

$$r_n.sign = r_c.sign \quad (5.23)$$

$$r_n.index = \begin{cases} i + \lambda_x \times (r_{c+1}.sign) \\ j + \lambda_y \times (r_{c+1}.sign) \\ k + \lambda_z \times (r_{c+1}.sign) \end{cases} \quad (5.24)$$

Where;

$$\lambda_x = \begin{cases} 1, & \text{if } r_{c+1}.type = x \\ 0, & \text{otherwise} \end{cases} \quad (5.25)$$

$$\lambda_y = \begin{cases} 1, & \text{if } r_{c+1}.type = y \\ 0, & \text{otherwise} \end{cases} \quad (5.26)$$

$$\lambda_z = \begin{cases} 1, & \text{if } r_{c+1}.type = z \\ 0, & \text{otherwise} \end{cases} \quad (5.27)$$

**The edge that needed to be marked as a zero-weighted edge is determined by:**

$$zwe.type \neq (r_c.type \text{ or } r_{c+1}.type) \quad (5.28)$$

$$zwe.index = \begin{cases} r_{c-1}.index & \text{if } r_c.sign = -ve \\ r_{c+1}.index & \text{if } r_c.sign = +ve \end{cases} \quad (5.29)$$

#### 5.4.3.5 Replacement of L-Shaped Segments

Replacement of the L-shaped segments can be carried out by depending upon the rules for U-shaped segments that were presented in the last section. Suppose that we want to replace the L-shaped residue pair, denoted as  $r_c$  and  $r_{c+1}$  in Figure 5.16, by two new artificial residues  $r_{n1}$  and  $r_{n2}$  as shown in the figure.



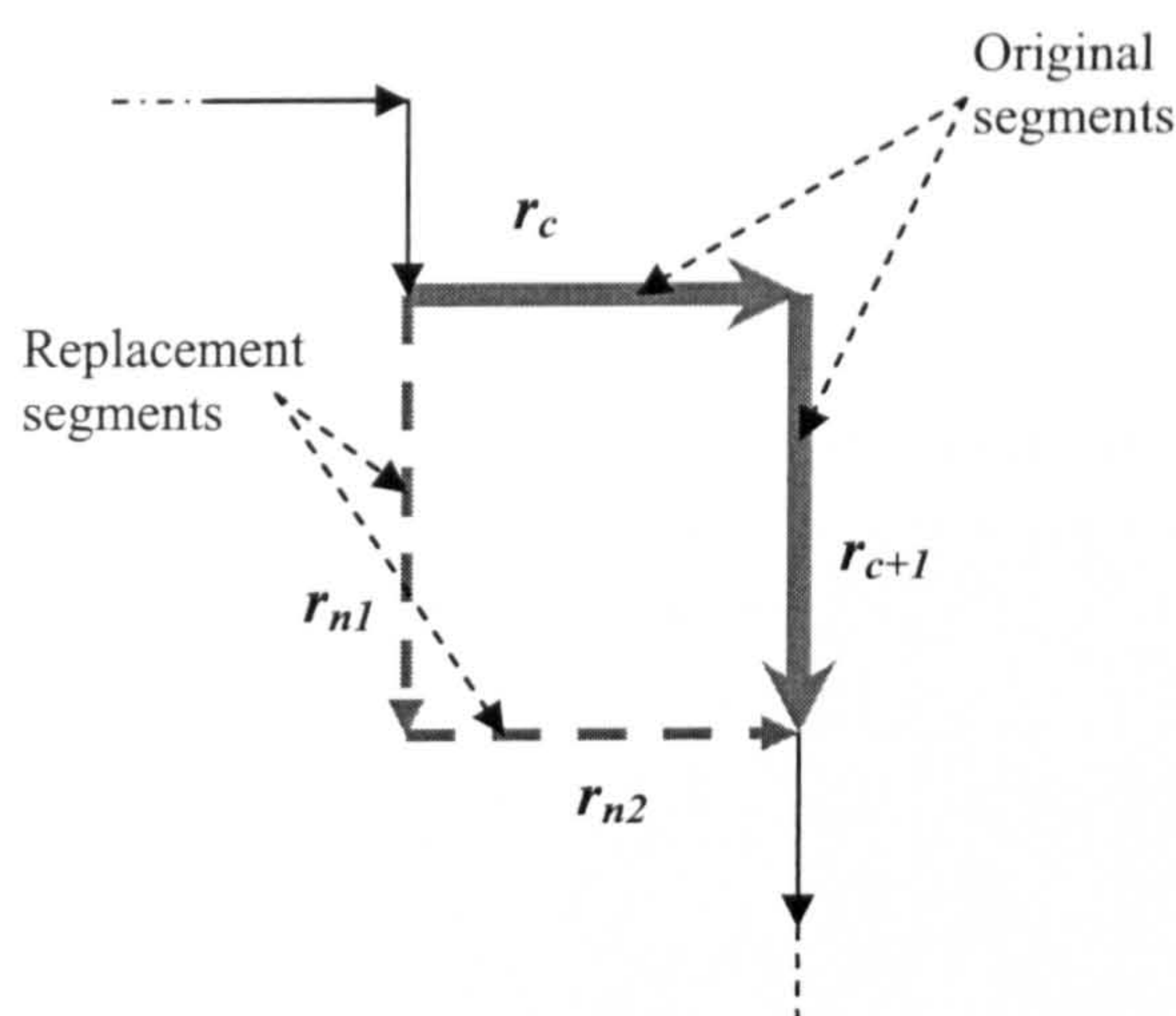


Figure 5.16: An L-shaped segment and its replacement.

From the previous section, we concluded that the new artificial residue is determined by the centre and the last residues in the U-shaped segment, *i.e.*  $r_c$  and  $r_{c+1}$ . This conclusion can be applied in the case of L-shaped residues. As shown in Figures 5.17(a) and 5.17(b), the artificial residue  $r_{n2}$  in the case of an L-shaped residue, is equivalent to the residue  $r_n$  in the U-shaped segment, so residue  $r_{n2}$  can be determined as if  $r_c$  and  $r_{c+1}$  form part of a U-shaped segment.

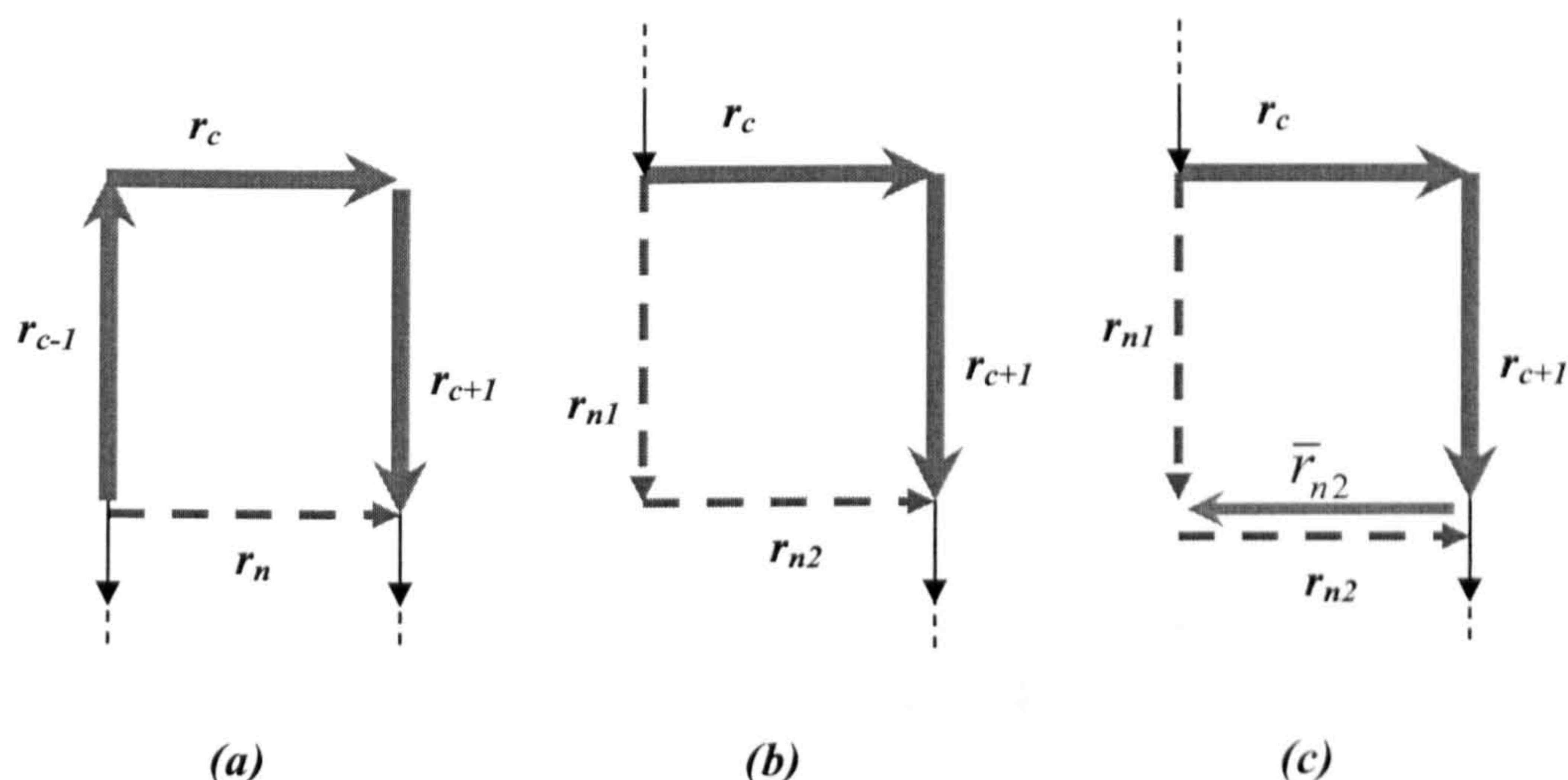


Figure 5.17: Identifying L-shaped segment replacement using U-shaped segments rules.  
 (a) set of U-shaped segment and its equivalent, (b) L-turn segment and its equivalent and  
 (c) new U-shaped segment and its equivalent



After determining  $r_{n2}$ , the residue  $r_{n1}$  can be determined by considering  $r_c$ ,  $r_{c+1}$  and  $\bar{r}_{n2}$  as a U-shaped segment, as shown in Figure 5.17(c). Noting that  $\bar{r}_{n2}$  has the opposite sign to  $r_{n2}$ .

As a result we can summarize the rules for replacing an L-shaped segment as follows:

**Given the following L-shaped segment:**

$$r_c \Rightarrow r_{c+1}$$

Then the new artificial residues  $r_{n1}$  and  $r_{n2}$  are determined by:

**Step1:** following the U-shaped rules, find  $r_{n2}$  from  $r_c \Rightarrow r_{c+1}$

**Step2:** from  $r_{c+1} \Rightarrow \bar{r}_{n2}$  find  $r_{n1}$ , where

$$\bar{r}_{n2}.type = r_{n2}.type$$

$$\bar{r}_{n2}.sign = -r_{n2}.sign$$

$$\bar{r}_{n2}.index = r_{n2}.index$$

**Step3:** find the corresponding  $zwe$  from  $r_c \Rightarrow r_{c+1}$  following the U-shaped rules defined in Equations. 5.28 and 5.29.

#### 5.4.4 Calculating Edges' Qualities.

After identifying all zero-weighted edges in the wrapped phase volume, the algorithm will proceed to calculate the qualities of each individual edge. The calculation of these qualities is carried out using one of the quality maps which were described earlier in this chapter, bearing in mind the fact that the qualities of the zero-weighted edges are assigned to zero.

Then, as in the case for the BP algorithm discussed above, all edges are sorted according to their qualities. The edges with higher quality values are placed in the front of the queue.

#### 5.4.5 Best Path Unwrapping.

After all edges' qualities are sorted, the algorithm will continue the unwrapping process following the discrete unwrapping path described earlier in section 5.3.2.



## 5.5 Results and Comparisons.

To evaluate the performance of the proposed algorithms, *i.e.* the BP and BPASL algorithms, two sets of wrapped-phase volumes were used, namely computer-generated and real wrapped-phase volumes. Both types of phase volumes were unwrapped using the proposed techniques. The results of these phase unwrapping algorithms were then compared with other existing three-dimensional phase unwrapping algorithms.

### 5.5.1 Simulated Results.

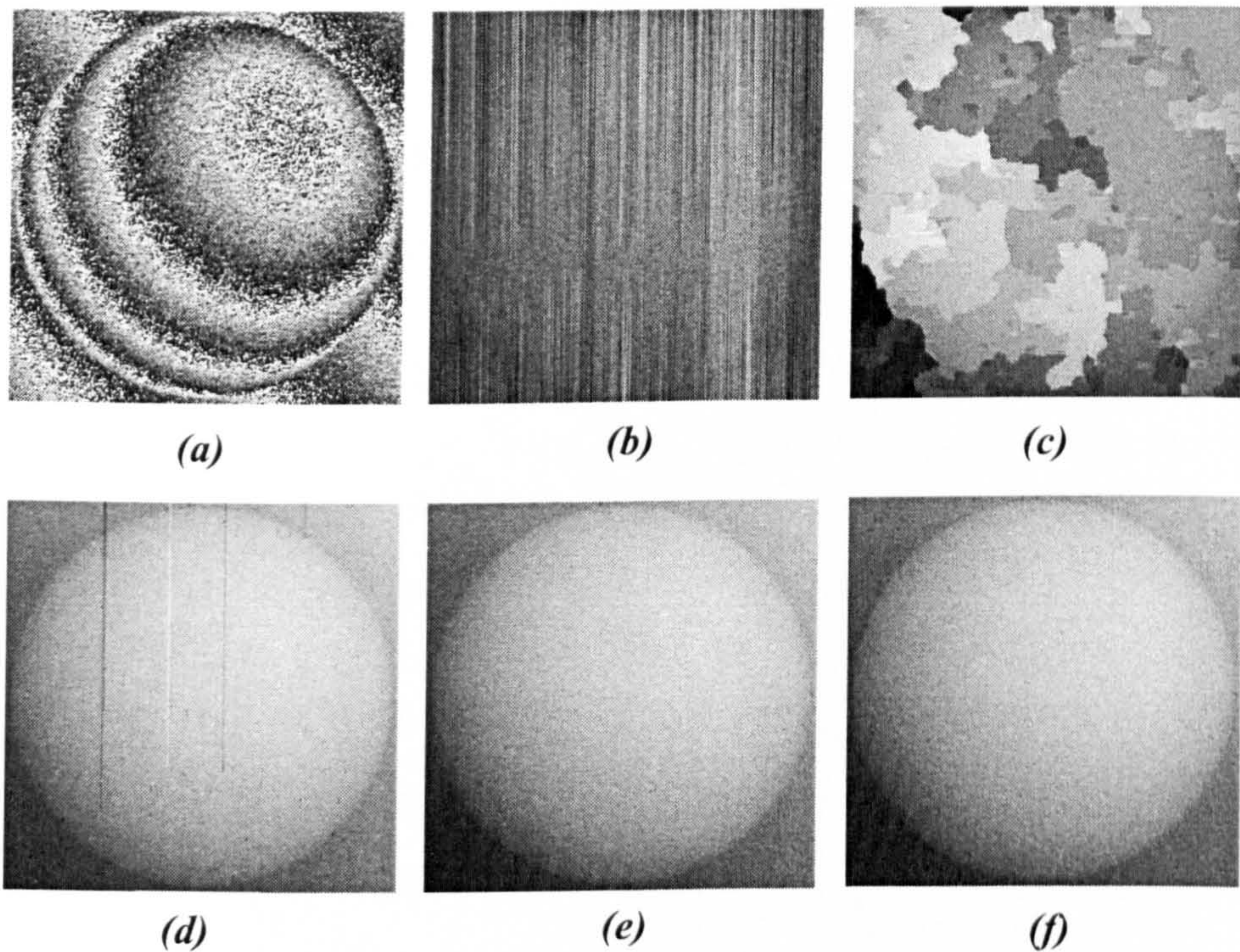
The proposed algorithms were tested using several simulated computer generated objects. The results of the proposed algorithm were compared with two well-known three-dimensional phase unwrapping algorithms. The algorithms used for comparison are considered to be state of the art in three-dimensional phase unwrapping and they have been used previously in many applications and have proven their robustness and high performance. The first algorithm is the three-dimensional phase unwrapping algorithm proposed by Cusack *et al.* and it is now in use for the efficient unwrapping of MRI phase volumes (Cusack and Papadakis, 2002). The second comparative technique is the three-dimensional noise immune phase unwrapping algorithm proposed by Huntley (Huntley, 2001).

The first computer generated object is a simple growing hemisphere whose radius is increasing with time, *i.e.*, with the frame number (see Appendix A.1). The sphere grows from a minimum radius to a maximum radius in 100 video frames, each frame consisting of  $256 \times 256$  pixels in the x and y dimensions respectively. Then Gaussian noise with zero mean and a standard deviation of 0.95 is added to each frame of the  $256 \times 256 \times 100$  voxel volume. The whole volume is subsequently wrapped between  $-\pi$  to  $\pi$  using the arctangent function and scaled between black and white for display purposes. The colour white here represents the maximum value and the colour black represents the minimum value of object height.

Figure 5.18 given below, shows the wrapped-phase and unwrapped-phase maps for the last frame ( $t=99$ ) of this sphere. The wrapped phase map for this frame is shown in Figure 5.18(a). As can be seen in the figure, the three-dimensional flood-fill phase



unwrapping algorithm and Cusack's algorithm completely fail to unwrap this phase volume, due to its high noise levels, as is shown in Figures 5.18(b) and 5.18(c) respectively. Huntley's algorithm produces an acceptable result for this frame, but with some error that has been propagated, as is shown in Figure 5.18(d). On the other hand, both the best path algorithm and the best path avoiding singularity loops algorithm successfully unwrap this wrapped-phase volume and deliver excellent results, as shown in Figures 5.18(e) and 5.18(f) respectively. Apparently Cusack's algorithm is not as robust against these very high noise levels when compared to the other algorithms considered here.



**Figure 5.18:** The results for the last frame ( $t=99$ ) of the simulated spherical object. (a) the wrapped-phase, the unwrapped-phase using: (b) flood-fill algorithm, (c) Cusack's algorithm, (d) Huntley's algorithm, (e) BP algorithm and (f) BPASL algorithm.

The second computer generated object considered is a dynamically moving object that is changing its shape with time, as described in Appendix A.2. A  $256 \times 256 \times 16$  Gaussian noise with zero mean and a standard deviation of 1.55 is added from frame 47 to frame 62 of this wrapped volume. Then, the whole volume is subsequently wrapped between  $-\pi$  to  $\pi$  using the arctangent function.



The wrapped phase volume is now divided into three discrete sets as follows: set 1 is a clean wrapped phase region located from frame 0 to frame 46, set 2 from frame 47 to frame 62 is very noisy region suffering from the added noise described above and finally, set 3 from frame 63 till frame 99 which is again a clean wrapped phase map with no added noise. The whole wrapped-phase volume is unwrapped using Cusack's, Huntley's, the best path and the best path avoiding singularity loops algorithms and the results are shown in Figure. 5.19

Row (a) in Figure 5.19 shows the wrapped-phase maps from each of three selected wrapped phase sets, the first map, the left-most, is for frame number 46, selected from set 1 of the volume where no noise is added. The second phase map, the middle in the row, is for frame number 55 chosen from set 2 with high noise levels present. The third map, the most right, is for frame 84 in set 3 where the noise is again absent.

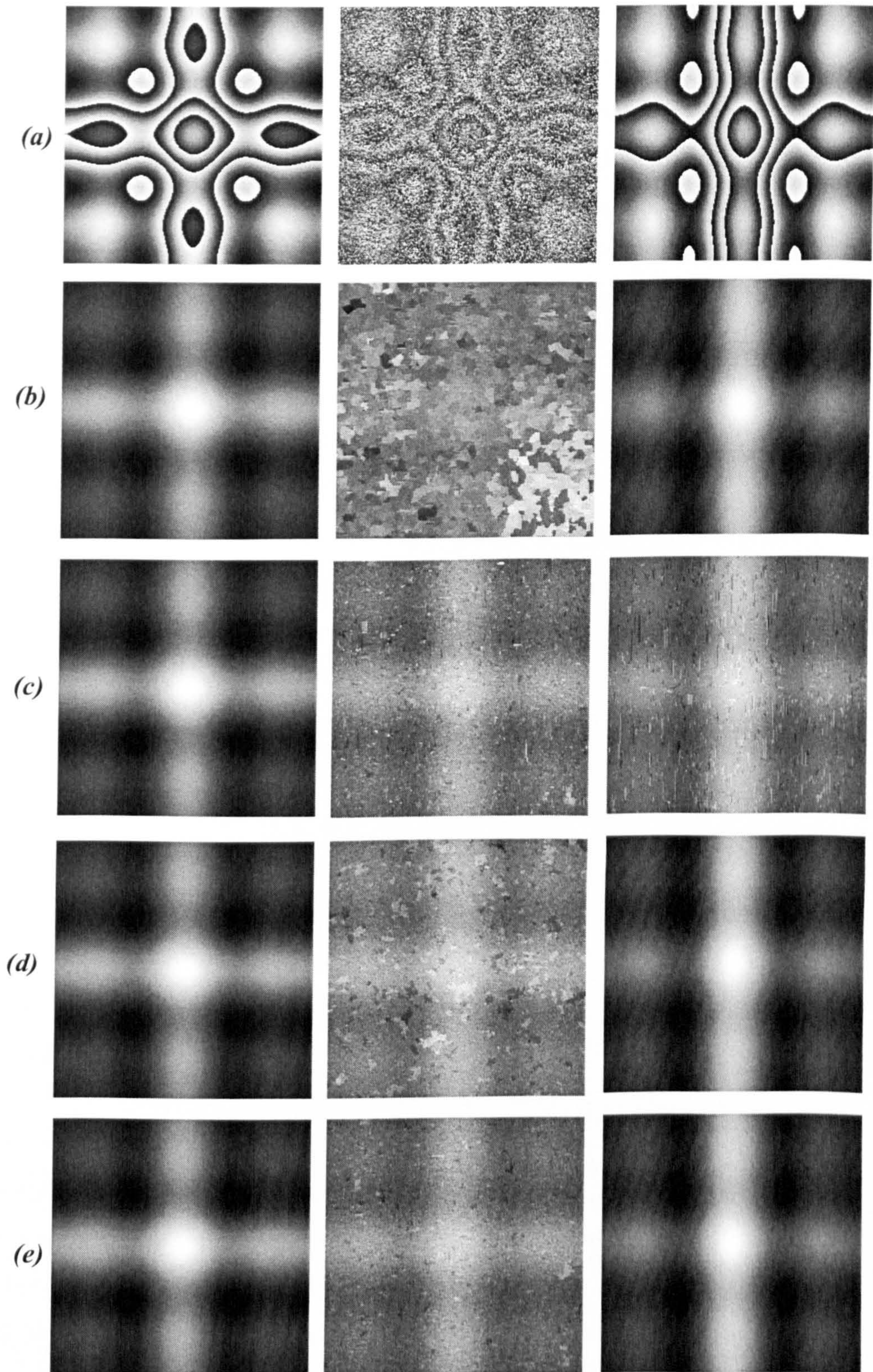
Row (b) shows the unwrapped phase maps resulting from using Cusack's algorithm. It can be easily seen that Cusack's algorithm succeeds in unwrapping frames 46 and 84. Nevertheless, this algorithm completely fails in unwrapping frame 55, as shown in the figure. Despite the complete failure of Cusack's algorithm to unwrap the noisy frames, it successfully isolates the noisy region and prevents errors propagating throughout the phase volume. This result is expected due to the fact that it relies upon a quality measure to guide the unwrapping procedure. This quality measure assigns lower qualities to those voxels located in the noisy region and thus this region was unwrapped last.

The results for Huntley's algorithm is shown in row (c) in Figure 5.19. As shown in that row, Huntley's algorithm succeeds in unwrapping frame 46, and it also manages to produce fair results in unwrapping the noisy frame. Indeed the results of Huntley's algorithm are much better in frame 55 than those which resulted from Cusack's algorithm for that particular frame. Indeed, Huntley's algorithm shows a high degree of robustness against noise. However, many ambiguities arise in identifying singularity loops, as discussed earlier in this chapter. These ambiguities violate the path independency and consequently a certain amount of errors propagate through to set 3. The unwrapped phase map of frame 84 is shown in the right-most corner in row (c) in the figure. Although this frame has no noise at all, Huntley's algorithm did not produce



a good result and we can easily see that this region is affected by some errors propagating through from the noisy region of set 2.





**Figure 5.19:** The results for frames 46, 55 and 84 of the second simulated object shown in first, second and third columns respectively. (row a) the wrapped-phase maps, the unwrapped-phase maps using: (row b) Cusack's algorithm, (row c) Huntley's algorithm, (row d) BP algorithm and (row e) BPASL algorithm.



Row (d) in Figure 5.19 shows the result of the best path algorithm. Because this algorithm depends on a quality measure to identify the unwrapping path, this algorithm succeeds in identifying the noisy region in order to minimize error propagation. The algorithm unwraps frames 46 and 85 successfully, without being affected with the noisy region. On the other hand, the algorithm produces fair results in unwrapping the noisy region as shown in the middle frame of this row. It is worth noticing that Huntley's algorithm manages to produce a better unwrapped phase map for frame 55 (noisy region) than the best path algorithm.

The results of the best path avoiding singularity loops algorithm are shown in row (e) in Figure 5.19. This algorithm combines the advantages of the best path algorithm, in finding the optimal unwrapping path, with the robustness of Huntley's algorithm in its robustness against noise. We can easily see that the BPASL algorithm gives the best results amongst all the other algorithms in this particular example. It indeed not only successfully isolates the noisy region and processes it last, but it also gives very good results in unwrapping the noisy regions, as is shown in row (e) of the figure.

### 5.5.2 Experimental Results.

The proposed algorithms (BP and BPASL) have also been tested experimentally on several three-dimensional wrapped-phase volumes that have resulted from the analysis of real fringe patterns projected onto the surface of dynamically moving objects. A video sequence of a moving object that dynamically modulates a fringe pattern was first captured. This video sequence was subsequently analysed frame-by-frame, using a two-dimensional Fourier fringe analysis algorithm, in order to produce a stack of two-dimensional wrapped-phase maps. Note that the extraction of wrapped-phase has not been optimised here and the wrapped-phase volumes are presented merely as vehicles for the testing of the unwrapping algorithms. This stack of two-dimensional wrapped-phase maps can then be considered to form a three-dimensional wrapped-phase volume. This has subsequently been unwrapped using the proposed and comparative algorithms.

The practical application that the phase unwrapping algorithm has been tested upon is that of dynamic human body measurement in radiotherapy treatment. This is an exacting application of 3D measurement technology, which requires accurate dynamic



measurements to be made in conditions that are far from ideal due to the effects of magnetic field produced by the x-rays machine which may affect the stability of the fringes.

Three wrapped-phase volumes have been used to show the performance of both proposed algorithms in unwrapping real wrapped phase volumes. The first two examples are taken from experiments carried out in the laboratory, simulating patient motion and the third example is from a real clinical setting. Each wrapped-phase volume consists of 25 frames with  $512 \times 512$  pixels in each frame. The first wrapped-phase volume has been obtained from analysing the chest/thorax region of a mannequin that has been manually raised and lowered in the mid-sagittal plane with a dorsal/ventral motion simulating respiration. The second wrapped-phase volume has been obtained by analysing the face of a RANDO Phantom, a synthetic human used in radiotherapy calibration, undergoing similar manual pseudo-respiratory motion in a laboratory setting. The third and final wrapped-phase volume is actual patient data acquired in a real medical clinical trial in a radiotherapy treatment room, exhibiting respiratory and ordinary levels of expected patient movement for a patient who is actually undergoing treatment for breast cancer.

Figure 5.20 shows three different arbitrary frames of the wrapped-phase volume resulting from measuring the mannequin's chest. Row (a) in the figure shows the wrapped-phase maps for frames numbered 0, 15 and 24 respectively. Row (b) in the figure shows the unwrapped-phase maps resulting from Cusack's algorithm. The unwrapped-phase maps resulting from Huntley's algorithm are shown in row (c). Row (d) and row (e) show the results of the BP and BPASL algorithms respectively. Clearly, all algorithms successfully unwrap this wrapped phase volume and they give satisfactory results. A 3D representation of the last frame is shown in Figure 5.21.



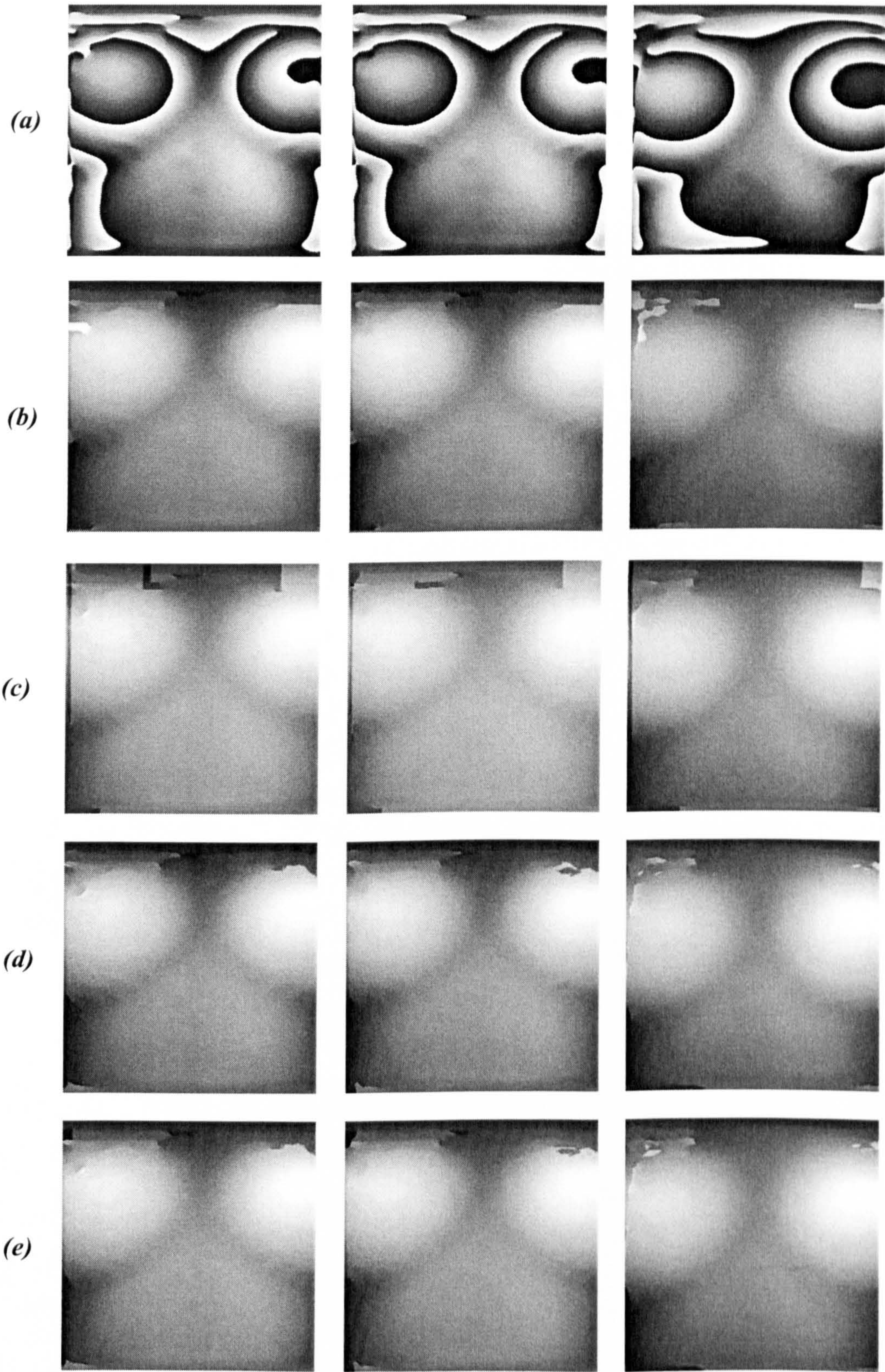


Figure. 5.20: The results for frames 0, 15 and 24 of the mannequin's chest shown in first, second and third columns respectively. (row a) the wrapped-phase maps, the unwrapped-phase maps using: (row b) Cusack's algorithm, (row c) Huntley's algorithm, (row d) BP algorithm and (row e) BPASL algorithm.



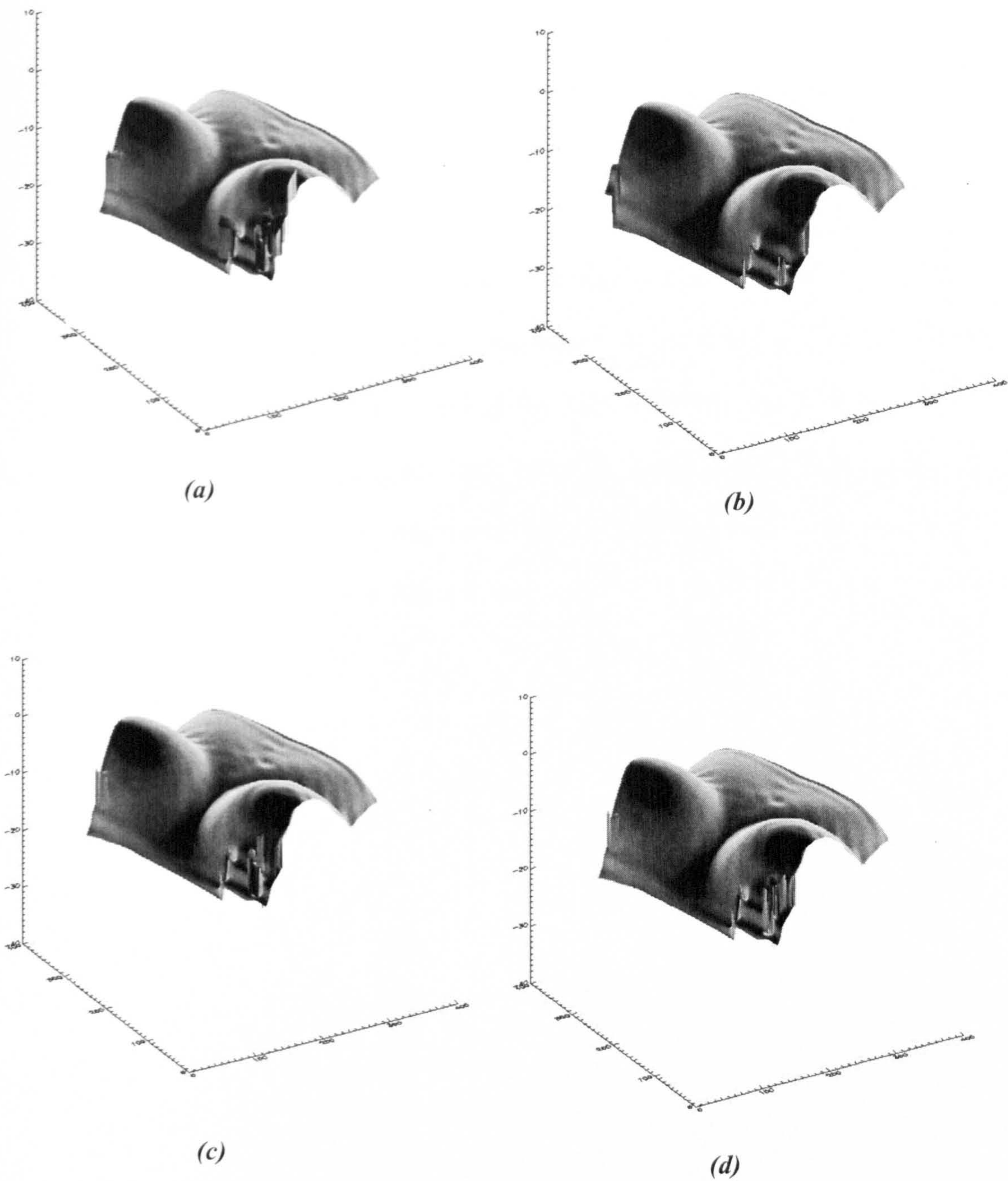


Figure 5.21: A three-dimensional view for the unwrapped-phase maps for frame 24 of the mannequin's chest using: (a) Cusack's algorithm, (b) Huntley's algorithm, (c) the BP algorithm and (d) the BPASL algorithm.



Figure 5.22 shows three different frames of the wrapped-phase volume resulting from measuring the radiotherapy RANDO Phantom dummy's face. Row (a) in the figure shows the wrapped-phase maps for frames 0, 15 and 24 respectively. The unwrapped phase maps using Cusack's, Huntley's, BP and BPASL algorithms are shown in rows (b), (c), (d) and (e) respectively.

Figure 5.22 shows that both of the proposed algorithms were capable of unwrapping the noisy regions last to prevent error propagation. Cusack's, and Huntley's algorithms create separated regions in unwrapping the dummy's face. On the other hand, the proposed algorithms unwrap the volume smoothly, without creating discontinuities in the unwrapped volume. Three-dimensional representations of the unwrapped-phase map for the last frame are shown in Figure 5.23.



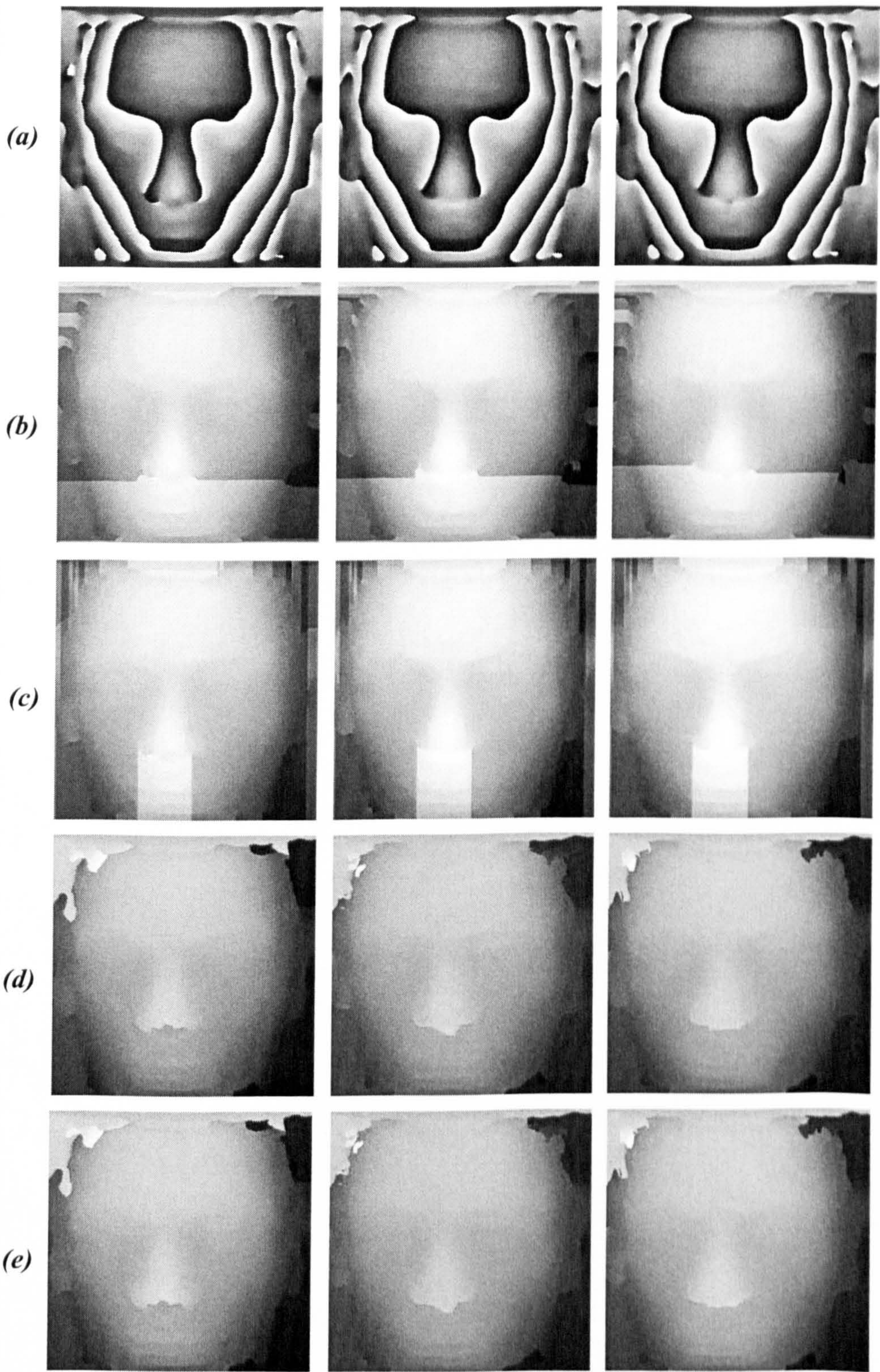
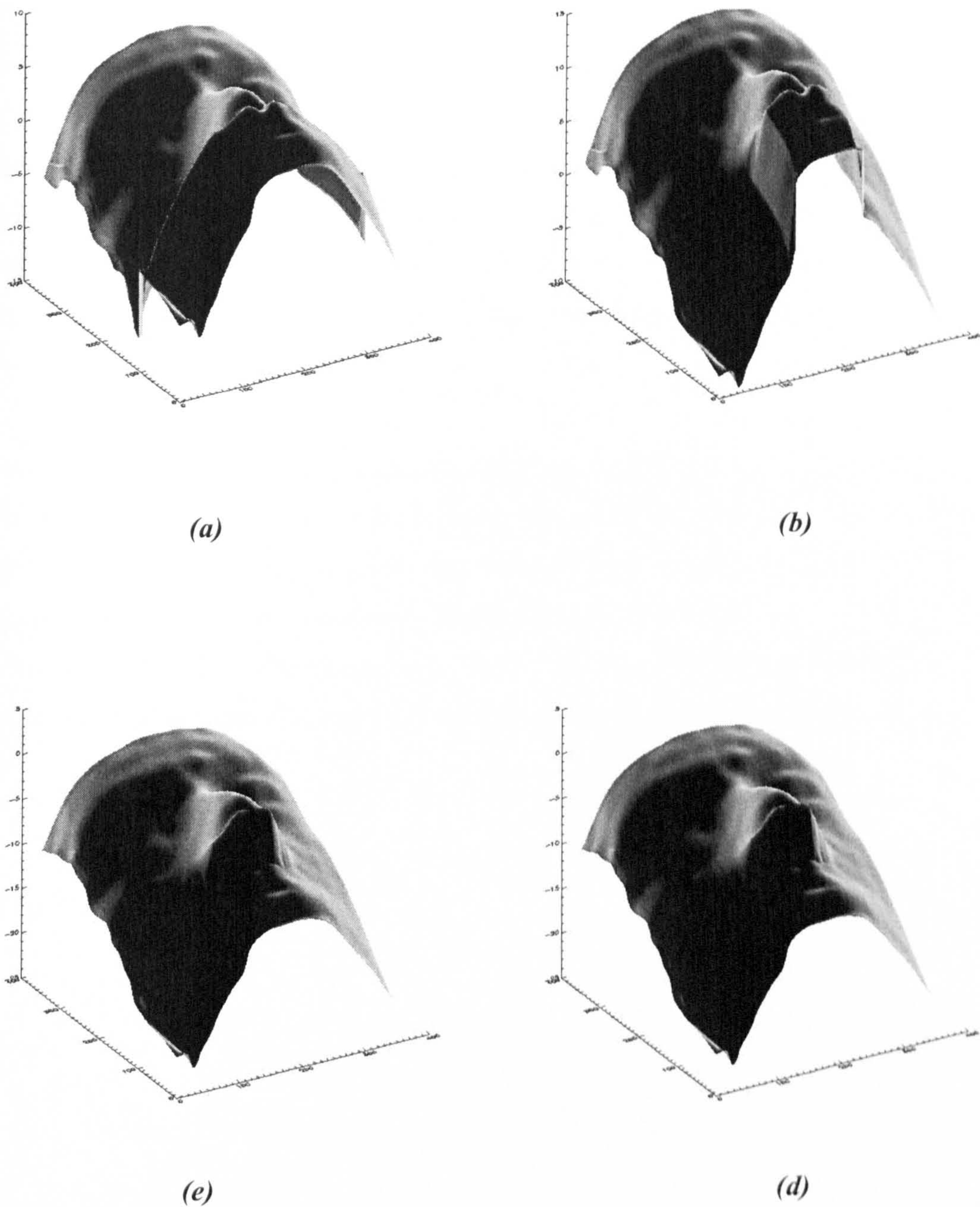


Figure 5.22: The results for frames 0, 15 and 24 of the dummy's face shown in first, second and third columns respectively. (row a) the wrapped-phase maps, the unwrapped-phase maps using: (row b) Cusack's algorithm, (row c) Huntley's algorithm, (row d) BP algorithm and (row e) BPASL algorithm.





**Figure 5.23:** A three-dimensional view for the unwrapped-phase maps for frame 24 of the mannequin’s chest using: (a) Cusack’s algorithm, (b) Huntley’s algorithm, (c) the BP algorithm and (d) the BPASL algorithm.



A  $256 \times 256 \times 100$  Gaussian noise term, with zero mean and a standard deviation of 0.85, is then added to the entire unwrapped phase volume of the dummy's face. This new phase volume is then rewrapped again using the arctangent function. Figure 5.24 shows both the wrapped phase and the unwrapped phase maps for this volume. Row (a) in Figure 5.24 shows the wrapped phase maps of the frames 0, 15, 24, sorted from left to right. The unwrapped phase maps using Cusack's, Huntley's, the BP and the BPASL algorithms are shown in rows (b), (c), (d) and (e) respectively. As shown in the figure, Cusack's algorithm and the BP algorithms give very poor results. Huntley's algorithm is still very powerful in dealing with noise, even with real objects, but due to ambiguities in the singularity loops, path dependency has been violated and so a separated region is created near to the dummy's nose.

The best path avoiding singularity loops algorithm again produces excellent results. The results show that this algorithm is very reliable. This algorithm is capable of finding an optimal path, whilst bearing in mind the effect of singularity loops. Although the ambiguities in identifying the singularity loops still exist in this algorithm, the effect of these ambiguities is minimized because of the fact that we are using a quality map to guide the unwrapping path. The results of this algorithm are shown in Figure 5.24 (e).



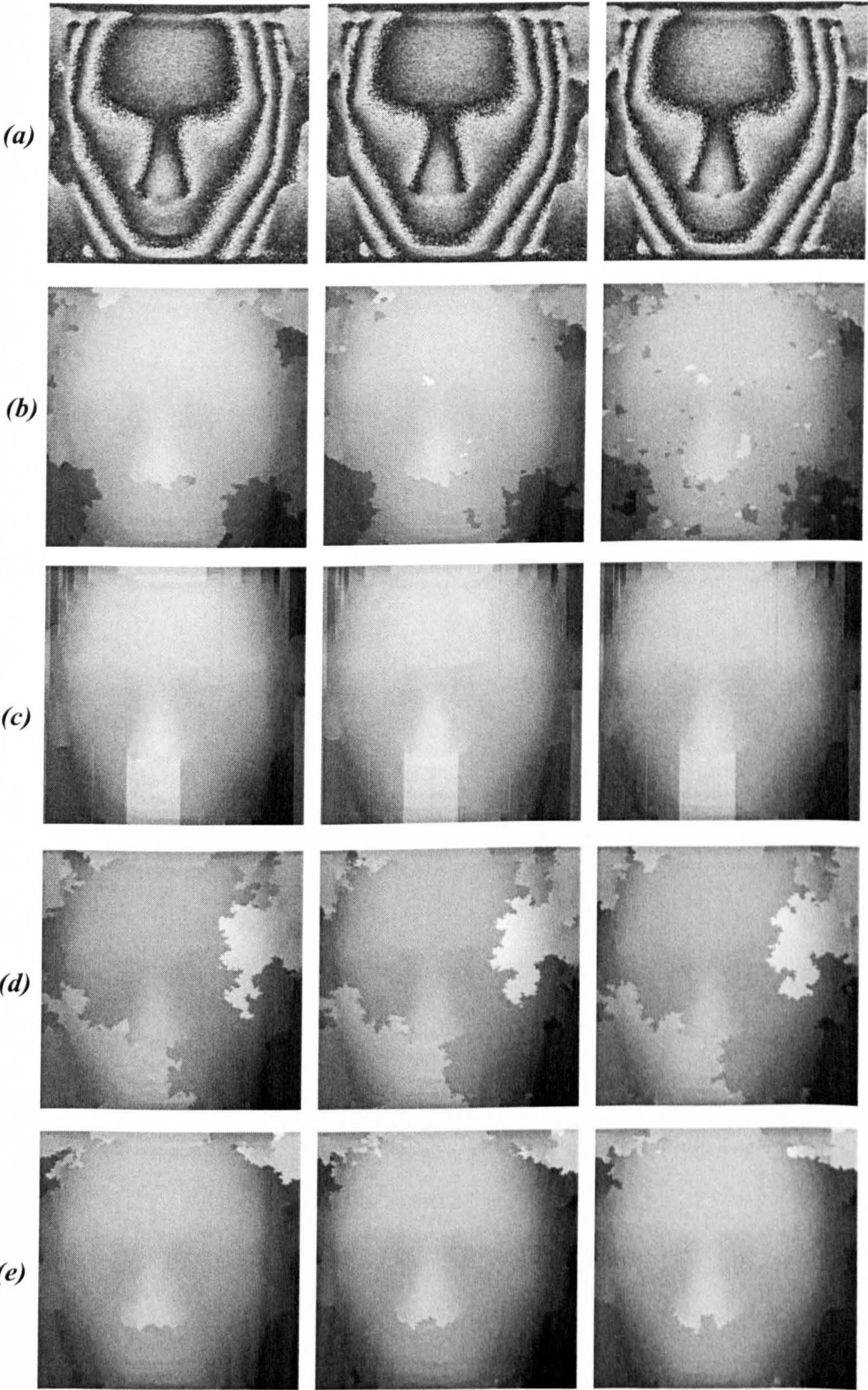
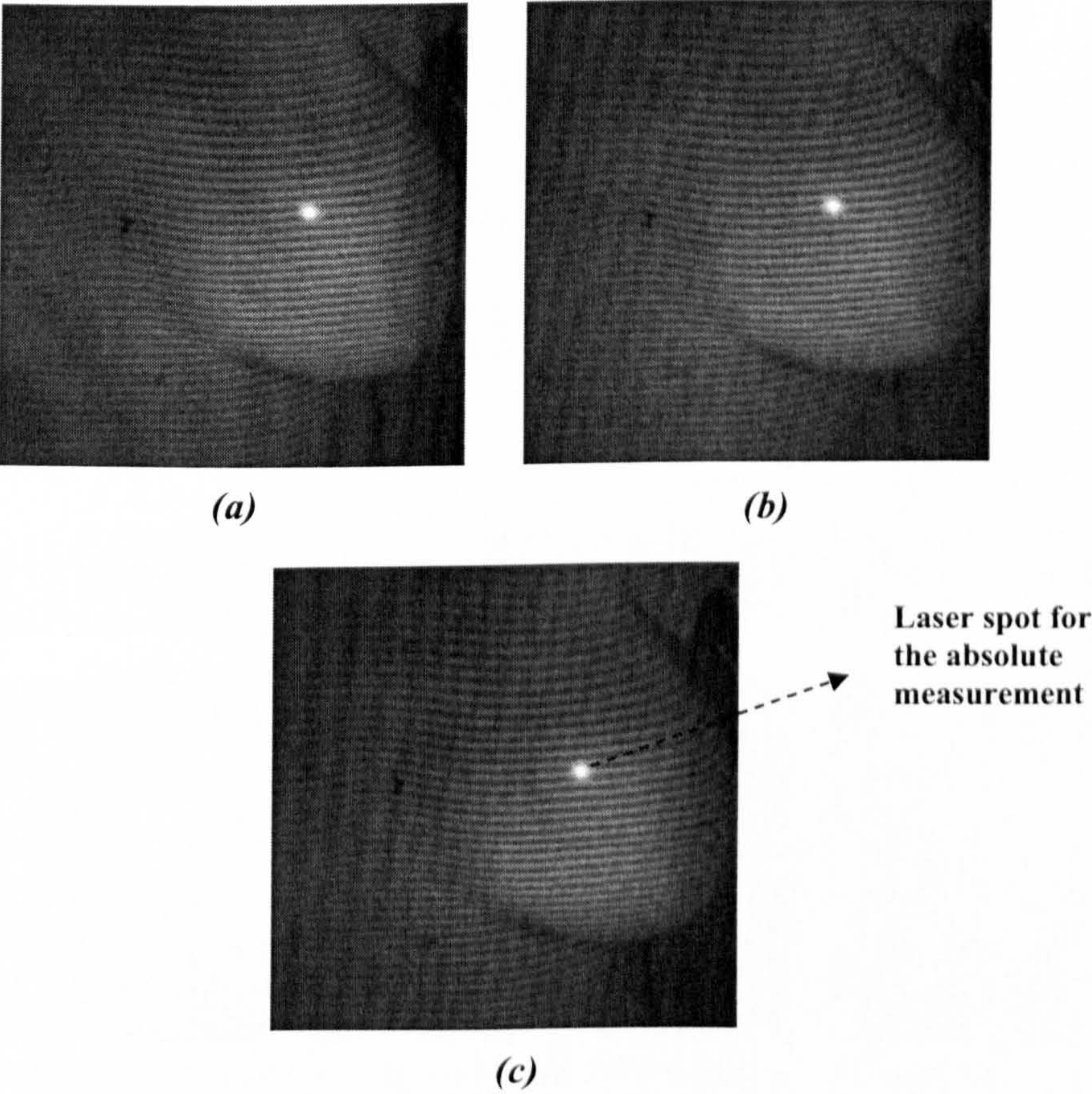


Figure. 5.24: The results for frames 0, 15 and 24 of the dummy's face with added noise shown in first, second and third columns respectively. (row a) the wrapped-phase maps, the unwrapped-phase maps using: (row b) Cusack's algorithm, (row c) Huntley's algorithm, (row d) BP algorithm and (row e) BPASL algorithm.

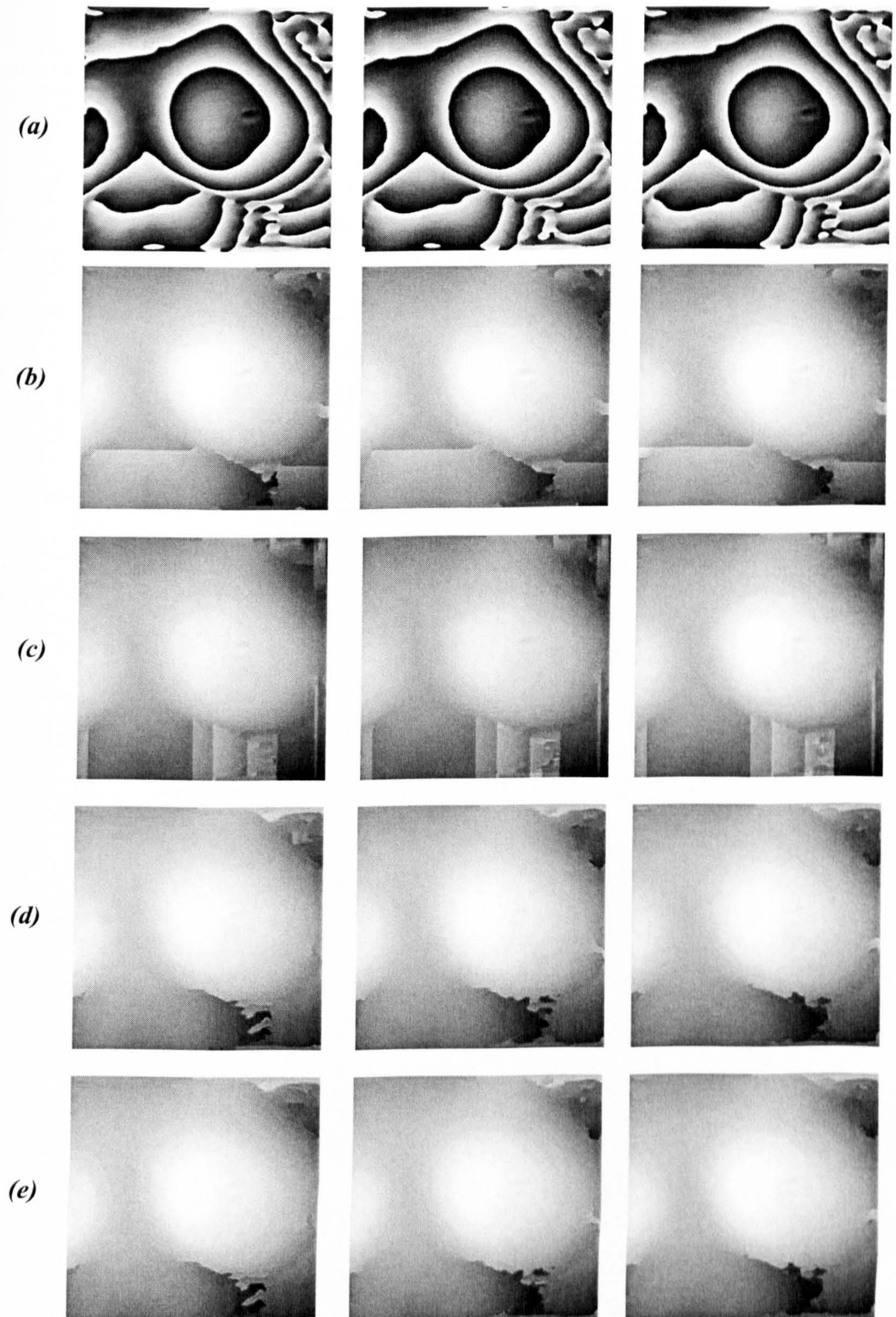


Figure 5.25 shows frames 0, 15 and 24, which have been taken from a video sequence of a patient undergoing radiotherapy treatment for breast cancer, displaying the real source fringe patterns which are to be analysed. The bright spot shown in this figure is a laser spot used to retrieve the absolute height. The top row of images presented in Figure 5.26 shows the processed wrapped-phase maps corresponding to the three fringe patterns shown in Figure 5.25. The second row shows the unwrapping results associated with these respective frames using Cusack’s algorithm. The third row shows similar results for Huntley’s algorithm. The forth row for the BP algorithm and the bottom row for the BPASL algorithm. It can be seen that the proposed algorithms outperform the existing algorithms for this example application. Both algorithms give approximately the same results in this particular example; this is because the best path algorithm already assigns low qualities to the zero-weighted edges. This may be seen more clearly in the 3D isometric plots for frame 24, as shown in Figure 5.27.



**Figure 5.25:** Three different frames of fringe patterns for a patient undergoing radiotherapy treatment for breast cancer. (a) frame 0, (b) frame (15) and (c) frame (24).





**Figure 5.26:** Results showing a region from a female human thorax, taken from a real clinical patient undergoing treatment for breast cancer. Wrapped-phase images of frames 0, 15 and 24 are shown in row (a) . The unwrapped-phase for those frames are shown underneath, using row(b) Cusack's algorithm, row (c) Huntley's algorithm, row (d) the BP algorithm and row (e) the BPASL algorithm.



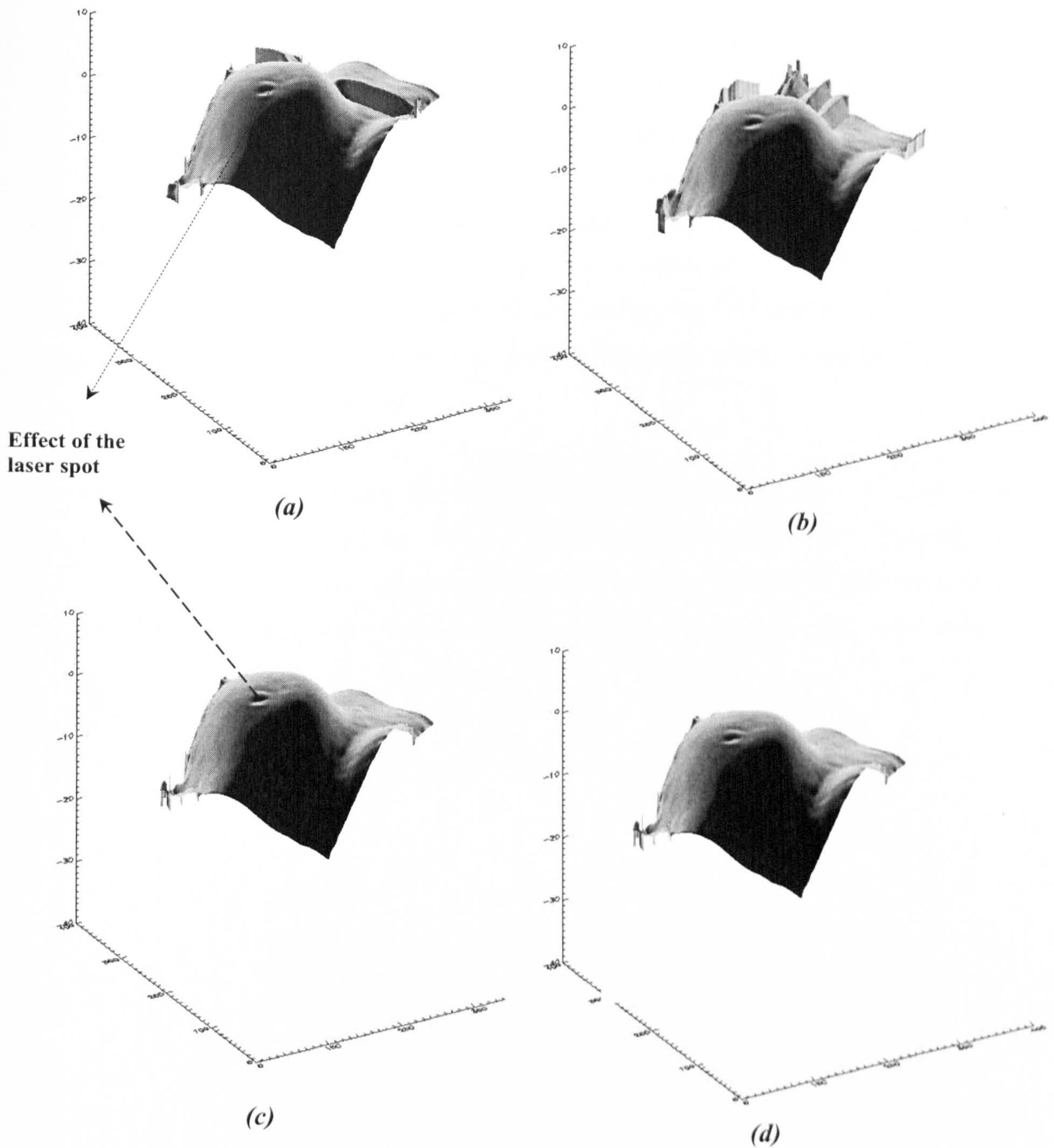


Figure 5.27: A three-dimensional view for the unwrapped-phase maps for frame 24, showing a clinical patient's breast using: (a) Cusack's algorithm, (b) Huntley's algorithm, (c) the BP algorithm and (d) the BPASL algorithm.

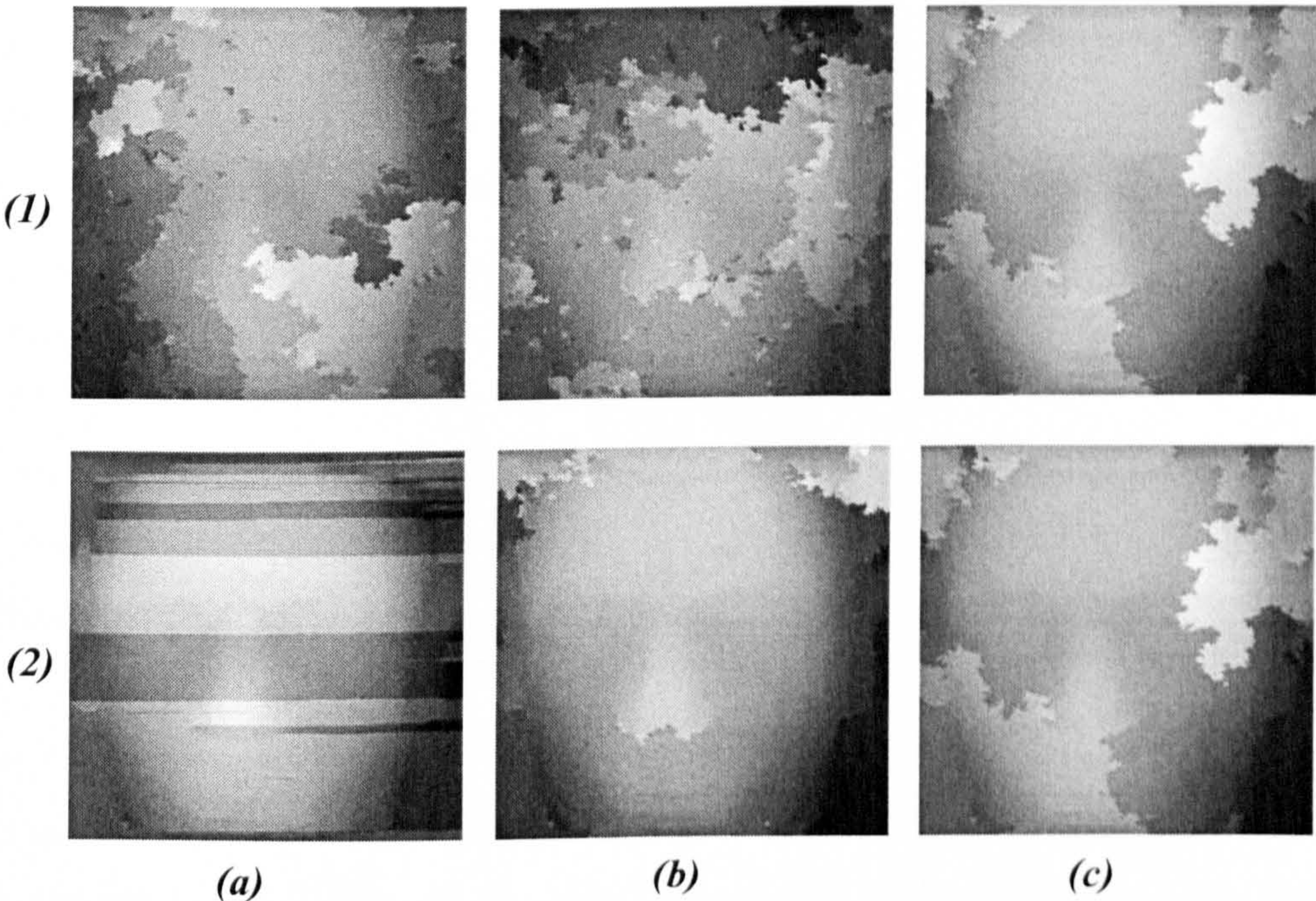


### 5.5.3 The effect of the quality map on the unwrapping path.

Choosing a different quality map leads the proposed algorithm into a different unwrapping path and gives different results. Figure 5.28 shows the effect of choosing different quality maps when unwrapping the noisy dummy's face which was described earlier. This figure shows the unwrapped-phase maps of frame number 15 taken from that wrapped-phase volume. The first row in the figure shows the results of the BP algorithm using different quality maps, while the second row shows the results of the BPASL algorithm using different quality maps. Column (a) in the figure shows the results of using a phase derivative variance quality map. The results obtained when using the maximum gradient and the second difference quality maps are shown in columns (b) and (c) respectively.

As can be seen in the figure, the second-difference quality map gives the best result amongst the other quality maps, when it is applied to guide the BP algorithm. On the other hand, the maximum-gradient quality map is the best map of choice for the case of the BPASL algorithm. There is no definitive rule to clarify which map has to be chosen.





**Figure 5.28:** The effect of quality map on the unwrapping path for a dummy’s face with added noise, row (1) the results of the BP algorithm and row (2) shows the results of the BPASL algorithm. The results of both algorithms using: column (a) phase derivative variance, column (b) the maximum gradient and column (c) the second difference.



#### 5.5.4 Comparisons between 2D and 3D Phase Unwrapping Algorithms.

In this section the author shows the advantages of three-dimensional phase unwrapping algorithms over two-dimensional phase unwrapping algorithms. To explain these advantages, the results of the four three-dimensional phase unwrapping algorithms discussed above are compared with six two-dimensional phase unwrapping algorithms.

All of the six two-dimensional phase unwrapping algorithms are considered to be state of the art in two-dimensional phase unwrapping and they have all been previously used in many applications and have proven their robustness and high performance in many cases. Chapter 3 describes these algorithms in greater detail.

The six two-dimensional algorithms are:

1. Goldstein branch cut phase unwrapping algorithm (Goldstein *et al.*, 1988).
2. Quality guided phase unwrapping algorithm (Bone, 1991).
3. Mask Cut phase unwrapping algorithm (Flynn, 1996).
4. Flynn's minimum discontinuity phase unwrapping algorithm (Flynn, 1997).
5. Herràez's sorting reliabilities phase unwrapping algorithm (Herraez *et al.*, 2002).
6. Least Square (Lp-norm) phase unwrapping algorithm (Ghiglia and Romero, 1996).

Two testing criteria are chosen; the first one is the ability of two-dimensional and three-dimensional phase unwrapping algorithms to unwrap very noisy wrapped phase maps. The second is the ability of two-dimensional and three-dimensional algorithms to unwrap objects which contain sharp discontinuities. Both conditions represent a challenge for all FFA systems and phase unwrapping algorithms.

##### 5.5.4.1 Noise Test.

The test object used here is a complicated steep surface whose height is increasing with time, as described in Appendix A.3. The object movement is represented using 100 frames, with 256×256 pixels in each frame.



Gaussian noise is added to the object's  $256 \times 256 \times 100$  frames as follows:

$$\text{Noise}_{(at\ frame\ t)} = \text{Noise}_{(at\ frame\ t-1)} + 256 \times 256 \text{ Gaussian noise}_{(mean=0, \sigma=0.175)}$$

Supposing that the noise at frame  $t = -1$  is equal to zero.

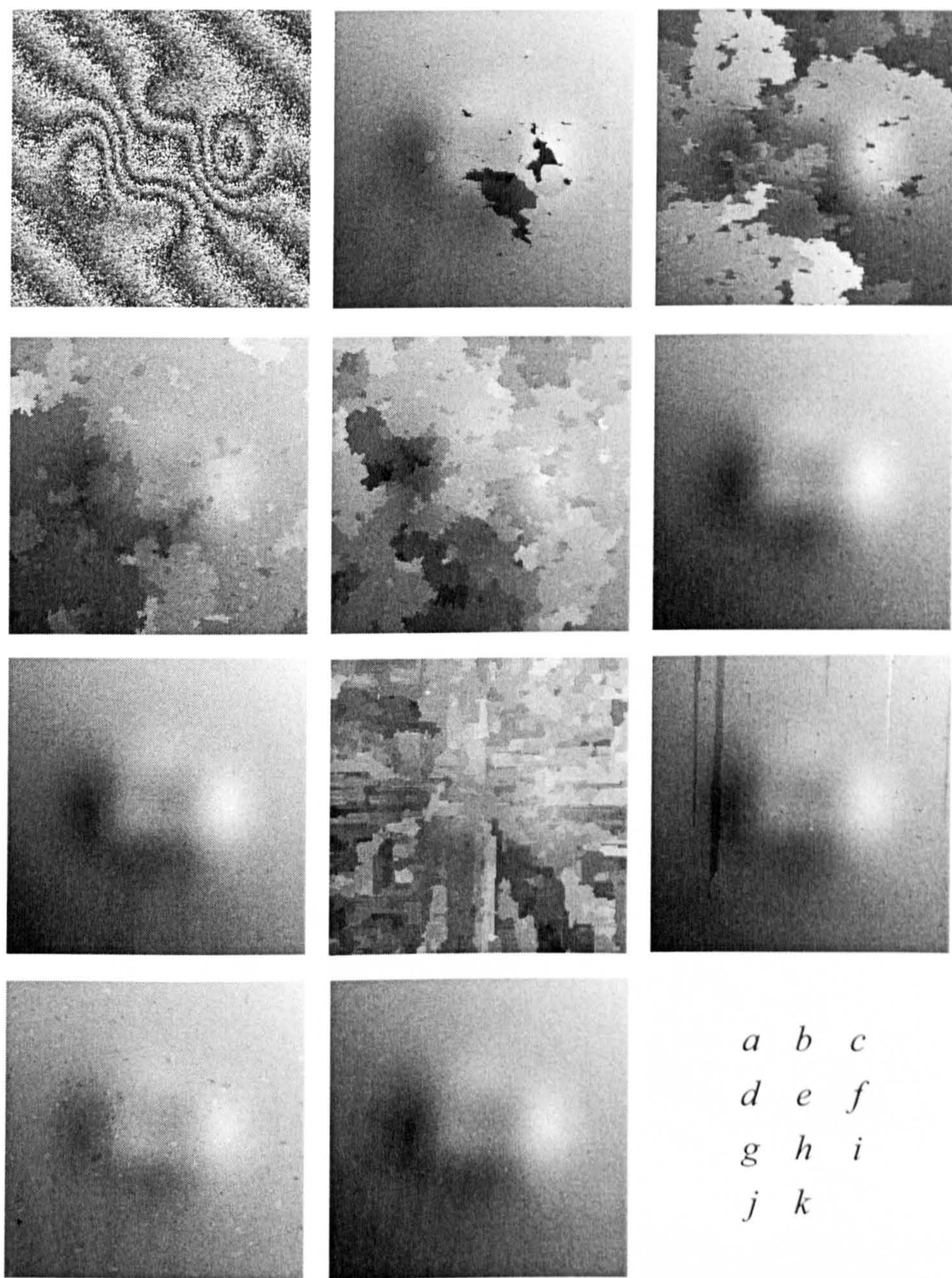
where  $\sigma$  is the standard deviation of the noise distribution.

This means that for this example the noise levels are not constant for each frame, as they were in the previous case, but instead the noise increases from a minimum in the first frame to a maximum level in the final frame of the sequence.

The whole phase volume is subsequently wrapped between the values  $-\pi$  to  $\pi$ , using the arctangent function and is then scaled between black and white for display purposes. The colour white represents the maximum height of the object and the colour black represents its minimum height.

Figure 5.29 shows the results of all unwrappers for frame 40. The wrapped phase of this frame is shown in Figure 5.29(a). The unwrapped phase maps for this frame using the Goldstein algorithm, mask-cut algorithm, quality-guided algorithm, Herràez's algorithm, Flynn's minimum discontinuity algorithm and the Lp-Norm algorithm are shown in Figures (b), (c), (d), (e), (f) and (g) respectively. As shown in the figure, most of these two-dimensional unwrappers fail in unwrapping such noisy wrapped phase. Only two two-dimensional phase unwrapping algorithms surprisingly produce an excellent result for this frame and these algorithms are the Flynn and Lp-Norm algorithms. The results of the three-dimensional phase unwrapping algorithms for this frame are shown in Figures (h) - (k). Cusack's algorithm, which is shown in Figure (h), completely fails in unwrapping this frame. Huntley's and the BP algorithms both produce a good result, shown in Figures (i) and (j) respectively. The BPASL algorithm gives the best result amongst the other three-dimensional algorithms, as is shown in Figure (k). A comparison between Herràez's algorithm, which is the 2D version of the BP algorithm, and the BP algorithm indicates the great benefit of going to the third dimension in order to avoid noise sources and to find more reliable unwrapping paths.



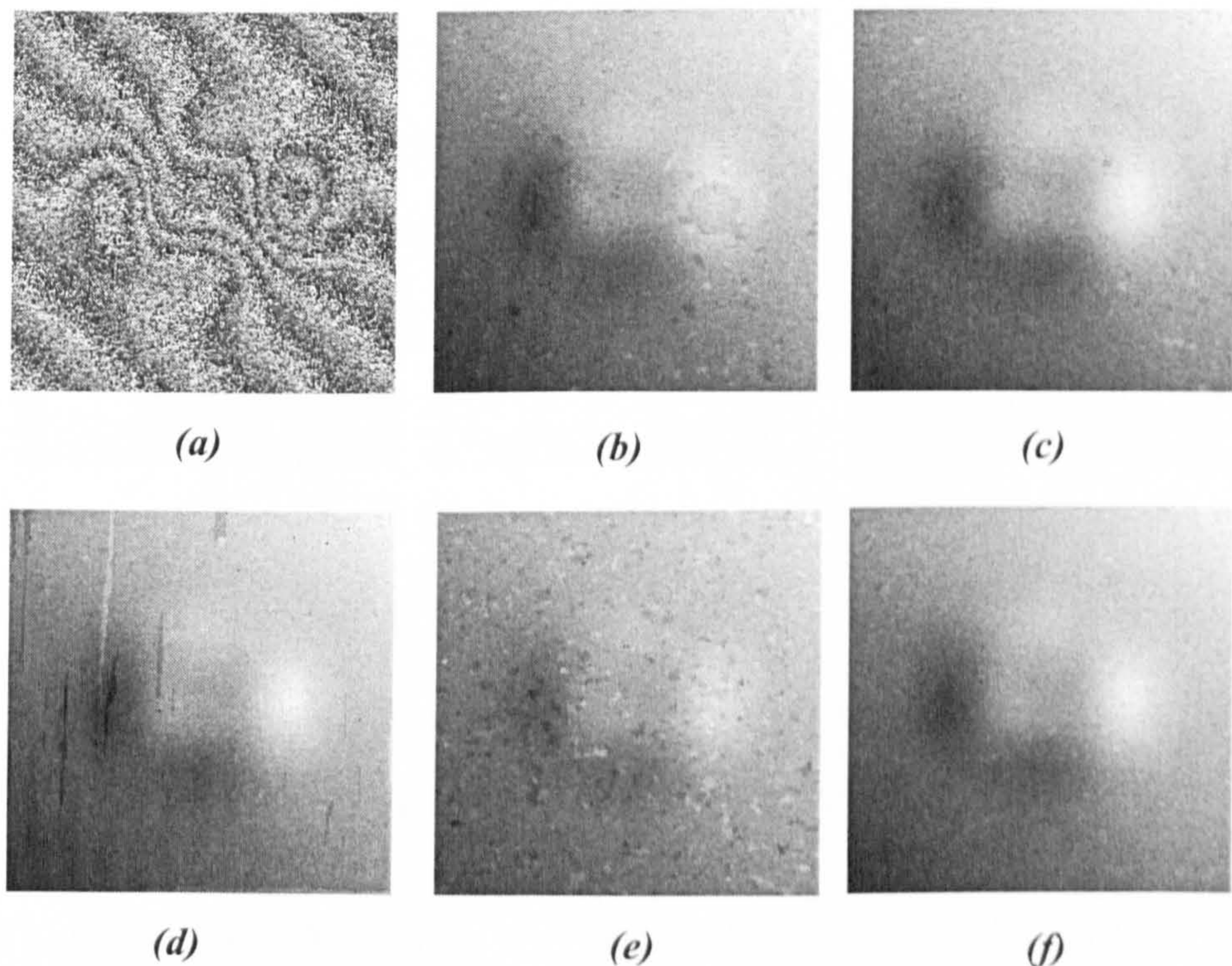


**Figure 5.29: A comparison between 2D and 3D phase unwrapping algorithms in a noisy object. (a) wrapped phase map for frame 40 of the noisy steep object. The unwrapped phase maps resulting form: (b) Goldstein’s algorithm, (c) Mask-cut algorithm, (d) Quality-guided algorithm, (e) Herràez’s algorithm, (f) Flynn’s algorithm, (g) Lp-norm algorithm, (h) Cusack’ s algorithm, (i) Huntley’s algorithm, (j) the BP algorithm and (k) the BPASL algorithm.**



All algorithms that failed to unwrap the previous frame in Figure 5.29 are now excluded from the comparison, the rest of the algorithms are re-considered using even higher levels of noise, as will be discussed subsequently.

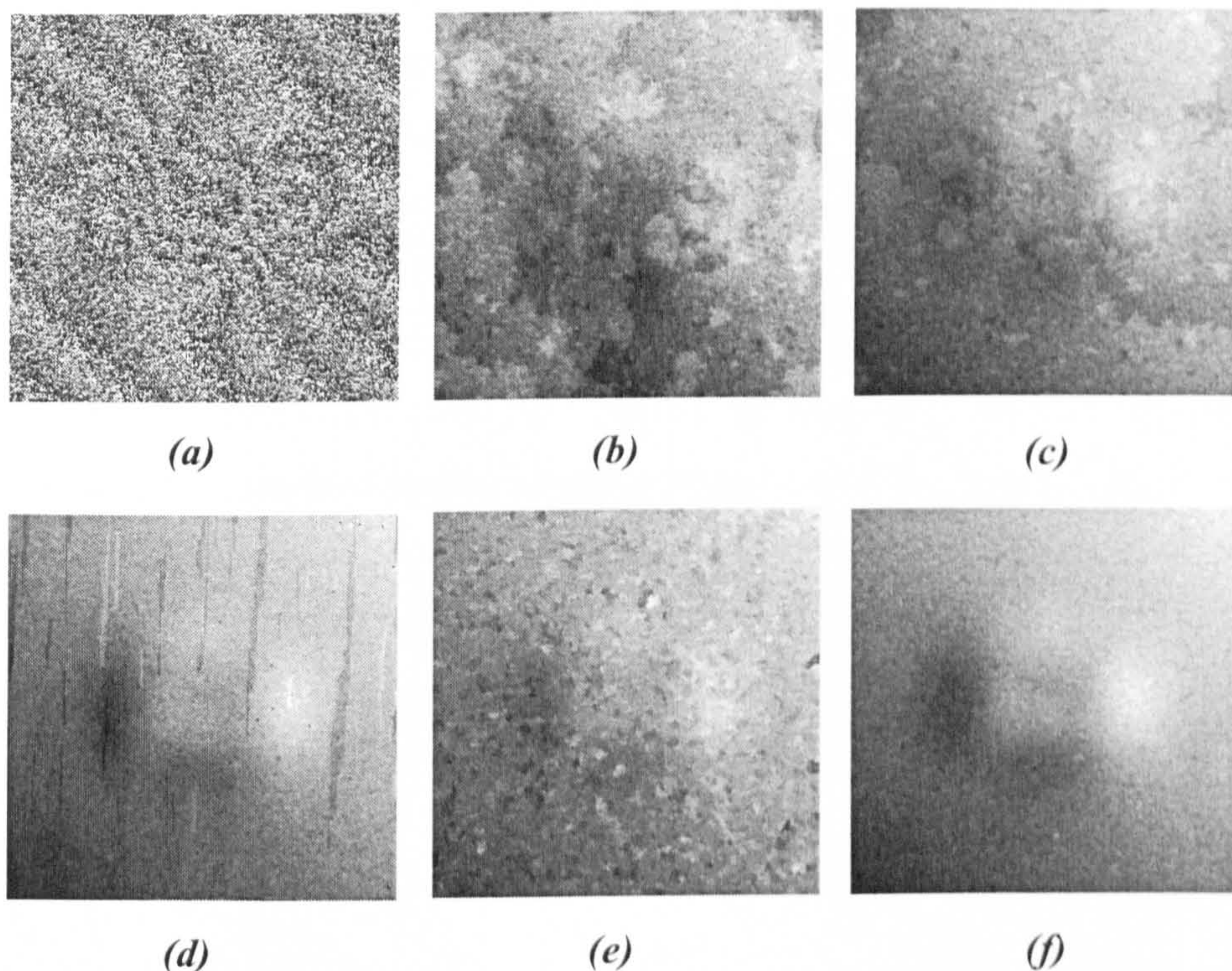
Figure 5.30(a) shows the wrapped phase map of the frame 65. This frame contains higher noise levels than in the previous frame which was shown in Figure 5.29. Unwrapping such a highly noisy frame is more difficult task than that encountered when unwrapping the frame in the previous example. Flynn's algorithm gives a good result in unwrapping this frame, as shown in Figure 5.30(b). The  $L^p$ -norm algorithm shows outstanding performance here and gives an excellent result, as shown in Figure 5.30(c). Huntley's algorithm also produces a good result for this frame, as is shown in Figure (d). The BP algorithm gives an acceptable result, as the object still can be seen behind the noise areas, as shown in Figure 5.30(e). The  $L_p$ -norm and the BPASL algorithms take the lead amongst all the other algorithms and produces the best overall results, as shown in Figures 5.30(c) and 5.30(f) respectively.



**Figure 5.30: A comparison between 2D and 3D phase unwrapping algorithms in a noisy object. (a) wrapped phase map for frame 65. The unwrapped phase maps resulting from: (b) Flynn's algorithm, (c)  $L_p$ -norm algorithm, (d) Huntley's algorithm, (e) the BP algorithm and (f) the BPASL algorithm.**



Testing the algorithm in more noisy frames shows that the BPASL outperforms all of two dimensional and three-dimensional phase unwrapping algorithms. Figure 5.31 (a) shows the wrapped phase map of the last frame, which experiences the highest level of noise. At this level of noise, both Flynn's and the Lp-norm algorithms fail to unwrap the frame due to its high noise levels, as shown in Figures (b) and (c) respectively. Huntley's algorithm still gives a very good result, with the exception of a minor error which has been propagated, as shown in Figure (d). The BP algorithm gives a poor result at this stage, whereas the BPASL still gives the best result amongst all the other algorithms, as shown in Figures (e) and (f) respectively.



**Figure 5.31: A comparison between 2D and 3D phase unwrapping algorithms in a noisy object. (a) Wrapped phase map for frame 99 of the noisy steep object. The unwrapped phase maps resulting from: (b) Flynn's algorithm, (c) Lp-norm algorithm, (d) Huntley's algorithm, (e) the BP algorithm and (f) the BPASL algorithm.**

#### 5.5.4.2 Discontinuously Test.

The synthetic object used in this test is a planar surface whose height is increasing with time. The object contains sharp discontinuous edges, as described in Appendix A.4. The object has 100 frames, with  $256 \times 256$  pixels in each frame. All the frames have been wrapped between  $-\pi$  and  $\pi$  using the arctangent function. The wrapped volume is



unwrapped using the previously mentioned two-dimensional phase unwrappers on a frame by frame basis, and is also unwrapped as a phase volume using the three-dimensional phase unwrappers.

Figure 5.32 shows the unwrapping results of all algorithms for frame number 65. Obviously, all algorithms easily detect the three surfaces and unwrap them perfectly. It can be seen that the wrapped phase in Figure 5.32(a) has three different regions, so two-dimensional phase unwrappers detect these regions easily. Note that for the case of the wrapped phase map that is shown in Figure 5.32(a), it can be seen that the phase wraps are easily differentiated across all three regions of the object. Despite the fact that the two-dimensional phase unwrapping algorithms detect three different surfaces, they fail to determine the exact difference between the levels of these surfaces. This can be seen by the low contrast between the three surfaces in Figures 5.32(b)- 5.32(e).

Figure 5.33 shows the unwrapped phase for the line 128 of frame 65. Figure 5.33(a) shows the phase distribution along this line for the original object. The results of the two-dimensional phase unwrapping algorithms are shown in Figures 5.33(b) – 5.33(e), whereas the results for the three-dimensional phase unwrapping algorithms are shown in Figures 5.33(f) – 5.33(i). This figure demonstrate that all the two-dimensional phase unwrapping algorithms fail to detect the original phase this is in the contrary of the three-dimensional phase unwrapping algorithms which successfully retrieve the phase.



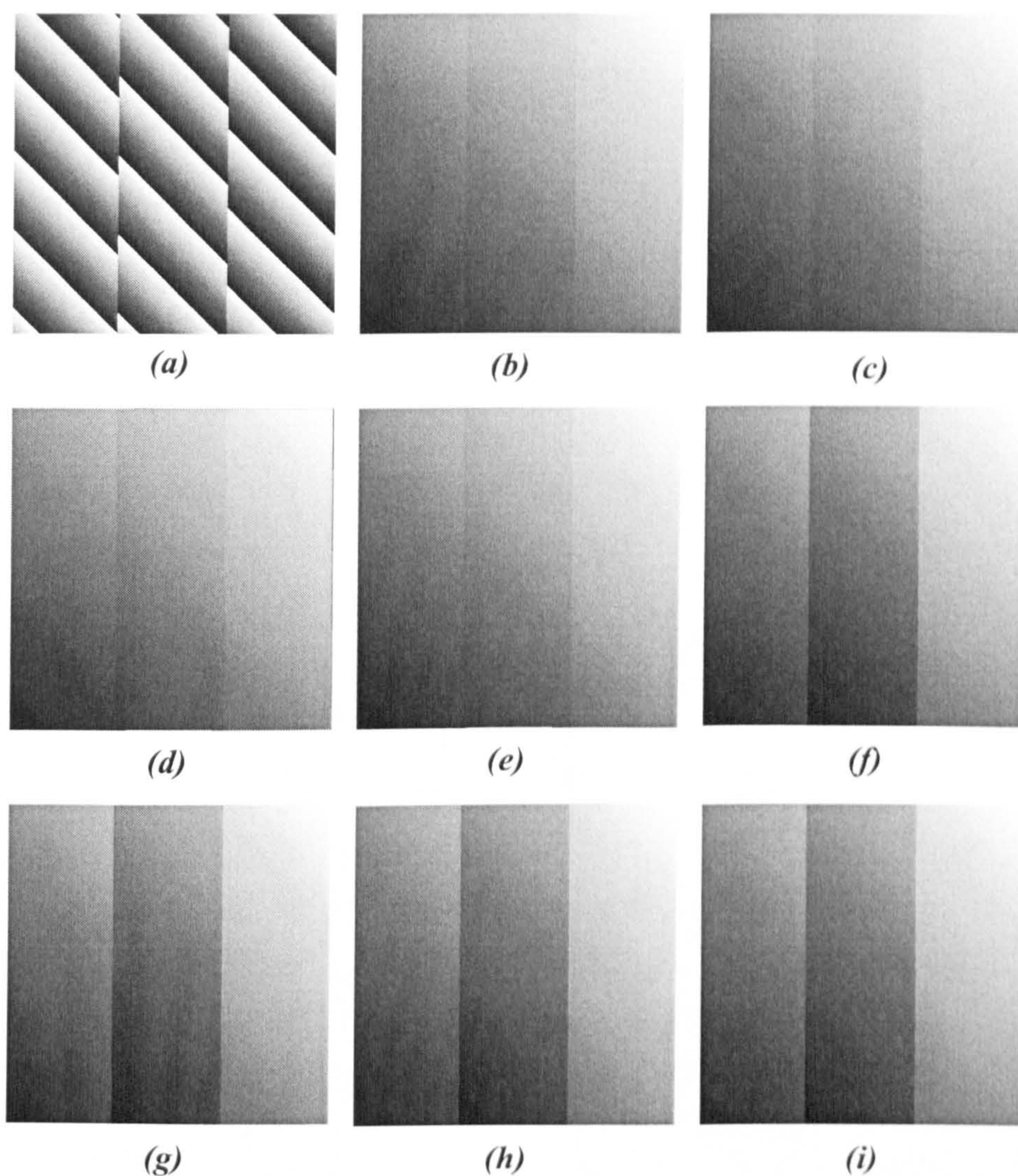
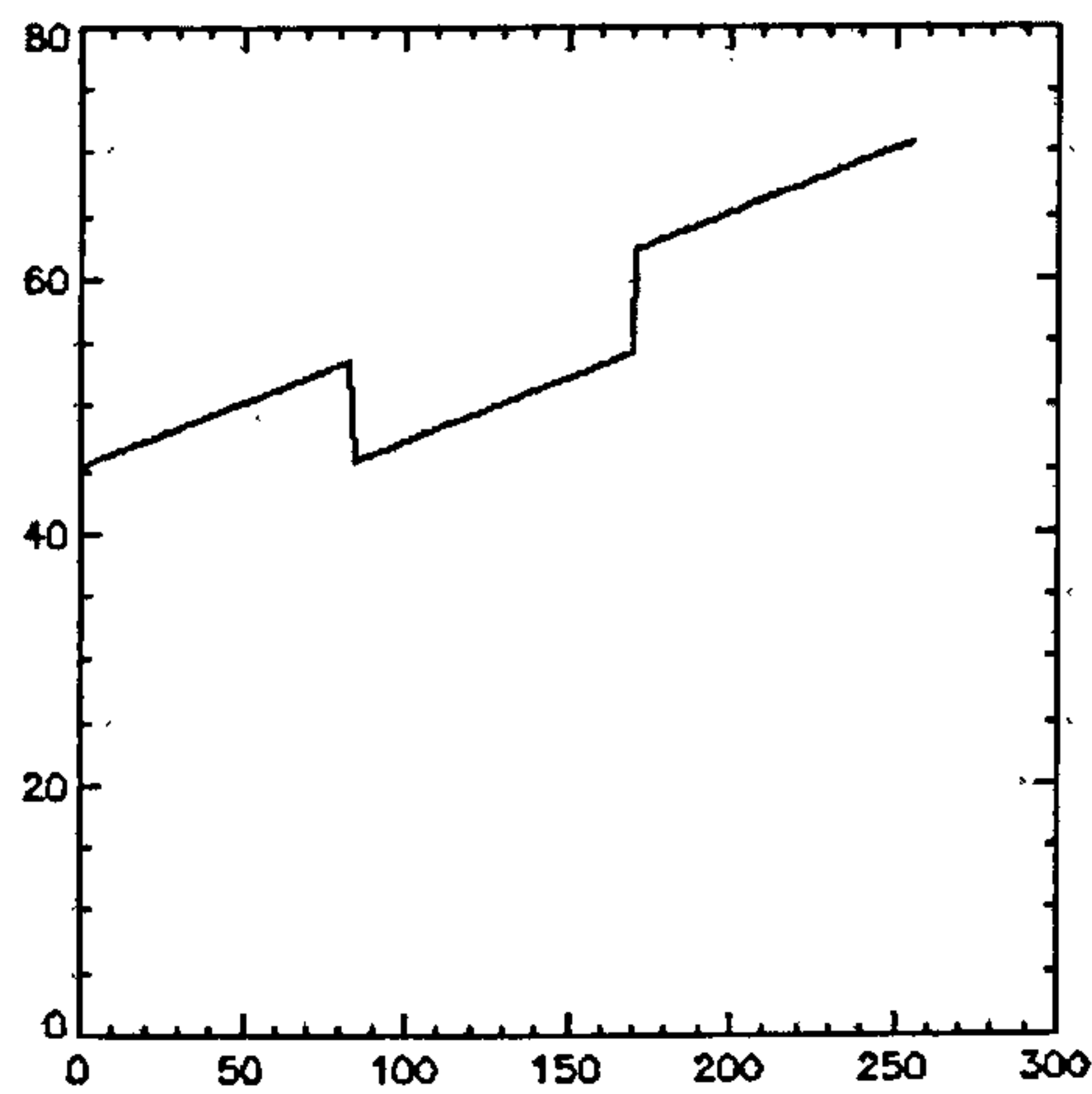
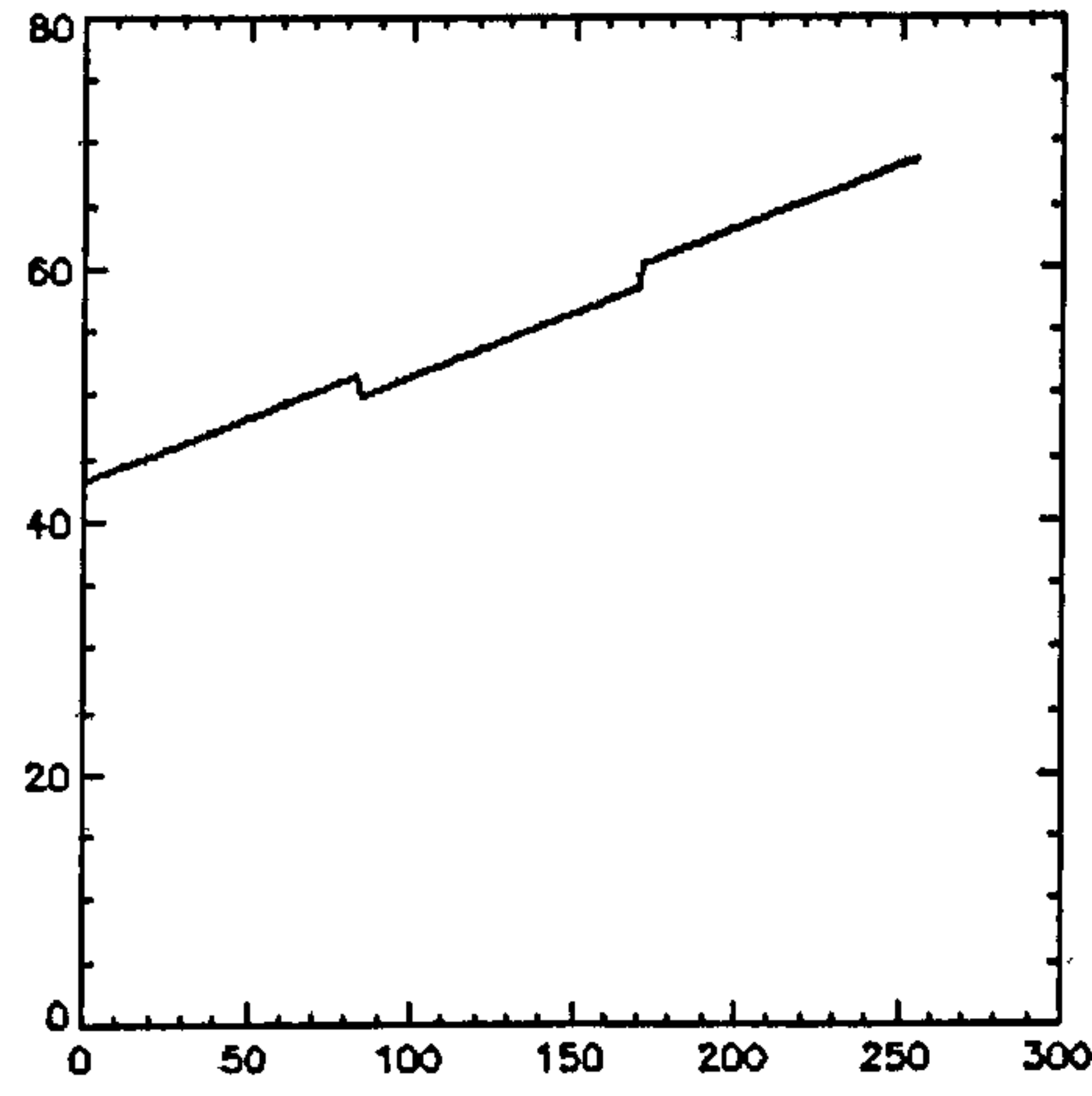


Figure 5.32: A comparison between 2D and 3D phase unwrapping algorithms in a discontinuity test. (a) wrapped phase map for frame 65 of the sharp-edges object. The unwrapped phase maps resulting from: (b) Goldstein's algorithm, (c) Quality-guided algorithm, (d) Flynn's algorithm, (e) Herràez's algorithm, (f) Cusack's algorithm, (g) Huntley's algorithm, (h) the BP algorithm and (i) the BPASL algorithm.

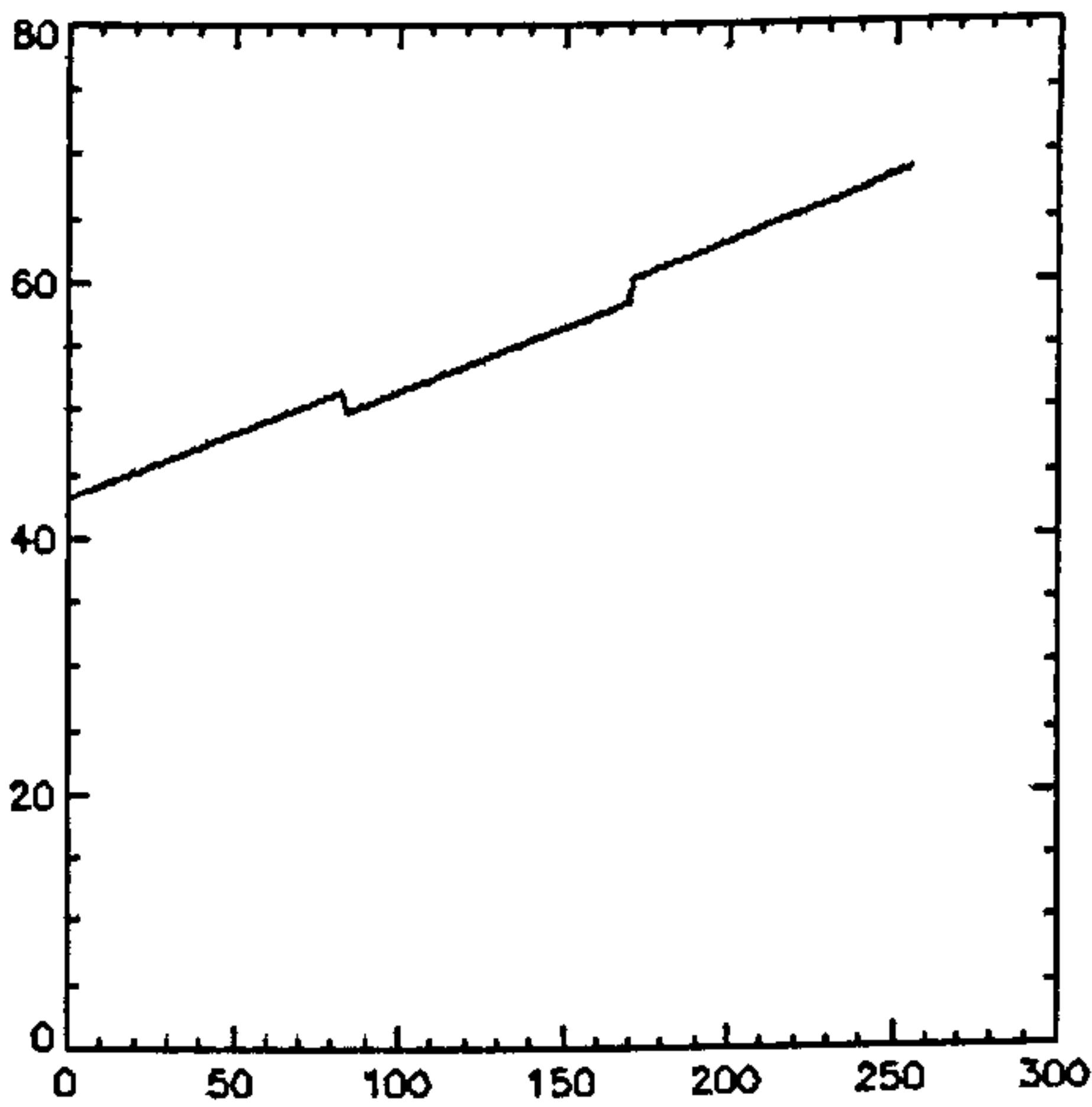




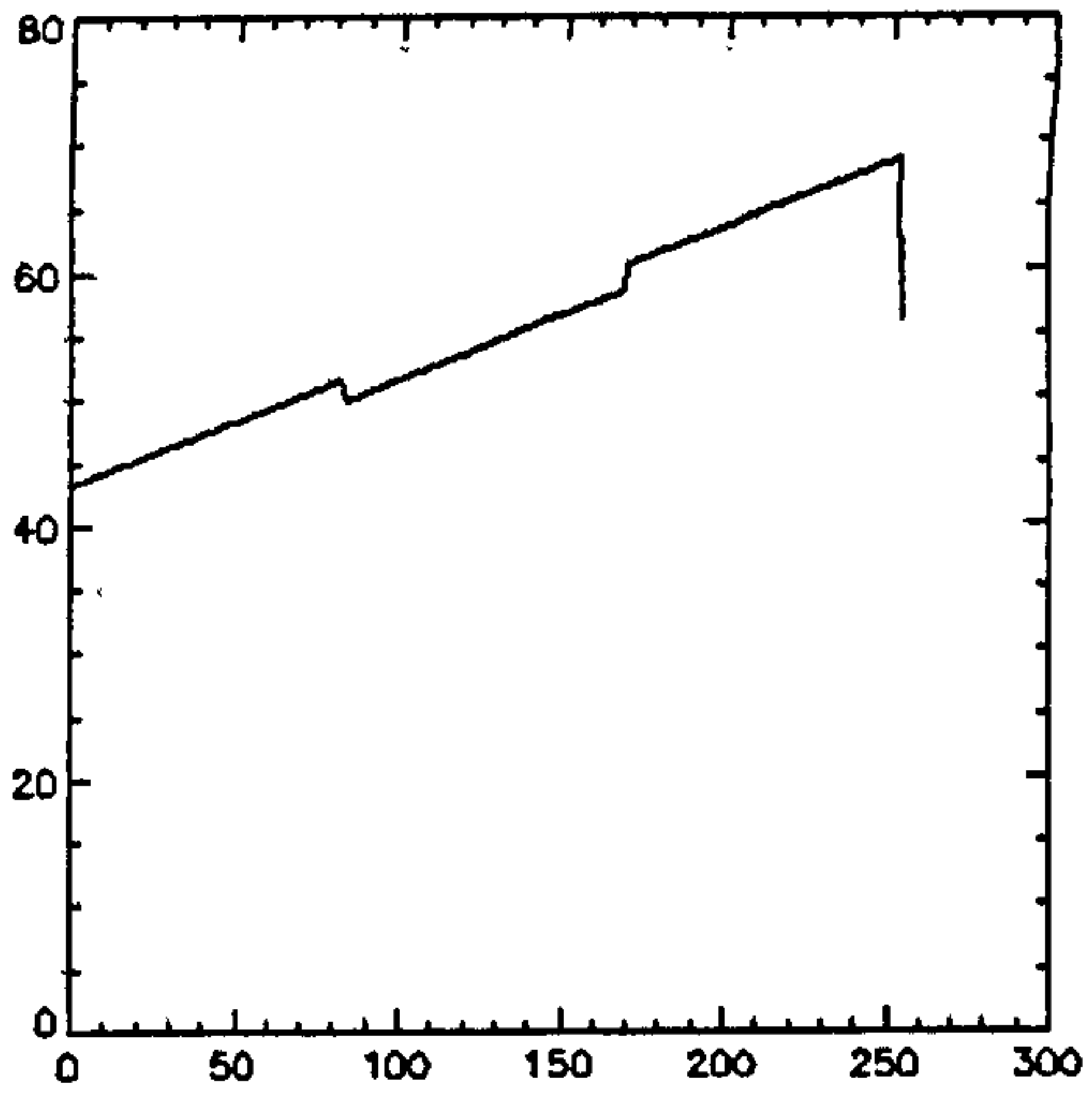
(a)



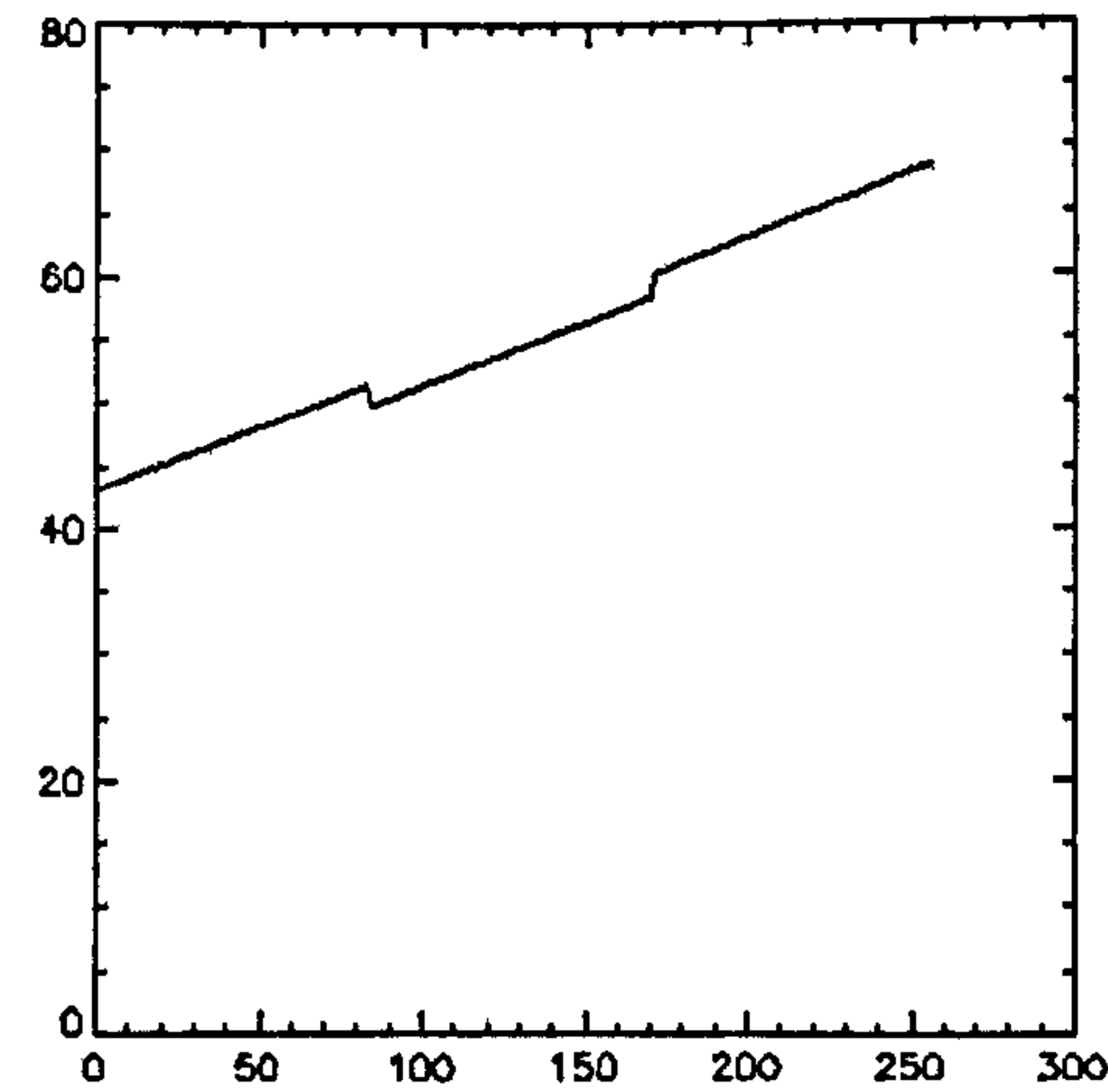
(b)



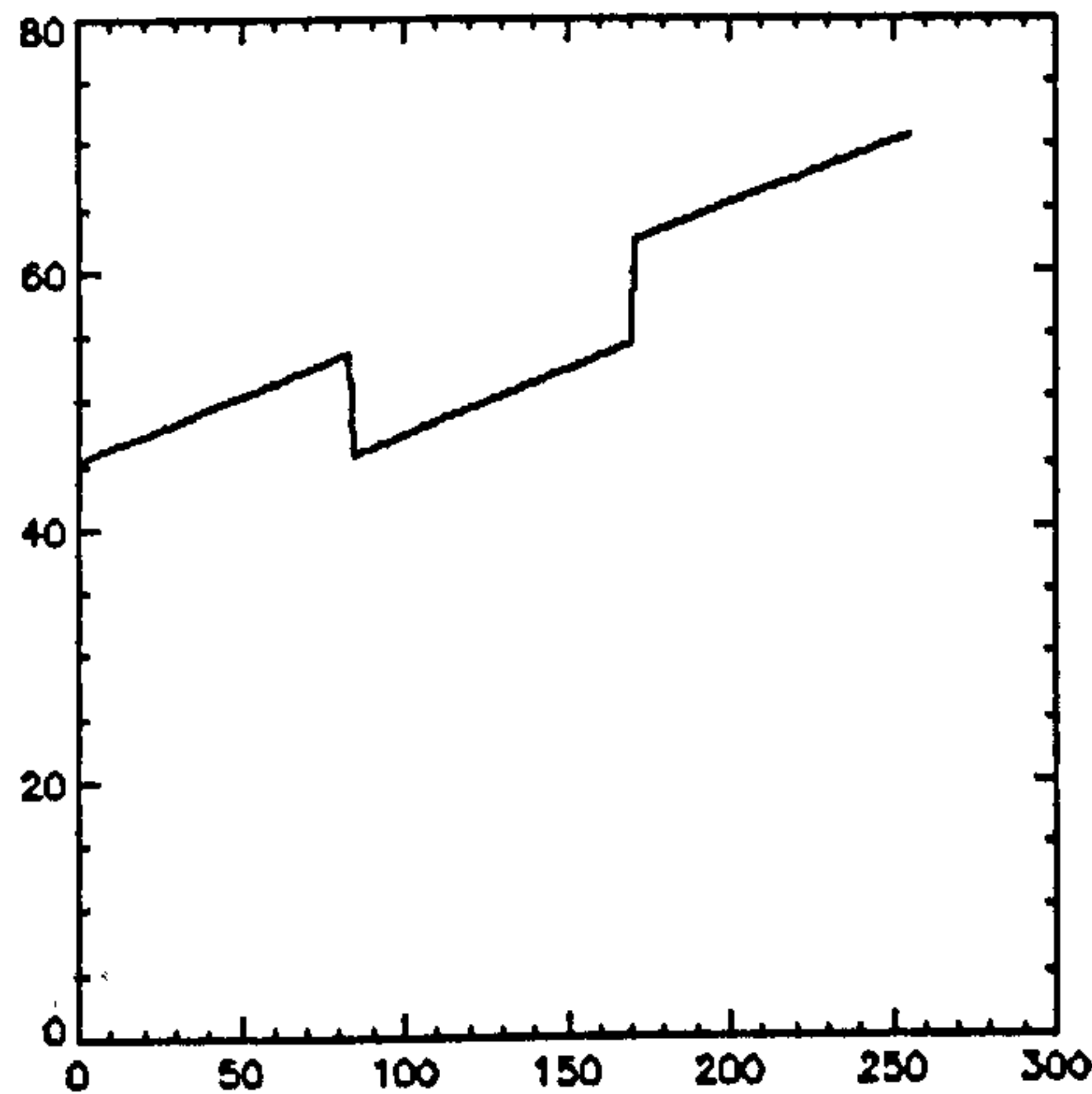
(c)



(d)

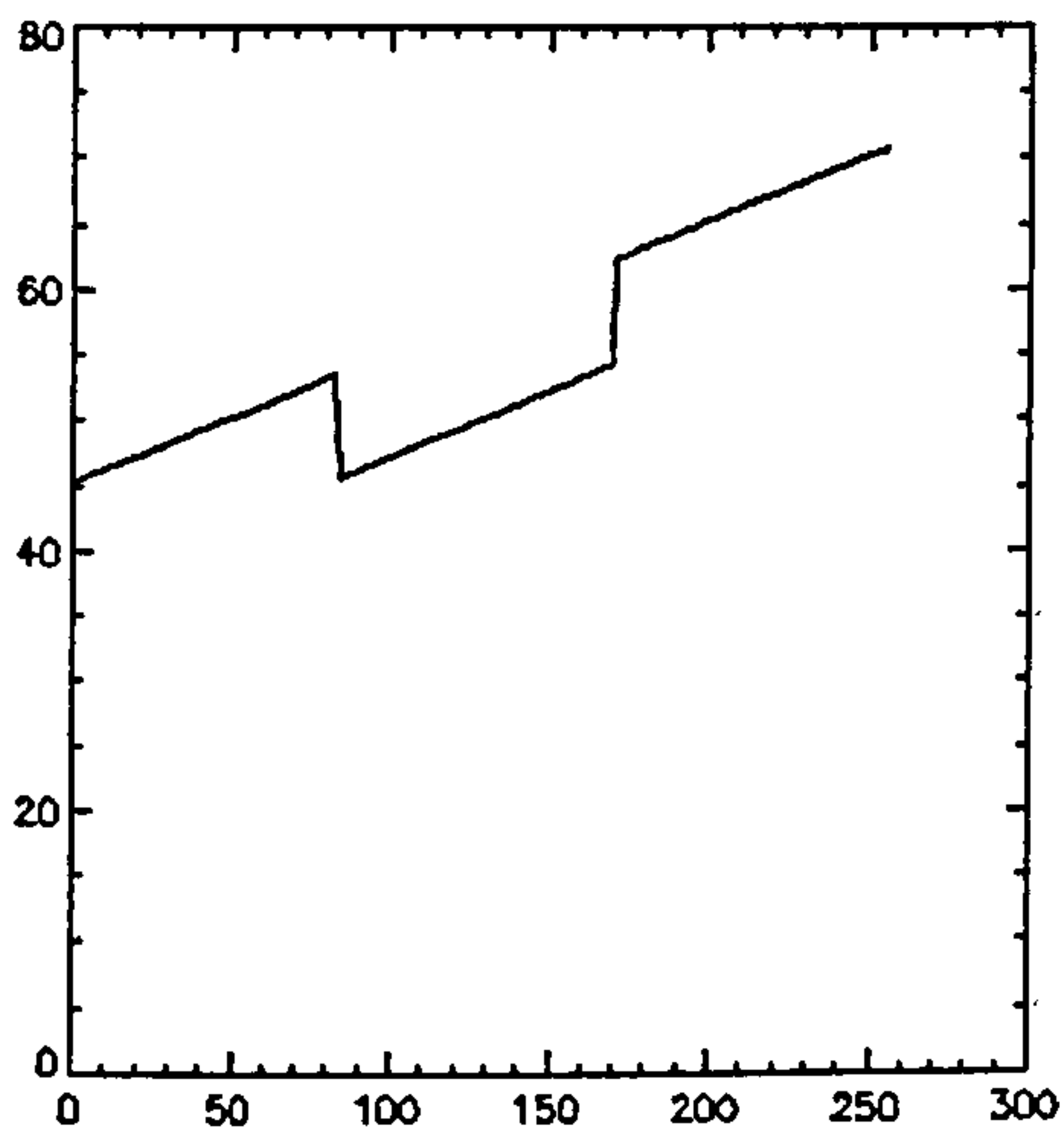


(e)

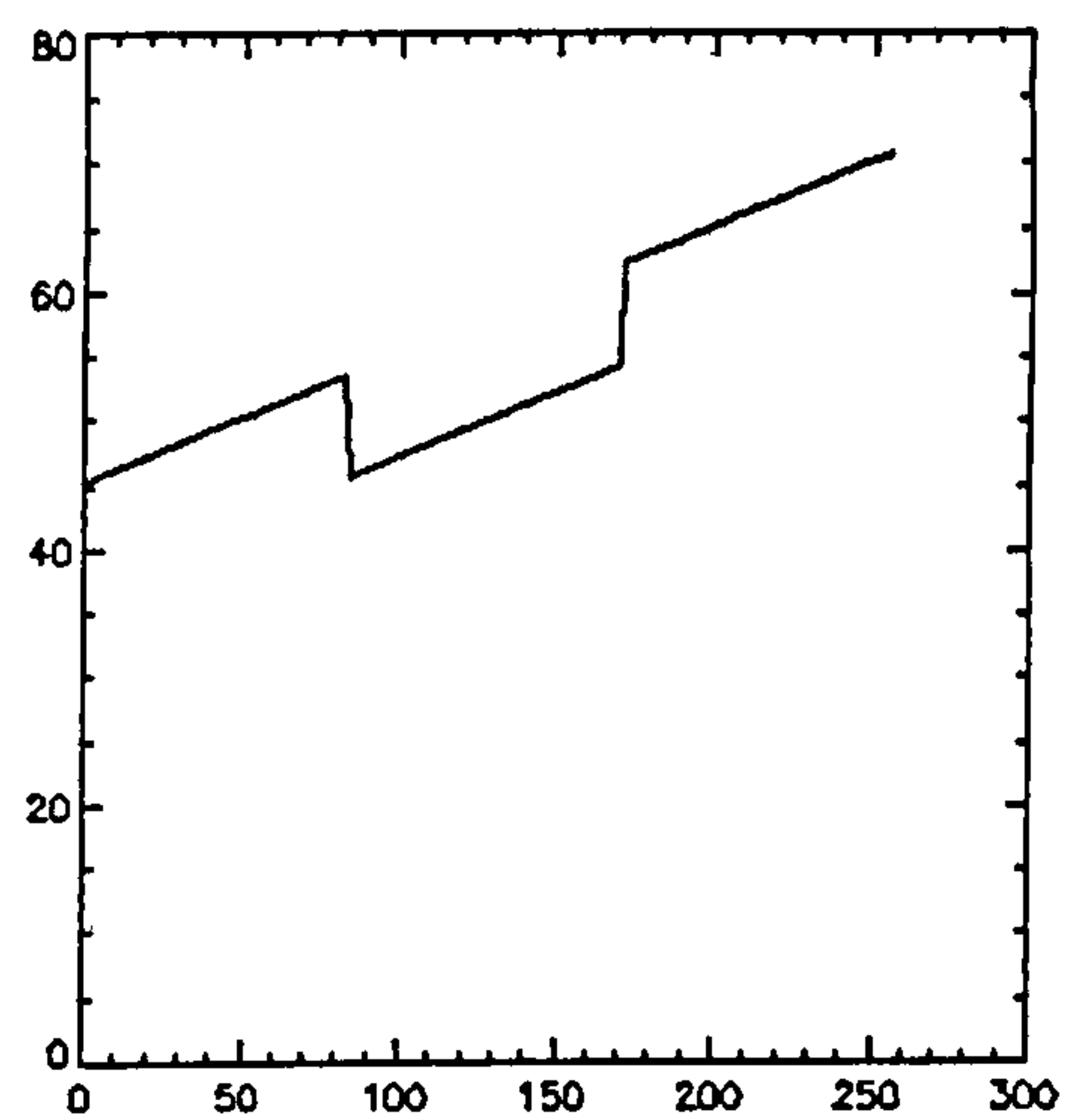


(f)

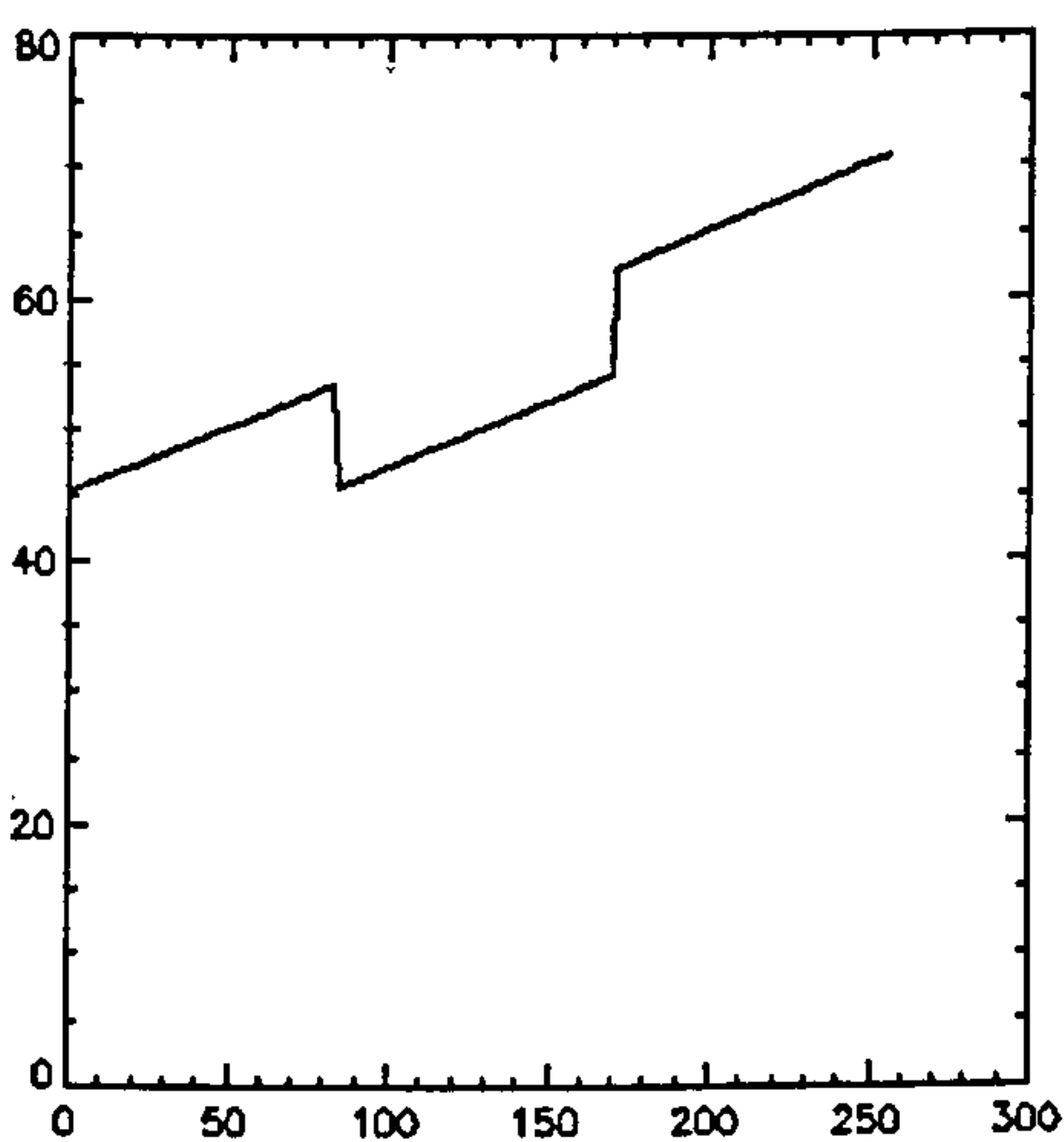




(g)



(h)



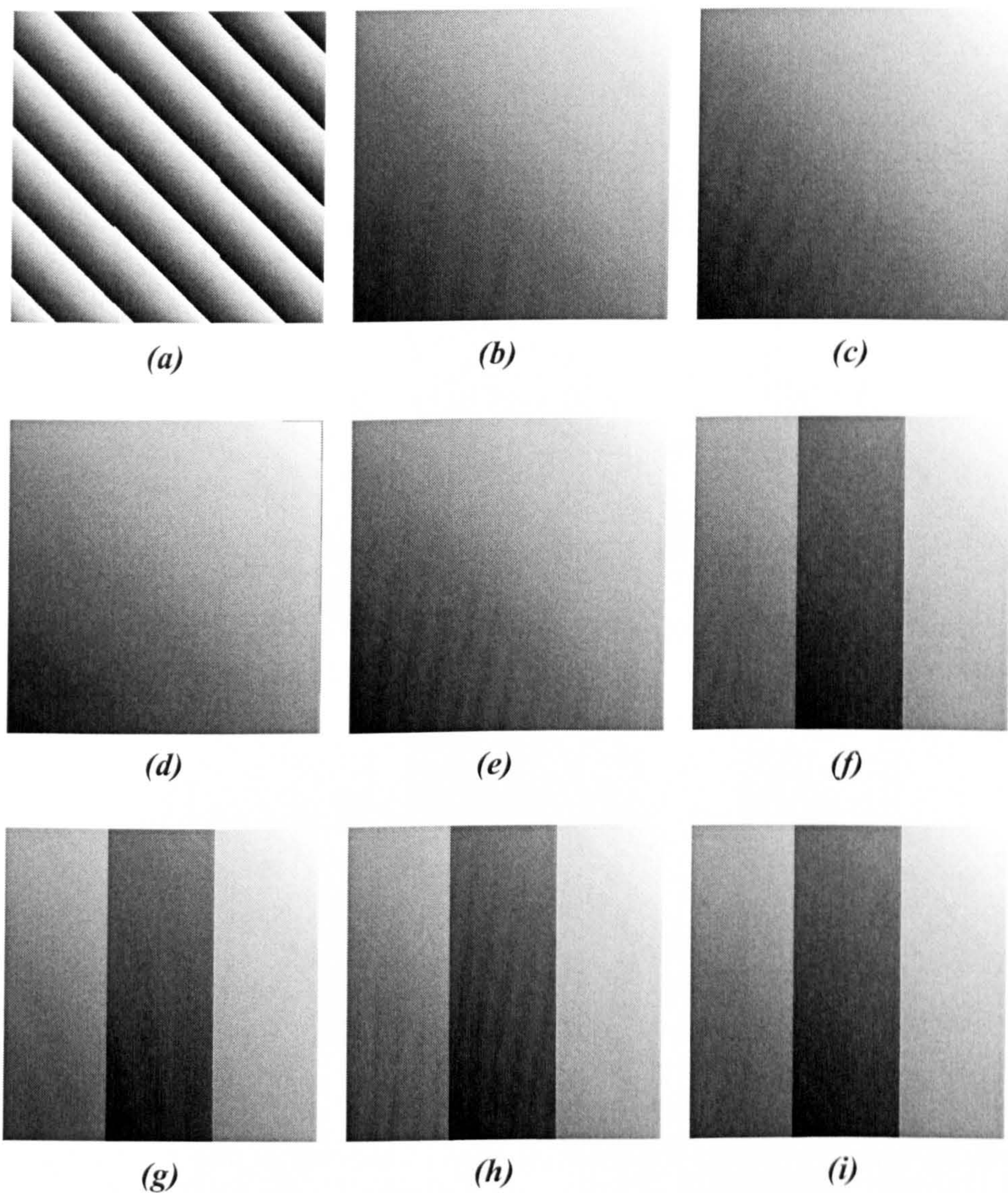
(i)

**Figure 5.33: A comparison between 2D and 3D phase unwrapping algorithms in a discontinuity test. (a) Line number 128 in the 65<sup>th</sup> frame of the sharp-edges object. The unwrapped phase for line 128 in for that frame resulting from: (b) Goldstein's algorithm, (c) Quality-guided algorithm, (d) Flynn's algorithm, (e) Herràez's algorithm, (f) Cusack's algorithm, (g) Huntley's algorithm, (h) the BP algorithm and (i) the BPASL algorithm.**

As the object is growing, the wrapped phase may appear as almost straight lines with almost undetectable discontinuities for two-dimensional phase unwrappers, as is shown in Figure 5.34 below. In this figure  $t = 87$ , and the wrapped phase almost appears as a single surface. All two-dimensional unwrappers completely failed to unwrap this case, and no single algorithm could detect the three individual surfaces. This is not the case



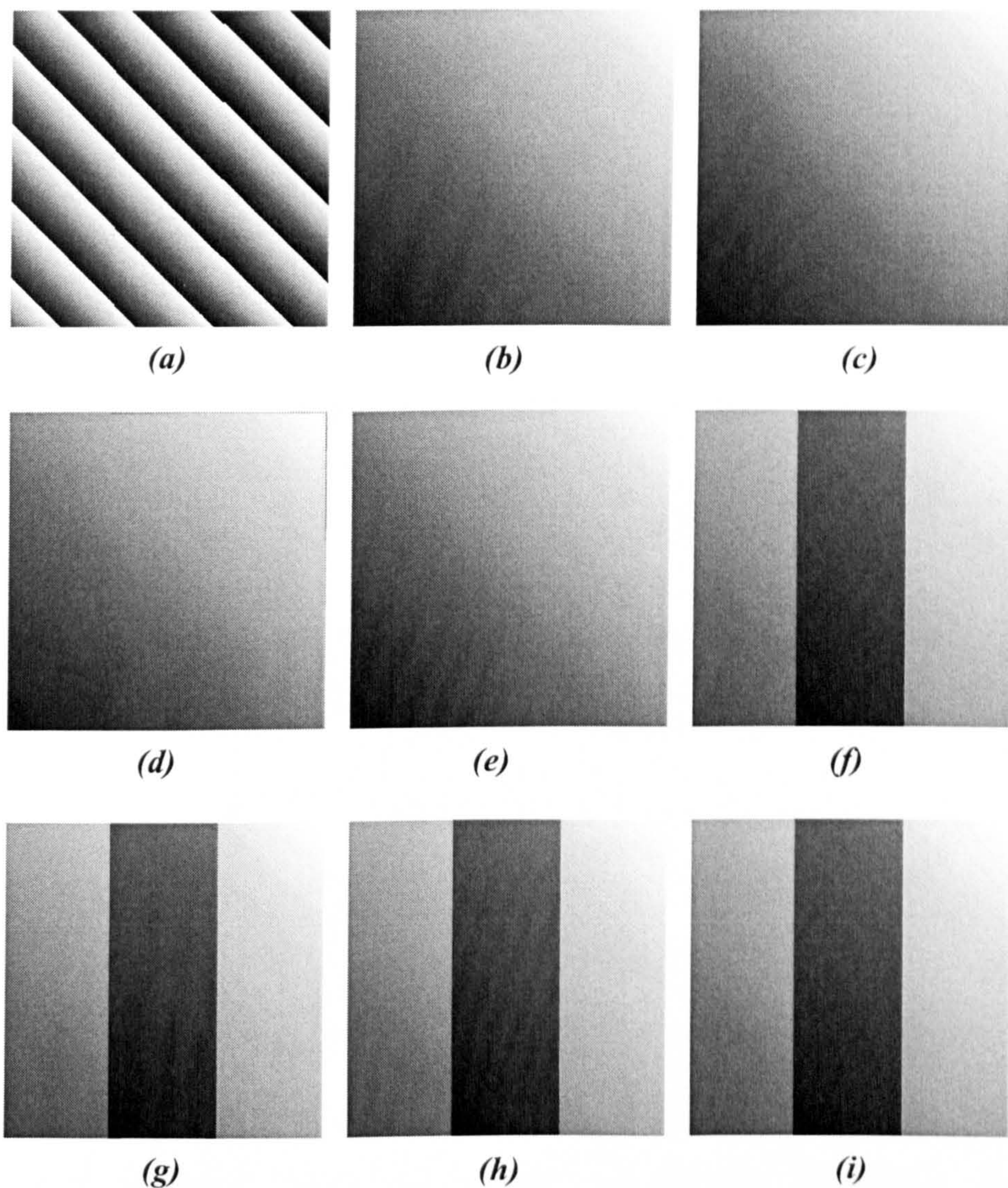
for the three-dimensional phase unwrappers, as is shown in Figures 5.34(g) and 5.34(h); the three-dimensional unwrappers successfully detect three separate surfaces.



**Figure 5.34: A comparison between 2D and 3D phase unwrapping algorithms in a discontinuity test. (a) wrapped phase map for frame 87 of the sharp-edges object. The unwrapped phase maps resulting from: (b) Goldstein's algorithm, (c) Quality-guided algorithm, (d) Flynn's algorithm, (e) Herràez's algorithm, (f) Cusack's algorithm, (g) Huntley's algorithm, (h) the BP algorithm and (i) the BPASL algorithm.**

Another example at  $t=99$  is shown in Figure 5.35, where the discontinuity lines are harder to detect. Again both three-dimensional unwrappers successfully detect the three surfaces, whilst all two-dimensional unwrappers failed to do this.





**Figure 5.35:** A comparison between 2D and 3D phase unwrapping algorithms in a discontinuity test. (a) Wrapped phase map for frame 99 of the sharp-edges object. The unwrapped phase maps resulting from: (b) Goldstein's algorithm, (c) Quality-guided algorithm, (d) Flynn's algorithm, (e) Herràez's algorithm, (f) Cusack's algorithm, (g) Huntley's algorithm, (h) the BP algorithm and (i) the BPASL algorithm.



Table 5.1 list the root mean square (RMS) error generated from the flood-fill, BP, BPASL, Cusack's and Huntley's algorithms. This RMS error is calculated between the original phase volumes and the unwrapped phase volumes for each computer simulated object used in this chapter as described by the following equation:

$$RMS(i, j, k) = \sqrt{\frac{\sum (unwrapped(i, j, k) - object(i, j, k))^2}{VolumeSize}} \quad (5.30)$$

**Table 5.1.** The RMS error for the BP algorithm, BPASL algorithm, Cusack's algorithm and the 3D noise immune algorithm (Huntley's algorithm).

Object	Flood Fill algorithm	The BP algorithm.	The BPASL algorithm	Huntley's algorithm.	Cusack's algorithm.
hemisphere	8.5844	0.9624	0.9675	1.2016	11.8580
Object with partial noise	4.3044	0.9997	0.7889	1.5181	6.9834
Object with increasing noise	6.0933	2.5125	1.3766	1.8705	7.8201
Three surfaces object	0.0147	$1.005 \times 10^{-5}$	$1.005 \times 10^{-5}$	0.0098	$1.003 \times 10^{-5}$

From the previous table, we can see the BP and BPASL algorithms give the least RMS error for the hemispherical object. Cusack's algorithm gives the worst result with the highest error. Huntley's algorithm gives good result where as the flood fill algorithm gives a poor result. These results are exactly matching the results shown in Figure 5.18.

For the second generated object, where the noise was added partial to some frames, we can also see that the BPASL has the lead among all others as it gives the lowest RMS error. Regarding the third simulated object, where the noise was increasing from frame to frame, the BPASL still has the least error. Huntley's algorithm gives a very good result in recovering this object where it comes directly after the BPASL algorithm. These results again match the results in Figures 5.29 – 5.31. In the case of the three-surface object, all 3D algorithms manage to recover the phase with very small amount of RMS error.



The previous discussion and Table 5.1 demonstrate the fantastic robustness of the BPASL as it is indeed outperforms other algorithms.

## 5.6 Summary.

In this chapter, two novel three-dimensional phase unwrapping algorithms, *i.e.* the Best Path (BP) and the Best Path Avoiding Singularity Loops (BPASL) algorithms, were proposed. The BP algorithm attempts to unwrap the wrapped phase volume by finding an optimal unwrapping path. The optimal unwrapping path is derived by calculating the quality of each voxel in the wrapped phase volume. The BP algorithm is found to be very robust and it manages to prevent error propagation throughout the phase volume. The disadvantages of this algorithm is that it does not take into account the effect of singularity loops during unwrapping, which may reduce the robustness of the algorithm. This algorithm has been tested on both simulated and real wrapped phase volumes. It was found that this algorithm is capable of unwrapping noisy objects and produces good results in these conditions.

The BPASL algorithm is a very powerful algorithm that combines the 3D noise-immune phase unwrapping algorithm proposed by Huntley in 2001, with the BP algorithm proposed by the author. The BPASL algorithm finds an optimal unwrapping path whilst also taking into account the effect of singularity loops. The concept of zero-weighted edges are introduced in order to adjust the optimal path and avoid these singularity loops. The algorithm has been tested on many objects and indeed it shows high robustness and high immunity against noise.

Both algorithms have been compared with two state-of-the art, robust, three-dimensional phase unwrapping algorithms (Huntley, 2001; Cusack and Papadakis, 2002). The proposed algorithms indeed outperform these two algorithms and produce very good results.

Also, a comparison between three-dimensional and two-dimensional phase unwrapping algorithms has been carried out, in terms of both robustness against noise and the capability to unwrap objects containing sharp edges. The comparison shows that three-



dimensional phase unwrapping algorithms are more robust against noise than their two-dimensional counterparts. Also, three-dimensional phase unwrapping algorithms were able to correctly detect the location of sharp edges in an object, whereas all the two-dimensional phase unwrapping algorithms failed in this regard. The success of the three-dimensional phase unwrapping algorithms in noisy and sharp discontinuity is due to the fact that three-dimensional phase unwrapping algorithms have plenty of choices of the unwrapping paths, so the avoiding errors and noisy regions become easier in the 3D case than the 2D case. Furthermore, 3D phase unwrapping algorithms unwrap a certain frame with the knowledge of the previous and next frames where the 2D algorithms have a huge lack in this regard. So in short, the key to why 3D phase unwrapping algorithms is better than 2D phase unwrapping algorithms is that the overall signal to noise ratio (SNR) in the wrapped phase volume is better than the SNR for certain frames.

A comparison between the best path (BP), the best path avoiding singularity loops (BPASL), flood-fill, Huntley's and Cusack's algorithms has been also carried out in this chapter. Comparison demonstrates that the BPASL has indeed managed to recover the unwrapped phase volume successfully with the minimum amount of RMS error. This comparison also demonstrates the robustness of the BPASL algorithm as it outperforms all other algorithms.

The best path (BP) algorithm, the best path avoiding singularity loops (BPASL) algorithm and the three-dimensional noise immune algorithm have been coded and implemented by the author using the C programming language and compiled using an Intel® 9.1 compiler. For comparison purposes, a precompiled executable program for Cusack's algorithm was obtained from Cusack himself. These algorithms have been executed on a PC with a Pentium 4 processor running at a 3.2 GHz clock speed. The memory of the PC is 4G-Byte RAM. The execution time of the proposed algorithm varies from volume to volume and depends on the particular phase distribution being analyzed. The execution times for simulated and real volumes discussed earlier for each algorithm are listed in Table 5.2.

A source code in C language for all two-dimensional phase unwrapping algorithms discussed earlier in this chapter was obtained from the Two-Dimensional Phase



Unwrapping book by Ghiglia *et al.* and a complete comparison between these algorithms in term of robustness and execution times can be found in their book (Ghiglia and Pritt, 1998).

**Table 5.2.** The execution times in seconds for the BP algorithm, BPASL algorithm, Cusack’s algorithm and the 3D noise immune algorithm (Huntley’s algorithm).

Object	Size	The BP algorithm.	The BPASL algorithm	Huntley’s algorithm.	Cusack’s algorithm.
Simulated hemisphere	256×256×100	20.47	115.13	35.85	23.24
Mannequin’s chest	512×512×25	17.81	90.03	22.45	20.86
RANDO Dummy’s face	512×512×25	17.68	85.63	21.23	21.81
Clinical patient	512×512×25	18.28	96.28	24.45	24.45



**References:**

- Abdul-Rahman, H.; Gdeisat, M.; Burton, D. and Lalor, M. (2005) Fast Three-Dimensional Phase-Unwrapping Algorithm Based on Sorting by Reliability Following a Non-Continuous Path. 1 ed. Munich, Germany, SPIE-Int. Soc. Opt. Eng. 5856 part 1, 32-40.
- Abdul-Rahman, H. S.; Gdeisat, M. A.; Burton, D. R. and Lalor, M. J. (2006) Three-Dimensional Phase Unwrapping Algorithms: A Comparison. *Photon06 conference*. Manchester, UK.
- Bone, D. J. (1991) Fourier Fringe Analysis: The Two-Dimensional Phase Unwrapping Problem. *Applied Optics*, 30, 3627-32.
- Cusack, R. and Papadakis, N. (2002) New Robust 3-D Phase Unwrapping Algorithms: Application to Magnetic Field Mapping and Undistorting Echoplanar Images. *Nero Image*, 16, 754-764.
- Fang, Q.; Meaney, P. and Paulsen, K. (2006) The Multidimensional Phase Unwrapping Integral and Applications to Microwave Tomographical Image Reconstruction. *IEEE Transactions on Image Processing*, 15, 3311-3324.
- Flynn, T. J. (1996) Consistent 2-D Phase Unwrapping Guided by a Quality Map. Lincoln, NE, USA, IEEE.
- Flynn, T. J. (1997) Two-Dimensional Phase Unwrapping with Minimum Weighted Discontinuity. *Journal of the Optical Society of America A (Optics, Image Science and Vision)*, 14, 2692-701.
- Ghiglia, D. C. and Romero, L. A. (1996) Minimum  $L^p$ -Norm Two-Dimensional Phase Unwrapping. *Journal of the Optical Society of America A (Optics, Image Science and Vision)*, 13, 1999-2013.
- Ghiglia, D. C. and Pritt, M. D. (1998) *Two-Dimensional Phase Unwrapping: Theory, Algorithms and Software*, A Willey-Interscience Publication.
- Goldstein, R. M.; Zebker, H. A. and Werner, C. L. (1988) Satellite Radar Interferometry: Two-Dimensional Phase Unwrapping. 4 ed. Springfield, VA, USA.
- Hasan, K. A. (2006) One-Dimensional Hilbert Transform Processing for N-Dimensional Phase Unwrapping. *WSEAS Transactions on Circuits and Systems*, 5, 783-789.



- Herraez, M. A.; Burton, D. R.; Lalor, M. J. and Gdeisat, M. A. (2002) Fast Two-Dimensional Phase-Unwrapping Algorithm Based on Sorting by Reliability Following a Noncontinuous Path. *Applied Optics*, 41, 7437-44.
- Huntley, J. M. (2001) Three-Dimensional Noise-Immune Phase Unwrapping Algorithm. *Applied Optics*, 40, 3901-8.
- Huntley, J.M and Saldner, H., (1993), Temporal phase-unwrapping algorithm for automated interferogram analysis, *Applied Optics*, vol. 32, no. 17, pp. 3047-3052.
- Jenkinson, M. (2003) Fast, Automated, N-Dimensional Phase-Unwrapping Algorithm. *Magnetic Resonance in Medicine*, 49, 193-7.
- Marklund, O.; Huntley, J. M. and Cusack, R. (2005) Robust Unwrapping Algorithm for 3-D Volumes of Arbitrary Shape Containing Knotted Phase Singularity Loops.
- Rajoub, B. A.; Burton, D. R. and Lalor, M. J. (2005) A New Phase-to-Height Model for Measuring Object Shape Using Collimated Projections of Structure Light. *Jornal of Optics A: Pure and Applied Optics*, 7, S368-S375.
- Salfity, M. F.; Ruiz, P. D.; Huntley, J. M.; Graves, M. J.; Cusack, R. and Beauregard, D. A. (2006) Branch Cut Surface Placement for Unwrapping of Undersampled Three-Dimensional Phase Data: Application to Magnetic Resonance Imaging Arterial Flow Mapping. *Applied Optics*, 45, 2711-22.



## *Chapter Six*

### *Conclusions and Future Work*



## Chapter six

### Conclusions and Future Work

---

#### 6.1 Conclusions

In this thesis, a novel three-dimensional Fourier fringe analysis algorithm was proposed to demodulate fringe pattern sequences that represent dynamic objects. The proposed technique extends state-of-the-art two-dimensional Fourier fringe analysis into three dimensions.

Previously, the two-dimensional Fourier fringe analysis technique was introduced to the measurement dynamic objects whose surface changes with time (Su *et al.*, 2001). But this technique is not very suitable for the measurement of dynamic objects. Measuring the surface shape of a dynamic object requires a time-sequence of fringe patterns and each fringe pattern is analysed individually using the two-dimensional Fourier fringe analysis method. This technique has a limitation characterised by the fact that it is inherently a two-dimensional image processing technique. Consequently, the use of this technique to analyse a sequence of fringe patterns for a dynamic object does not utilise any relationship between consecutive fringe patterns. On that basis, the proposed three-dimensional Fourier fringe analysis technique proposed in this thesis is a novel technique that does utilise the relationships between frames of the fringe pattern sequence.

The proposed three-dimensional Fourier fringe analysis algorithm has been evaluated using both simulated and real dynamic objects. Computer simulation results show that the proposed technique can be applied to track the movement of a dynamic object at any time. Moreover, two real fringe pattern volumes for dynamic objects have also been used to evaluate the performance of the proposed algorithm. The first object is a moving dummy that mimics a patient in a treatment room and the second is a real patient. Results show that the proposed technique is able to demodulate fringe pattern volumes successfully.



The results of the proposed algorithm were compared with the state-of-art two-dimensional Fourier fringe analysis technique. This comparison shows that the current proposed algorithm is equivalent in performance to the two-dimensional Fourier fringe analysis method, despite the fact that two-dimensional Fourier fringe analysis has undergone many developments since 1986.

The main advantages of three-dimensional Fourier fringe analysis can be summarized as follows:

1. Three-dimensional Fourier fringe analysis considered all frames in the fringe patterns sequence as a single unit, not as individual frames.
2. Three-dimensional Fourier fringe analysis gives more frequency separation between the frequency components. So, filtering in frequency domain becomes easier.
3. Three-dimensional Fourier fringe analysis produces three-dimensional wrapped phase volumes. Three-dimensional wrapped phase volume gives more options and unwrapping paths than is the case for the two-dimensional wrapped phase maps. Consequently noisy regions can be avoided more easily.

On the other hand, three-dimensional Fourier fringe analysis has the following drawbacks:

1. Analysing video frame sequences necessitates larger computer memory requirements and utilises very complicated computations. Therefore, only a limited number of frames can be processed. In this thesis, the size of the processed videos is  $512 \times 512 \times 50$ . In other words, we can currently analyse dynamic objects only for short periods of time, *e.g.*, 2 seconds in our case.
2. Three-dimensional Fourier fringe analysis is difficult to automate. Each dynamic object has a different three-dimensional filter to select the needed components.

The three-dimensional Fourier fringe analysis algorithm can be divided into two stages: phase extraction and phase unwrapping. The first stage extracts the wrapped phase volume which is wrapped and limited to the  $-\pi$  and  $\pi$  range. The second stage involves the three-dimensional phase unwrapping process.



Two novel three-dimensional phase unwrapping algorithms were proposed in this thesis. The first algorithm is the **Best Path algorithm (BP)** which attempts to unwrap the wrapped phase volume by finding an optimal unwrapping path. The optimal unwrapping path is derived by calculating the quality of each voxel in the wrapped phase volume. The BP algorithm is found to be very robust and it manages to prevent error propagation throughout the phase volume. However, this algorithm does not take into account the effect of singularity loops during unwrapping, which may reduce the robustness of the algorithm. This algorithm has been tested on both simulated and real wrapped phase volumes. It was found that this algorithm is capable of unwrapping noisy wrapped phase volumes and produces good results.

The advantages of the BP algorithm can be outlined in the following points:

1. The BP algorithm utilises a quality map to unwrap reliable regions first and leaves noisy regions to be unwrapped last in order to prevent error propagation.
2. The BP algorithm does not rely on the quality of the voxels; it relies on the quality of the edges that connect two neighbouring voxels. Depending upon edge quality to guide the unwrapping path produces better results than relying upon voxel quality (Herraez *et al.*, 2002).
3. The BP algorithm follows discrete unwrapping paths to ensure the processing of the highest quality regions even if they are separated from each other.

The main disadvantage of the BP algorithm is:

- The BP algorithm does not identify the singularity loops. Singularity loops are the source of errors in three dimensional wrapped phase volumes as explained by Huntley (Huntley, 2001). Therefore, despite the fact that the BP algorithm gives very robust results, as shown in chapter 5 of this thesis, there is no guarantee that the BP algorithm will not penetrate any of the singularity loops that are located within the wrapped phase volume.

To overcome this problem, the **Best Path Avoiding Singularity Loop algorithm (BPASL)** is proposed by the author. The BPASL is very powerful algorithm that combines the 3D noise-immune phase unwrapping algorithm proposed by Huntley (Huntley, 2001) with the BP algorithm proposed by the author (Abdul-Rahman *et al.*, 2005). The BPASL algorithm finds the optimal unwrapping path taking into account the



effect of any singularity loops. The concept of **zero-weighted edges** is introduced to adjust the optimal path in order to avoid these singularity loops. The algorithm has been tested on many objects and indeed it shows high robustness and high immunity against noise.

The BPASL algorithm has an important advantage over the three-dimensional noise-immune algorithm (Huntley, 2001). The three-dimensional noise-immune algorithm identifies singularity loops then starts to unwrap around these loops. However, this algorithm does not calculate the quality of each individual voxel so that all voxels have the same weight and this may lead to error propagation. Moreover, there are still many ambiguities in identifying the loops, which may also cause errors to be propagated.

Both algorithms the (BP and BPASL) have been compared with two state-of-the-art, robust three-dimensional phase unwrapping algorithms; Huntley's algorithm (Huntley, 2001) and Cusack's algorithm (Cusack and Papadakis, 2002). The proposed algorithms indeed outperform the other two algorithms and produce very good results as shown previously in chapter five in this thesis.

Moreover, a comparison between three-dimensional and two-dimensional phase unwrapping algorithms has been carried out, in terms of robustness against noise and capability of detecting discontinuities. The comparison shows that three-dimensional phase unwrapping algorithms are more robust against noise than the two-dimensional phase unwrapping algorithms. Also, three-dimensional phase unwrapping algorithms were able to detect discontinuity lines in conditions where all the two-dimensional phase unwrapping algorithms failed.

## 6.2 Recommendations for Future Work

Many suggestions and recommendations for future work can be outlined for both three-dimensional Fourier fringe analysis and three-dimensional phase unwrapping research topics:



### I. Three-dimensional Fourier fringe analysis:

The field of three-dimensional Fourier fringe analysis is still a very new area of research, and it is expected to be an appropriate choice in analysing fringe patterns sequence for a dynamic object. The three-dimensional Fourier fringe analysis algorithm proposed in this thesis represents an initial prototype which may be developed in the future. The author would like to carry on the following investigations as future work in order to develop the proposed system:

1. The frequency filtering is one of the critical steps in the analysis. In this thesis only two types of filtering volumes were applied, ideal cubic and Butterworth filters. The author recommends the investigation of the use of other three-dimensional filters such as; spherical and three-dimensional exponential filters, to select the required frequency components.
2. Removing the DC term from the frequency spectrum has improved two-dimensional Fourier fringe analysis (Chen *et al.*, 2005; Gdeisat *et al.*, 2006). The author recommends extending the methods of DC removal into three dimensions and applying these techniques to three-dimensional Fourier fringe analysis.
3. In this thesis, only full-field fringe patterns were considered. Non full-field fringe patterns are more complicated and require the use of more sophisticated algorithms. The author suggests investigating how to analyse non full-field fringe patterns sequence using three-dimensional Fourier fringe analysis.
4. Finally, all dynamic objects used in this thesis were changing their heights with time. The author would like to consider more general movements such as, rotating and translating objects.



## II. Three-dimensional phase unwrapping:

The field of three-dimensional phase unwrapping algorithms is still new and only a few algorithms have been proposed so far. As shown in this thesis, three-dimensional phase unwrapping algorithms are more robust than many two-dimensional phase unwrapping algorithms. Many issues in the field of three-dimensional phase unwrapping algorithms should to be investigated, such as:

1. Extending the most reliable two-dimensional phase unwrapping algorithms into three dimensions. Many two-dimensional phase unwrapping algorithms are very robust such as Flynn's minimum discontinuity algorithm (Flynn, 1997) and the  $L^P$ -Norm algorithm (Ghiglia and Romero, 1996). The author recommends extending these algorithms into three dimensions and investigating their robustness.
2. In three-dimensional wrapped phase volumes, singularity loops exist. Only a few researchers have clarified why these loops occur. The author recommends further mathematical investigations of these singularity loops. Also, many ambiguities are found in identifying these loops. Therefore an investigation into methods of overcoming these ambiguities is recommended.
3. The performance of the proposed phase unwrapping algorithms, *i.e.* Best Path and Best Path Avoid Singularity Loops, are tested on the interferometry applications to extract objects' height. The author recommends testing the performance of these algorithms on different applications. For example, unwrapping MRI phase volumes. Unwrapping MRI phase volumes is considered to be a real challenge for any phase unwrapping algorithm due to its high noise environment and irregular volume shapes. The author also recommends adapting the proposed algorithms to deal with such cases.



4. Many applications produce a 4D wrapped phase volume. For example, measuring liquid flow in a certain volume with a certain period consider to be 4D problem. The author recommends an investigation to extend the proposed algorithm into four dimensions. Extending to four dimensions requires serious investigations about the behaviour of the singularity loops in 4D space. The theory of singularity loops may be extended to be singularity surfaces instead of loops and the 3D case becomes a special case of the more general 4D case.



**References:**

- Abdul-Rahman, H.; Gdeisat, M.; Burton, D. and Lalor, M. (2005) Fast three-dimensional phase-unwrapping algorithm based on sorting by reliability following a non-continuous path. 1 ed. Munich, Germany, SPIE-Int. Soc. Opt. Eng. 5856 part 1, 32-40.
- Chen, W.; Su, X.; Cao, Y.; Qc, Z. and Xiang, L. (2005) Method of eliminating zero spectrum in Fourier transform profilometry. *Optics and Lasers in Engineering*, 43, 1267-1276.
- Cusack, R. and Papadakis, N. (2002) New Robust 3-D Phase Unwrapping Algorithms: Application to Magnetic Field Mapping and Undistorting Echoplanar Images. *Nero Image*, 16, 754-764.
- Flynn, T. J. (1997) Two-dimensional phase unwrapping with minimum weighted discontinuity. *Journal of the Optical Society of America A (Optics, Image Science and Vision)*, 14, 2692-701.
- Gdeisat, M. A.; Burton, D. R. and Lalor, M. J. (2006) Eliminating the zero spectrum in Fourier transform profilometry using a two-dimensional continuous wavelet transform. *Optics Communications*, 266, 482-489.
- Ghiglia, D. C. and Romero, L. A. (1996) Minimum  $L^p$ -norm two-dimensional phase unwrapping. *Journal of the Optical Society of America A (Optics, Image Science and Vision)*, 13, 1999-2013.
- Herraez, M. A.; Burton, D. R.; Lalor, M. J. and Gdeisat, M. A. (2002) Fast two-dimensional phase-unwrapping algorithm based on sorting by reliability following a noncontinuous path. *Applied Optics*, 41, 7437-44.
- Huntley, J. M. (2001) Three-dimensional noise-immune phase unwrapping algorithm. *Applied Optics*, 40, 3901-8.
- Su, X.; Chen, W.; Qc, Z. and Chao, Y. (2001) Dynamic 3-D shape measurement method based on FTP. *optics and Lasers in Engineering*, 36, 46-64.



# *Appendix A*

## *The Simulated Objects Used in the Thesis*



## Appendix A

### The Simulated Objects Used in the Thesis

#### A.1 The First Object (Growing Hemisphere)

The first computer generated dynamic object used in this thesis is a growing hemisphere whose radius is increasing with time, *i.e.*, with the frame number. The sphere grows from the minimum radius of 105 pixels to the maximum radius of 125 pixels in 100 frames as described in Equation A.1; each frame consisting of  $256 \times 256$  pixels.

$$z(i, j, t) = 0.01 \times \sqrt{r^2(t) - [(x(i, j) - 128)^2 + (y(i, j) - 128)^2]} + [0.05 \times (x(i, j) + y(i, j))] \quad (\text{A.1})$$

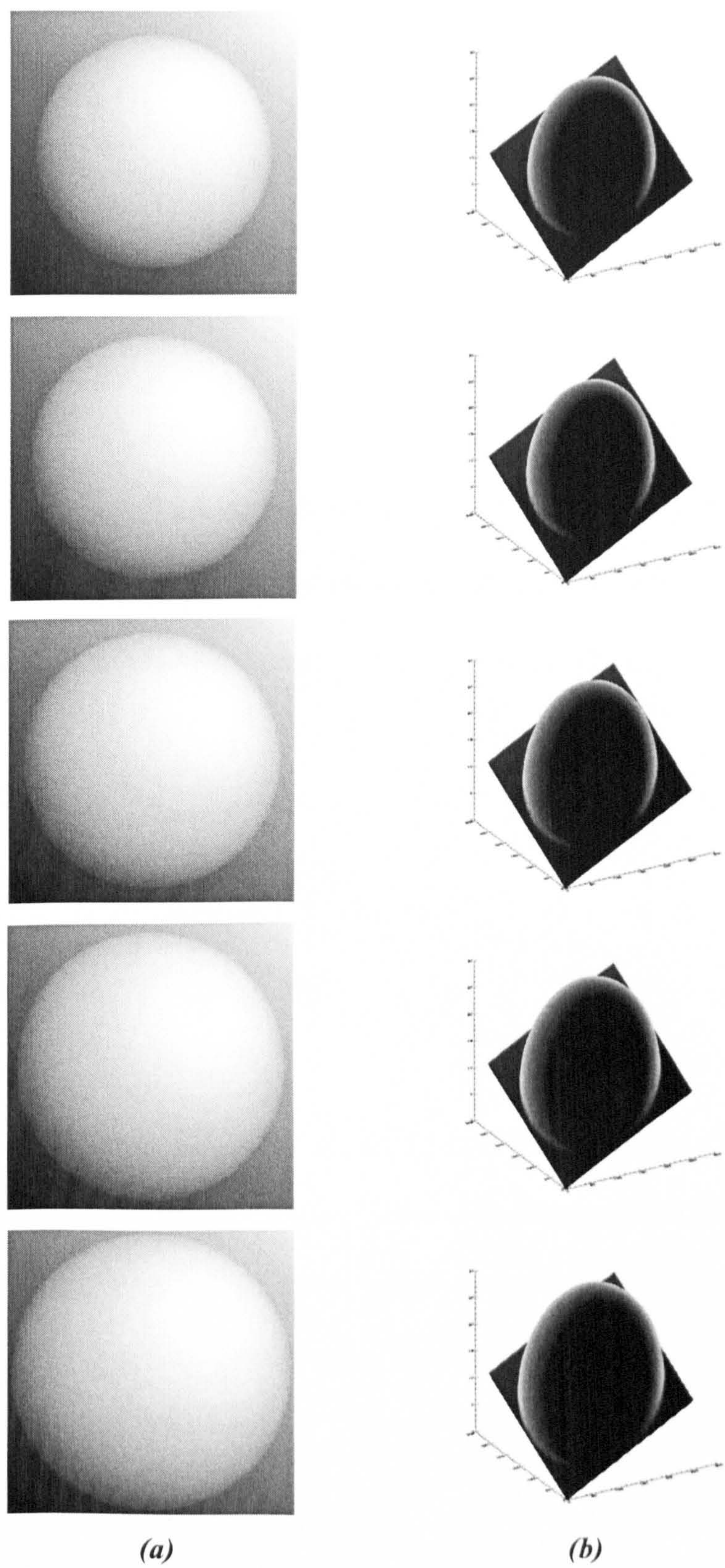
where,  $x(i, j)$  and  $y(i, j)$  are defined in the range  $[0, 255]$ .  $z(i, j, t)$  is the height of the pixel  $(i, j)$  at time  $t$  (actually,  $t$  represents the frame number).  $r(t)$  is the radius of the hemisphere at time  $t$  and its given by:

$$r(t) = 105 + (0.2 \times t) \quad (\text{A.2})$$

Where  $t$  is defined in the range  $[0, 99]$ .

Figure A.1 shows frames numbers 0, 25, 50, 75 and 99 of this object. Column (a) shows two-dimensional scaled images for that frames, these images scaled between black and white for display purposes. The colour white represents the maximum height of the object and the colour black represents its minimum height. Column (b) shows three-dimensional views for these frames.





**Figure A.1:** Frames 0, 25, 50, 75 and 99 of the first simulated object (growing sphere). Column(a) 2D scaled images for these frames and column (b) 3D views for these frames.



## A.2 The Second Object

The second computer generated dynamic object used in this thesis is a complicated surface whose shape is changing with time, *i.e.*, with the frame number. Each frame consists of  $256 \times 256$  pixels. The shape of this surface at time  $t$  is given by:

$$z(i, j, t) = 10 \times \left[ \sigma_1(t) \cdot \frac{\sin[x(i, j)]}{x(i, j)} + \sigma_2(t) \cdot \frac{\sin[y(i, j)]}{y(i, j)} \right] \quad (\text{A.3})$$

where,  $x(i, j)$  and  $y(i, j)$  are defined in the range  $[0, 255]$ .  $z(i, j, t)$  is the height of the pixel  $(i, j)$  at time  $t$  (actually,  $t$  represents the frame number).  $\sigma_1(t)$  and  $\sigma_2(t)$  are time varying functions that are given by:

$$\sigma_1(t) = 1.50 - (0.01 \times (t + 1)) \quad (\text{A.4})$$

$$\sigma_2(t) = 0.49 + (0.01 \times (t + 1)) \quad (\text{A.5})$$

where,  $t$  is defined in the range  $[0, 99]$ .

Figure A.2 shows frames numbers 0, 25, 50, 75 and 99 of this object. Column (a) shows two-dimensional scaled images for these frames, the images scaled between black and white for display purposes. The colour white represents the maximum height of the object and the colour black represents its minimum height. Column (b) shows three-dimensional views for these frames.



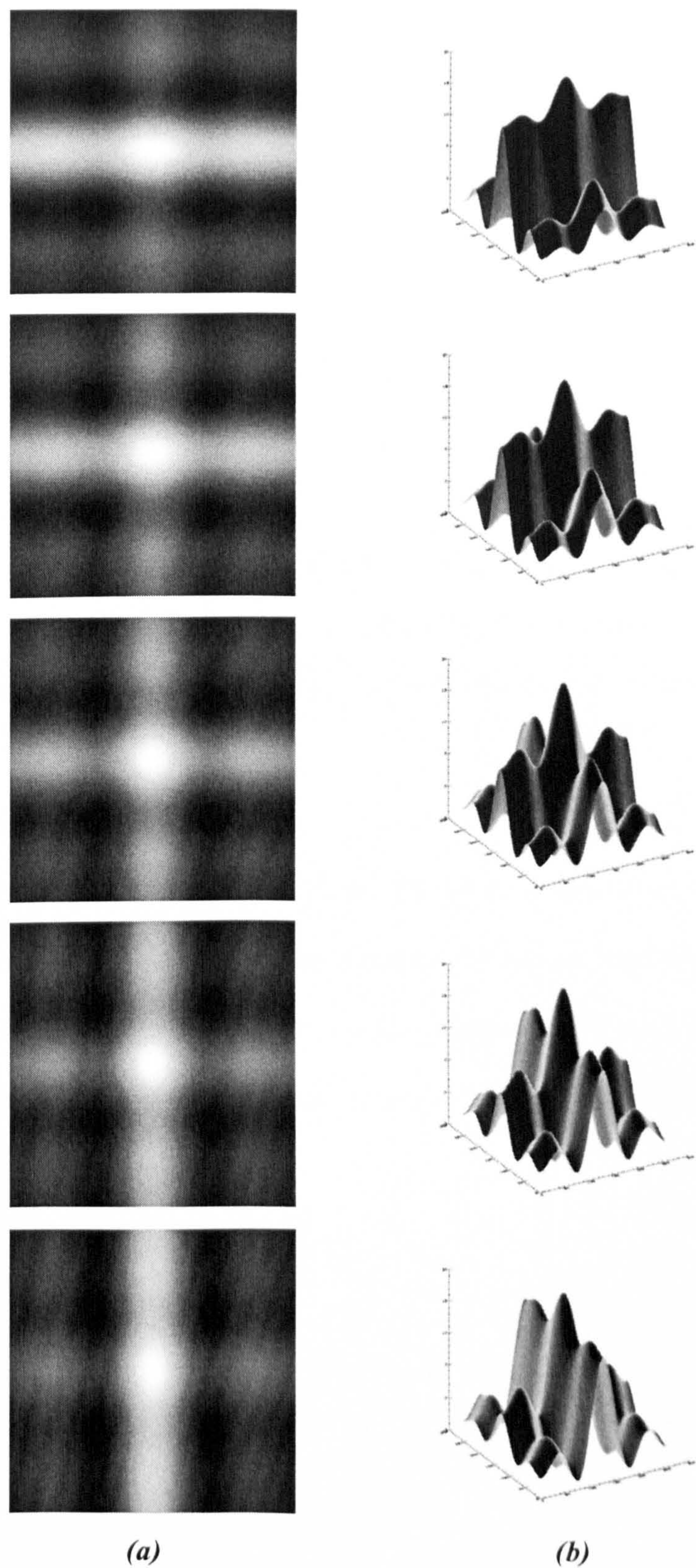


Figure A.2: Frames 0, 25, 50, 75 and 99 of the second simulated object. Column(a) 2D scaled images for these frames and column (b) 3D views for these frames.



## A.2 The Third Object

The third dynamic object is complicated steep surface whose height is increasing with time, as described in equation A.6. The object moves from its minimum height to its maximum heights in 100 frames, with  $256 \times 256$  pixels in each frame.

$$z(i, j, t) = 6.12 \left[ (1 - x_1^2) \exp(-x_1^2 - (y_1 + 1)^2) \right] - 20.6 \left[ \left( \frac{x_1}{5} - x_1^3 - y_1^5 \right) \exp(-x_1^2 - y_1^2) \right] \quad (\text{A.6}) \\ - 0.68 \left[ \exp(-(x_1 + 1)^2 - y_1^2) \right] + 0.1(x_2 + y_2) + (0.01 \ t)$$

where,  $x_1$  and  $y_1$  are defined in the range  $[-3.5, 3.5]$ , whereas  $x_2$  and  $y_2$  are defined in the range  $[0, 255]$ .  $z(i, j, t)$  is the height of the pixel  $(i, j)$  at time  $t$  (actually,  $t$  represents the frame number). The height difference between two successive frames is 0.01 units.  $t$  is defined in the range  $[0, 99]$

Figure A.3 shows frames numbers 0, 25, 50, 75 and 99 of this object. Column (a) shows two-dimensional scaled images for these frames, the images scaled between black and white for display purposes. The colour white represents the maximum height of the object and the colour black represents its minimum height. Column (b) shows three-dimensional views for these frames.



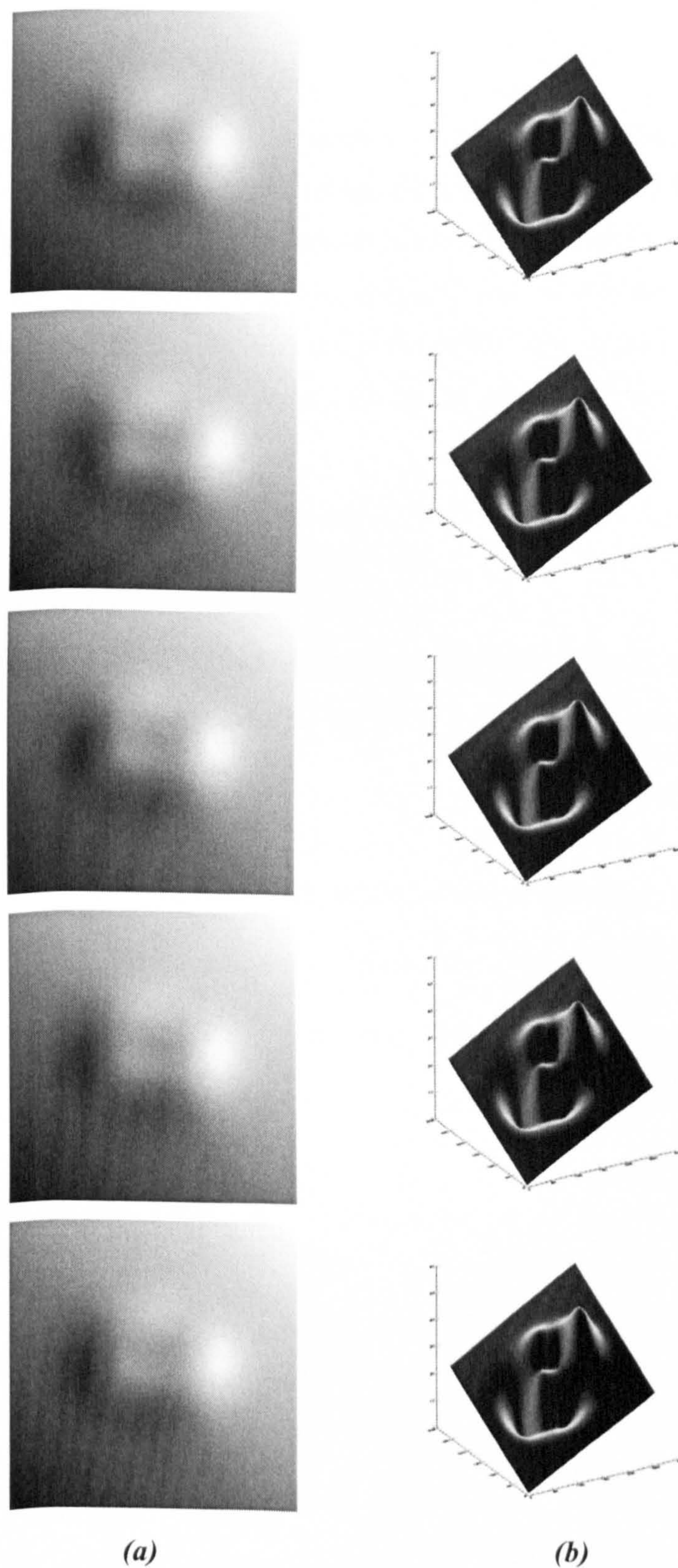


Figure A.3: Frames 0, 25, 50, 75 and 99 of the third simulated object. Column(a) 2D scaled images for these frames and column (b) 3D views for these frames.



## A.4 The Fourth Object

The fourth dynamic object is a planar surface whose height is increasing with time, as described in equation A.7. The object moves from its minimum height to its maximum height in 100 frames, with  $256 \times 256$  pixels in each frame. In the first 50 frames (from frame 0 to frame 49), the object is moving normally according to the equation, but from frame 50 until frame 99 a mask is applied to the middle area to prevent it from moving and allowing the two other outer areas to continue growing forming a central slotted depression.

$$z(i, j, t) = 0.5 \times (x(i, j) + y(i, j)) + (0.5 \times t) \quad (\text{A.7})$$

where,  $x(i, j)$  and  $y(i, j)$  are defined in the range  $[0, 255]$ .  $z(i, j, t)$  is the height of the pixel  $(i, j)$  at time  $t$  (actually,  $t$  represents the frame number) and  $t$  is defined in the range  $[0, 99]$

Figure A.4 shows frames numbers 0, 25, 50, 75 and 99 of this object. Column (a) shows two-dimensional scaled images for these frames, the images scaled between black and white for display purposes. The colour white represents the maximum height of the object and the colour black represents its minimum height. Column (b) shows three-dimensional views for these frames.



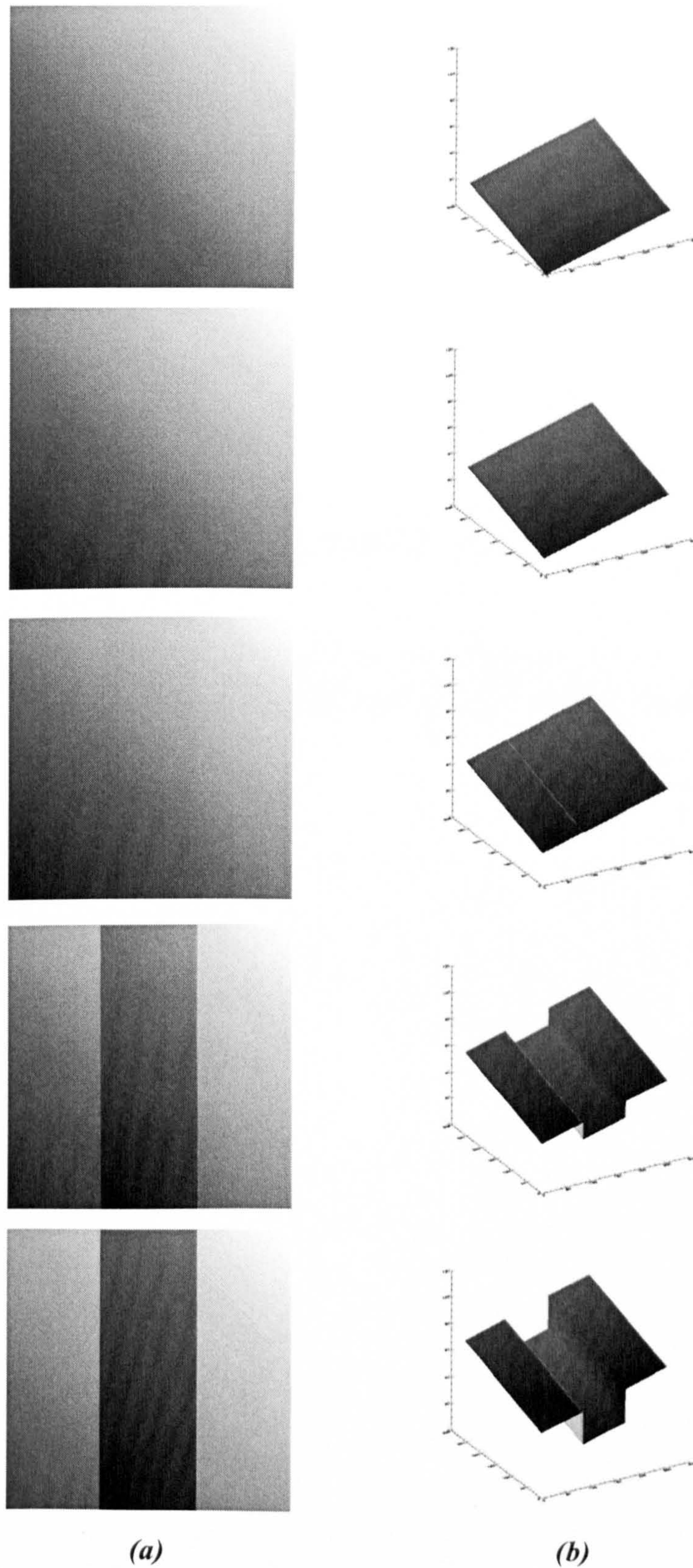


Figure A.4: Frames 0, 25, 50, 75 and 99 of the fourth simulated object. Column (a) 2D scaled images for these frames and column (b) 3D views for these frames.



## *Appendix B*

### *List of the Author's Publications*

.



## Appendix B

### List of the Author's Publications

---

Abdul-Rahman, H.; Gdeisat, M.; Burton, D.; Lalor, M.; Lilley F. and Moore C (2007) Fast and Robust Three-dimensional Best Path Phase Unwrapping Algorithm. *Applied Optics* (submitted and accepted for publication).

Abdul-Rahman, H. S.; Gdeisat, M. A.; Burton, D. R. and Lalor, M. J. (2006) Three-dimensional phase unwrapping algorithms: a comparison. *Photon06 conference*. Manchester, UK.

Abdul-Rahman, H.; Gdeisat, M.; Burton, D. and Lalor, M. (2005) Fast three-dimensional phase-unwrapping algorithm based on sorting by reliability following a non-continuous path. 1 ed. Munich, Germany, SPIE-Int. Soc. Opt. Eng. 5856 part 1, 32-40.