



**UNIVERSIDADE FEDERAL FLUMINENSE**  
**INSTITUTO DE FÍSICA**

**Desempacotamento de fase aplicado à perfilometria  
óptica.**

**JÚLIO CESAR DIAS DE SOUZA**

Dissertação apresentada ao Instituto de Física da Universidade Federal Fluminense como requisito parcial à obtenção do título de mestre em Física.

Orientador:

Prof. Dr. Paulo Acioly Marques dos Santos

Niterói  
2014

# **Desempacotamento de fase aplicado à perfilometria óptica.**

**Júlio Cesar Dias de Souza**

Dissertação apresentada ao Instituto de Física da Universidade Federal Fluminense como requisito parcial à obtenção do título de mestre em Física.

**Aprovada em 16 de dezembro de 2014.**

## **COMISSÃO EXAMINADORA**

---

**Prof. Dr. Paulo A. M. dos Santos**  
**Universidade Federal Fluminense**

---

**Prof.<sup>a</sup> Maria Priscila Pessanha de Castro**  
**Universidade Estadual do Norte Fluminense Darcy Ribeiro**

---

**Prof. José Augusto Huguenin**  
**Universidade Federal Fluminense**

Niterói

2014

S729 Souza, Júlio Cesar Dias de  
Desempacotamento de fase aplicado à perfilometria óptica /  
Júlio Cesar Dias de Souza; orientador: Paulo Acioly Marques  
dos Santos -- Niterói, 2014.  
163 p. : il.

Dissertação (Mestrado) - Universidade Federal Fluminense,  
Instituto de Física, 2014.  
Bibliografia: p.157-163.

1.MATÉRIA CONDENSADA. 2.PERFILOMETRIA. 3.RESÍDUO.  
I. Santos, Paulo Acioly Marques dos, Orientador.  
II.Universidade Federal Fluminense. Instituto de Física,  
Instituição responsável. III.Título.

CDD 530.41

Dedicatória

À minha esposa Ana Paula,  
meus pais e meu irmão.

# Agradecimentos

À minha esposa pela paciência, compreensão e companheirismo nas horas mais difíceis e a meus pais pelo carinho, apoio e dedicação durante todos estes anos.

Ao Prof. Dr. Paulo Acioly Marques dos Santos pela orientação e apoio, com incomparável boa vontade em auxiliar e contribuir para meu crescimento e pelos conhecimentos compartilhados, os quais foram muito além do conteúdo acadêmico, ao longo do período de convivência neste Mestrado.

Ao Mestre e Doutorando Marcos Eduardo de Oliveira pela inigualável boa vontade, ajuda prestada e conhecimentos compartilhados nesses anos de convivência.

Ao Mestre e Doutorando Rosembergue Brasileiro por seu companheirismo e ajuda nos momentos de dificuldade.

Aos funcionários da coordenação da pós graduação, em especial à Valéria pela prontidão e competência ao solucionar os mais variados problemas.

Aos funcionários da biblioteca pela competência e profissionalismo durante toda a graduação e mestrado.

E, finalmente, aos meus companheiros de trabalho, em especial a André Martins e Fábio Ramos Cordeiro por seu apoio incondicional à minha formação, mesmo em períodos de maior volume de trabalho.

# Resumo

Muitas aplicações de imageamento como perfilometria interferométrica ou topografia de superfície por Radar de Abertura Sintética (inSAR) utilizam a análise da informação de fase. Um problema que surge naturalmente decorrente do processo é o empacotamento da fase nos intervalo de  $-\pi$  a  $\pi$ .

O objetivo do presente trabalho é apresentar uma proposta para desempacotamento bidimensional de fase (*2D Phase Unwrapping*), com base na teoria de resíduos, que seja eficiente na obtenção de seus valores verdadeiros, quando aplicada a mapas gerados pela técnica de Perfilometria por Transformação de Fourier, utilizada para a obtenção de perfis de superfícies iluminadas com padrões de franjas.

Para isso, abordaremos a teoria sobre o desempacotamento e um estudo sobre o impacto dos ruídos no processo, apresentaremos algumas abordagens atuais para a solução do problema, como os algoritmos de inundação (*flood-fill*), multifrequências e escaneamento rápido de linha (*fast scan-line*) e abordaremos com maior profundidade a teoria de resíduos aplicada ao desempacotamento de fase em algoritmos de balanceamento de resíduos (*branch-cut*).

Posteriormente, demonstraremos matematicamente nossa proposta por balanceamento de cargas de resíduos e analisaremos os resultados obtidos com nosso algoritmo ao desempacotar mapas de fase criados pela técnica de Perfilometria por Transformação de Fourier sobre padrões de franjas de moiré geradas por cristal fotorrefrativo, padrões de franjas de interferência geradas por interferômetro de Michelson e padrões de franjas gerados por computador e projetados por um projetor digital, analisando o impacto dos ruídos e a eficiência quando comparado a abordagens similares.

# Abstract

Many applications of imaging such as interferometric surface profilometry or topography by Synthetic Aperture Radar (InSAR) uses the analysis of phase information. A problem that arises naturally in the process of extracting this information is the wrapping phase in the interval from  $-\pi$  to  $\pi$ .

The objective of the present work is to demonstrate a proposal for two-dimensional phase unwrapping based on the residue's theory, which is efficient in obtain the true values of phase in maps generated by the technique of Profilometry by Fourier Transform, that is used to obtaining profiles of surfaces illuminated with fringe patterns.

To achieve this, we'll do a theoretical explanation of unwrapping and a study of the impact of noises in the process. We'll present some current approaches to a solution for the problem, as flooding (*flood-fill*), multifrequency and fast line scanning (*fast scan-line*) algorithms and discuss the theory of residues applied to the phase unwrapping residues balancing algorithms (*branch-cut*).

Subsequently, we'll do the mathematical demonstration of our proposal for balancing residue charges and analyze the results obtained with our algorithm to unwrap phase maps created by the technique of Fourier Transform Profilometry on moiré fringe patterns generated by photorefractive crystal, patterns of interference fringes generated by a Michelson interferometer and fringe patterns generated by a computer and projected by a digital projector, analyzing the impact of noise and the efficiency of our algorithm when compared to similar approaches.

# SUMÁRIO

1. INTRODUÇÃO .....	19
1.1 ESTRUTURA DA DISSERTAÇÃO .....	21
2. A IMPORTÂNCIA DA FASE NO PROCESSAMENTO DE IMAGENS .....	24
2.1 FUNÇÕES PERIÓDICAS E REPRESENTAÇÕES COMPLEXAS .....	24
2.1.1 FUNÇÕES PERIÓDICAS .....	24
2.1.2 NÚMEROS COMPLEXOS .....	26
2.2 RERESSENTAÇÃO DE FUNÇÕES EM ESPECTROS DE FREQUÊNCIA .....	28
2.3 ANÁLISE DE ESPECTROS DE FASE NO PROCESSAMENTO DE IMAGENS .....	31
2.4 A FASE EMPACOTADA .....	37
3. FUNDAMENTOS DO DESEMPACOTAMENTO DE FASE.....	39
3.1 O DESEMPACOTAMENTO BIDIMENSIONAL DE FASE ( <i>2D PHASE UNWRAPPING</i> ) .....	40
3.2 DESAFIOS DO DESEMPACOTAMENTO DE FASE .....	46
3.3 ALGORITMOS DE DESEMPACOTAMENTO DE FASE .....	52
3.3.1 ALGORITMOS DE MULTIFREQUÊNCIAS .....	53
3.3.2 ALGORITMOS DE CAMINHO NÃO GUIADOS .....	56
3.3.3 ALGORITMOS GUIADOS POR QUALIDADE.....	57
3.3.4 ESTRATÉGIAS DE CAMINHO .....	59

3.3.4.1 ALGORITMOS DE INUNDAÇÃO.....	59
3.3.4.2 ALGORITMOS DE ESCANEAMENTO RÁPIDO DE LINHA.....	62
3.3.5 ALGORITMOS DE BALANÇAMENTO DE RESÍDUOS.....	64
3.3.5.1 A LOCALIZAÇÃO DOS RESÍDUOS.....	72
3.3.5.2 A CONEXÃO ENTRE OS RESÍDUOS.....	76
3.3.6 OUTROS ALGORITMOS DE DESEMPACOTAMENTO DE FASE.....	78
4. PERFILOMETRIA POR TRANSFORMAÇÃO DE FOURIER.....	80
5. PROPOSTA PARA UM ALGORITMO DE DESEMPACOTAMENTO DE FASE POR BALANCEAMENTO DE RESÍDUOS.....	89
5.1 BASE TEÓRICA.....	90
5.2 ALGORITMO PROPOSTO.....	94
6. RESULTADOS OBTIDOS.....	104
7. CONCLUSÕES E PERSPECTIVAS FUTURAS.....	121
7.1 CONCLUSÕES.....	121
7.2 PERSPECTIVAS FUTURAS.....	125
APÊNDICE A.....	126
A.1 – GERAÇÃO DE PADRÕES DE FRANJAS DE MOIRÉ.....	126
A.2 – GERAÇÃO DE PADRÕES DE MOIRÉ DINÂMICOS POR CRISTAIS FOTORREFRATIVOS.....	129
A.3 – MONTAGEM EXPERIMENTAL.....	133

APÊNDICE B .....	136
B.1 – CÓDIGO FONTE DO ALGORITMO DESENVOLVIDO .....	136
REFERÊNCIAS.....	157

# LISTA DE FIGURAS

<b>Fig. 2.1</b> Representação da função periódica $g(x)=\cos(x)$ , com o período $T$ indicado. ....	25
<b>Fig. 2.2</b> Representação do plano $z=x+iy$ no plano complexo.....	26
<b>Fig. 2.3</b> Formas de onda resultantes da soma de duas componentes senoidais de frequência $f$ e $2f$ . (A) As duas componentes possuem a mesma amplitude e estão em fase; (B) A componente de frequência $2f$ tem o dobro da amplitude da componente de frequência $f$ e está defasada em $\pi/2$ . ....	29
<b>Fig. 2.4</b> Representação no domínio da frequência das formas de onda dos harmônicos ilustrados na Fig. 2.3 (A) e (B). ....	30
<b>Fig. 2.5</b> Imagem original representada no domínio da posição. ....	35
<b>Fig. 2.6</b> Imagem da Fig. 2.5 representada no domínio da frequência espacial.....	35
<b>Fig. 2.7</b> Espectro de amplitude da Fig. 2.5 no espaço de frequência espacial.....	35
<b>Fig. 2.8</b> Espectro de fase da Fig. 2.5 na representação de frequência espacial. ....	35
<b>Fig. 2.9</b> Imagem com informação de fase corrompida, resintetizada para o domínio da posição. ....	36
<b>Fig. 2.10</b> Imagem com informação de amplitude corrompida, resintetizada para o domínio da posição. ....	37
<b>Fig. 3.1</b> Gráfico da fase empacotada de uma amostra com $N$ pontos.....	41
<b>Fig. 3.2</b> Fase de ajuste a ser somada à fase empacotada.....	42
<b>Fig. 3.3</b> Gráfico comparativo entre a fase de ajuste e a fase empacotada. A soma das duas funções produz a função de fase verdadeira da Fig 2.4.....	42
<b>Fig. 3.4</b> Função de fase empacotada e função de fase verdadeira (resultado da soma da fase empacotada com a fase de ajuste), para comparação. ....	43

<b>Fig. 3.5</b> Função empacotada com ruído e sem ruído para comparação. ....	47
<b>Fig. 3.6</b> Fase de ajuste para a fase empacotada sem a presença de ruído e com a presença de ruído. ....	47
<b>Fig. 3.7</b> Fase desempacotada com e sem ruído.....	48
<b>Fig. 3.8</b> (a-i) Exemplo de uma estratégia de caminho por inundação. ....	61
<b>Fig. 3.9</b> Estratégia de um algoritmo de escaneamento rápido de linha .....	64
<b>Fig. 3.10</b> (a) e (b) com caminhos de desempacotamento do ponto A para o ponto B, sem a presença de barreira conectando os resíduos (a) e com a presença de barreira conectando os resíduos (b).....	71
<b>Fig. 3.11</b> Parte de uma imagem contendo um mapa de fases empacotado. As setas indicam o sentido anti-horário para a soma dos elementos. ....	75
<b>Fig. 3.12</b> Uma das possíveis configurações para conexão entre os resíduos. As regiões hachuradas representam áreas excluídas do processo de desempacotamento, os círculos brancos os resíduos de carga positiva (+1) e os círculos pretos, os de carga negativa (-1). ....	77
<b>Fig. 3.13</b> Outra das possíveis configurações para conexão entre os resíduos. Nesta, as regiões hachuradas compreendem uma área maior do que na figura anterior. ....	77
<b>Fig. 4.1</b> Geometria óptica para a projeção de franjas em um objeto. ....	82
<b>Fig. 4.2</b> Espectro de Fourier de um padrão de franjas projetado sobre a superfície de um objeto e um único espectro selecionado e transladado para a origem. ....	86
<b>Fig. 4.3</b> Representação do plano complexo. ....	87
<b>Fig. 5.1</b> Mapa de fases empacotado gerado por computador. ....	97
<b>Fig. 5.2</b> Mapa de fases com a localização dos saltos deslocada, obtido após a execução do passo 1.....	97

<b>Fig. 5.3</b> Mapa contendo os pontos com salto de fase do mapa da Fig. 5.1, obtidos após a execução dos passos 2 e 3.....	97
<b>Fig. 5.4</b> Mapa com as linhas de salto de fase do mapa da Fig. 5.2, obtidas após a execução do passo 4.....	97
<b>Fig. 5.5</b> Resíduos sobrepostos às linhas de salto de fase..	98
<b>Fig. 5.6</b> Resíduos detectados no mapa de fases deslocado em relação ao original. ....	98
<b>Fig. 5.7</b> Resíduos cruzados por linhas de salto de fase do mapa original.....	99
<b>Fig. 5.8</b> Resíduos cruzados por linhas de salto de fase do mapa deslocado. ....	99
<b>Fig. 5.9</b> Após verificado o tamanho das linhas que conectam os resíduos, são selecionadas aquelas que possuem a menor extensão.....	99
<b>Fig. 5.10</b> Caminhos de contorno sobre as linhas que conectam os resíduos..	100
<b>Fig. 5.11</b> Mapa de fases desempacotado obtido após o processamento do passo 8. ....	101
<b>Fig. 5.12</b> Resultado, para comparação, de desempacotamento utilizando um algoritmo de Goldstein que efetua o balanceamento de resíduos.....	101
<b>Fig. 5.13</b> Resultado, para comparação, de desempacotamento linear utilizando um algoritmo simples de caminho não guiado, que desconsidera a existência dos resíduos. .....	101
<b>Fig. 6.1</b> Imagem com 700x900 píxeis, do padrão de franjas de moiré dinâmico projetado sobre a superfície de um semicilindro. ....	107
<b>Fig. 6.2</b> Mapa de fases empacotado, obtido através da aplicação da técnica de PTF sobre o padrão de franjas da Fig. 6.1.....	107
<b>Fig. 6.3</b> Mapa de fases da Fig. 6.2, visto sob uma perspectiva tridimensional. ....	107
<b>Fig. 6.4</b> Mapa de fases desempacotado após a utilização de um algoritmo simples que fez a verredura linear do mapa da Fig. 6.2.....	108

<b>Fig. 6.5</b> Mapa de fases da Fig. 6.4, visto sob uma perspectiva tridimensional. ....	108
<b>Fig. 6.6</b> Mapa de fase desempacotado obtido através da utilização do algoritmo de Goldstein sobre o mapa da Fig. 6.2.....	108
<b>Fig. 6.7</b> Mapa de fases da Fig. 6.6 visto sob uma perspectiva tridimensional. ....	108
<b>Fig. 6.8</b> Mapa de fases desempacotado obtido através da utilização do algoritmo proposto neste trabalho sobre o mapa da Fig. 6.2.....	109
<b>Fig. 6.9</b> Mapa de fases da Fig. 6.8, visto sob uma perspectiva tridimensional. ....	109
<b>Fig. 6.10</b> Configuração de barreiras para balanceamento de cargas de resíduos efetuada pelo algoritmo proposto nesse programa. ....	109
<b>Fig. 6.11</b> Configuração de barreiras para balanceamento de cargas de resíduos efetuada pelo algoritmo de Goldstein.....	109
<b>Fig. 6.12</b> Imagem com 1900x2200 píxeis, do padrão de franjas gerado por computador e projetado por um projetor digital sobre uma semi-rosca. ....	110
<b>Fig. 6.13</b> Mapa de fases empacotado, obtido através da aplicação da técnica de PTF sobre o padrão de franjas da Fig. 6.12. ....	110
<b>Fig. 6.14</b> Mapa de fases da Fig. 6.13 visto sob uma perspectiva tridimensional. ....	110
<b>Fig. 6.15</b> Mapa de fases desempacotado após a utilização de um algoritmo simples que fez a verredura linear do mapa da Fig. 6.13.....	111
<b>Fig. 6.16</b> Mapa de fases da Fig. 6.15 visto sob uma perspectiva tridimensional. ....	111
<b>Fig. 6.17</b> Mapa de fases desempacotado obtido após a utilização do algoritmo de Goldstein sobre o mapa da Fig. 6.13.....	111
<b>Fig. 6.18</b> Mapa de fases da Fig. 6.17 visto sob uma perspectiva tridimensional. ....	111
<b>Fig. 6.19</b> Mapa de fases desempacotado obtido através da utilização do algoritmo proposto neste trabalho sobre o mapa da Fig. 6.13.....	112

<b>Fig. 6.20</b> Mapa de fases da Fig. 6.19, visto sob uma perspectiva tridimensional. ....	112
<b>Fig. 6.21</b> Configuração de barreiras para balanceamento de cargas de resíduos efetuada pelo algoritmo proposto nesse programa. ....	112
<b>Fig. 6.22</b> Configuração de barreiras para balanceamento de cargas de resíduos efetuada pelo algoritmo de Goldstein.....	112
<b>Fig. 6.23</b> Imagem com 200x300 píxeis, do padrão de franjas de interferência gerado por um interferômetro de Michelson e projetado sobre a superfície de um semicilindro....	114
<b>Fig. 6.24</b> Mapa de fases empacotado, obtido através da aplicação da técnica de PTF sobre o padrão de franjas da Fig. 6.23. ....	114
<b>Fig. 6.25</b> Mapa de fases da Fig. 6.24 visto sob uma perspectiva tridimensional. ....	114
<b>Fig. 6.26</b> Mapa de fases desempacotado após a utilização de um algoritmo simples que fez a verredura linear do mapa da Fig. 6.24.....	115
<b>Fig. 6.27</b> Mapa de fases da Fig. 6.26 visto sob uma perspectiva tridimensional. ....	115
<b>Fig. 6.28</b> Mapa de fases desempacotado obtido após a utilização do algoritmo de Goldstein sobre o mapa da Fig. 6.23.....	115
<b>Fig. 6.29</b> Mapa de fases da Fig. 6.28 visto sob uma perspectiva tridimensional. ....	115
<b>Fig. 6.30</b> Mapa de fases desempacotado obtido através da utilização do algoritmo proposto neste trabalho sobre o mapa da Fig. 6.24.....	116
<b>Fig. 6.31</b> Mapa de fases da Fig. 6.30, visto sob uma perspectiva tridimensional. ....	116
<b>Fig. 6.32</b> Configuração de barreiras para balanceamento de cargas de resíduos efetuada pelo algoritmo proposto nesse programa. ....	118
<b>Fig. 6.33</b> Configuração de barreiras para balanceamento de cargas de resíduos efetuada pelo algoritmo de Goldstein.....	118
<b>Fig. 6.34</b> Ampliação de região com configuração de barreiras de balanceamento dos resíduos, em nosso algoritmo (a) e o algoritmo de Goldstein (b).....	118

**Fig. A.1.1** Padrão de moiré resultante da superposição de redes de Ronchi.....126

**Fig. A.3.1** Montagem experimental básica de mistura de duas ondas (2WM) utilizada neste trabalho .....133

# LISTA DE TABELAS

<b>Tabela A.3.1</b> Alguns parâmetros físicos do cristal $\text{Bi}_{12}\text{TiO}_{20}$ .....	136
--	-----

# LISTA DE QUADROS

<b>Quadro 1</b> Condições para a independência de caminho.....	66
<b>Quadro 2</b> Comparação entre os tempos de processamento do algoritmo de Goldstein e o proposto nesse trabalho.....	119
<b>Quadro 3</b> Lista comparativa entre os fatores de desempenho do algoritmo de Goldstein e o proposto nesse trabalho.....	120

# Capítulo 1

## INTRODUÇÃO

*Neste capítulo será efetuada uma apresentação da motivação e dos objetivos gerais dessa dissertação, além de um resumo da estrutura de organização dos capítulos.*

Técnicas de processamento e análise de imagens digitais têm aplicações em diversas áreas como perfilometria interferométrica [1], ressonância magnética [2], rádio astronomia [3], etc.

Em muitas dessas aplicações é possível relacionar um valor de fase a ser extraído da imagem, com quantidades físicas de interesse como a topografia de uma superfície em interferometria [4] ou a distorção de uma frente de onda em óptica adaptativa [5].

Essas técnicas, ao extrair-se o valor da fase, deparam-se com um problema: a operação matemática para obtenção da informação envolve o cálculo de funções periódicas trigonométricas como a *arcotangente*. Como computacionalmente não é possível distinguir se o argumento de funções desta natureza está acrescido de um múltiplo de seu período, os cálculos por computador retornam valores no intervalo de  $-\pi$  a  $\pi$ , dito estarem empacotados, sendo necessária uma operação de desempacotamento sobre a fase obtida para remover as descontinuidades criadas e recuperar seu valor verdadeiro.

O processo de desempacotamento de fase, basicamente, analisa os pontos da função com descontinuidade de módulo igual a  $2\pi$  e acrescenta um valor múltiplo de  $2\pi$  a todos os pontos necessários, de forma a recuperar seu caráter contínuo.

Apesar de inicialmente parecer uma tarefa simples, a presença de ruídos provenientes do meio físico e dos instrumentos de medição utilizados para a obtenção da imagem, causam enorme dificuldade na criação de um algoritmo definitivo que apresente resultados satisfatórios no desempacotamento de qualquer mapa de fases<sup>1</sup>. Por exemplo, no caso da análise de franjas provenientes de processos interferométricos, podemos encontrar ruídos causados por problemas de iluminação ou refletância disforme de superfícies [6], iluminação de fundo, descontinuidades do objeto, frequência insuficiente de franjas projetadas [7], variações de corrente elétrica e térmicas nos sensores dos instrumentos de captura [8, 9], falseamento na conversão de sinal analógico para digital, entre outros.

A variedade de formas, configurações e densidades de ruídos que podem ser encontrados em mapas de fase obtidos de sistemas físicos reais eleva em muito a complexidade do problema, o que tem movimentado um grande esforço em pesquisas para o desenvolvimento de inúmeros algoritmos com o objetivo de superar essas dificuldades.

Atualmente o que existe é uma gama de aplicações, cada qual, especializada no desempacotamento de mapas com determinadas características [10].

Na busca por uma solução, nas últimas décadas, diversas abordagens e interpretações acerca do problema foram desenvolvidas [11]. Existem aquelas que realizam uma análise local, para verificar a ocorrência de fatores que possam interferir no processo e outras que estimam de forma global, uma solução para a minimização do impacto causado pelos ruídos no processo [12].

A verificação global envolve a utilização de algoritmos de minimização de norma, que por utilizarem cálculos complexos, apesar de geralmente produzirem bons resultados,

---

<sup>1</sup> Representação da distribuição dos valores de fase de uma função.

consomem tempo e recursos muito maiores para o término do processamento do que os algoritmos de análise local e por essa razão não serão abordados neste trabalho.

Focaremos nos aspectos teóricos e práticos dos algoritmos de análise local, que ao longo das últimas décadas apresentaram significativo avanço em qualidade e redução do tempo de processamento necessário para obtenção dos valores verdadeiros da fase. Esses algoritmos, basicamente, verificam quais os pontos no mapa têm potencial para interferir no desempacotamento e realizam um tratamento, seja isolando ou até substituindo-os por inferência ou outro método, de forma a minimizar a propagação de erros durante o processo [10].

## **1.1 ESTRUTURA DA DISSERTAÇÃO**

Após a introdução e apresentação da estrutura da dissertação, no capítulo 2 explicaremos a respeito da importância da fase no processamento de imagens, com a proposição de um experimento prático que evidencia qualitativamente o impacto da alteração dessa informação sobre a imagem original.

Além disso, como introdução para o capítulo 3, aproveitaremos o experimento demonstrado para destacar a ocorrência do empacotamento decorrente do processo computacional de extração da informação da fase.

No capítulo 3 introduziremos os fundamentos do desempacotamento de fase e exporemos a necessidade de sua realização para a obtenção dos valores verdadeiros da fase.

Como forma de apresentar uma visão atual das pesquisas na área, também apresentaremos alguns dos algoritmos clássicos e atuais, apontando as vantagens,

desvantagens e cenários apropriados para sua utilização. Ademais, utilizando a teoria de resíduos, introduziremos o conceito de cargas de resíduo como indicativo de problemas no mapa de fases, a independência de caminho como condição a ser buscada para a realização do desempacotamento e o balanceamento de resíduos como forma de atender a essa condição. Nesse ponto, chamaremos a atenção para a liberdade de escolha das possíveis configurações para o balanceamento de resíduos e o impacto que a estratégia dessa escolha pode ter na qualidade do resultado final e no tempo necessário para o processamento.

No capítulo 4 apresentaremos a técnica de Perfilometria por Transformação de Fourier (PTF) como método para a geração de mapas empacotados, que após passarem por um processo de desempacotamento, fornecem um perfil do objeto iluminado com um padrão de franjas. A aplicação desse método associado ao desempacotamento de fase permite a realização de diversos experimentos úteis na área de perfilometria.

No capítulo 5 apresentaremos uma proposta para um algoritmo de desempacotamento que utiliza a teoria de resíduos para a recuperação da independência de caminho, com uma estratégia alternativa para escolha da configuração das conexões para balanceamento de cargas, que minimiza a perda de informação e reduz o tempo de processamento, quando comparada à forma tradicional de análise de todas as configurações possíveis.

No capítulo 6 apresentaremos os resultados obtidos com a utilização de nosso algoritmo sobre mapas gerados pela técnica de PTF em imagens obtidas de forma experimental através de processos distintos como projeção direta de franjas por projetor digital, projeção de franjas obtidas em um experimento com um interferômetro de Michelson ou a geração de padrões de moiré dinâmicos por cristal fotorrefrativo.

Nesse íterim, compararemos os resultados de nosso algoritmo com um tradicional que também utiliza a teoria de resíduos e um simples, que executa o desempacotamento de forma linear sem considerar a presença de fatores potenciais para a propagação de erros. Dessa forma, buscamos estabelecer um parâmetro que indique se houve progresso na qualidade dos mapas gerados e melhora na eficiência de nosso algoritmo em termos de tempo de processamento.

No capítulo 7, para encerrar essa dissertação, apresentaremos as conclusões acerca do trabalho desenvolvido e as perspectivas futuras quanto ao desenvolvimento da pesquisa.

## Capítulo 2

# A IMPORTÂNCIA DA FASE NO PROCESSAMENTO DE IMAGENS

*Neste capítulo será apresentada uma breve introdução à importância da fase no processamento de imagens e como a necessidade de desempacotá-la surge naturalmente em decorrência do processo matemático para extração dessa informação da imagem.*

## 2.1 FUNÇÕES PERIÓDICAS E REPRESENTAÇÕES COMPLEXAS

### 2.1.1 FUNÇÕES PERIÓDICAS

Funções periódicas são aquelas que se repetem ao longo da variável independente com um período constante, ou seja, existe um  $T$ , tal que  $g(x) = g(x + nT)$ , com  $n \in \mathbb{Z}$ , para qualquer  $x$  no domínio de  $g$ . A variável  $T$ , chamada de período, é o comprimento do intervalo em  $x$  para a função se repetir<sup>2</sup>. A Fig. 2.1 contém um exemplo de função periódica, com seu respectivo período indicado.

Definimos também a variável  $f$  como o inverso do período, que representa a frequência fundamental, ou o número de repetições (ou ciclos) por intervalo unitário em  $x$ .

Entre as funções periódicas mais conhecidas, podemos citar as trigonométricas seno, cosseno e tangente. Por exemplo, a função  $g(x) = \cos(x)$  é periódica, com período

---

<sup>2</sup> Observe que após o comprimento  $nT$ , a função também se repete.

$T = 2\pi$  e frequência  $f = T^{-1} = 1/2\pi$ , uma vez que  $\cos(x) = \cos(x + 2\pi) = \cos(x + 2\pi n)$

O argumento de funções dessa natureza é chamado de fase  $\phi$ , que no caso do exemplo anterior,  $g(x) = \cos(x)$ , é  $\phi = x$ .

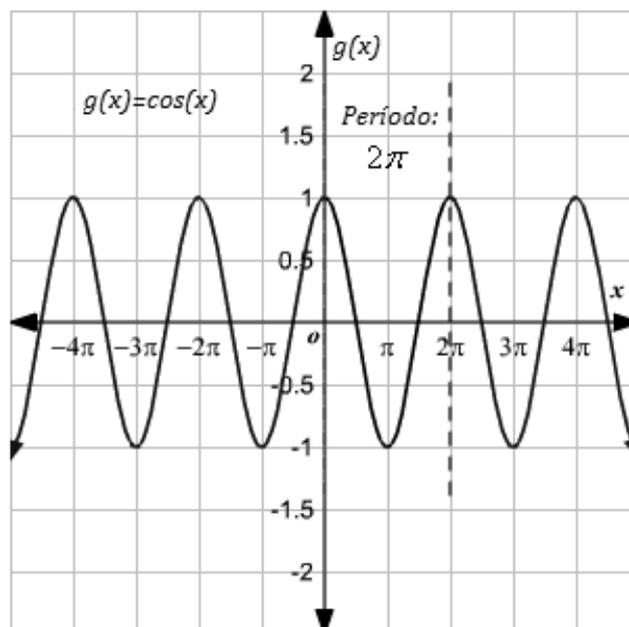


Fig. 2.1 Representação da função periódica  $g(x)=\cos(x)$ , com o período  $T$  indicado.

## 2.1.2 NÚMEROS COMPLEXOS

Um número complexo é um número da forma  $z = x + iy$ , com  $x$  e  $y$  reais. Nessa representação,  $x$  é a parte real de  $z$ ,  $y$  a parte imaginária e  $i = \sqrt{-1}$ .

Os números complexos podem ser representados por pontos em um plano cartesiano denominado diagrama de Argand<sup>3</sup> (Fig. 2.2).

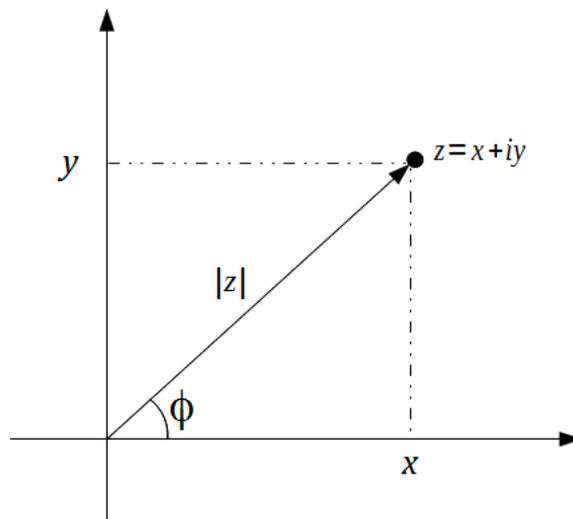


Fig. 2.2 Representação do plano  $z=x+iy$  no plano complexo.

Neste diagrama, o eixo horizontal representa a parte real de  $z$  e o eixo vertical a parte imaginária, de forma que cada ponto do diagrama esteja associado biunivocamente a um único valor de  $z$ .

---

<sup>3</sup> Jean Robert Argand (1768-1822), Matemático francês. Seu artigo sobre o plano complexo foi publicado em 1806.

Dado um número complexo podemos calcular a distância  $|z|$  da origem ao ponto que representa o número no diagrama. Essa quantidade chamada amplitude (magnitude, valor absoluto ou módulo) do número complexo, é definida pela relação

$$|z| = \sqrt{x^2 + y^2} \quad (2.1)$$

Ao observar o diagrama, podemos notar que existe um ângulo entre o segmento que representa  $|z|$  e o eixo real positivo. Esse valor é conhecido como argumento do número complexo, denotado por  $Arg(z)$  ou fase  $\emptyset$ , e pode ter seu valor representado pela expressão

$$\emptyset = \tan^{-1}\left(\frac{y}{x}\right), \quad x \neq 0 \quad (2.2)$$

Se notarmos que na Fig. 2.2  $x = |z|\cos(\emptyset)$  e  $y = |z|\sen(\emptyset)$ , o número complexo  $z = x + iy$  pode ser reescrito na forma

$$z = x + iy = |z|\cos(\emptyset) + i|z|\sen(\emptyset), \quad (2.3)$$

ou seja,

$$z = |z|[\cos(\emptyset) + i\sen(\emptyset)],$$

conhecida como forma polar ou trigonométrica de  $z$ .

Além disso, se utilizarmos a fórmula de Euler<sup>4</sup> ( $e^{i\phi} = \cos(\phi) + i\text{sen}(\phi)$ ) combinada com a Eq. (2.3), obtemos a expressão

$$z = |z|[\cos(\phi) + i \text{sen}(\phi)] = |z|e^{i\phi}, \quad (2.4)$$

conhecida como a forma exponencial de  $z$ .

Neste ponto convém mencionar que no subcapítulo anterior definimos a fase  $\phi$  como o argumento de uma função periódica. Se observarmos que na representação do plano complexo,  $\phi = \phi + 2n\pi$ , e a representação polar e exponencial do número complexo tem  $\phi$  como argumento, concluímos que ambas apresentam caráter periódico, justificando a utilização dessa terminologia.

## 2.2 REPRESENTAÇÃO DE FUNÇÕES EM ESPECTROS DE FREQUÊNCIA

Qualquer função periódica, por mais complexa que seja, pode ser decomposta como a soma de uma série de funções senoidais (ou cossenoidais) cujas frequências são múltiplos inteiros da frequência fundamental  $1/T$ . Esses componentes de onda são conhecidos como harmônicos e obedecem a relação  $f_n = n/T$  (com  $n=1,2,3,\dots$ ).

A Fig. 2.3 A e B contém um exemplo de função representada como a soma de seus harmônicos.

---

<sup>4</sup> Matemático e físico suíço, falecido em 1783, fez descobertas em campos variados como análise matemática, cálculos e grafos além de contribuições em óptica, mecânica, e astronomia.

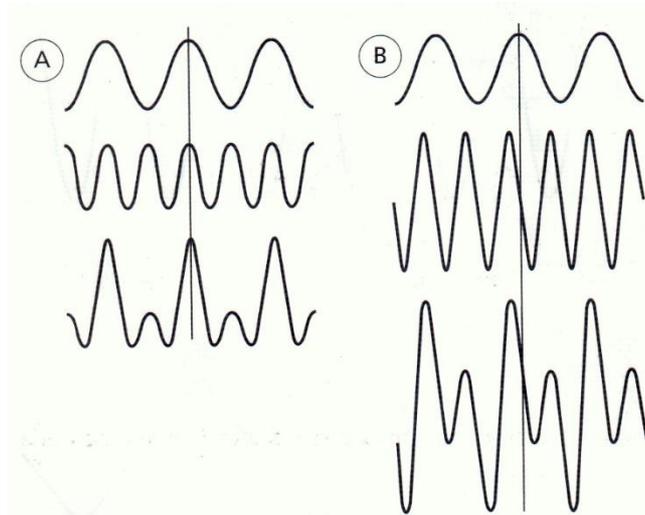


Fig. 2.3 Formas de onda resultantes da soma de duas componentes senoidais de frequência  $f$  e  $2f$ . (A) As duas componentes possuem a mesma amplitude e estão em fase; (B) A componente de frequência  $2f$  tem o dobro da amplitude da componente de frequência  $f$  e está defasada em  $\pi/2$ .

Para expressar qualquer forma de onda em termos de suas componentes de ondas senoidais, é necessário definir, além da frequência de cada componente, sua amplitude e fase.

Conseqüentemente, uma onda periódica pode ser expressa como função do tempo ou no domínio da frequência, neste segundo caso, expressando a amplitude e fase das ondas senoidais constituintes como uma função da frequência.

Na Fig. 2.4 estão representadas as formas de onda senoidais da Fig. 2.3 em termos de seus espectros de amplitude e fase.

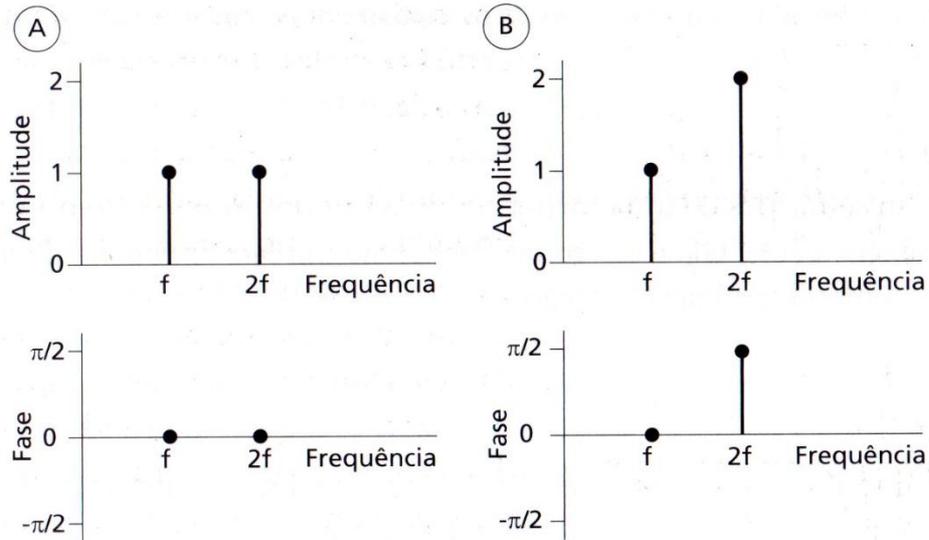


Fig. 2.4 Representação no domínio da frequência das formas de onda dos harmônicos ilustrados na Fig. 2.3 (A) e (B), mostrando seus respectivos espectros de amplitude e fase.

A operação conhecida como Transformação (ou Transformada) de Fourier [13] pode ser utilizada para converter uma função  $g(t)$ , no domínio do tempo, em suas amplitudes e espectros de fase equivalentes  $A(f)$  e  $\phi(f)$  ou em uma função complexa da frequência  $G(f)$ , chamada de espectro de frequência:

$$G(f) = A(f)e^{i\phi(f)} \quad (2.5)$$

A Transformada de Fourier também pode ser utilizada para a análise de funções dependentes da posição. Neste caso, o tempo é substituído pela posição e a frequência por uma frequência espacial. Como veremos adiante, essa forma de representação será particularmente útil para os objetivos propostos neste trabalho.

## 2.3 ANÁLISE DE ESPECTROS DE FASE NO PROCESSAMENTO DE IMAGENS

A representação e análise de imagens em espectros de amplitude e fase tem sido extensivamente utilizada em diversas aplicações como detecção de bordas [14], segmentação de imagem [15], ressonância magnética [16] e perfilometria interferométrica [17].

Neste contexto, em 1981, Oppenheim e Lim [18] publicaram um trabalho clássico ilustrando a importância da fase como componente essencial no processamento de imagens.

Para demonstrar esse fato, apresentaremos um experimento simples que consiste na alteração dos valores de fase e amplitude de uma imagem, de forma a corrompê-los, para posterior análise qualitativa do impacto dessas alterações.

Para isso, utilizaremos transformações que permitem a representação da imagem em valores de amplitude e fase, alteraremos essas representações e as resintetizaremos para reconstruir a imagem e comparar o resultado obtido com a informação original.

De forma mais detalhada, podemos definir uma transformação contínua bidimensional de Fourier [13, 19], capaz de transportar a informação contida em um espaço de representação para outro através das relações

$$F(q, p) = \frac{1}{2\pi} \iint_{-\infty}^{\infty} f(x, y) e^{-i2\pi(qx+py)} dx dy \quad (2.6)$$

$$f(x, y) = \frac{1}{2\pi} \iint_{-\infty}^{\infty} F(q, p) e^{i2\pi(qx+py)} dq dp \quad (2.7)$$

Onde  $x$  e  $y$  podem representar, por exemplo, um deslocamento espacial na direção horizontal e vertical respectivamente e  $q$  e  $p$  uma frequência espacial da grandeza física representada.

Similarmente, definiremos uma transformação para espaços discretos com  $N \times M$  elementos fundamentais.

Neste caso, a transformação conhecida como Transformação (ou Transformada) Discreta de Fourier (TDF), poderá ser representada pelas relações

$$F(k, l) = \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} f(n, m) e^{-i2\pi\left(\frac{kn}{N} + \frac{lm}{M}\right)}, \quad \begin{array}{l} k=0, 1, \dots, N-1 \\ l=0, 1, \dots, M-1 \end{array} \quad (2.8)$$

$$f(n, m) = \frac{1}{NM} \sum_{k=0}^{N-1} \sum_{l=0}^{M-1} F(k, l) e^{i2\pi\left(\frac{kn}{N} + \frac{lm}{M}\right)}, \quad \begin{array}{l} n=0, 1, \dots, N-1 \\ m=0, 1, \dots, M-1 \end{array} \quad (2.9)$$

Onde  $n$  e  $m$  representam os índices dos elementos fundamentais no domínio da posição e  $k$  e  $l$  no domínio da frequência.

Podemos entender  $f(n, m)$  como a  $n, m$ -ésima parte da função contínua  $f(x, y)$  no intervalo  $x = n\Delta x$  e  $y = m\Delta y$ ,  $F(k, l)$  como uma aproximação do espectro contínuo  $F(q, p)$ , com frequência espacial  $f_k = k/N\Delta x$  e  $f_l = l/M\Delta y$ , sendo  $\Delta x$  e  $\Delta y$  as medidas dos elementos fundamentais do espaço de representação de posição e  $N$  e  $M$  a quantidade

total desses elementos na direção horizontal e vertical respectivamente, em ambos os domínios.

Definiremos também o operador  $\mathfrak{F}$  de Transformação de Fourier e  $\mathfrak{F}^{-1}$  de Transformação Inversa de Fourier de modo que para o caso contínuo, teremos:

$$\mathfrak{F}[f(x, y)] = F(q, p) \quad (2.10)$$

$$\mathfrak{F}^{-1}[F(q, p)] = f(x, y) \quad (2.11)$$

$$\mathfrak{F}^{-1}\{\mathfrak{F}[f(x, y)]\} = f(x, y) \quad (2.12)$$

E para o caso discreto:

$$\mathfrak{F}[f(n, m)] = F(k, l) \quad (2.13)$$

$$\mathfrak{F}^{-1}[F(k, l)] = f(n, m) \quad (2.14)$$

$$\mathfrak{F}^{-1}\{\mathfrak{F}[f(n, m)]\} = f(n, m) \quad (2.15)$$

De forma mais concreta, podemos interpretar a imagem digital da Fig. 2.5 como uma representação bidimensional discreta com  $N \times M$  elementos no domínio da posição

$f(n, m)$  (com as variáveis  $n$  e  $m$  representando respectivamente, a posição horizontal e vertical de seus elementos).

Se efetuarmos uma Transformada Discreta de Fourier Bidimensional dessa função<sup>5</sup>, obteremos o espectro de frequência espacial  $F(k, l)$ , representado na Fig. 2.6, que pode ser separado em outros dois espectros, um de amplitude  $A(k, l)$  e outro de ângulo de fase  $\varphi(k, l)$ , conforme as Fig. 2.7 e Fig. 2.8.

O espectro de amplitude pode ser obtido fazendo  $A(k, l) = |F(k, l)| = \sqrt{\{Im[F(k, l)]\}^2 + \{Re[F(k, l)]\}^2}$  onde  $Im[F(k, l)]$  e  $Re[F(k, l)]$  são operadores que retornam números reais que correspondem à parte imaginária e real de  $F(k, l)$  respectivamente.

O espectro de fase  $\varphi(k, l)$  pode ser obtido pela atuação do operador *arcotangente*<sup>6</sup>, através da relação  $\varphi(k, l) = \tan^{-1} \left\{ \frac{Im[F(k, l)]}{Re[F(k, l)]} \right\}$ , cujo valor calculado por computador varia entre  $(-\pi, \pi]$ , de modo que  $-\pi < \varphi(k, l) \leq \pi$  ou  $|\varphi(k, l)| \leq \pi$ .

---

<sup>5</sup> Esse cálculo pode ser efetuado por computador utilizando as relações 2.3 e 2.4 definidas anteriormente.

<sup>6</sup> Neste ponto, chamo a atenção para o fato do operador *arcotangente* retornar apenas valores de ângulo entre  $-\pi$  e  $\pi$  e não o valor real de  $\varphi$ . Em breve abordaremos este assunto com maior atenção.

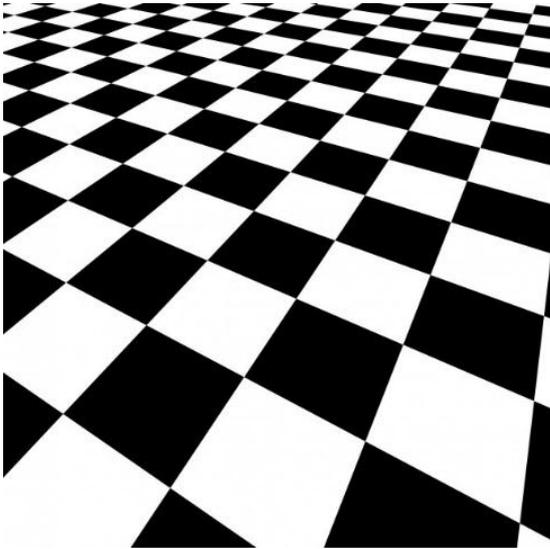


Fig. 2.5 Imagem original representada no domínio da posição.

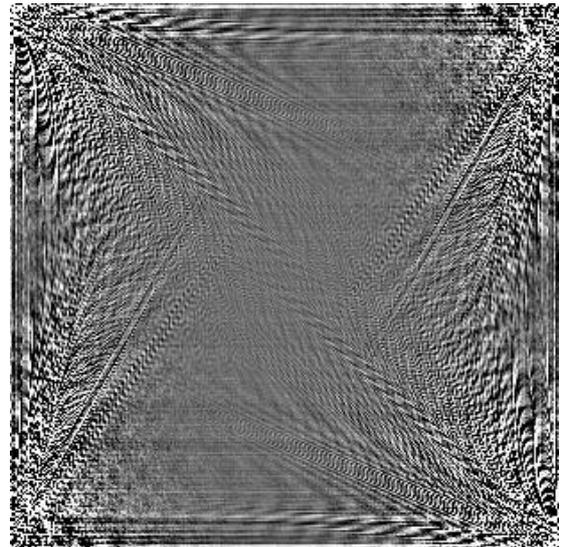


Fig. 2.6 Imagem da Fig. 2.5 representada no domínio da frequência espacial.

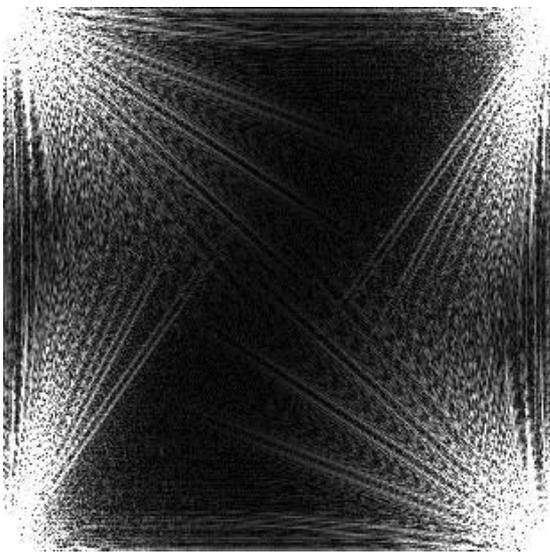


Fig. 2.7 Espectro de amplitude da Fig. 2.5 no espaço de frequência espacial.

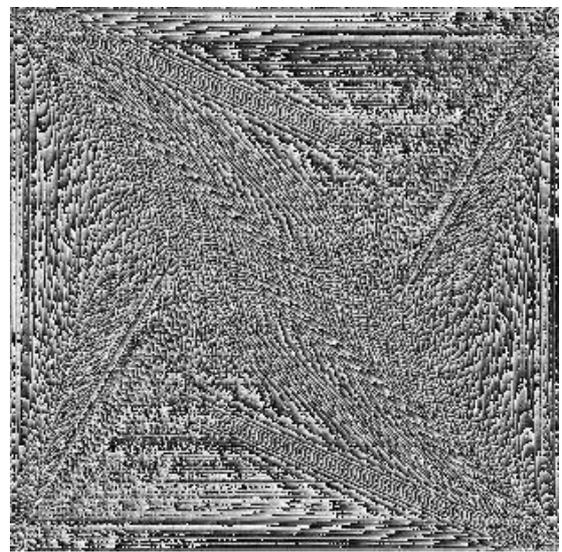


Fig. 2.8 Espectro de fase da Fig. 2.5 na representação de frequência espacial.

Ao utilizarmos uma Transformada Inversa de Fourier apenas sobre o espectro de amplitudes, ou seja, ao corromper a informação de fase e transportarmos a imagem para o domínio da posição, obtemos o resultado apresentado na Fig. 2.9.

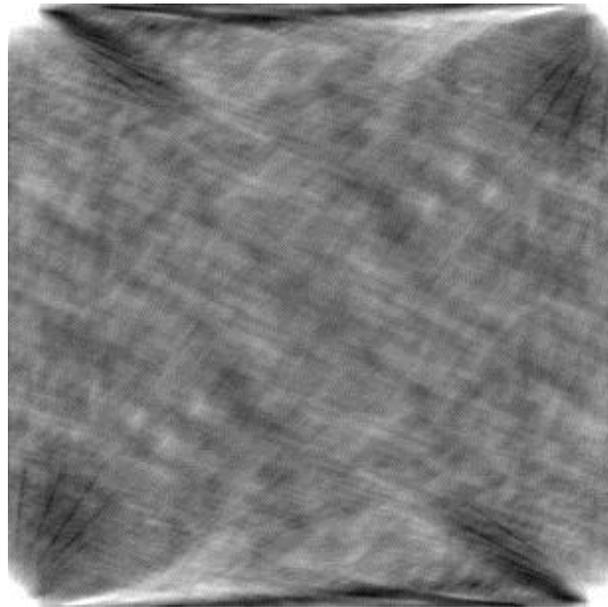


Fig. 2.9 Imagem com informação de fase corrompida, resintetizada para o domínio da posição.

Porém, ao sintetizarmos a imagem para o domínio da posição utilizando apenas a informação do espectro de fases, ou seja, corrompendo a informação de amplitude, obtemos o resultado apresentado na Fig. 2.10.

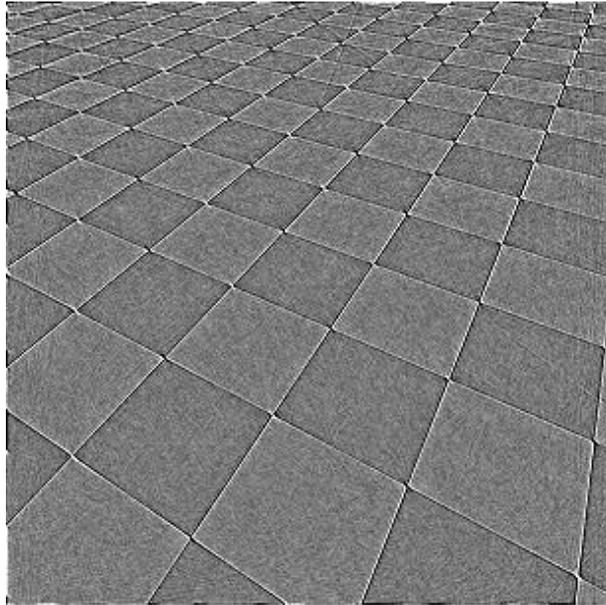


Fig. 2.10 Imagem com informação de amplitude corrompida, ressintetizada para o domínio da posição.

Qualitativamente é possível observar que a alteração dos valores de fase da imagem produziu um impacto muito maior sobre o resultado final. A alteração da amplitude diminuiu a nitidez e adicionou ruído à imagem, mas não comprometeu totalmente a sua legibilidade.

Esse simples experimento evidencia o impacto da alteração da fase no processamento de imagens e o cuidado que devemos ter ao tratar essa propriedade para evitar a perda de informações relevantes para a análise efetuada.

No próximo subcapítulo chamaremos a atenção para um problema advindo do processo de obtenção da fase, bem como indicaremos algumas aplicações físicas em que essa informação torna-se particularmente útil.

## **2.4 A FASE EMPACOTADA**

No experimento mencionado acima, para representar uma imagem em um espectro de valores de fase, efetuamos uma operação matemática que consistia na atuação do operador *arcotangente*  $\left(\varphi(k, l) = \tan^{-1} \left\{ \frac{\text{Im}[F(k, l)]}{\text{Re}[F(k, l)]} \right\}\right)$  sobre os pontos do espectro de frequência espacial.

O problema reside no fato de que computacionalmente, ao realizarmos uma operação dessa natureza, obteremos sempre um valor de fase no intervalo  $(-\pi, \pi]$  e não o seu valor verdadeiro<sup>7</sup>. A fase, neste caso, é dita estar empacotada (*wrapped*).

Em diversas aplicações o valor verdadeiro da fase está relacionado a quantidades físicas interessantes como o perfil de uma superfície ou à distorção de uma frente de onda [5] e a obtenção de seu valor verdadeiro pode tornar-se essencial para a análise do sistema físico, sendo necessário, portanto, a realização de um processo de desempacotamento de fase (*Phase Unwrapping*).

Atualmente o desempacotamento de fase tem aplicação em tecnologias como perfilometria interferométrica [1], imageamento por ressonância magnética (MRI) [2], topografia de superfície por radar de abertura sintética (inSAR) [4], medições não destrutivas de deformação e vibração de componentes eletrônicos [20] entre outras [21-24].

Devido à sua importância, o desenvolvimento de técnicas para o desempacotamento da fase é objeto de vasta pesquisa. No próximo capítulo veremos seus fundamentos bem como algumas das técnicas mais utilizadas e os principais desafios envolvidos no processo.

---

<sup>7</sup> Isso acontece, pois como vimos no capítulo anterior, para uma função periódica trigonométrica, por exemplo, a cosseno, temos  $\cos(\vartheta) = \cos(\vartheta + 2\pi n)$  com  $n \in \mathbb{Z}$ . Devido a essa impossibilidade matemática de distinção, ao efetuar o cálculo computacional do arcotangente de um ângulo, os valores são interpretados como tendo seu módulo sempre menor ou igual a  $2\pi$ , ou seja, a parte  $2\pi n$  é desconsiderada já que numericamente o resultado é o mesmo.

## Capítulo 3

# FUNDAMENTOS DO DESEMPACOTAMENTO DE FASE

*Neste capítulo serão apresentados os fundamentos do desempacotamento de fase, bem como alguns dos métodos mais utilizados atualmente, além de apontar as vantagens e dificuldades de cada um como forma de apresentar ao leitor uma visão do estado da arte das pesquisas na área.*

No capítulo anterior demonstramos um processo para obtenção de informação da fase que consistia na atuação do operador *arcotangente* sobre os pontos de uma imagem representada no domínio da frequência espacial. Esse operador retornava valores dito estarem empacotados no intervalo de  $-\pi$  a  $\pi$  e não os valores verdadeiros da fase.

Foi mencionado também que a fase pode estar relacionada a quantidades físicas como o perfil de uma superfície ou medições não destrutivas em deformações de componentes, sendo necessário, portanto, o desenvolvimento de técnicas para seu desempacotamento.

Neste capítulo iremos explicitar os fundamentos do desempacotamento de fase e como forma de fornecer um panorama das pesquisas na área, detalharemos algumas das técnicas mais utilizadas, juntamente com as dificuldades e desafios envolvidos no processo, além de apontar vantagens, desvantagens e indicar os cenários mais apropriados para sua utilização.

Por fim, como grande parte das aplicações trabalham com a análise de imagens, as demonstrações apresentadas, em sua maioria, envolverão funções em espaços bidimensionais, mas que podem ser generalizadas para N dimensões.

### **3.1 O DESEMPACOTAMENTO BIDIMENSIONAL DE FASE (2D PHASE UNWRAPPING)**

De forma geral, em duas dimensões, podemos representar o processo de extração da informação da fase como a atuação de um operador de empacotamento  $\mathcal{W}$  sobre uma função  $\psi(x, y)$  através da relação:

$$\varphi(x, y) = \mathcal{W}[\psi(x, y)], \quad (3.1)$$

onde  $\varphi(x, y)$  é uma função empacotada no intervalo  $(-\pi, \pi]$  e  $\psi(x, y)$  é uma função que contém informação da fase verdadeira a ser determinada.

Como um exemplo de  $\mathcal{W}$ , podemos citar o operador *arcotangente*, utilizado no capítulo 2.

A fase empacotada no intervalo  $(-\pi, \pi]$  também pode ser representada através da relação:

$$\varphi(x, y) = \psi(x, y) - 2\pi g(x, y), \quad (3.2)$$

onde  $g(x, y)$  é uma função inteira que faz com que  $-\pi < \varphi(x, y) \leq \pi$ .

O processo de desempacotamento de fase basicamente procura, tendo por base os valores de fase  $\varphi(x, y)$ , estimar os valores para a função  $g(x, y)$  de modo a obtermos a fase verdadeira  $\psi(x, y)$  que, como já mencionada antes, é objeto de interesse físico em inúmeras aplicações.

Um caso simples é o demonstrado na Fig. 3.1, com uma função unidimensional discretizada  $\varphi(n)$ , cuja variável  $n$  varia de 0 a N e representa uma fase empacotada no intervalo  $(-\pi, \pi]$ .

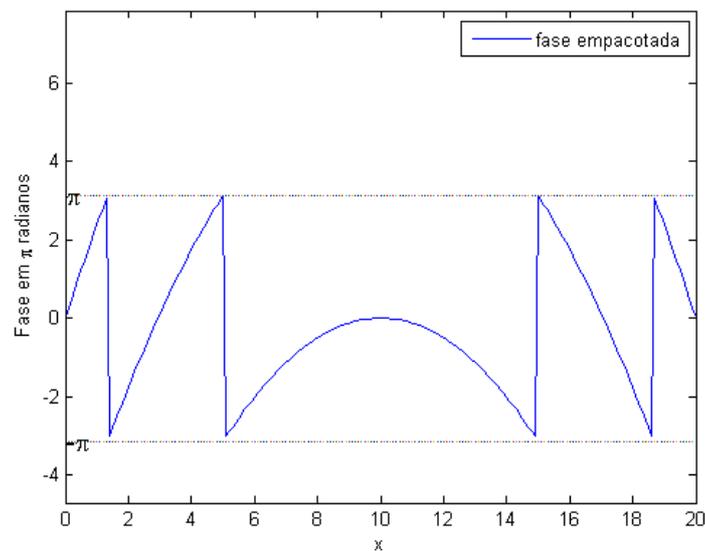


Fig. 3.1 Gráfico da fase empacotada de uma amostra com N pontos.

Para este caso não é difícil estimar o valor da função  $g(n)$  e obter o valor verdadeiro da fase  $\psi(n)$ , uma vez que basta identificar as regiões com descontinuidades de módulo igual a  $2\pi$  e somar a uma fase de ajuste.

Na Fig. 3.2 é possível observar a fase de ajuste, neste caso, uma função degrau cujos saltos são múltiplos de  $2\pi$ .

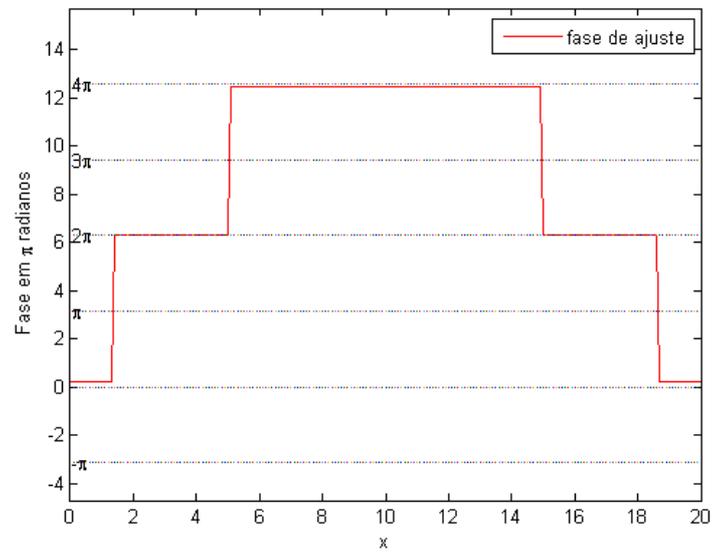


Fig. 3.2 Fase de ajuste a ser somada à fase empacotada.

Na Fig. 3.3 e Fig. 3.4, podemos observar que a fase empacotada ao ser somada à fase de ajuste, recupera seu valor verdadeiro.

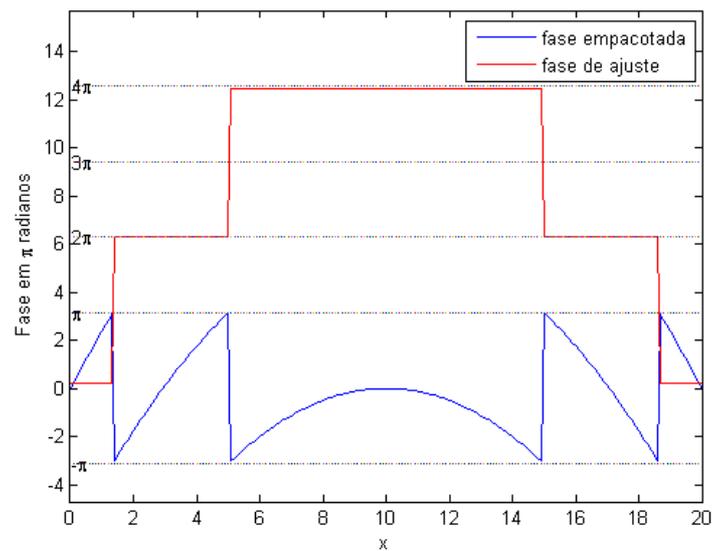


Fig. 3.3 Gráfico comparativo entre a fase de ajuste e a fase empacotada. A soma das duas funções produz a função de fase verdadeira da Fig 2.4.

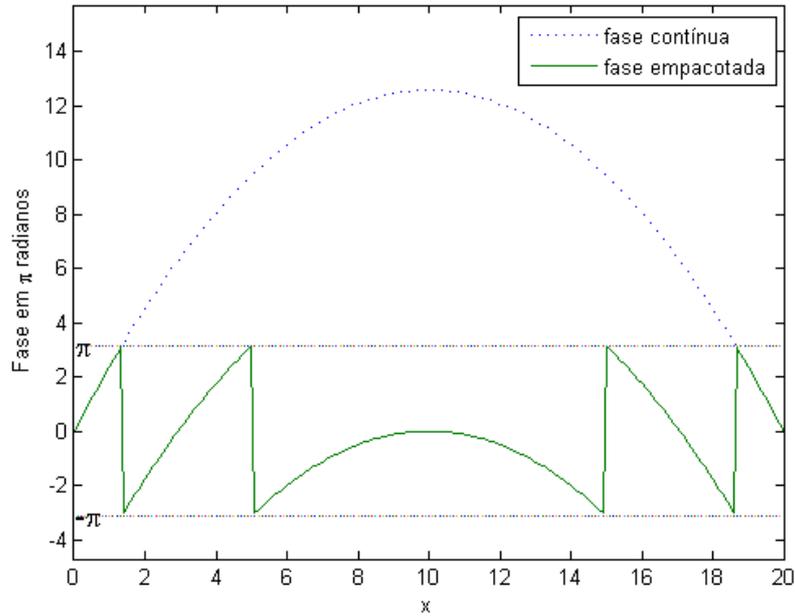


Fig. 3.4 Função de fase empacotada e função de fase verdadeira (resultado da soma da fase empacotada com a fase de ajuste), para comparação.

Matematicamente, o que fizemos foi estimar uma função de ajuste  $\varphi_a(n)$  de modo que:

$$\varphi_a(n) = 2\pi g(n), \quad (3.3)$$

e soma-la à função de fase empacotada  $\varphi(n)$  para obtermos  $\psi(n)$  conforme as relações abaixo:

$$\varphi(n) + \varphi_a(n) = \psi(n) - 2\pi g(n) + 2\pi g(n) = \psi(n) \quad (3.4)$$

$$\varphi(n) + \varphi_a(n) = \psi(n) \quad (3.5)$$

De forma mais detalhada, para obtenção da fase de ajuste, calculamos a diferença de fase entre o  $n$ -ésimo ponto e o seu precedente:

$$\Delta\varphi_d(n) = \varphi(n) - \varphi(n - 1) \quad (3.6)$$

com  $n$  variando de 1 a  $N$ .

Como a variação da fase é lenta se comparada ao intervalo de amostragem, o módulo da diferença de fase  $|\Delta\varphi_d(n)|$  é muito menor do que  $2\pi$  para pontos onde a distribuição de fase é contínua. Nos pontos onde uma descontinuidade (salto de fase) acontece, esse valor torna-se aproximadamente  $2\pi$ . Portanto, configurando-se um critério apropriado para a diferença de fase absoluta, pode-se especificar todos os pontos para os quais o salto de fase de  $2\pi$  acontece e também a direção de cada salto de fase que é definida pelo sinal de  $\Delta\varphi_d(n)$ .

Dessa forma, o próximo passo é a determinação da fase de ajuste  $\varphi_a(n)$  para cada ponto de  $\varphi(n)$ . Como somente uma distribuição de fase relativa necessita ser determinada, define-se inicialmente  $\varphi_a(0) = 0$  e configura-se:

$$\varphi_a(n) = \varphi_a(n - 1), \quad (3.7)$$

para  $n = 1, 2, 3, \dots, k - 1$ . Procedendo desta maneira até que o primeiro salto de fase seja detectado no  $k$ -ésimo ponto. Se a direção do salto de fase é positiva, ou seja,  $\Delta\varphi_d(k) > 0$ , configura-se:

$$\varphi_a(k) = \varphi_a(k - 1) - 2\pi. \quad (3.8)$$

Se é negativa ( $\Delta\varphi_d(k) < 0$ ), temos:

$$\varphi_a(k) = \varphi_a(k - 1) + 2\pi \quad (3.9)$$

Novamente, inicia-se o processo fazendo  $\varphi_a(n) = \varphi_a(n - 1)$  para  $n = k + 1, k + 2, \dots, m - 1$ , até que o próximo salto de fase ocorra no  $m - \acute{e}simo$  ponto, onde será realizada a mesma operação de adição ou subtração de  $2\pi$  feita no  $k - \acute{e}simo$  ponto, com  $k$  agora sendo trocado por  $m$ .

Repetindo este procedimento para todos os pontos com salto de fase de  $2\pi$  pode-se determinar a distribuição de fase de ajuste, conforme a Fig. 3.2.

A distribuição de fase contínua, resultado da soma da distribuição de fase descontínua com a distribuição de fase de ajuste é mostrada na Fig. 3.4.

No caso da medição sobre um plano bidimensional completo, uma operação de continuidade de fase na direção  $m$  seria necessária, pois inicialmente foi configurado  $\varphi_a(0, m) = 0$  para todo  $m$  sem se preocupar com a distribuição de fase nesta direção. Desta forma é suficiente determinar uma distribuição de fase de ajuste adicional apenas em uma linha ao longo do eixo  $m$ , ou seja, uma linha que passe por um  $n=c$ , onde  $c$  é uma constante arbitrária. Isso pode ser feito pelo mesmo procedimento descrito para a direção de variação de  $n$ , com o valor inicial agora definido por  $\varphi_a(c, 0) = 0$ .

Quando realizado sobre funções que, desconsiderando as descontinuidades de módulo  $2\pi$ , são bem comportadas, este procedimento é suficiente para a obtenção da função contínua com os valores verdadeiros da fase.

## 3.2 DESAFIOS DO DESEMPACOTAMENTO DE FASE

Conforme vimos no subcapítulo anterior, o desenvolvimento de um algoritmo para desempacotamento da fase é uma tarefa relativamente fácil se considerarmos a análise de funções que, à exceção dos saltos de fase de  $2\pi$ , são bem comportadas.

O problema surge quando, ao invés de analisarmos situações idealizadas, passamos a considerar a informação obtida de forma experimental.

No campo de imagens, por exemplo, há a presença de ruídos provenientes de interferência atmosférica em experimentos com Radar de Abertura Sintética (inSAR) [4], descontinuidades e variações na refletância de superfícies, falta ou iluminação disforme e falseamento (*aliasing*) devido à baixa taxa amostragem na conversão de sinal analógico para digital em experimentos de interferometria óptica [25], além de limitações técnicas dos instrumentos de medição, etc, que podem comprometer o processo, ocasionando a propagação de erro e deformação da informação original, podendo até gerar a falha global do desempacotamento.

Para ilustrar o impacto destes fatores sobre o desempacotamento, adicionamos um ruído gaussiano à fase empacotada da Fig. 3.1, conforme mostrado nas Fig. 3.5 a Fig. 3.7. É perceptível a diferença nos resultados obtidos ao realizar o desempacotamento com e sem ruído. A presença de ruído gerou falsas descontinuidades<sup>8</sup> e alterou as já existentes, afetando significativamente o resultado final do processo de desempacotamento.

---

<sup>8</sup> Neste caso, por falsas descontinuidades, entenda como aquelas que não foram produzidas pelo processo de empacotamento nem estavam presentes na função antes desta ser empacotada .

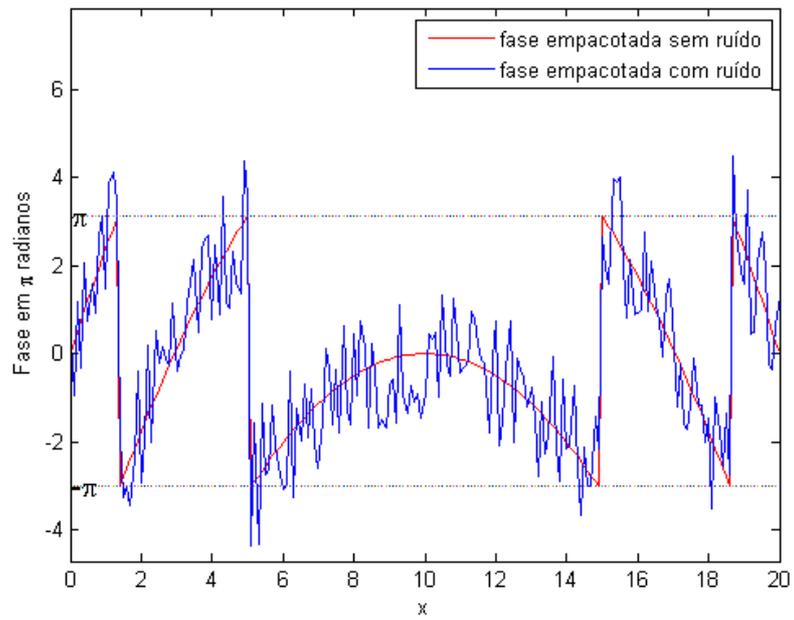


Fig. 3.5 Função empacotada com ruído e sem ruído para comparação.

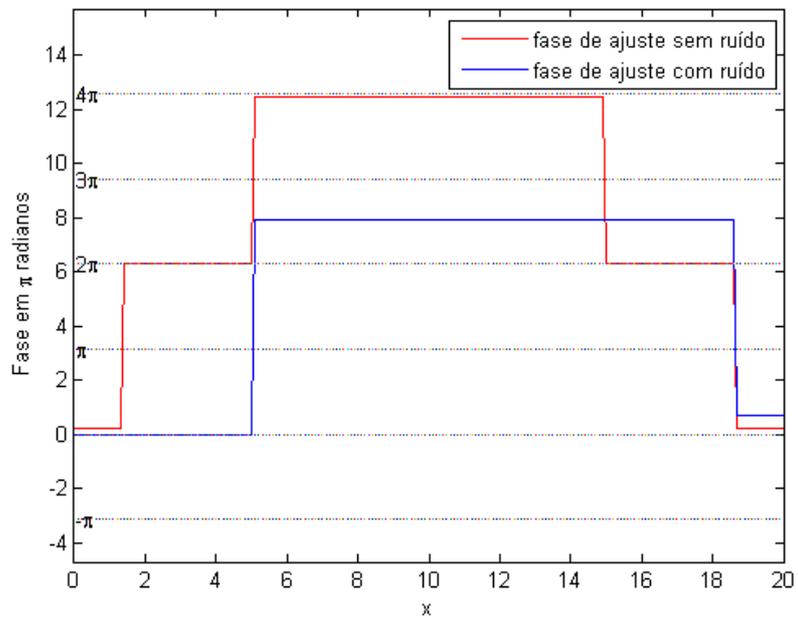


Fig. 3.6 Fase de ajuste para a fase empacotada sem a presença de ruído e com a presença de ruído.

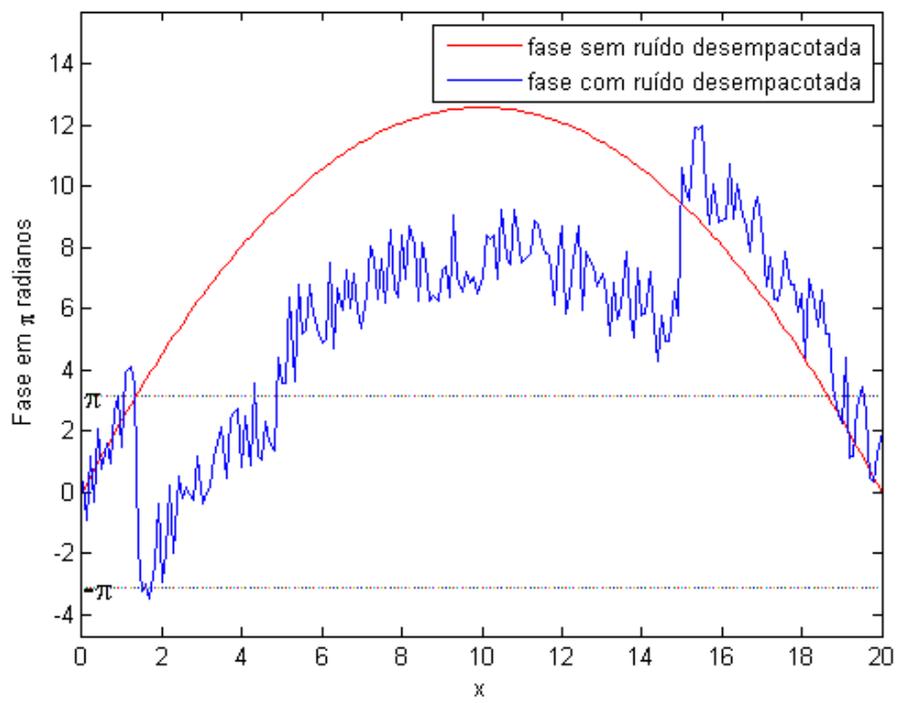


Fig. 3.7 Fase desempacotada com e sem ruído.

O problema apresentado pode ser matematizado de forma mais rigorosa, através da chamada condição de Itoh [26] que garante que o desempacotamento resultará no valor correto da fase apenas se o módulo da diferença de fase verdadeira entre os píxeis vizinhos for menor ou igual a  $\pi$ , ou seja:

$$|\Delta\psi(n)| \leq \pi \quad (3.10)$$

Partindo dessa premissa, ao atuar com o operador de empacotamento  $\mathcal{W}$  anteriormente definido pelas equações (3.1) e (3.2), sobre uma função  $\psi(n)$  contendo a fase verdadeira, obtemos:

$$\mathcal{W}\{\psi(n)\} = \varphi(n) = \psi(n) - 2\pi k(n), \quad n = 0, 1, \dots, N - 1, \quad (3.11)$$

Onde  $k(n)$  é uma função inteira de modo que

$$-\pi < \varphi(n) \leq \pi. \quad (3.12)$$

Também podemos definir o operador  $\Delta$  de forma que:

$$\begin{aligned} \Delta\{\psi(n)\} &= \psi(n + 1) - \psi(n), \\ \Delta\{k(n)\} &= k(n + 1) - k(n), \quad n = 0, 1, \dots, N - 2. \end{aligned} \quad (3.13)$$

Ao calcular a diferença de fase empacotada através da atuação de  $\Delta$  sobre a equações (3.11), conforme as relações em (3.13), obtemos

$$\Delta\{\mathcal{W}\{\psi(n)\}\} = \Delta\{\psi(n)\} - 2\pi\Delta\{k_1(n)\} \quad (3.14)$$

Se atuarmos novamente com operador de empacotamento sobre a (3.14), de forma similar à equação (3.11), obtemos:

$$\begin{aligned} \mathcal{W}\{\Delta\{\mathcal{W}\{\psi(n)\}\}\} &= \Delta\{\mathcal{W}\{\psi(n)\}\} - 2\pi k_2(n) \\ &= \Delta\{\psi(n)\} - 2\pi\Delta\{k_1(n)\} - 2\pi k_2(n) \\ &= \Delta\{\psi(n)\} - 2\pi[\Delta\{k_1(n)\} - k_2(n)] \end{aligned} \quad (3.15)$$

Onde os índices em  $k_1$  e  $k_2$  foram utilizados para distinguir as funções  $k(n)$  geradas pelas operações distintas de empacotamento.

Porém, como o operador  $\mathcal{W}$  retorna apenas valores no intervalo  $(-\pi, \pi]$ , a condição de Itoh definida pela Eq. (3.10), que pode ser escrita na forma

$$-\pi < \Delta\{\psi(n)\} \leq \pi,$$

implica que o termo  $2\pi[\Delta\{k_1(n)\} + k_2(n)]$  na Eq. (3.15) deve ser nulo, de modo que podemos escrever

$$\mathcal{W}\{\Delta\{\mathcal{W}\{\psi(n)\}\}\} = \mathcal{W}\{\Delta\{\varphi(n)\}\} = \Delta\{\psi(n)\}, \quad (3.16)$$

e se observarmos que da Eq. (3.13) podemos escrever

$$\Delta\{\psi(n)\} = \psi(n) - \psi(n-1), \quad n = 1, \dots, N-1, \quad (3.17)$$

sendo que, se calcularmos a soma da diferença de fase  $\Delta\{\psi(n)\}$  sobre os  $m$  termos de um

mapa de fases, obtemos

$$\begin{aligned}\sum_{n=1}^m \Delta\{\psi(n)\} &= \psi(1) - \psi(0) + \psi(2) - \psi(1) + \dots + \psi(m-1) + \\ &\psi(m) - \psi(m-1) = \psi(m) - \psi(0)\end{aligned}\quad (3.18)$$

ou

$$\psi(m) = \psi(0) + \sum_{n=1}^m \Delta\{\psi(n)\} = \psi(0) + \sum_{n=1}^m \mathcal{W}\{\Delta\{\varphi(n)\}\}.\quad (3.19)$$

A equação acima também pode ser estendida para N dimensões, inclusive em espaços contínuos. Para isso, assumindo que o gradiente de fase é conhecido juntamente com a fase verdadeira em algum ponto inicial  $\mathbf{r}_0$ , a fase verdadeira em algum outro ponto  $\mathbf{r}$  pode ser obtida através da integral de caminho:

$$\psi(\mathbf{r}) = \psi(\mathbf{r}_0) + \int_C \nabla\psi(\mathbf{r}) \cdot d\mathbf{r}\quad (3.20)$$

Onde C é qualquer caminho que conecte os pontos  $\mathbf{r}$  e  $\mathbf{r}_0$  em um espaço N-dimensional e  $\nabla\psi$  é o gradiente de fase neste espaço.

As Eqs. (3.19) e (3.20) definem um método para desempacotamento de fase conhecido como método de Itoh [26] no qual a fase verdadeira em qualquer ponto pode ser obtida através da soma ou integração dos valores de fase empacotados ou do gradiente de fase verdadeiro ao longo de um caminho, desde que conhecido o valor verdadeiro da fase para um ponto inicial do mapa de fases ( $\psi(\mathbf{r}_0)$ ).

Estas equações também estabelecem a condição de Itoh<sup>9</sup> (Eq. (3.10)) como requisito obrigatório para obtenção dos valores corretos da fase.

O método de Itoh consiste em um importante instrumento de base para algoritmos de desempacotamento, porém, conforme apresentado no início do capítulo, a presença de ruídos e falsas descontinuidades pode fazer com que a diferença entre píxeis vizinhos seja maior do que  $\pi$ , violando a condição de Itoh e provocando a falha do processo de desempacotamento, conforme exemplificado na Fig. 3.7.

Por este motivo, durante as últimas décadas foram desenvolvidas diversas pesquisas para a criação de algoritmos para estimar os valores verdadeiros da fase onde a condição de Itoh falha [10]. Estes algoritmos diferem basicamente em tempo de processamento e qualidade do resultado final. Em geral, uma maior imunidade a ruídos e fatores diversos exige maior poder computacional, acarretando em uma maior demora no processamento.

No próximo subcapítulo serão apresentados alguns algoritmos atualmente utilizados como forma de fornecer uma noção do progresso das pesquisas em desempacotamento de fase.

### **3.3 ALGORITMOS DE DESEMPACOTAMENTO DE FASE**

Devido ao grande número e variedade, torna-se inviável, mesmo que interessante, a apresentação de todos os algoritmos de desempacotamento de fase atualmente

---

<sup>9</sup> A condição de Itoh pode ser entendida como uma versão do teorema de amostragem de Nyquist [13], em que um sinal contínuo pode ser completamente reconstruído apenas se a frequência da amostragem possuir, no mínimo, o dobro da frequência do sinal, ou seja, os pontos amostrados devem apresentar uma diferença de fase máxima de  $\pi$  radianos.

disponíveis.

Como forma de fornecer uma perspectiva das atuais pesquisas na área, neste capítulo serão apresentados alguns dos algoritmos frequentemente referenciados na comunidade acadêmica, que por suas características, serão classificados em duas grandes categorias: algoritmos espaciais e temporais.

Nos algoritmos temporais [27-30] o mapa de fases verdadeiras é obtido através da análise da relação entre mapas de fase empacotados resultantes da projeção de padrões com diferentes frequências sobre o objeto de interesse.

Nos algoritmos espaciais [31-35] o mapa de fases verdadeiras é obtido através da análise de um único mapa de fase empacotado, seguindo por caminhos ao longo do mapa para realizar o desempacotamento.

### **3.3.1 ALGORITMOS DE MULTIFREQUÊNCIAS**

Entre os algoritmos temporais, podemos destacar os de multifrequências [36, 37]. Basicamente, nesses algoritmos, são projetados sobre o objeto vários conjuntos de padrões de franjas, deslocados entre si por uma fase e com diferentes frequências espaciais.

Com o auxílio da técnica de Deslocamento de Fase Perfilométrico (*Phase Shifting Profilometry*) [38-41], de cada grupo é obtido um mapa de fases empacotado. Ao analisar a relação entre os mapas obtidos, é possível obter um mapa de fases verdadeiras sem ruídos ou propagação de erro, uma vez que o processo de desempacotamento não analisa a relação do pixel a ser desempacotado com os píxeis vizinhos.

De forma mais detalhada, supondo que as imagens dos padrões de franjas projetados sobre um objeto tenham uma resolução de  $H \times V$  e que as franjas do  $n$ -ésimo

conjunto estejam dispostas verticalmente com uma frequência espacial  $f_n$  de acordo com as relações abaixo:

$$f_1 = \frac{1}{H} \quad (3.21)$$

e

$$f_n = 2f_{n-1}. \quad (3.22)$$

Como as franjas do primeiro grupo tem período igual à largura da imagem, não é necessário efetuar o desempacotamento do mapa de fases gerado e podemos definir a relação:

$$\psi_1(x, y) = \psi_1^R(x, y) \quad (3.23)$$

Onde  $\psi_1(x, y)$  refere-se ao mapa de fases verdadeiras e  $\psi_1^R(x, y)$  ao mapa de fases empacotado, obtido do primeiro conjunto de padrões de franjas.

Para o segundo conjunto de padrões de franjas temos:

$$\psi_2(x, y) = \psi_2^R(x, y) + 2\pi k_2(x, y) \quad (3.24)$$

Onde  $k_2(x, y)$  é um inteiro que representa um múltiplo de  $2\pi$  de uma fase de ajuste a ser adicionada à fase empacotada para obtermos a fase verdadeira.

Além disso, da técnica de Deslocamento de Fase Perfilométrico [38-41], para cada conjunto de franjas projetadas, temos a relação:

$$h(x, y) = \frac{l_o \psi(x, y)}{\psi(x, y) - 2\pi f_o d_o}, \quad (3.25)$$

onde  $l_o$  é a distância entre a câmera de captura e o plano de referência,  $d_o$  é a distância entre a câmera e a fonte de projeção das franjas,  $f_o$  é a frequência espacial do padrão de franjas e  $h(x, y)$  é a distância da superfície do objeto iluminado ao plano de referência.

Utilizando as Eqs. (3.22) e (3.25), temos:

$$\psi_2(x, y) = 2\psi_1(x, y) \quad (3.26)$$

De modo que, de acordo com a Eq. (3.27):

$$k_2(x, y) = R \left[ \frac{\psi_1(x, y)}{\pi} - \frac{\psi_2^R(x, y)}{2\pi} \right], \quad (3.27)$$

onde o operador R retorna o inteiro mais próximo de seu argumento (uma vez que a função  $k_2(x, y)$  deve ser inteira).

Após obter  $\psi_2(x, y)$  o processo é repetido para a obtenção de  $\psi_3(x, y)$  e assim sucessivamente até que todos os conjuntos de mapas tenham sido desempacotados conforme as relações abaixo:

$$k_n(x, y) = R \left[ \frac{\psi_{n-1}(x, y)}{\pi} - \frac{\psi_n^R(x, y)}{2\pi} \right] \quad (3.28)$$

$$\psi_n(x, y) = \psi_n^R(x, y) + 2\pi k_n(x, y) \quad (3.29)$$

Ao observarmos as Eqs. (3.28) e (3.29), podemos perceber que o valor de fase de cada pixel é obtido sem a interferência espacial dos píxeis vizinhos, de modo que ruídos e falsas descontinuidades não são propagados pelo mapa durante o processo de desempacotamento.

Apesar da resistência a ruídos, este método exige alto poder computacional, uma vez que as pesquisas atuais apontam a necessidade de aproximadamente 3 grupos com frequências espaciais diferentes, cada um com no mínimo 2 padrões de franjas com fases deslocadas entre si para obtenção de um mapa de fases verdadeiras com resultados satisfatórios [36].

Essa técnica não é indicada para situações dinâmicas, com vibrações ou movimento, uma vez que seria necessário a geração de mapas de fase deslocados em um intervalo de tempo muito curto de forma que a alteração da posição do objeto interfira minimamente na obtenção dos padrões de franjas.

### **3.3.2 ALGORITMOS DE CAMINHO NÃO GUIADOS**

São os algoritmos de desempacotamento de fase espaciais mais simples. Um exemplo é o algoritmo utilizado no subcapítulo 3.1. O processo basicamente consiste em uma varredura em direções horizontais e verticais para estimar uma fase de ajuste que somada ao mapa de fases empacotado, produz o mapa de fases verdadeiras.

Por sua simplicidade, exigem muito pouco tempo de processamento, mas devido à sua forma linear de execução do desempacotamento, conforme demonstrado no subcapítulo 3.2, são muito sensíveis a ruídos e distúrbios, uma vez que estes podem ocasionar a geração de falsas descontinuidades que acarretam a propagação de erros de

desempacotamento que podem, inclusive, causar a falha global do processo.

Geralmente são utilizados em mapas com muito pouco ou nenhum ruído e sem falsas descontinuidades.

### **3.3.3 ALGORITMOS GUIADOS POR QUALIDADE**

Nos algoritmos espaciais guiados por qualidade (*Quality Guided Algorithms*) [10, 42-48], cada ponto do mapa de fases é analisado segundo critérios pré-definidos de forma a estabelecer um mapa de qualidade, ou seja, uma matriz com parâmetros para mensurar o quanto o valor de cada pixel pode estar distorcido em relação ao valor que teria sem a presença de ruídos.

O desempacotamento do mapa de fases é efetuado com base nessa matriz para priorizar os píxeis com maior qualidade, uma vez que estes tendem a minimizar a propagação de erros por terem sido afetados em menor escala por fatores diversos.

Atualmente existem diversos algoritmos utilizados para a geração do mapa de qualidade, como os de Variância Derivativa de Fase [10], de Modulação de Fase [48] e os de Gradiente de Fase [45, 48].

Os algoritmos de Variância Derivativa de Fase avaliam a variação das diferenças de fase empacotadas em regiões do mapa de fase para detectar falsas descontinuidades de  $2\pi$  que podem prejudicar o processo de desempacotamento. Produzem bons resultados na detecção de falsos saltos de fase mas exigem muitos cálculos com alto custo computacional, não sendo indicados para processos que exigem um processamento rápido.

Os algoritmos de Modulação de Fase procuram estimar a refletância da superfície

através da análise dos píxeis da imagem do padrão de franjas projetado, para gerar uma matriz que indica em quais pontos há menor ou maior probabilidade do valor da fase estar corrompido por ruídos de iluminação.

Apesar de ser eficiente para localizar regiões com baixa refletividade e possuir baixo custo computacional por não exigir cálculos complexos, sendo assim indicado para situações em que se necessite de rápido processamento, não é efetivo na detecção de falsos saltos de fase.

Os algoritmos de Gradiente de Fase analisam o valor de fase de cada ponto em relação aos pontos vizinhos para avaliar se a fase teve seu valor alterado devido a presença de ruídos diversos de modo a indicar quais pontos devem ser priorizados no processo de desempacotamento. São efetivos na detecção de pontos com baixa qualidade para desempacotamento e seu custo computacional é menor do que os de Variância Derivativa de Fase, porém, são mais sensíveis a ruídos.

Existem diversos outros algoritmos para a geração de mapas de qualidade mas, em geral, basicamente indicam quais pontos são mais suscetíveis à propagação de erro e quais devem ser priorizados no processo de desempacotamento de modo a minimizar as distorções causadas por ruídos, falsas discontinuidades, iluminação disforme dentre outros fatores que venham a interferir na análise do mapa empacotado para a geração do mapa de fases verdadeiras.

Tendo estabelecido o mapa de qualidade, são utilizados algoritmos com estratégias de caminho para o desempacotamento.

### 3.3.4 ESTRATÉGIAS DE CAMINHO

Após a geração do mapa de qualidade, que indica quais pontos foram afetados em maior ou menor escala, são utilizadas estratégias de caminho para a realização do desempacotamento de fase.

Um bom algoritmo guiado por qualidade, utilizando-se desse mapa, procura minimizar distorções e propagações de erro durante o processo de desempacotamento. A estratégia adotada possui papel crucial na qualidade e confiabilidade do processo.

Entre as diversas estratégias de caminho desenvolvidas, apresentaremos a de inundação (*flood-fill*) [31, 33, 34] e escaneamento rápido de linha (*fast scan-line*) [35], por serem largamente utilizados em pesquisas na área e fornecerem uma ideia da forma de trabalho e complexidade de algoritmos dessa natureza.

#### 3.3.4.1 ALGORITMOS DE INUNDAÇÃO

Conforme já mencionado, os algoritmos de estratégia de caminho por inundação (*flood-fill*) utilizam o mapa de qualidade para desempacotar a fase através de caminhos que priorizem os píxeis afetados em menor escala por fatores diversos.

A escolha do mapa de qualidade utilizado como base para o *flood-fill* possui grande impacto no resultado final do desempacotamento e deve levar em conta fatores como o tipo de ruído envolvido, as características da superfície e do meio em que se encontra, além do tempo disponível ou desejado para a geração do mapa.

Para elucidar o processo, serão apresentados abaixo os passos básicos envolvidos

em um algoritmo de *flood-fill*.

Para isso, supondo que com base no mapa de fases, um mapa de qualidade  $Q(x,y)$  tenha sido previamente gerado por algum método, como por exemplo, os mencionados no subcapítulo anterior, temos:

Passo 1. Através do mapa  $Q(x,y)$  é efetuada a identificação do pixel empacotado com melhor qualidade e utilizado seu valor como sendo igual ao valor verdadeiro da fase.

O pixel é marcado como tendo sido desempacotado, conforme a Fig. 3.8 (a) e o cursor é então posicionado neste ponto para dar continuidade ao processo conforme os passos seguintes.

Passo 2. É observado o valor de qualidade dos quatro píxeis vizinhos ao cursor (superior, inferior, direito e esquerdo), que é então posicionado no pixel empacotado com maior valor de qualidade, conforme Fig. 3.8 (b) e (c).

Passo 3. Denotando por  $\phi_c(n)$  o valor da fase empacotada na atual posição do cursor,  $\phi_c(n-1)$  e  $\psi(n-1)$  o valor da fase empacotada e desempacotada na posição anterior, é calculado o valor da fase desempacotada  $\psi(n)$  na posição atual do cursor através da Eq. (3.30):

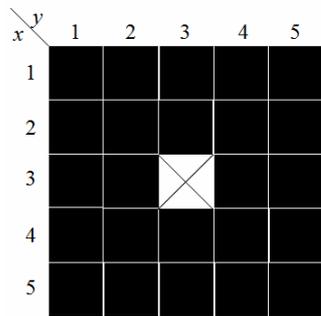
$$\psi(n) = \begin{cases} \psi(n-1) + \Delta\phi_c(n) + 2\pi, & \Delta\phi_c(n) \leq -\pi \\ \psi(n-1) + \Delta\phi_c(n), & -\pi < \Delta\phi_c(n) < \pi \\ \psi(n-1) + \Delta\phi_c(n) - 2\pi, & \Delta\phi_c(n) \geq \pi \end{cases} \quad (3.30)$$

Onde

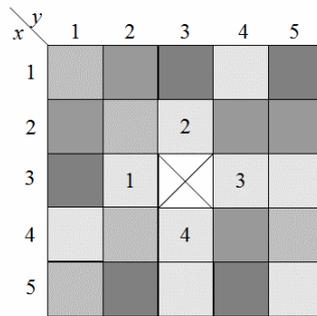
$$\Delta\phi_c(n) = \phi_c(n) - \phi_c(n-1), \quad (3.31)$$

conforme mostrado na Fig. 3.8 (d) a (f).

Após este cálculo, o pixel é marcado como tendo passado por desempacotamento e os passos 2 e 3 são repetidos, como demonstrado nas Fig. 3.8 (g) a (i), até que todos os



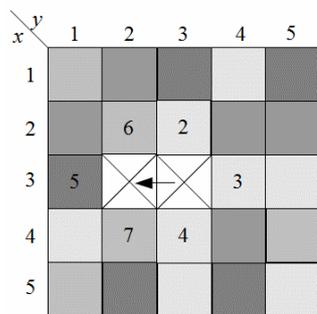
(a)



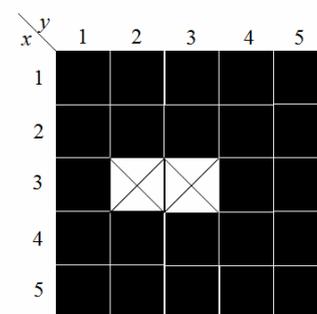
(b)

índice	posição (x,y)	valor de qualidade
3	(3,4)	0,69
4	(4,3)	0,76
2	(2,3)	0,80
1	(3,2)	0,90

(c)



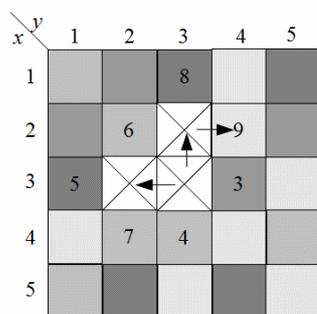
(d)



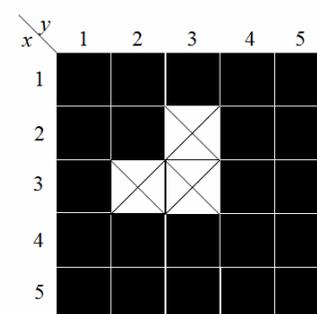
(e)

índice	posição (x,y)	valor de qualidade
7	(4,2)	0,55
5	(3,1)	0,63
3	(3,4)	0,69
6	(2,2)	0,72
4	(4,3)	0,76
2	(2,3)	0,80

(f)



(g)



(h)

índice	posição (x,y)	valor de qualidade
8	(1,3)	0,50
7	(4,2)	0,55
5	(3,1)	0,63
3	(3,4)	0,69
6	(2,2)	0,72
4	(4,3)	0,76
9	(2,4)	0,84

(i)

Fig. 3.8 (a-i) Exemplo de uma estratégia de caminho por inundação.

píxeis da imagem tenham sido desempacotados.

A vantagem dessa estratégia é que os pontos com melhor qualidade são priorizados, minimizando a propagação de erros no processo. A desvantagem reside no alto custo computacional necessário uma vez que para cada pixel são avaliados os píxeis vizinhos e o melhor caminho dentro do mapa de fases a ser seguido.

### **3.3.4.2 ALGORITMOS DE ESCANEAMENTO RÁPIDO DE LINHA**

Os algoritmos de escaneamento rápido de linha [35], também pertencentes à categoria de algoritmos espaciais, destacam-se pela rápida geração do mapa desempacotado.

Com base em faixas de valores de um mapa de qualidade  $Q(x,y)$  previamente gerado, os píxeis são classificados em níveis ou camadas. Para cada camada o processo de desempacotamento é realizado a partir de seu centro geométrico. Por exemplo, em um mapa de qualidade cuja a graduação dos píxeis seja decimal entre 0 e 1, sendo 1 para os píxeis com maior qualidade; os píxeis com qualidade entre 0,8 e 1 podem ser agrupados em uma camada, enquanto os píxeis com qualidade entre 0,60 e 0,79 em outra camada e assim sucessivamente, até que todos estejam agrupados por suas respectivas classificações.

O desempacotamento é efetuado varrendo a primeira camada, a partir de seu centro geométrico, de forma que sejam verificados apenas os píxeis com qualidade entre 0,80 e 1,00. Posteriormente, a próxima camada com píxeis cujo valor de qualidade esteja entre 0,60 e 0,79 é desempacotada e assim sucessivamente, até que todas as camadas tenham

sido verificadas e todos os píxeis desempacotados.

De forma mais detalhada, para a execução do algoritmo de escaneamento rápido são adotados os seguintes passos:

Passo 1. Com base no mapa de qualidade  $Q(x,y)$  os píxeis são classificados em camadas de qualidade. O cursor é então colocado em uma posição inicial  $(x_0, y_0)$  próxima ao centro da camada inicial, conforme mostrado na Fig. 3.9 (retirada de [35]).

Passo 2. A partir de  $(x_0, y_0)$  o cursor se move horizontalmente em direção às bordas da camada efetuando o desempacotamento de cada pixel de forma sequencial. Se o cursor encontrar algum pixel que pertença à outra camada, os píxeis restantes são colocados em uma lista e toda a linha é considerada escaneada.

Passo 3. Após toda a linha ter sido escaneada, o cursor retorna para o pixel inicial, move-se verticalmente para outra linha e efetua o escaneamento horizontal conforme descrito no passo 2. Este processo é repetido até que todas as linhas da camada atual tenham sido escaneadas.

Passo 4. Após o escaneamento de toda a camada, a fase verdadeira dos píxeis que foram colocados na lista do passo 2 são estimadas tendo por base os píxeis vizinhos que foram desempacotados.

Passo 5. O cursor vai para a próxima camada, em uma posição inicial  $(x_0, y_0)$ , próxima ao centro, e os passos 2, 3 e 4 são repetidos até que todas as camadas tenham sido escaneadas e todos os píxeis tenham sido desempacotados.

Em comparação com os algoritmos de inundação que fazem a análise de cada pixel comparando com outros do mapa, as estratégias de escaneamento rápido de linha, por utilizarem um esquema de desempacotamento por níveis, têm um gasto computacional e consequentemente o tempo de processamento necessário para a tarefa drasticamente reduzido, porém, essa estratégia é mais sensível à propagação de erros do que a de

inundação, não sendo indicado para a análise de superfícies com muitos distúrbios.

No próximo subcapítulo apresentaremos os algoritmos de balanceamento de resíduos (*branch-cut*), que por servirem de base para o trabalho desenvolvido nessa dissertação, terão uma explicação mais detalhada do que os apresentados até o momento.

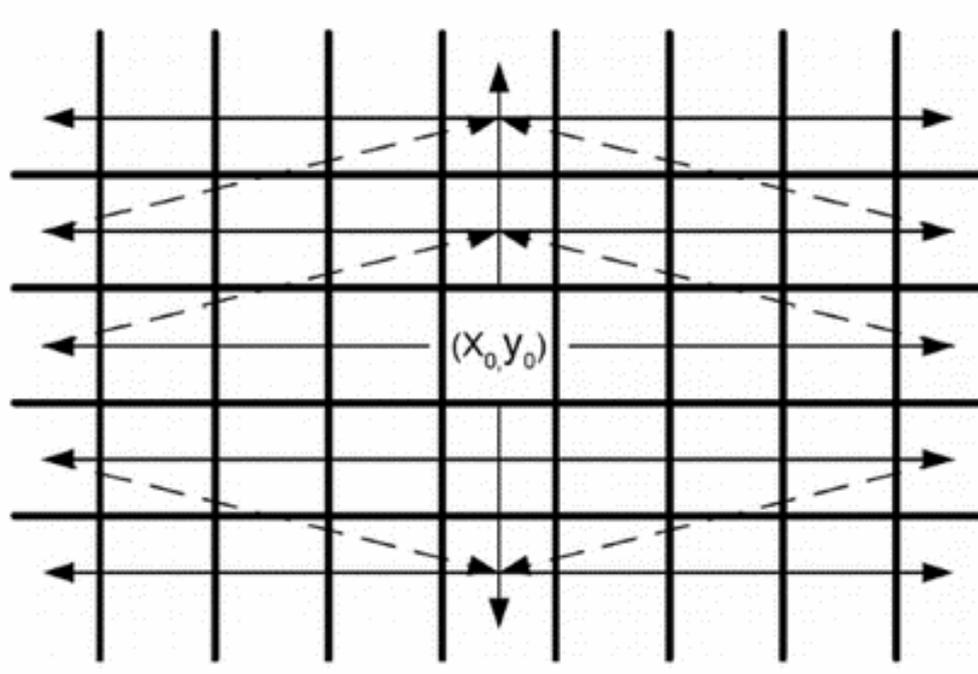


Fig. 3.9 Estratégia de um algoritmo de escaneamento rápido de linha (retirada de [35]).

### 3.3.5 ALGORITMOS DE BALANÇAMENTO DE RESÍDUOS

Também conhecidos como métodos de balanceamento de resíduos os algoritmos espaciais de *branch-cut* utilizam a teoria de resíduos para detectar e marcar a presença de distúrbios que possam comprometer o processo de desempacotamento.

Utilizando o método inicialmente proposto por Goldstein, Zebcker e Werner [49],

que através da teoria de resíduos, procura descontinuidades e ruídos que possam afetar o processo de desempacotamento, Huntley [32] descreve um método que utiliza barreiras, ou *branch-cut*, para delimitar regiões que não podem ser cruzadas durante o processo de desempacotamento, afim de minimizar a propagação de erros.

Para entrar em mais detalhes, vamos relembrar a Eq. (3.20) apresentada no subcapítulo 3.2, que serve de base para a obtenção da fase desempacotada:

$$\psi(\mathbf{r}) = \psi(\mathbf{r}_0) + \int_C \nabla\psi(\mathbf{r}) \cdot d\mathbf{r} , \quad (3.20)$$

onde a integral depende de um caminho  $C$  que liga  $\mathbf{r}_0$  a  $\mathbf{r}$  no domínio da função.

Uma pergunta que surge naturalmente ao observar a equação acima é se existem condições nas quais a integral é independente de caminho, ou seja, depende apenas dos pontos inicial e final de  $C$ , o que facilitaria em muito o processo de desempacotamento.

Podemos observar que a Eq. (3.20), a menos de uma constante aditiva, tem a forma:

$$I = \int_C \mathbf{F}(\mathbf{r}) \cdot d\mathbf{r} \quad (3.32)$$

Sendo que em duas dimensões podemos escrever  $\mathbf{F}(\mathbf{r}) = f(x, y)\hat{\mathbf{x}} + g(x, y)\hat{\mathbf{y}}$  e  $d\mathbf{r} = dx\hat{\mathbf{x}} + dy\hat{\mathbf{y}}$  onde  $\hat{\mathbf{x}}$  e  $\hat{\mathbf{y}}$  são vetores unitários na direção  $x$  e  $y$  respectivamente, de modo que a Eq. (3.32) pode ser reescrita na forma:

$$I = \int_C \mathbf{F}(\mathbf{r}) \cdot d\mathbf{r} = \int_C [f(x, y)dx + g(x, y)dy] \quad (3.33)$$

Utilizando a definição acima, Ghiglia [10] resume um conjunto de condições, mostradas no Quadro 1, que ao serem atendidas, garantem a independência de caminho para a Eq. (3.20):

**Quadro 1**  
Condições para a independência de caminho

<p><b>1.</b> <math>\mathbf{F}(\mathbf{r}) \cdot d\mathbf{r} = f(x, y)dx + g(x, y)dy</math>    <math>\mathbf{F}(\mathbf{r})</math> deve ser uma diferencial exata;</p>
<p><b>2.</b> <math>\mathbf{F}(\mathbf{r}) = \nabla\psi(r)</math>    <math>\mathbf{F}(\mathbf{r})</math> deve poder ser escrita como o gradiente de uma função escalar <math>\psi</math>;</p>
<p><b>3.</b> <math>\oint \mathbf{F}(\mathbf{r}) \cdot d\mathbf{r} = 0</math>    a integral de <math>\mathbf{F}(\mathbf{r})</math> ao longo de qualquer caminho fechado deve ser zero;</p>
<p><b>4.</b> <math>\nabla \times \mathbf{F}(\mathbf{r}) = 0</math>    o rotacional da função <math>\mathbf{F}(\mathbf{r})</math> deve ser identicamente zero.</p>

Na verdade, atender a uma destas condições implica que as outras também serão atendidas [13].

Seguindo esse raciocínio, se associarmos  $\psi(\mathbf{r})$  com a fase e  $\mathbf{F}(\mathbf{r})$  com o seu gradiente, imediatamente podemos ver a conexão entre a independência de caminho e o desempacotamento de fase, ou seja, se a fase satisfizer às quatro condições definidas, a Eq. (3.20) torna-se independente de caminho e o processo de desempacotamento torna-se trivial.

Infelizmente, os mapas de fase obtidos de forma experimental apresentam ruídos

e descontinuidades que frequentemente violam estas condições. O desempacotamento de fase torna-se desafiante uma vez que é necessário detectar inconsistências, estimar seus impactos e procurar formas de recuperar a independência de caminho para facilitar o processo de obtenção da fase verdadeira.

Como uma abordagem na busca da solução para este problema, vamos nos atentar à terceira condição de independência de caminho e chamar a atenção para o fato de que da teoria de funções de variáveis complexas, é sabido que o problema de calcular integrais de linha ao longo de caminhos fechados é equivalente ao problema algébrico de avaliar resíduos contornados por um caminho fechado.

Esta equivalência, conhecida como “teorema dos resíduos” para funções de variáveis complexas, pode ser definida conforme abaixo:

$$\oint f(z)dz = 2\pi i \sum_{k=1}^n r_k \quad (3.34)$$

Onde  $z$  é uma variável complexa da forma  $z = x + iy$ ,  $i = \sqrt{-1}$  e os termos  $r_k$  são os  $n$  resíduos ou coeficientes dos termos  $(z - z_k)^{-1}$  da expansão de Laurent [13] da função  $f(z)$  em torno dos pontos singulares  $z_k$ , envolvidos pelo contorno de integração. Além disso, a função  $f(z)$  é definida como analítica<sup>10</sup> nas vizinhanças do ponto singular. A palavra resíduo advém do fato de que apenas os termos  $(z - z_k)^{-1}$  contribuem para a integral de linha.

Utilizando-se dessa teoria, Ghiglia [10] faz um paralelo entre a teoria de resíduos de variáveis complexas e o problema de desempacotamento de fase bidimensional, ao

---

<sup>10</sup> A grosso modo, funções analíticas são aquelas que podem ser representadas por séries de potências.

definir o “*Teorema de Resíduos para o Desempacotamento Bidimensional de Fase*”, através da relação:

$$\oint \nabla\psi(\mathbf{r}) \cdot d\mathbf{r} = 2\pi \sum_{i=1}^n q_i , \quad (3.35)$$

onde  $q_i$  são  $n$  as cargas de resíduos envolvidas pelo contorno de integração.

A Eq. (3.35) introduz o conceito de cargas de resíduos, inicialmente proposto por Goldstein [49].

Diferentemente da Eq. (3.34), nesta os resíduos  $q_i$  não assumem qualquer valor, mas apenas os valores +1 ou -1. Isto ocorre devido à natureza típica das funções com informação de fase, cuja integração ao longo de um caminho fechado resulta em valores múltiplos inteiros, positivos ou negativos de  $2\pi$ , de forma que podemos fazer uma analogia com o conceito de cargas em eletromagnetismo, considerando os resíduos de valor +1 portadores de carga positiva e os de valor -1, de carga negativa.

Por exemplo, ao analisarmos o comportamento da função  $\psi(x, y)$ , que no capítulo 2 continha a informação da fase obtida em um espaço de representação de frequência espacial através da relação:

$$\psi(x, y) = \tan^{-1} \left( \frac{y}{x} \right), \quad (3.36)$$

com  $x$  e  $y$  reais, podemos observar que a equação é analítica em todos os pontos de seu domínio, com exceção da origem, onde a derivada não é definida. Também não é difícil demonstrar que a integral da Eq. (3.33) ao longo de qualquer caminho fechado que não envolva a origem é zero.

Se calcularmos as derivadas da Eq. (2.36) em relação a  $x$  e  $y$ , obtemos:

$$\begin{aligned}\frac{\partial\psi}{\partial x} &= \frac{-y}{x^2 + y^2}, \\ \frac{\partial\psi}{\partial y} &= \frac{x}{x^2 + y^2},\end{aligned}\tag{3.37}$$

que em coordenadas polares podem ser escritas como:

$$\begin{aligned}\frac{\partial\psi}{\partial x} &= \frac{-\text{sen}(\theta)}{r}, \\ \frac{\partial\psi}{\partial y} &= \frac{\text{cos}(\theta)}{r},\end{aligned}\tag{3.38}$$

onde  $x = r\text{cos}(\theta)$ ,  $y = r\text{sen}(\theta)$  e  $\theta$  é o ângulo entre o raio  $r$  e o eixo  $x$ .

Se utilizarmos o teorema de resíduos da Eq. (3.35):

$$\oint \nabla\psi(\mathbf{r}) \cdot d\mathbf{r} = \oint \left( \frac{\partial\psi}{\partial x} \hat{\mathbf{x}} + \frac{\partial\psi}{\partial y} \hat{\mathbf{y}} \right) \cdot (dx\hat{\mathbf{x}} + dy\hat{\mathbf{y}}) = \oint \left( \frac{\partial\psi}{\partial x} dx + \frac{\partial\psi}{\partial y} dy \right).\tag{3.39}$$

Utilizando coordenadas polares e seguindo por um caminho fechado de integração, no sentido anti-horário, com um raio  $r$ , que envolva a origem, obtemos:

$$\begin{aligned}\oint \left( \frac{\partial\psi}{\partial x} dx + \frac{\partial\psi}{\partial y} dy \right) &= \int_0^{2\pi} \left\{ \left[ \frac{-\text{sen}(\theta)}{r} \right] [-r\text{sen}(\theta)] + \left[ \frac{\text{cos}(\theta)}{r} \right] [r\text{cos}(\theta)] \right\} d\theta = \\ &= \int_0^{2\pi} [\text{sen}^2(\theta) + \text{cos}^2(\theta)] d\theta = \int_0^{2\pi} d\theta = 2\pi,\end{aligned}\tag{3.40}$$

onde o resultado obtido indica a presença de resíduo envolvido pelo caminho de integração e ratifica a validade do teorema definido por Ghiglia em (3.35).

Além do exposto, conforme já mencionado, Goldstein demonstrou que problemas que dificultam o desempacotamento estão associados a resíduos, os quais são originados basicamente de duas fontes [50]:

De regiões com ruídos e *aliasing*<sup>11</sup> que violam a condição de Itoh, e descontinuidades de objeto como quinas e orifícios cuja variação seja maior do que  $\pi$ , que ao serem empacotadas geram falsos saltos de fase.

A tarefa de encontrar problemas que afetam o desempacotamento, se resume à detecção desses resíduos.

Além disso, é interessante observar que o que está relacionado à fonte do problema, também indica uma solução.

Enquanto que os resíduos gerados, com carga positiva (+1) ou negativa (-1), indicam a presença de problemas no mapa de fase, não é difícil perceber que se a soma dos envolvidos pela curva da Eq. (3.35) for igual a zero, ou seja, se estiverem “balanceados”, a integral de linha desta região será zero e a terceira condição do Quadro 1 será novamente satisfeita.

Uma forma simples de fazer isso é escolher um caminho de contorno que envolva uma carga positiva e outra negativa, de forma que a soma das duas seja zero. Para isso, podemos conectá-los através de barreiras (*branch-cut*) que delimitam caminhos que não podem ser cruzados durante o desempacotamento.

Uma vez que todos os resíduos estejam conectados de forma a impedir que qualquer caminho de integração cruze uma região com resíduo desbalanceado, o

---

<sup>11</sup> *Aliasing* ou falseamento é um efeito de serrilhado em imagens digitais, causado por baixa taxa de amostragem na conversão de sinal analógico para digital. O serrilhado pode possuir descontinuidades de módulo maior do que  $\pi$ , violando a condição de Itoh.

desempacotamento pode ser efetuado sem violar as condições de independência de caminho impostas.

A Fig. 3.10 (a) e (b) contém um exemplo que ilustra a relação dos resíduos com as barreiras e a independência de caminho.

Na Fig. 3.10 (a), para efetuar o desempacotamento do ponto A até o ponto B, são mostrados três caminhos possíveis. Ao percorrer o caminho à esquerda do resíduo positivo (círculo branco) ou à direita do resíduo negativo (círculo preto), o ponto B será acrescido em  $6\pi$ , porém, ao passar entre os dois resíduos, o acréscimo no ponto B será de apenas  $4\pi$ , evidenciando a dependência de caminho no resultado final do desempacotamento.

Na Fig. 3.10 (b) há uma barreira conectando o resíduo positivo ao negativo e delimitando uma região que não pode ser cruzada durante o desempacotamento. Qualquer outro caminho fará com que B seja acrescido de  $6\pi$ , ou seja, o valor será o mesmo, independentemente do caminho de integração escolhido. A independência de caminho, neste caso, foi recuperada.

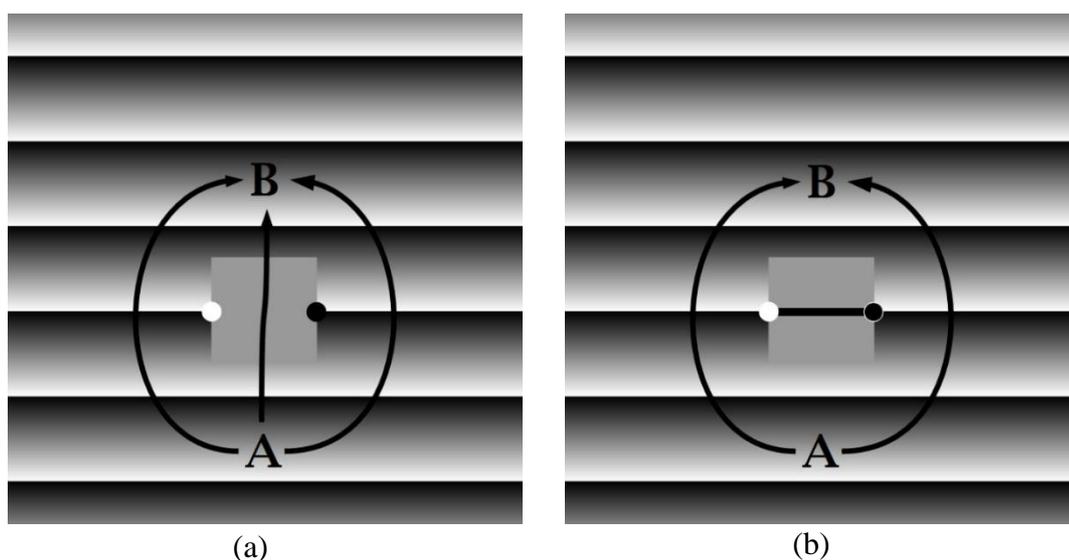


Fig. 3.10 (a) e (b) com caminhos de desempacotamento do ponto A para o ponto B, sem a presença de barreira conectando os resíduos (a) e com a presença de barreira conectando os resíduos (b).

### 3.3.5.1 A LOCALIZAÇÃO DOS RESÍDUOS

Evidenciada a importância dos resíduos como indicativos de fatores no mapa de fases que interferem no processo de desempacotamento, torna-se imprescindível determinar uma forma de localizá-los, uma vez que, além de servirem de base para os algoritmos de *branch-cut*, sua posição e densidade indicam a gravidade do problema a ser tratado e graficamente demonstram em quais regiões os erros de desempacotamento são mais suscetíveis de ocorrerem.

Uma forma simples de efetuar essa verificação é realizar integrações ao longo de caminhos fechados. Se o resultado do cálculo for diferente de zero, pela equação (3.35), deverá haver uma carga de resíduo na área circulada pelo caminho.

Para minimizar a região onde o resíduo possa estar e facilitar sua localização, podemos utilizar o menor caminho de integração possível.

Para isso, vamos trabalhar com espaços discretos bidimensionais<sup>12</sup>, no qual ao utilizarmos a definição (3.16), obtemos:

$$\Delta\{\psi(m, n)\} = \mathcal{W}\{\Delta\{\varphi(m, n)\}\} \quad (3.41)$$

Onde  $\psi(m, n)$  representa a fase verdadeira,  $\varphi(m, n)$  a fase empacotada e o operador  $\Delta$  pode ser definido conforme as relações:

$$\Delta_1[\varphi(m, n)] = \varphi(m, n + 1) - \varphi(m, n) \quad (3.42)$$

---

<sup>12</sup> Como o mapa de fases geralmente é uma imagem digital bidimensional, trabalharemos com espaços discretos em 2-D na apresentação do método.

$$\Delta_2[\varphi(m, n)] = \varphi(m + 1, n + 1) - \varphi(m, n + 1)$$

$$\Delta_3[\varphi(m, n)] = \varphi(m + 1, n) - \varphi(m + 1, n + 1)$$

$$\Delta_4[\varphi(m, n)] = \varphi(m, n) - \varphi(m + 1, n)$$

Substituindo a Eq. (3.41) na versão bidimensional discreta do teorema de resíduos definido pela Eq. (3.35):

$$\sum_{i=1}^4 \Delta_i\{\psi(m, n)\} = \sum_{i=1}^4 \mathcal{W}\{\Delta_i\{\varphi(m, n)\}\} = 2\pi q(m, n), \quad (3.43)$$

onde  $q(m, n)$  é a carga de resíduo envolvida pelo contorno da soma.

Rearranjando a Eq. (3.43), podemos observar que a carga de resíduo  $q$  pode ser encontrada através da soma dos elementos ao longo de um caminho fechado, através da relação:

$$q(m, n) = R \left[ \frac{1}{2\pi} \sum_{i=1}^4 \mathcal{W}\{\Delta_i\{\varphi(m, n)\}\} \right], \quad (3.44)$$

onde o operador  $R$  retorna o inteiro mais próximo de seu argumento, para garantir que as cargas de resíduos tenham os valores 1, -1 ou 0.

Por exemplo, a Fig. 3.11 representa parte de um mapa de fases empacotado discretizado, com os índices  $m$  e  $n$  representando respectivamente a posição horizontal e vertical dos píxeis no mapa. Ao aplicarmos a Eq. (3.42) aos elementos marcados na figura, obtemos os resultados:

$$\begin{aligned}
\Delta_1 &= -\frac{4\pi}{5}, \\
\Delta_2 &= -\frac{2\pi}{5}, \\
\Delta_3 &= \frac{7\pi}{5}, \\
\Delta_4 &= -\frac{\pi}{5}.
\end{aligned}
\tag{3.45}$$

Ao utilizarmos a Eq.(3.44), obtemos:

$$q(m, n) = R \left[ \frac{1}{2\pi} \left( -\frac{4\pi}{5} - \frac{2\pi}{5} - \frac{3\pi}{5} - \frac{\pi}{5} \right) \right] = -1
\tag{3.46}$$

onde chamamos atenção para o fato de que, como  $\Delta_3 > \pi$ , temos:

$$\mathcal{W}\{\Delta_3\} = \frac{7\pi}{5} - 2\pi = \frac{7\pi - 10\pi}{5} = -\frac{3\pi}{5}.
\tag{3.47}$$

O valor de carga  $q(m, n) = -1$ , indica a presença de resíduo na posição  $m, n$ .

Ao repetirmos o processo para todos os píxeis da imagem, é possível detectar todos os resíduos presentes no mapa.

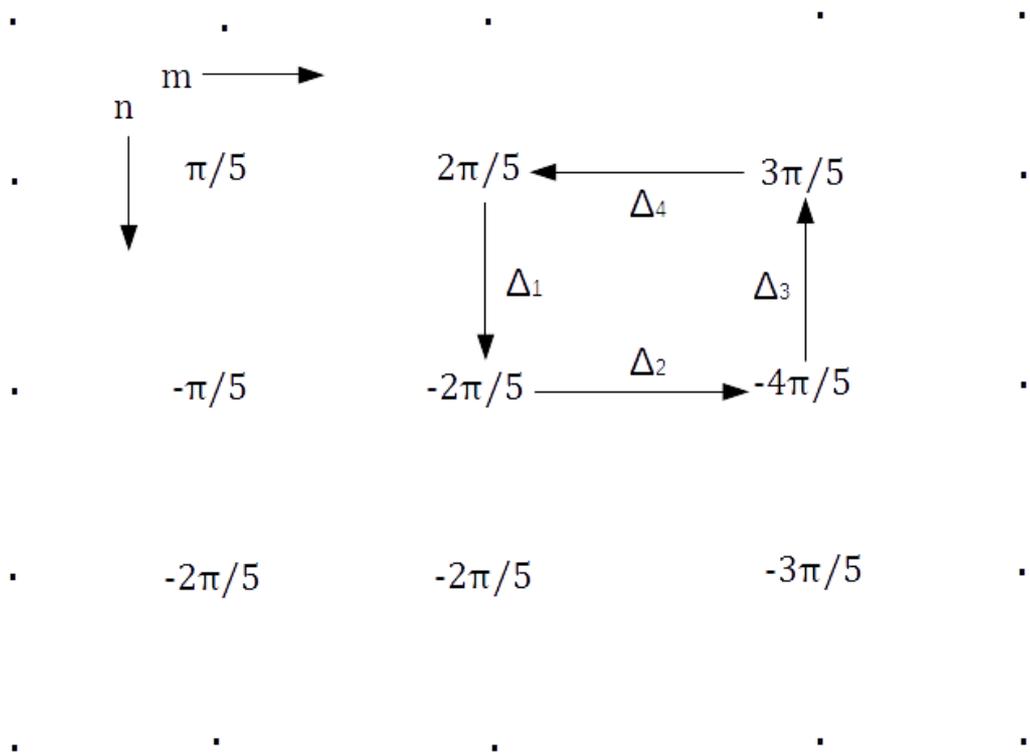


Fig. 3.11 Parte de uma imagem contendo um mapa de fases empacotado. As setas indicam o sentido anti-horário para a soma dos elementos.

### 3.3.5.2 A CONEXÃO ENTRE OS RESÍDUOS

Tendo localizado a posição de todos os resíduos do mapa, o próximo passo é conectá-los de modo que possam ser envolvidos por um caminho fechado de integração (ou soma, para espaços discretos) e a soma das cargas envolvidas seja zero, em acordo com a Eq (3.35). Dessa forma, as condições para independência de caminho podem ser novamente satisfeitas.

Uma forma simples de fazer isso é conectar os resíduos em pares de carga oposta, utilizando barreiras (*branch-cuts*) que delimitam regiões no mapa que não podem ser cruzadas durante o processo de desempacotamento.

A estratégia parece simples mas esbarra em um grande problema de otimização computacional, pois ao conectar os resíduos, deve-se procurar envolver as menores regiões possíveis, de forma a minimizar a informação perdida, uma vez que as regiões que contém os *branch-cuts* são isoladas dos caminhos de desempacotamento.

Como a quantidade possível de configurações de conexões entre cargas opostas é  $(S/2)!$ , onde  $S$  é o número de cargas de resíduos presentes no mapa, dependendo do número de cargas, verificar, dentre todas as configurações, aquela que possui os menores *branch-cut*, pode demandar um grande custo computacional e conseqüentemente muito tempo para ser concluída.

Para demonstrar o impacto da escolha da configuração de conexão dos resíduos, as Fig. 3.12 e Fig. 3.13 contém duas possibilidades para uma mesma imagem. É possível observar que a soma das regiões isoladas (hachuradas) na Fig. 3.12 é menor do que a soma das regiões isoladas na Fig. 3.13. Estas regiões não são, ao menos inicialmente, incluídas no processo de desempacotamento e podem conter informações importantes

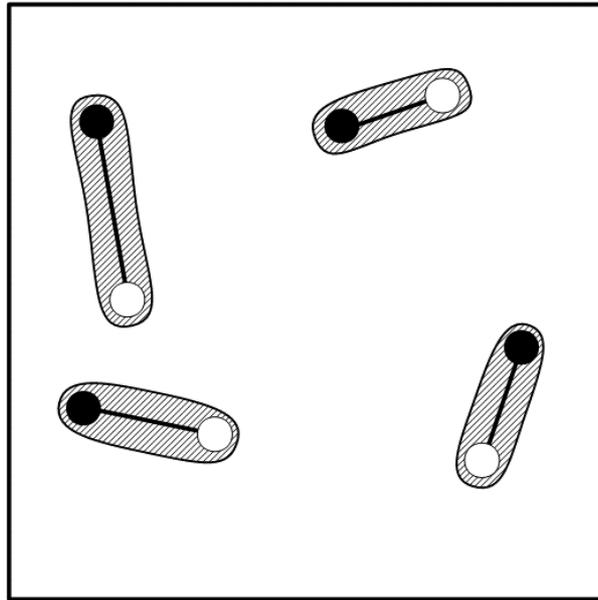


Fig. 3.12 Uma das possíveis configurações para conexão entre os resíduos. As regiões hachuradas representam áreas excluídas do processo de desempacotamento, os círculos brancos os resíduos de carga positiva (+1) e os círculos pretos, os de carga negativa (-1).

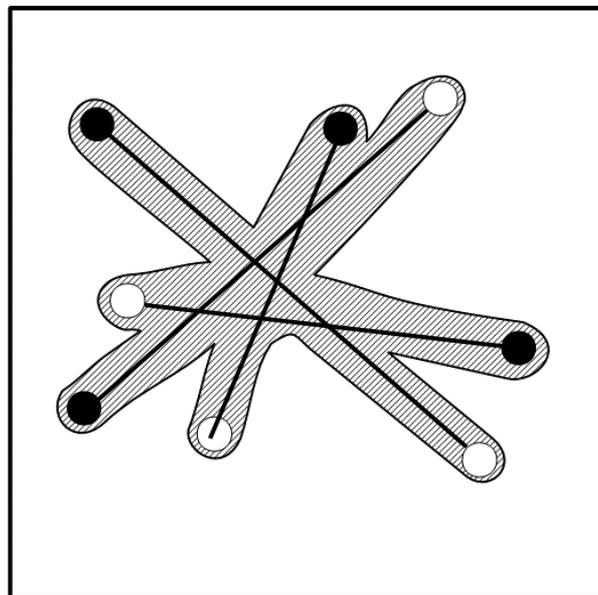


Fig. 3.13 Outra das possíveis configurações para conexão entre os resíduos. Nesta, as regiões hachuradas compreendem uma área maior do que na figura anterior.

acerca da análise efetuada sobre o mapa. Um bom algoritmo de *branch-cut* deve poder distinguir, no menor tempo possível, entre os cenários possíveis, aquele que trará a menor perda de informação na reconstrução do mapa.

Frente a esse problema, o desenvolvimento de algoritmos mais eficientes, com menor custo computacional, redução no tempo de processamento e com perdas mínimas de informação é alvo de diversas pesquisas na área e será objeto de estudo nesse trabalho.

Para isso, desenvolvemos uma abordagem alternativa à tradicional análise de todas as configurações possíveis, com vistas a obter uma melhor relação entre tempo de processamento e qualidade do mapa de fases verdadeiras obtido.

### **3.3.6 OUTROS ALGORITMOS DE DESEMPACOTAMENTO DE FASE**

Optamos, nesse trabalho, por apresentar alguns dos algoritmos mais referenciados na área, posto que a apresentação de um quadro completo extrapolaria os objetivos práticos dessa dissertação.

Mas conforme mencionado no início do subcapítulo 2.3, existem diversos estudos e abordagens para o problema do desempacotamento, como os Algoritmos de Corte por Máscara (*mask-cut*) [51, 52], que trabalham de forma similar aos de *branch-cut*, porém, associados a mapas de qualidade; de Minimização de Norma ( $L^p$  – *norm*) [53], Bayesiano [54], de Crescimento de Grupos ou Baseado em Correlação (*group-growing*) [55], por Projecção de Franjas Coloridas (*color-fringe projection*) [56] entre outros.

Cada um, com vantagens, desvantagens e situações em que são mais indicados a serem utilizados, produzem resultados dentro de suas limitações técnicas.

A busca por um algoritmo definitivo para desempacotamento de fase, mesmo que tenha apresentado avanços significativos, permanece a décadas.

Nosso principal objetivo nesse trabalho será, utilizando a abordagem de *branch-cut*, contribuir com essa busca, visando a criação de um algoritmo rápido, eficiente, que supere muitas das dificuldades encontradas na área e consiga desempacotar, com resultados satisfatórios, os mapas de fase obtidos sob as condições experimentais dos sistemas do Laboratório de Óptica Não Linear e Aplicada da Universidade Federal Fluminense – UFF (*Lab. Óptica UFF*).

Além disso, esse algoritmo poderá ser utilizado em diversas aplicações que envolvam a geração de mapas de fase, configurando-se, portanto, em ferramenta muito útil na resolução de problemas e superação de dificuldades presentes em áreas diversas do conhecimento, como Engenharia, Medicina, Física, Biologia etc.

Como forma de demonstrar a utilidade e necessidade do desempacotamento de fase, no próximo capítulo, apresentaremos uma técnica que utiliza Transformações de Fourier para a geração de um mapa, que após passar por um processo de desempacotamento, fornecerá o perfil tridimensional da superfície analisada.

## Capítulo 4

# PERFILOMETRIA POR TRANSFORMAÇÃO DE FOURIER

*Neste capítulo será apresentada a técnica de perfilometria por transformação de Fourier. O procedimento consiste na utilização da Transformação de Fourier sobre um padrão de franjas projetado sobre superfície de um objeto e geração de um mapa de fases, que através de um algoritmo de desempacotamento de fase, fornece uma representação tridimensional do objeto iluminado com o padrão de franjas.*

Técnicas não destrutivas para levantamento do perfil da superfície de objetos por processos ópticos têm sido extensivamente estudadas [57-64] devido à sua importância em áreas como engenharia mecânica, robótica, microbiologia, geologia, entre outras [59, 64-67], pois fornecem boa resolução, densidade de dados e alta precisão, além de causarem mínimo dano à estrutura analisada [68].

A base para a utilização destas técnicas é a projeção de um padrão senoidal regular de franjas sobre a superfície a ser medida. Este padrão pode ser gerado por processos interferométricos, holográficos, por projeção de redes físicas ou uma imagem de franjas gerada digitalmente em computador. O padrão projetado é deformado pela superfície do objeto iluminado e a análise da fase dessas franjas deformadas fornece informação para determinação de seu perfil.

Muitos algoritmos vêm sendo desenvolvidos para a extração de um mapa que forneça a fase de cada ponto da imagem do padrão deformado.

Para técnicas que utilizam múltiplas imagens, temos os já mencionados algoritmos de deslocamento de fase (“Phase-Shifting”) de três passos, quatro passos, cinco passos,

entre outros [1, 67, 69-75]. Para as técnicas que utilizam apenas uma imagem de um padrão de franjas projetado sobre a superfície de interesse, dentre as principais, podemos mencionar a por Transformação de Fourier, Transformação de Wavelet e Transformação Janelada de Fourier [63].

Neste capítulo teremos como objeto de estudo a técnica por Transformação de Fourier [76-78] que devido à sua alta precisão, densidade de análise de dados e possibilidade de processamento dinâmico da informação [68], será utilizada para a obtenção dos mapas de fase analisados neste trabalho.

A Perfilometria por Transformação de Fourier (PTF) [79] se baseia na aplicação da transformada de Fourier sobre o padrão de franjas projetado e sua manipulação no domínio da frequência espacial para geração de um mapa de fases que, como mostraremos neste capítulo, necessita passar por um processo de desempacotamento de fase para recuperação de sua fase verdadeira e obtenção do perfil do objeto iluminado.

O padrão de franjas projetado sobre a superfície de um objeto, por exemplo, por uma rede física ou projetor digital, pode ser matematicamente representado pela expressão

$$g(x, y) = a(x, y) + b(x, y) \cos(2\pi f_0 x + \varnothing(x, y)), \quad (4.1)$$

onde  $a(x, y)$  e  $b(x, y)$  representam respectivamente as variações de irradiância produzidas pela reflexão não-uniforme da luz na superfície do objeto, provenientes da iluminação ambiente e padrão de franjas projetado,  $f_0$  representa a frequência portadora do padrão de franjas e  $\varnothing(x, y)$  é a fase, que caracteriza a deformação das franjas e está relacionada à forma do objeto.

Ao mapearmos a distribuição de irradiância para cada ponto e relacionarmos a fase destes pontos à distância de um plano de referência, podemos construir o perfil da superfície iluminada.

A Fig. 4.1 mostra a geometria óptica que descreve a projeção de franjas na superfície de um objeto tridimensional.

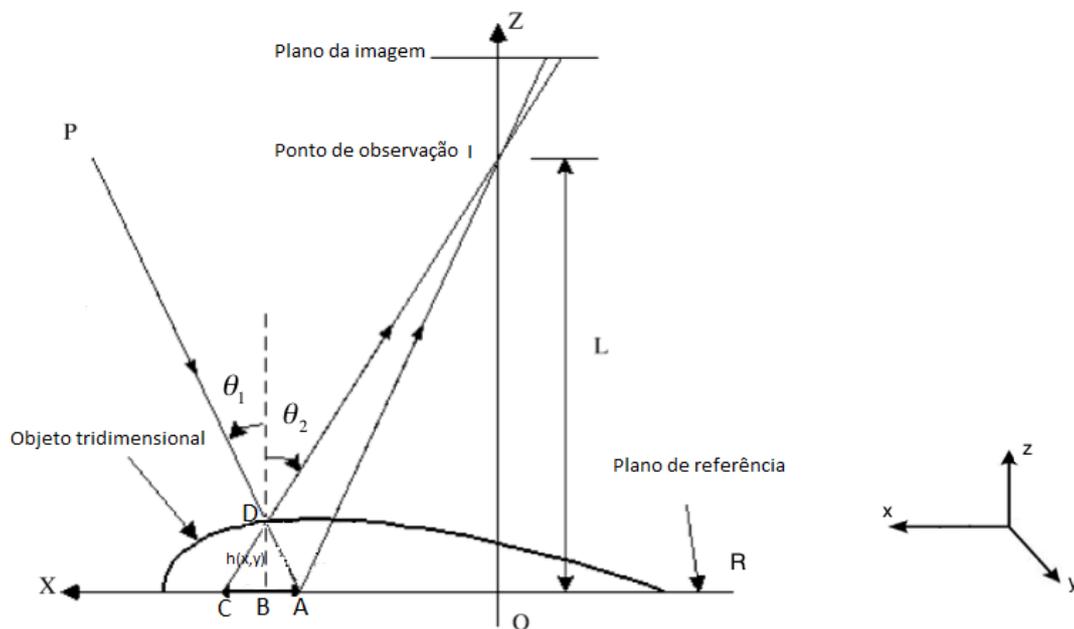


Fig. 4.1 Geometria óptica para a projeção de franjas em um objeto.

É possível estabelecer uma relação entre a fase  $\phi(x, y)$  e a distância  $h(x, y)$  do ponto iluminado a um plano de referência  $R$ . O padrão luminoso projetado neste plano, que serve como um referência para a medição da profundidade do objeto, possui uma frequência espacial  $f_0$ . A intensidade produzida no ponto  $C$  ( $x = \overline{OC}$ ) em  $R$  é dada por:

$$g_C(x, y) = a(x, y) + b(x, y)\cos(2\pi f_0 \overline{OC}) \quad (4.2)$$

A intensidade observada no ponto  $D$  no objeto é a mesma do ponto  $A$  ( $x = \overline{OA}$ ) em  $R$  se o objeto não estivesse presente, modificada pela refletividade  $r(x, y)$  do objeto:

$$g_D(x, y) = r(x, y)(a(x, y) + b(x, y)\cos(2\pi f_0 \overline{OA})) \quad (4.3)$$

e a diferença de fase entre os pontos  $C$  e  $D$ , observada no ponto  $I$ , pode ser escrita como

$$\phi_{CD} = 2\pi f_0 (\overline{OC} - \overline{OA}) = 2\pi f_0 \overline{AC} \quad (4.4)$$

Ao observarmos que

$$\overline{AC} = \overline{AB} + \overline{BC} \quad (4.5)$$

Obtemos, de acordo com a Fig. 4.1:

$$\begin{cases} \overline{AB} = h(x, y)\tan\theta_1 \\ \overline{BC} = h(x, y)\tan\theta_2 \end{cases} \quad (4.6)$$

onde  $\theta_1$  e  $\theta_2$  representam as respectivas direções de iluminação e observação.

Combinando as Eqs. (4.5) e (4.6) podemos estabelecer a relação entre  $\overline{AC}$  e a profundidade  $h(x, y)$  do objeto através da equação:

$$\overline{AC} = h(x, y)(\tan\theta_1 + \tan\theta_2) \quad (4.7)$$

que combinada com a Eq.(4.4) fornece a relação desejada entre fase e profundidade:

$$h(x, y) = \frac{1}{2\pi f_0(\tan\theta_1 + \tan\theta_2)} \phi_{CD}(x, y) = k\phi_{CD}(x, y), \quad (4.8)$$

onde o valor de  $k$  pode ser estimado por um processo de calibração, com a ajuda de um software adequado como o Matlab® versão R2012a da MathWorks® ou ImageJ (desenvolvido por Wayne Rasband do Research Services Branch, National Institute of Mental Health, Bethesda, Maryland).

Como nosso objetivo é obter  $h(x, y)$ , o próximo passo é isolar as variações de amplitude indesejadas, encontrar o valor de  $\phi(x, y)$  para cada ponto da superfície e aplicar na Eq.(4.8).

Para isso, utilizando a representação complexa de cossenos

$$\cos(\theta) = \frac{1}{2} [\exp(i\theta) + \exp(-i\theta)], \quad (4.9)$$

podemos reescrever a Eq. (4.1) na forma:

$$g(x, y) = a(x, y) + c(x, y) \exp(2\pi i f_0 x) + c^*(x, y) \exp(-2\pi i f_0 x), \quad 4.10$$

onde

$$c(x, y) = \frac{1}{2}b(x, y) \exp(i\phi(x, y)), \quad (4.11)$$

e  $c^*(x, y)$  é o conjugado complexo de  $c(x, y)$ .

Ao aplicarmos a transformação de Fourier na Eq.(4.10) em relação à variável  $x$ , tratando a variável  $y$  como um parâmetro fixo e utilizando a propriedade de translação na frequência da transformada:

$$\mathcal{F}\{f(x)\exp(if_0x)\} = F(f - f_0), \quad (4.12)$$

obtemos:

$$G(f, y) = A(f, y) + C(f - f_0, y) + C^*(f - f_0, y) \quad (4.13)$$

onde as letras capitais denotam o espectro de Fourier no domínio da frequência espacial,  $f$  a própria frequência espacial na direção  $x$  e  $f_0$  a frequência portadora.

Neste estágio, utilizando-se do auxílio de software adequado como o Matlab® versão R2012a da MathWorks®, é possível selecionar e extrair a variação de fundo indesejada,  $A(f, y)$ , do espectro de Fourier, obtendo:

$$G(f, y) = C(f - f_0, y) + C^*(f - f_0, y). \quad (4.14)$$

A Eq. (4.14) possui duas componentes deslocadas pela frequência portadora  $f_0$ . Se isolarmos e transladarmos  $C(f - f_0, y)$  para a origem do eixo das frequências, conforme representado na Fig. 4.2, obtemos  $C(f, y)$  e com o auxílio de um software adequado como

o Matlab, extrairmos e aplicarmos a transformada inversa de Fourier a  $C(f, y)$  em relação a  $f$ , obtemos  $c(x, y)$ , definido pela Eq. (4.11)

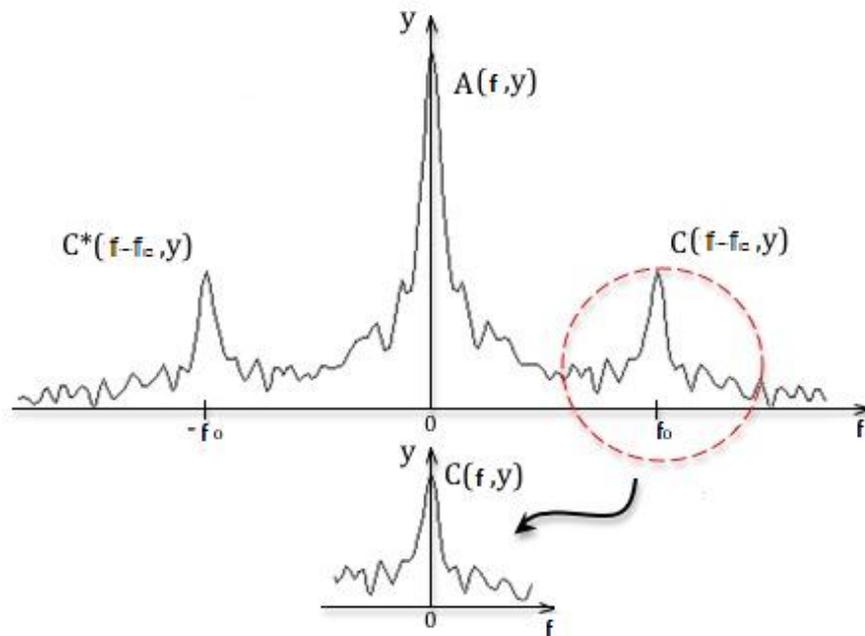


Fig. 4.2 Espectro de Fourier de um padrão de franjas projetado sobre a superfície de um objeto e um único espectro selecionado e transladado para a origem.

Ao utilizarmos a representação exponencial de uma função complexa

$$Z = \arg(Z) e^{i\phi + 2n\pi} = \arg(Z) e^{i\phi} = r e^{i\phi} \quad \forall n \in N, \quad (4.15)$$

comparando com a Eq.(4.11), podemos supor  $Z=c(x,y)$  e  $r = \frac{1}{2}b(x, y)$ , o que nos permite reescrevê-la como

$$c(x, y) = \frac{1}{2}b(x, y)[\cos(\phi(x, y)) + i\text{sen}(\phi(x, y))]. \quad (4.16)$$

Neste ponto, ao representar a função  $c(x,y)$  no plano complexo (Fig. 4.3),

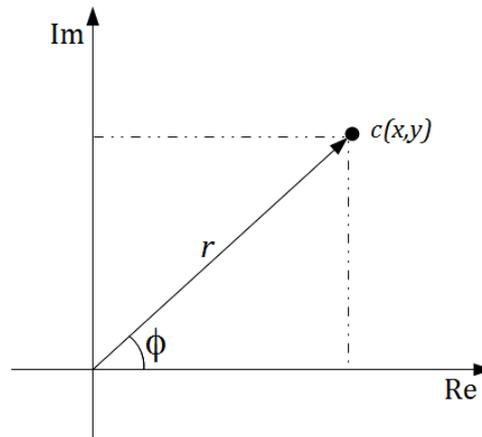


Fig. 4.3 Representação do plano complexo. O ângulo da figura pode ser obtido calculando o arco tangente do quociente entre a projeção do raio  $r$  no eixo imaginário e a projeção no eixo real.

vemos que  $\phi(x,y)$  pode ser determinado pelo arco-tangente do quociente entre sua parte imaginária e sua parte real:

$$\phi(x,y) = \tan^{-1} \left( \frac{\text{Im}[c(x,y)]}{\text{Re}[c(x,y)]} \right). \quad (4.17)$$

Porém, ao realizar este cálculo, encontramos um problema computacional a ser contornado.

Assim como em exemplos anteriores, o cálculo da fase por computador (Eq.(4.17)), devido à impossibilidade de distinguir se o valor do arco tangente está acrescido ou não de um múltiplo de  $2\pi$ , retorna valores que variam de  $-\pi$  a  $\pi$ , constituindo um mapa de distribuição de fases empacotado neste intervalo.

Contudo, é mais fácil entender o mapa se este não apresentar descontinuidades, o que torna necessária a utilização de algoritmos de desempacotamento de fase para a obtenção da fase verdadeira do mapa.

Dadas as características do sistema e do mapa obtido, além da necessidade de tempo de processamento e qualidade almejada, a escolha do algoritmo de desempacotamento apropriado torna-se essencial.

Assim, tendo obtido o mapa de fases desempacotado, utilizando a Eq. (4.8), que relaciona a fase  $\phi(x, y)$  a  $h(x, y)$ , podemos construir o perfil da superfície iluminada com o padrão de franjas.

Por fim, neste capítulo foi visto que o método por Transformação de Fourier fornece um mapa com a separação completa da informação de fase da imagem de um padrão de franjas deformado pela superfície de um objeto tridimensional. O mapa gerado necessita passar por um processo de desempacotamento de fase para recuperação de seus valores verdadeiros e produção de uma representação que possua a discriminação entre elevação e depressão da superfície do objeto iluminada com o padrão de franjas.

No próximo capítulo apresentaremos uma proposta para desempacotamento de fase aplicado à PTF, com vistas a obtermos uma aplicação que forneça resultados de forma rápida, eficiente, com precisão de detalhes e que poderá ser utilizada, inclusive, na obtenção de perfis nos experimentos desenvolvidos no *Lab. Óptica UFF*.

## Capítulo 5

# PROPOSTA PARA UM ALGORITMO DE DESEMPACOTAMENTO DE FASE POR BALANCEAMENTO DE RESÍDUOS

*Neste capítulo será apresentada a base teórica para uma proposta de desempacotamento de fase por balanceamento de resíduos que utiliza a geração de mapas empacotados com os saltos de fase deslocados, com vistas a minimizar os caminhos de conexão entre os resíduos. Também descreveremos detalhadamente os passos envolvidos no processamento do algoritmo, com um exemplo ilustrativo de sua aplicação, para facilitar o entendimento.*

Dentre as técnicas de desempacotamento de fase atualmente utilizadas, a de balanceamento de resíduos é consagrada pela qualidade dos resultados obtidos e tempo reduzido de processamento quando comparada a outros algoritmos como o de inundação (*flood-fill*) ou *multifrequências*.

Neste capítulo apresentaremos uma proposta de algoritmo de balanceamento de resíduos com uma abordagem mais rápida e direta do que a de Goldstein e Huntley [32], que faz a análise de todas as configurações possíveis de conexão entre os resíduos para utilizar aquela que possui os menores caminhos.

A abordagem de Huntley et. al., apesar de produzir bons resultados, uma vez que garante a minimização das regiões do mapa de fase que são isoladas do desempacotamento, demanda alto custo computacional por analisar uma grande quantidade de cenários, além do fato de que as conexões podem ser efetuadas através de

retas que não levam em conta as características do mapa de fase, podendo gerar a perda de informações importantes para a análise a ser efetuada.

Nossa abordagem, mais rápida e direta, utiliza a análise das linhas do mapa com salto de fase como referência para a minimização dos caminhos de conexão entre os resíduos.

Dessa forma reduzimos drasticamente a quantidade de cálculos a ser efetuada e o tempo necessário para o término do processamento, além de, ao levar em consideração as características do mapa, priorizamos, sem a utilização de um mapa de qualidade, as regiões que possuem as informações mais importantes para a obtenção da fase.

## 5.1 BASE TEÓRICA

Para demonstrar nossa técnica, inicialmente vamos supor uma função empacotada  $\varphi(x, y)$  bidimensional discretizada:

$$\varphi(m, n) = \mathcal{W}[\psi(m, n)] = \psi(m, n) - 2\pi k_1(m, n), \quad (5.1)$$

onde  $\mathcal{W}$  é um operador de empacotamento,  $\psi(m, n)$  contém os valores verdadeiros da fase e  $k_1(m, n)$  é uma função inteira que faz com que  $-\pi < \varphi(m, n) \leq \pi$ .

Se adicionarmos uma fase  $\emptyset$  a todos os pontos do mapa de fases empacotado  $\varphi(m, n)$ , temos

$$\varphi'(m, n) = \varphi(m, n) + \emptyset = \psi(m, n) - 2\pi k_1(m, n) + \emptyset, \quad (5.2)$$

e atuarmos com o operador  $\mathcal{W}$  sobre a função  $\varphi'(m, n)$ :

$$\mathcal{W}[\varphi'(m, n)] = \xi(m, n) = \varphi'(m, n) - 2\pi k_2(m, n), \quad (5.3)$$

onde  $k_2(m, n)$  é uma função inteira que faz com que  $-\pi < \xi(m, n) \leq \pi$ .

Se substituirmos a Eq. (5.2) na Eq. (5.3):

$$\xi(m, n) = \varphi(m, n) + \emptyset - 2\pi k_2(m, n) \quad (5.4)$$

Reagrupando os termos:

$$\xi(m, n) = \varphi(m, n) - 2\pi \left[ k_2(m, n) - \frac{\emptyset}{2\pi} \right] \quad (5.5)$$

ou

$$\xi(m, n) = \varphi(m, n) - 2\pi k'(m, n) \quad (5.6)$$

com  $k'(m, n) = k_2(m, n) - \emptyset/2\pi$ .

A função  $\xi(m, n)$  difere da função  $\varphi(m, n)$  pela presença do termo  $2\pi k'(m, n)$ . Na prática, como veremos adiante, esse termo equivale a uma variação global na fase do mapa  $\varphi(m, n)$  e altera as regiões em que os saltos de fase ocorrem, mas sem comprometer a variação relativa entre os pontos vizinhos.

Desse modo, o resultado final do desempacotamento, tanto para  $\xi(m, n)$  quanto para  $\varphi(m, n)$ , é o mesmo.

Para demonstrar esse segundo fato, seguindo o raciocínio do subcapítulo 2.2, se calcularmos a diferença de fase em  $\xi(m, n)$  através da atuação do operador  $\Delta$ , anteriormente definido pela Eq. (3.13), e utilizarmos as Eqs. (5.2) e (5.4), temos:

$$\begin{aligned}\Delta[\xi(m, n)] &= \Delta[\varphi(m, n) + \emptyset - 2\pi k_2(m, n)] = & (5.7) \\ &= \Delta[\psi(m, n) - 2\pi k_1(m, n) + \emptyset - 2\pi k_2(m, n)]\end{aligned}$$

$$\Delta[\xi(m, n)] = \Delta\psi(m, n) - 2\pi\Delta[k_1(m, n) + k_2(m, n)] + \Delta\emptyset \quad (5.8)$$

Como  $\emptyset$  é constante em todo o domínio da função,  $\Delta\emptyset = 0$ , de forma que:

$$\Delta[\xi(m, n)] = \Delta\psi(m, n) - 2\pi\Delta[k_1(m, n) + k_2(m, n)] \quad (5.9)$$

Se atuarmos novamente com o operador  $\mathcal{W}$  sobre a Eq. (5.9), temos

$$\begin{aligned}\mathcal{W}\{\Delta[\xi(m, n)]\} &= \Delta\psi(m, n) - 2\pi\Delta[k_1(m, n) + k_2(m, n)] - 2\pi k_3(m, n) & (5.10) \\ &= \Delta\{\psi(m, n)\} - 2\pi[\Delta\{k_1(m, n) + k_2(m, n)\} + k_3(m, n)].\end{aligned}$$

Porém, como  $\mathcal{W}$  retorna apenas valores no intervalo  $(-\pi, \pi]$ , a condição de Itoh definida pela Eq. (3.10), que pode ser escrita na forma  $-\pi < \Delta\{\psi(m, n)\} \leq \pi$ , implica que o termo  $2\pi[\Delta\{k_1(m, n) + k_2(m, n)\} + k_3(m, n)]$  na Eq. (5.10) deve ser nulo, de forma que podemos escrever

$$\mathcal{W}\{\Delta[\xi(m, n)]\} = \Delta\{\psi(m, n)\}. \quad (5.11)$$

Ao compararmos a Eq. (5.11) com a Eq. (3.16) do subcapítulo 2.2, podemos perceber que o procedimento de adicionar a fase  $\emptyset$  ao mapa empacotado e atuar novamente com o operador  $\mathcal{W}$ , não altera o resultado final do desempacotamento, uma vez que ambas as funções são equivalentes à atuação do operador  $\Delta$  sobre a função  $\psi(m, n)$ .

Além disso, utilizando a relação (5.11) e o mesmo raciocínio das Eqs. (3.17) e (3.18), podemos chegar à relação:

$$\psi(m, n) = \psi(0, 0) + \sum_{n=1}^m \mathcal{W}\{\Delta\{\xi(m, n)\}\} = \psi(0, 0) + \sum_{n=1}^m \Delta\{\psi(m, n)\}. \quad (5.12)$$

O procedimento de adicionar a fase  $\emptyset$  e posteriormente efetuar novo empacotamento revela ser um método válido para produzir mapas de fase cujas regiões de saltos de  $2\pi$  estejam deslocadas em relação ao original, sem alterar o resultado final do desempacotamento.

Além disso, é possível variar o valor de  $\emptyset$  para controlar o deslocamento dessas regiões, o que pode ser útil para analisar diferentes configurações do mapa empacotado.

Por fim, se substituirmos a Eq. (5.11) na versão discreta bidimensional do teorema de resíduos dado pela Eq. (3.43), obtemos:

$$\sum_{i=1}^4 \Delta_i\{\psi(m, n)\} = \sum_{i=1}^4 \mathcal{W}\{\Delta_i\{\xi(m, n)\}\} = 2\pi q(m, n), \quad (5.13)$$

ou

$$q(m, n) = R \left[ \frac{1}{2\pi} \sum_{i=1}^4 \mathcal{W}\{\Delta_i\{\xi(m, n)\}\} \right]. \quad (5.14)$$

Devido à relação (5.11), fica evidente que a função  $q(m, n)$  para o mapa de fases  $\xi(m, n)$  possui os mesmos valores de cargas de resíduo do que para o mapa de fases  $\varphi(m, n)$ , o que nos leva a concluir que a localização dos resíduos é invariante pela operação que efetuamos para a obtenção de  $\xi(m, n)$ .

Com isso, demonstramos a criação de um método para deslocar as regiões de salto de fase no mapa empacotado e garantimos que a localização dos resíduos é invariante sob a aplicação deste método. Ambos os resultados obtidos servirão de base para a criação de nosso algoritmo de *branch-cut*.

## 5.2 ALGORITMO PROPOSTO

Tendo estabelecido as bases para nossa proposta de desempacotamento de fase, neste subcapítulo descreveremos de forma detalhada os passos para sua execução (para detalhes sobre o código fonte, ver Apêndice B).

Para isso, considerando que um mapa de fases  $\varphi(m, n)$  com  $M \times N$  pontos tenha sido gerado digitalmente, por exemplo, pela técnica de Perfilometria por Transformação de Fourier (PTF) detalhada no capítulo 4, temos:

Passo 1. Sobre o mapa de fases  $\varphi(m, n)$  é adicionada uma fase  $\emptyset$  a cada um de seus pontos e realizado novo empacotamento para a obtenção de outro mapa  $\xi(m, n)$ , cujos saltos de fase estão deslocados em relação aos saltos de  $\varphi(m, n)$ .

Passo 2. Para  $m$  inicialmente igual a 0 e  $n$  variando de 1 até  $N-1$ , calcula-se a diferença de fase entre o  $n$ -ésimo ponto de  $\varphi(0, n)$  e seu precedente através da relação:

$$\Delta\varphi(0, n) = \varphi(0, n) - \varphi(0, n - 1) \quad (5.15)$$

Se  $|\Delta\varphi(0, n)|$  for igual a  $2\pi$ , as coordenadas do ponto são armazenadas em uma matriz  $\varphi'$ .

Passo 3. Após armazenar as coordenadas dos pontos com salto de fase em  $\varphi'$ , variando  $m$  de 0 a  $M - 1$ , para cada valor de  $m$ , o passo 2 é repetido de modo que, se  $|\Delta\varphi(m, n)|$  for igual a  $2\pi$ , as coordenadas do ponto são adicionadas à matriz  $\varphi'$ .

Passo 4. Após a matriz  $\varphi(m, n)$  ter sido verificada e as coordenadas de todos os pontos com salto de fase armazenadas na matriz auxiliar, os passos 2 e 3 são repetidos, só que com  $\xi(m, n)$  no lugar de  $\varphi(m, n)$  e as coordenadas dos saltos de fase adicionadas a uma matriz  $\xi'$ .

Neste ponto teremos dois mapas de fase com os saltos deslocados entre si e duas matrizes com a localização dos saltos de fase em cada mapa.

Passo 5. Na matriz  $\varphi(m, n)$  ou  $\xi(m, n)$ , utilizando a Eq. (5.14), com  $m$  variando de 0 a  $M - 1$  e  $n$  variando de 0 a  $N - 1$ , os valores e as posições das cargas de resíduos são detectados e armazenados em uma matriz  $Q$ <sup>13</sup>,

Passo 6. Para cada resíduo positivo da matriz  $Q$  é verificado se sua coordenada coincide com a existência de um ponto marcado na matriz  $\varphi'$  ou  $\xi'$ , ou seja, se o resíduo encontra-se sobre uma linha de salto de fase. Caso positivo, todos os outros pontos

---

<sup>13</sup> Observe que foi criada apenas uma matriz para armazenar a localização dos resíduos. Isso se deve ao fato de que como a localização é a mesma para os dois mapas, não é necessário refazer o procedimento para o mapa com a fase deslocada. Basta utilizar a mesma matriz  $Q$ .

conectados ao ponto inicial são marcados até que a posição coincida com a localização de um resíduo negativo ou a borda da imagem. Dessa forma é traçada uma linha que se inicia em um resíduo positivo e termina em um resíduo negativo ou na borda da imagem.

Se ambas as matrizes  $\varphi'$  ou  $\xi'$  conterem pontos que possibilitem o traçado de uma linha que liga os resíduos, é verificada qual linha possui o menor comprimento para ser marcada como barreira no mapa de fases.

Se apenas uma das matrizes  $\varphi'$  ou  $\xi'$  conter pontos que possibilitem o traçado da linha que conecta os resíduos, estes são marcados como barreira para o desempacotamento.

Caso a localização dos resíduos não coincida com nenhum ponto nas matrizes  $\varphi'$  ou  $\xi'$ , nada é feito e o resíduo é marcado para ser verificado posteriormente.

Passo 7. Tendo conectado os resíduos ao longo das linhas de salto de fase, os resíduos remanescentes, que não tenham sido conectados a nenhum outro são conectados entre si através de uma linha reta, priorizando sempre o resíduo de carga oposta mais próximo. Caso não seja possível conectar a outro resíduo, aquele que sobrou é conectado à borda da imagem.

Passo 8. Com todas as barreiras de conexão entre os resíduos tendo sido estabelecidas, é efetuado o desempacotamento de forma linear e contínua do mapa, utilizando a Eq (3.19), através de caminhos que contornem as barreiras definidas.

Como exemplo, a Fig. 5.1 contém um mapa de fases empacotado gerado em computador.

Ao executar o passo 1, utilizando o valor  $\emptyset = \pi$  para a fase adicionada ao mapa, obtemos o mapa de fase deslocado apresentado na Fig. 5.2.

Ao executar os passos 2,3 e 4, são localizados os pontos com salto de fase para os dois mapas representados na Fig. 5.1 e Fig. 5.2.

Com base nesses pontos, construímos dois mapas de salto de fase, exibidos na Fig. 5.3 e Fig. 5.4, para demonstrar visualmente as linhas que servem de base para a criação das barreiras.

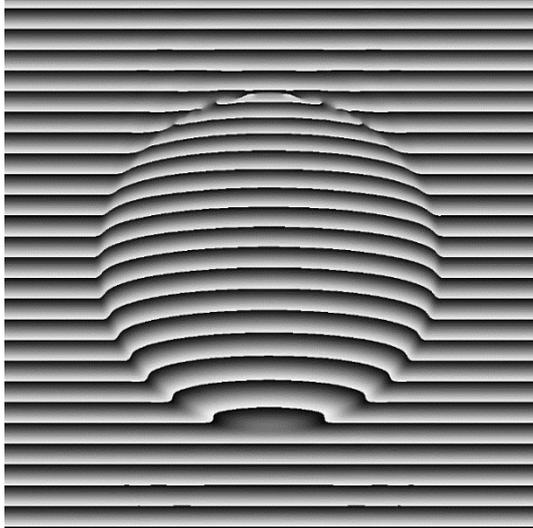


Fig. 5.1 Mapa de fases empacotado gerado por computador.

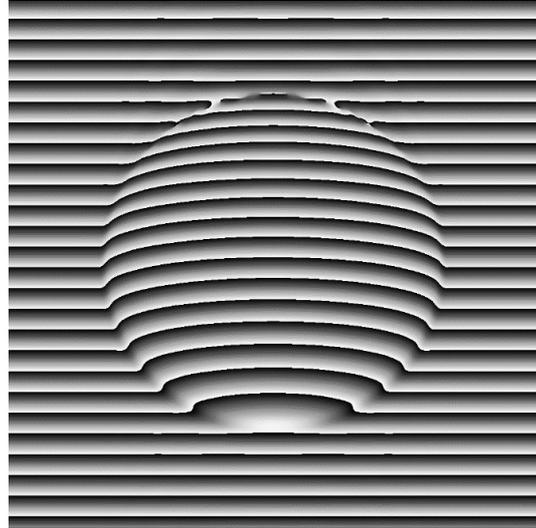


Fig. 5.2 Mapa de fases com a localização dos saltos deslocada, obtido após a execução do passo 1.

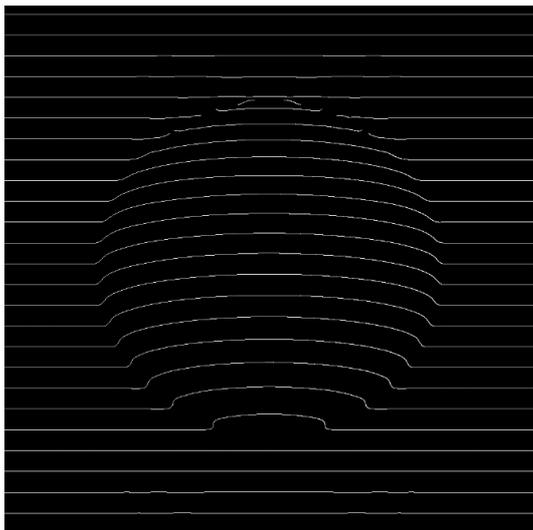


Fig. 5.3 Mapa contendo os pontos com salto de fase do mapa da Fig. 5.1, obtidos após a execução dos passos 2 e 3.

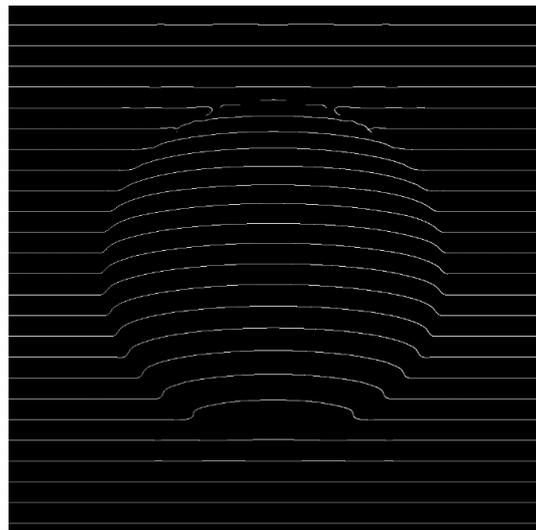


Fig. 5.4 Mapa com as linhas de salto de fase do mapa da Fig. 5.2, obtidas após a execução do passo 4.

O resultado do passo 5 efetuado sobre os dois mapas de fases é evidenciado na Fig. 5.5 e Fig. 5.6. Observe que a localização dos resíduos é a mesma, ainda que os mapas tenham regiões de salto de fase diferentes.

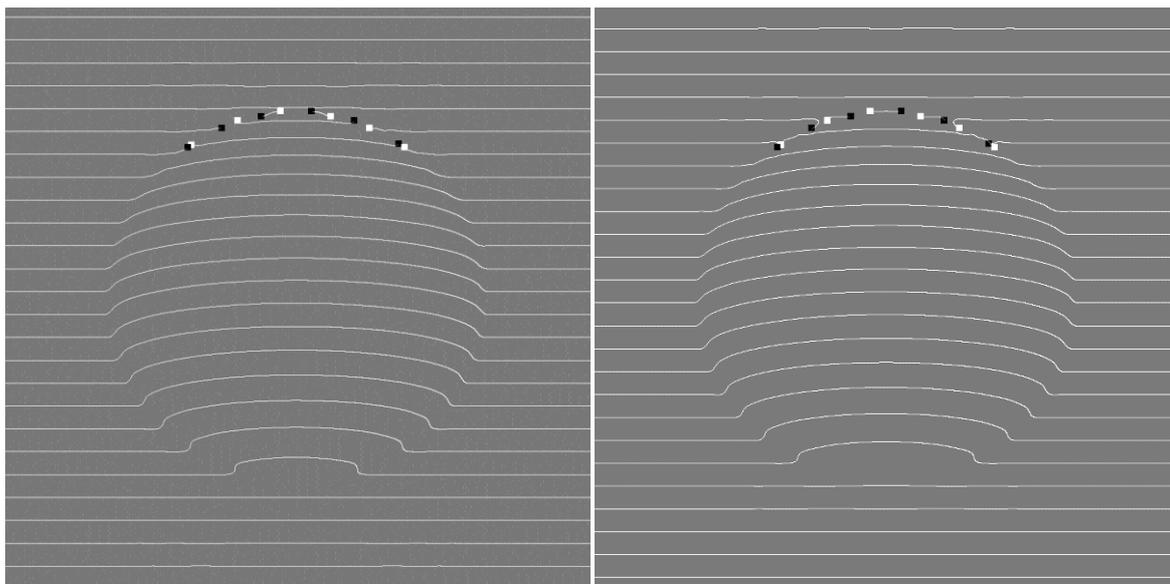


Fig. 5.5 Resíduos sobrepostos às linhas de salto de fase. Os pontos brancos referem-se a resíduos de carga positiva (+1) e os negros, aos resíduos de carga negativa (-1). Para melhorar a visualização, o tamanho dos pontos foi ampliado.

Fig. 5.6 Resíduos detectados no mapa de fases deslocado em relação ao original. É possível observar que a localização dos resíduos para os dois mapas é a mesma.

Na Fig. 5.7 e Fig. 5.8 é possível perceber que para os dois mapas as linhas entre dois resíduos de cargas opostas possuem configurações diferentes. No passo 6 essas linhas são verificadas e aquelas com o menor comprimento são utilizadas como barreira para o desempacotamento.

Ainda sobre o passo 6, na Fig. 5.9 é mostrado o resultado da verificação do tamanho das linhas e a escolha da configuração com as de menor extensão.

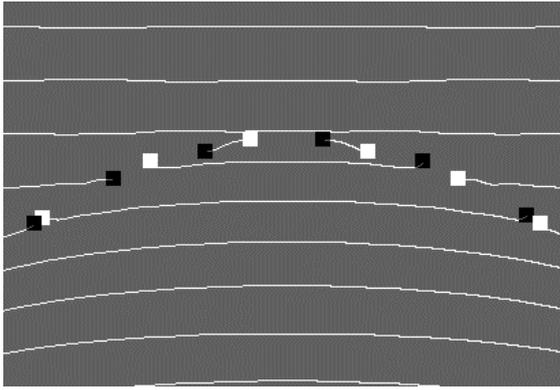


Fig. 5.7 Resíduos cruzados por linhas de salto de fase do mapa original. Na região entre dois resíduos, as linhas podem ser utilizadas como referência para a criação das barreiras.

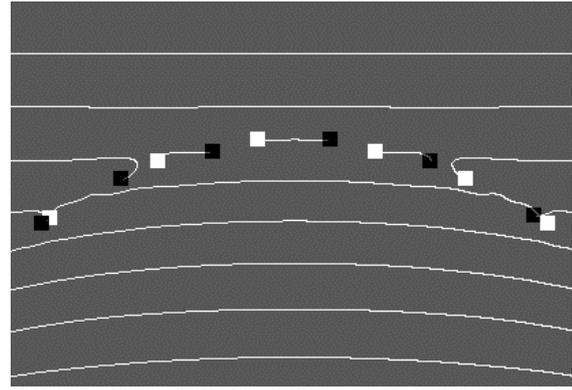


Fig. 5.8 Resíduos cruzados por linhas de salto de fase do mapa deslocado. Apesar da localização dos resíduos ser a mesma, a configuração das linhas que cruzam dois resíduos é diferente, logo, seu tamanho também pode variar.

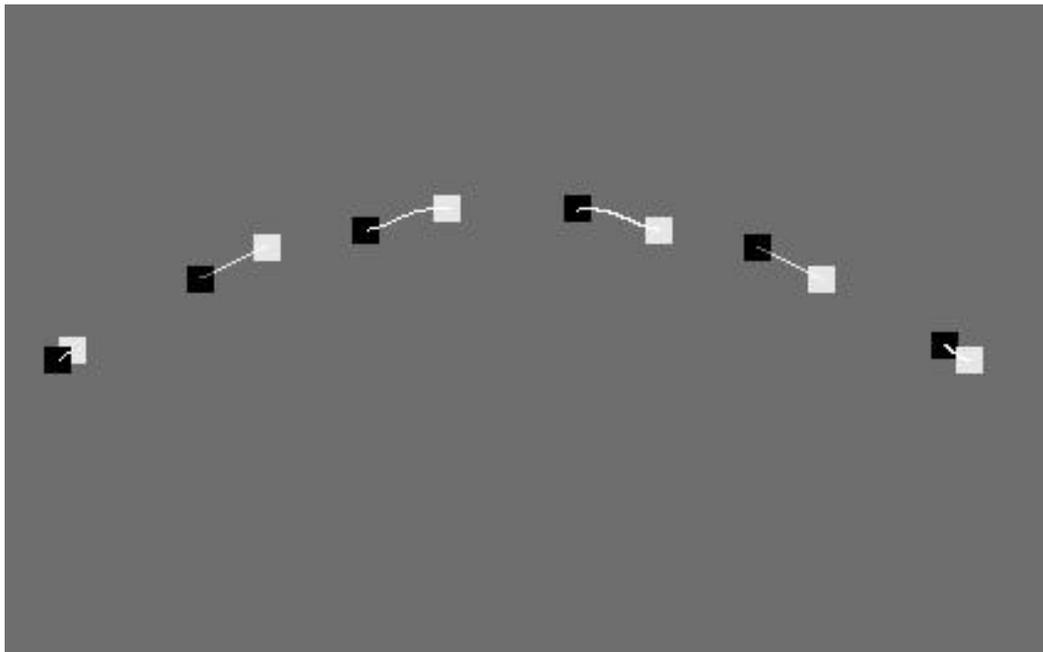


Fig. 5.9 Após verificado o tamanho das linhas que conectam os resíduos, são selecionadas aquelas que possuem a menor extensão.

Quanto ao passo 7, com a configuração apresentada no exemplo desenvolvido, não foi necessário conectar resíduos às bordas ou a outros que tenham sobrado, por linhas retas.

Por fim, quanto ao passo 8, na Fig. 5.10 são demonstradas as regiões de contorno das barreiras.

Durante o desempacotamento, ao encontrar uma barreira, o algoritmo a contorna seguindo por uma dessas regiões de contorno. Dessa forma, é garantido que os caminhos de integração circularão todas as regiões cuja soma das cargas de resíduo seja zero.

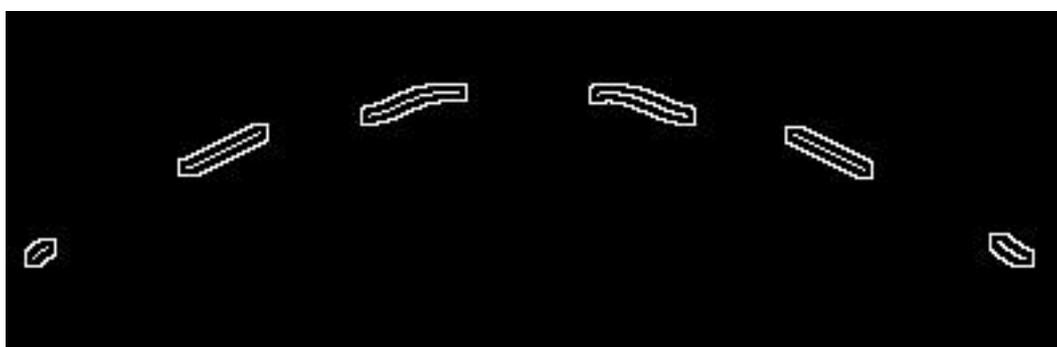


Fig. 5.10 Caminhos de contorno sobre as linhas que conectam os resíduos. Durante o desempacotamento, ao encontrar algum desses caminhos, o algoritmo os utiliza para seguir por um caminho de integração que não cruze as barreiras entre os resíduos.

É importante perceber que a área do mapa no interior dos contornos é inicialmente isolada do processo de desempacotamento e por isso, deve-se procurar minimizar o tamanho das conexões entre os resíduos, para justamente minimizar as regiões isoladas e consequentemente reduzir a quantidade de informação do mapa que não é desempacotada.

A Fig. 5.11 exibe o resultado final do processamento do passo 8, com o mapa de fase desempacotado.

Para efeito de comparação, na Fig. 5.12 o mesmo mapa original foi desempacotado utilizando um algoritmo de Goldstein e na Fig. 5.13 o mapa foi desempacotado, utilizando

um algoritmo de caminho não guiado, com uma varredura linear sem considerar a existência dos resíduos.

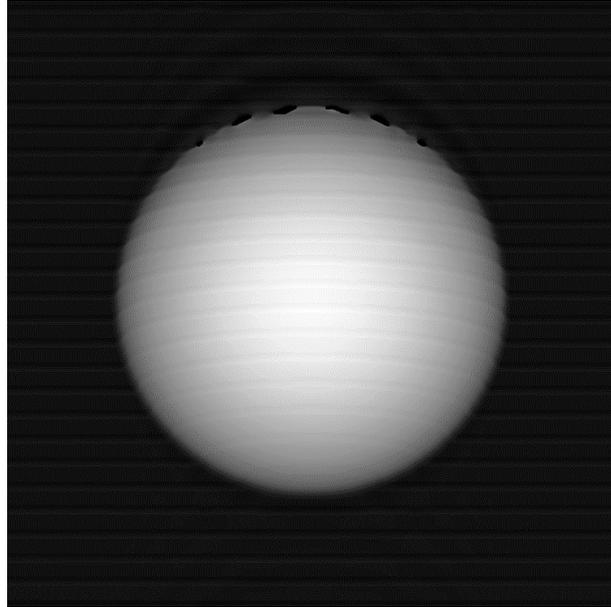


Fig. 5.11 Mapa de fases desempacotado obtido após o processamento do passo 8.

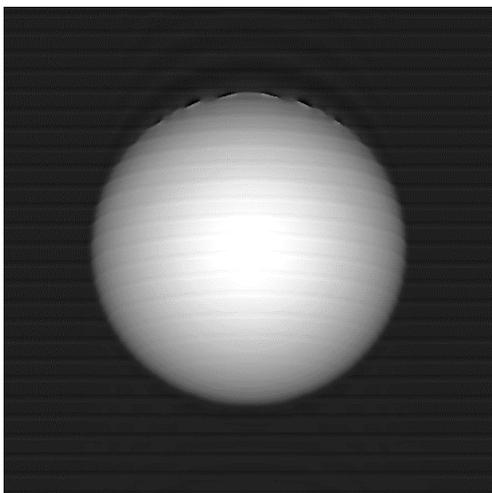


Fig. 5.12 Resultado, para comparação, de desempacotamento utilizando um algoritmo de Goldstein que efetua o balanceamento de resíduos.

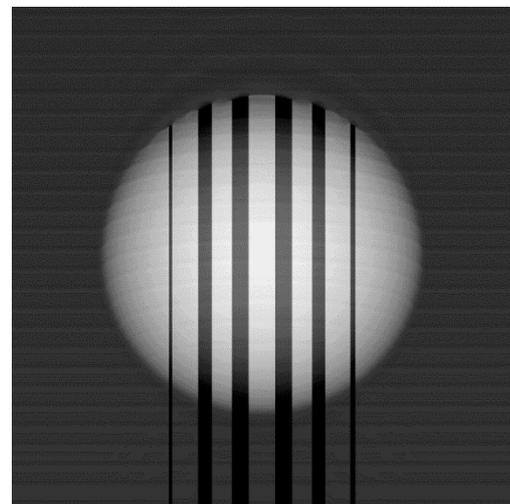


Fig. 5.13 Resultado, para comparação, de desempacotamento linear utilizando um algoritmo simples de caminho não guiado, que desconsidera a existência dos resíduos.

É fácil perceber que, com o algoritmo de caminho não guiado, houve propagação de erros, tendo como origem justamente as regiões que em nosso algoritmo foram identificadas com a presença de resíduos.

Quanto ao mapa gerado pelo algoritmo de Goldstein, que também efetua o balanceamento de resíduos, é possível perceber grande semelhança com os resultados alcançados por nosso algoritmo.

Ao comparar os resultados fica evidente que a abordagem dos resíduos conseguiu identificar e isolar fatores de interferência no mapa, minimizando a perda de informação sem apresentar a propagação de erro ocorrida em um processo linear de desempacotamento e nossa estratégia de balanceamento obteve sucesso ao estimar a fase verdadeira com a mesma qualidade do Goldstein, algoritmo similar consagrado da área.

Por fim, neste capítulo apresentamos a base teórica para nosso algoritmo bem como os passos necessários para sua execução. Para facilitar o entendimento, utilizamos um exemplo ilustrativo da técnica aplicada a um mapa de fase empacotado gerado por computador e comparamos o resultado obtido com uma técnica simples de desempacotamento para evidenciar sua validade.

O mapa de fases utilizado, por ter sido sintetizado artificialmente em computador, tendo todos os seus parâmetros controlados, estava livre de fatores de interferência que não os desejados para a exemplificação efetuada, o que simplificou enormemente a tarefa de execução do desempacotamento.

No próximo capítulo utilizaremos mapas de fase obtidos de forma experimental, cujo valor verdadeiro da fase estará relacionado a quantidades físicas do sistema analisado.

Diferentemente do mapa sintetizado em computador, nestes há a presença de fatores como ruídos e limitações técnicas dos instrumentos de medição que podem afetar

o processo. Neste ponto, a eficácia e as limitações de nosso algoritmo serão colocadas à prova.

Além disso, para demonstrar o avanço em relação às pesquisas na área, compararemos, de forma breve, os resultados obtidos com outro algoritmo de abordagem semelhante, com vistas a ter um parâmetro para avaliação da qualidade dos mapas gerados e o tempo total gasto no processamento.

## Capítulo 6

### RESULTADOS OBTIDOS

*Neste capítulo serão apresentados os resultados da utilização de nosso algoritmo em experimentos práticos, sobre mapas de fase com ruídos inerentes ao sistema físico analisado, além de compararmos com outros algoritmos de forma a avaliar sua qualidade e eficiência.*

Tendo definido a base teórica e detalhado os passos necessários para a execução de nosso algoritmo, o próximo passo é avaliar sua eficiência e limitações ao lidar com mapas obtidos de forma experimental por diferentes métodos e com a presença de ruídos inerentes a cada sistema físico analisado.

Para isso, utilizaremos mapas de fase gerados pela técnica de PTF efetuada sobre padrões de franja de moiré dinâmicos obtidos holograficamente, onde foi utilizado, como meio holográfico, um cristal fotorrefrativo (para maiores detalhes, ver Apêndice A), por projeção direta utilizando um projetor digital e por padrões de franja produzidos por um experimento com um interferômetro de Michelson.

Todos os padrões gerados, por possuírem caráter periódico senoidal, são apropriados para utilização da técnica de PTF para a geração de mapas de fase empacotados.

Além de formas distintas de geração do mapa, para verificar a validade de nossa proposta, compararemos com um algoritmo simples de caminho não guiado e um algoritmo tradicional de Goldstein (disponível para download em [80]) que utiliza balanceamento de resíduos para desempacotar a fase.

Esse último, utilizando a teoria de resíduos, localiza e efetua o balanceamento de cargas verificando, dentre todas as configurações possíveis<sup>14</sup>, aquela com as menores barreiras entre os resíduos e efetua o desempacotamento utilizando um algoritmo de inundação para contornar as regiões proibidas.

O motivo para a escolha do algoritmo de Goldstein para comparação se deve principalmente ao estudo realizado por Ghiglia [10] que efetuou uma comparação com outras propostas e recomenda sua utilização, antes de outras alternativas, pela diversidade de mapas em que apresenta bons resultados e o tempo de processamento reduzido quando comparado aos outros algoritmos analisados por ele.

Inicialmente, a Fig. 6.1 apresenta um padrão de moiré dinâmico projetado sobre um semicilindro e a Fig. 6.2 o mapa de fases empacotado correspondente gerado pela técnica de PTF, desenvolvida em algoritmo próprio na linguagem Matlab.

Se compararmos o mapa com o do capítulo anterior (Fig. 5.1), gerado por computador, podemos observar que o atual apresenta ruídos e descontinuidades diversas a serem tratadas, uma vez que o padrão de franjas utilizado, obtido de forma experimental, sofreu a ação de fatores como a refletância disforme da superfície iluminada e regiões com pouca iluminação, além de flutuações de correntes elétricas no sensor da câmera CCD utilizada para a gravação da imagem e falseamento (*aliasing*) ocasionado pelo processo de conversão do sinal analógico para o digital.

O desempacotamento do mapa utilizando um algoritmo simples de caminho não guiado, que faz uma varredura do mapa de forma linear, sem analisar a presença de fatores que possam comprometer o processo, produziu o resultado apresentado na Fig. 6.4. Como

---

<sup>14</sup> Neste ponto, relembramos que o número de configurações possíveis é  $(S/2)!$ , onde  $S$  é o número de resíduos presentes no mapa. Dependendo do valor de  $S$ , o processamento do algoritmo pode demandar um alto custo computacional.

não localiza nem trata as regiões problemáticas, houve a propagação de erros, que prejudicou severamente a legibilidade do mapa.

A utilização do algoritmo de Goldstein produziu o resultado apresentado na Fig. 6.6. Ao compararmos com o mapa da Fig. 6.4, gerado pelo algoritmo simples de desempacotamento, fica evidenciada a eficiência da teoria dos resíduos como indicativo de problemas no mapa de fase.

Na Fig. 6.8, apresentamos o mapa desempacotado por nosso algoritmo proposto. Ao compararmos o resultado obtido com o mapa da Fig. 6.6, é possível perceber que, além de conseguir minimizar a propagação de erros, nosso algoritmo gerou um mapa com qualidade superior, uma vez que conseguiu estimar a fase verdadeira corretamente em regiões em que o de Goldstein falhou, minimizando a perda de informação durante o processo.

Na Fig. 6.10 e na Fig. 6.11, apresentamos a configuração de barreiras do nosso algoritmo e o de Goldstein. Se compararmos, podemos observar que apesar da localização dos resíduos ser a mesma para ambos, a forma como as cargas foram conectadas é diferente.

Outro método utilizado para a obtenção do mapa de fases foi a projeção de um padrão regular de franjas senoidais gerado por computador, projetado sobre a superfície de uma semi-rosca (Fig. 6.12) e capturado por uma câmera CCD.

A Fig. 6.13 e a Fig. 6.14 contêm o mapa de fases gerado pela técnica de PTF sobre o padrão de franjas, a Fig. 6.15 e a Fig. 6.16 apresentam o mapa de fases desempacotado utilizando o algoritmo linear de caminho não guiado, a Fig. 6.17 e a Fig. 6.18 contêm o mapa gerado pelo algoritmo de Goldstein e a Fig. 6.19 e a Fig. 6.20 apresentam o mapa gerado pelo algoritmo proposto nesse trabalho.

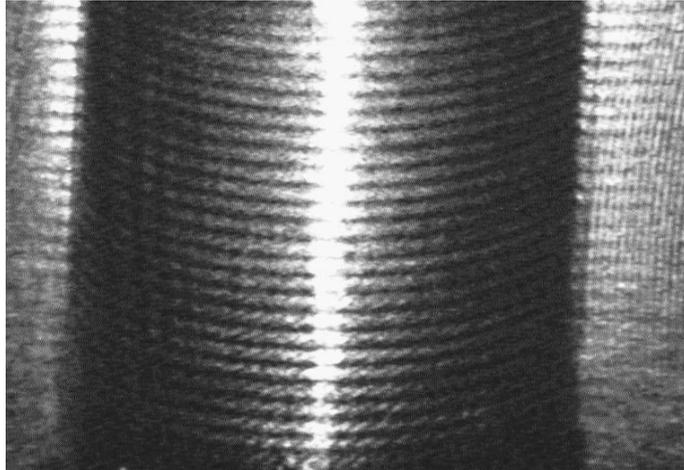


Fig. 6.1 Imagem com 700x900 píxeis, do padrão de franjas de moiré dinâmico projetado sobre a superfície de um semicilindro.

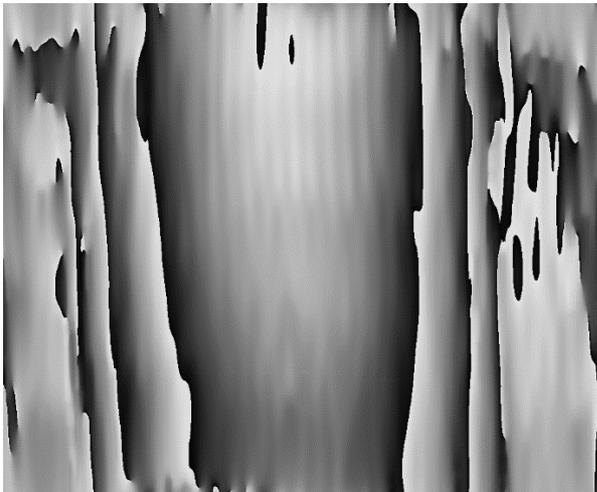


Fig. 6.2 Mapa de fases empacotado, obtido através da aplicação da técnica de PTF sobre o padrão de franjas da Fig. 6.1.

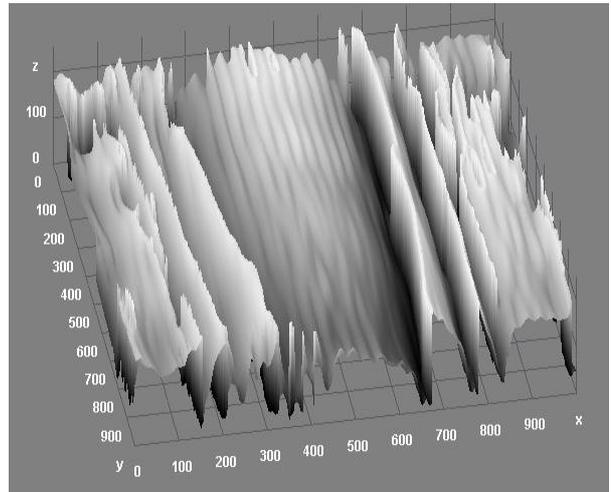


Fig. 6.3 Mapa de fases da Fig. 6.2, visto sob uma perspectiva tridimensional.

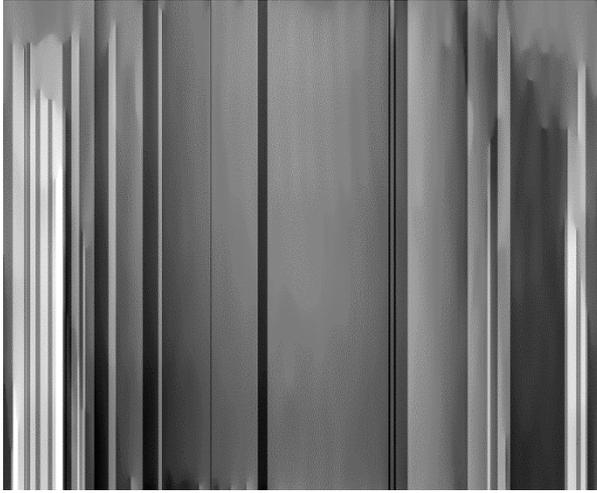


Fig. 6.4 Mapa de fases desempacotado após a utilização de um algoritmo simples que fez a verredura linear do mapa da Fig. 6.2.

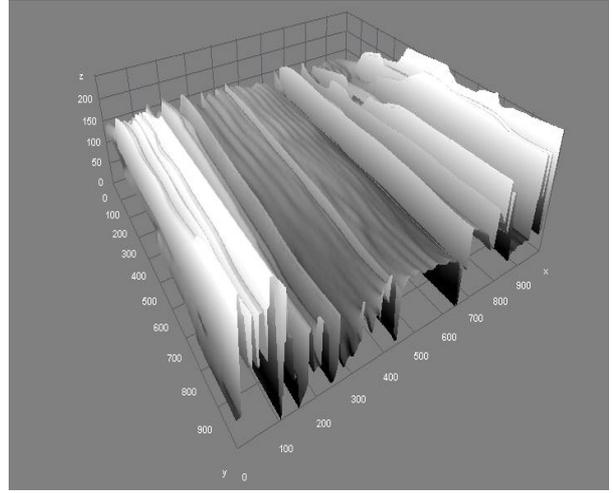


Fig. 6.5 Mapa de fases da Fig. 6.4, visto sob uma perspectiva tridimensional.

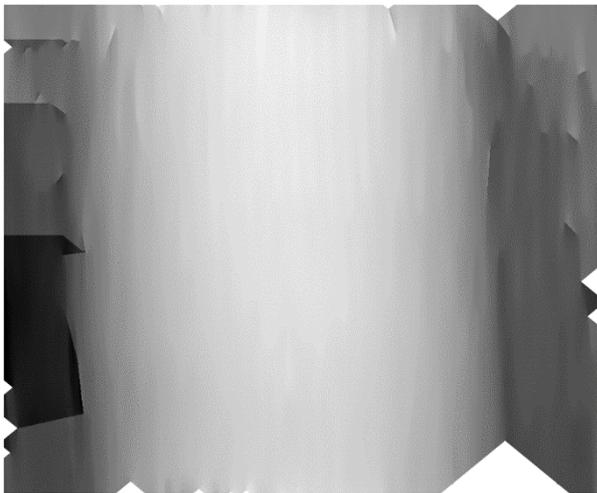


Fig. 6.6 Mapa de fase desempacotado obtido através da utilização do algoritmo de Goldstein sobre o mapa da Fig. 6.2

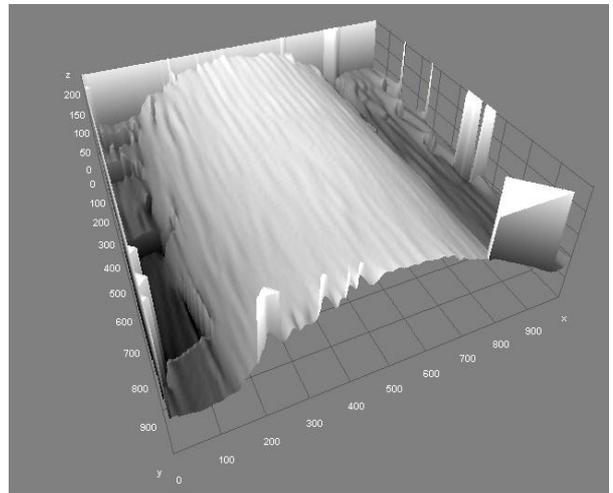


Fig. 6.7 Mapa de fases da Fig. 6.6 visto sob uma perspectiva tridimensional.

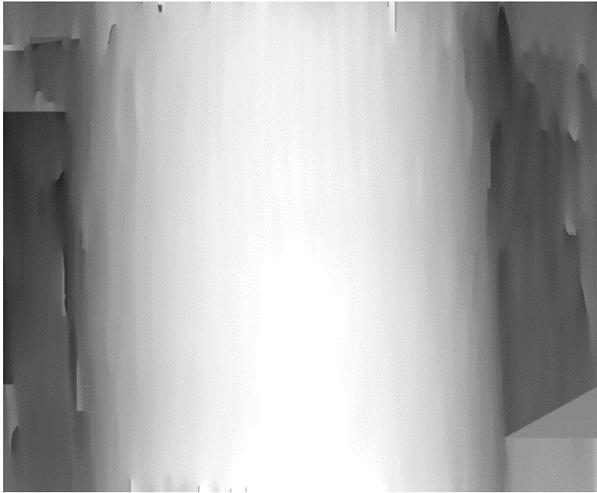


Fig. 6.8 Mapa de fases descompactado obtido através da utilização do algoritmo proposto neste trabalho sobre o mapa da Fig. 6.2.

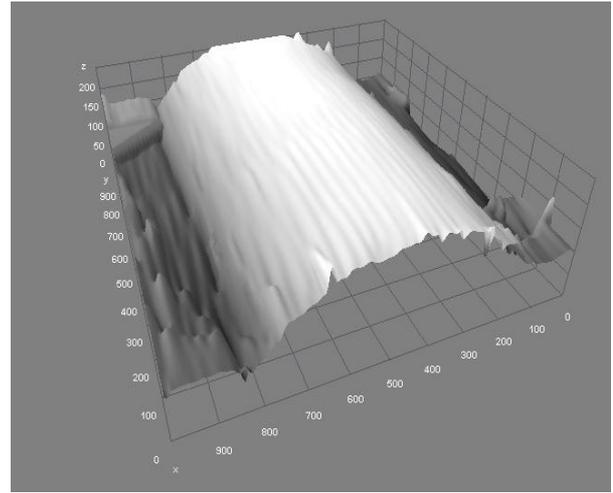


Fig. 6.9 Mapa de fases da Fig. 6.8, visto sob uma perspectiva tridimensional.

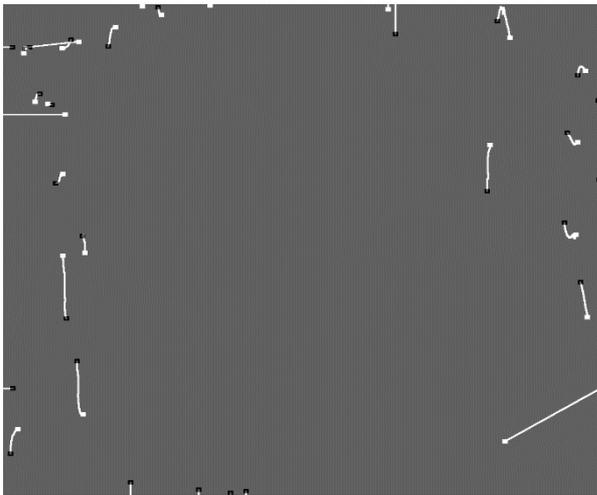


Fig. 6.10 Configuração de barreiras para balanceamento de cargas de resíduos efetuada pelo algoritmo proposto nesse programa.

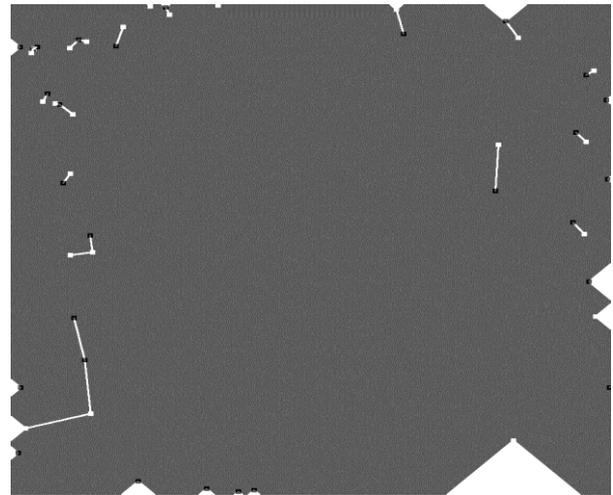


Fig. 6.11 Configuração de barreiras para balanceamento de cargas de resíduos efetuada pelo algoritmo de Goldstein.

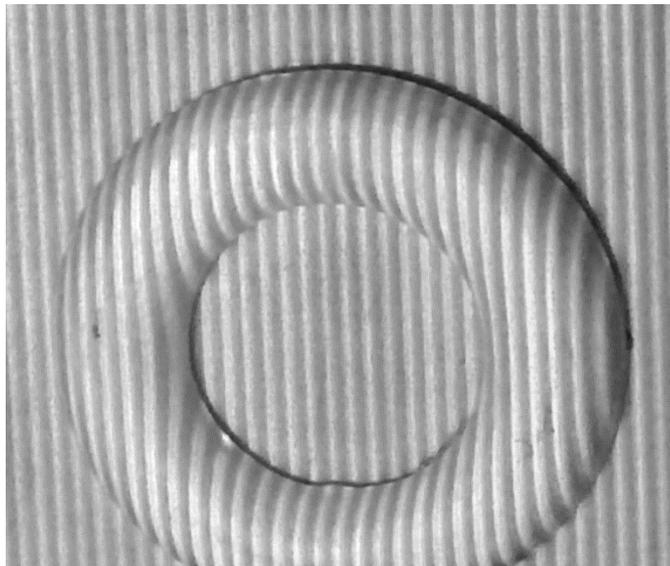


Fig. 6.12 Imagem com 1900x2200 píxeis, do padrão de franjas gerado por computador e projetado por um projetor digital sobre uma semi-rosca.

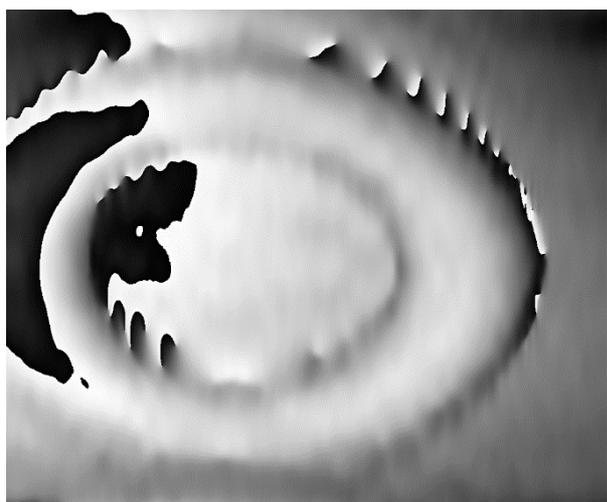


Fig. 6.13 Mapa de fases empacotado, obtido através da aplicação da técnica de PTF sobre o padrão de franjas da Fig. 6.12.

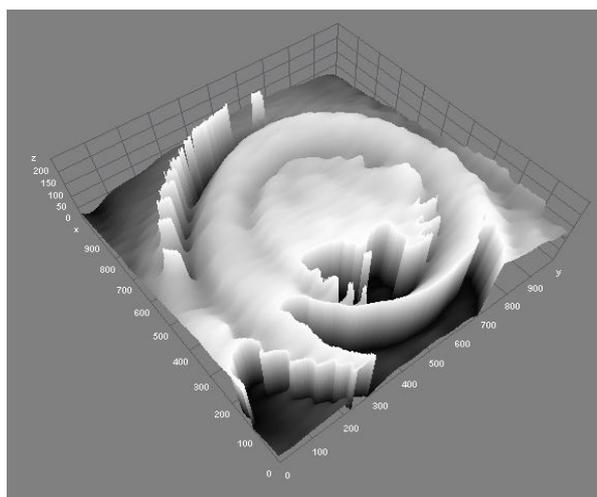


Fig. 6.14 Mapa de fases da Fig. 6.13 visto sob uma perspectiva tridimensional.

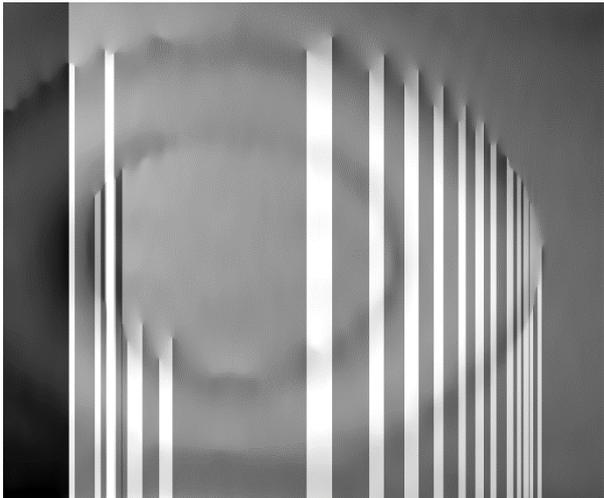


Fig. 6.15 Mapa de fases desempacotado após a utilização de um algoritmo simples que fez a verredura linear do mapa da Fig. 6.13.

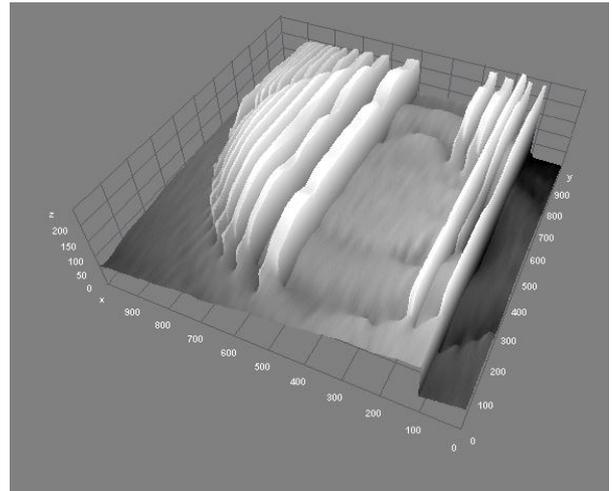


Fig. 6.16 Mapa de fases da Fig. 6.15 visto sob uma perspectiva tridimensional.

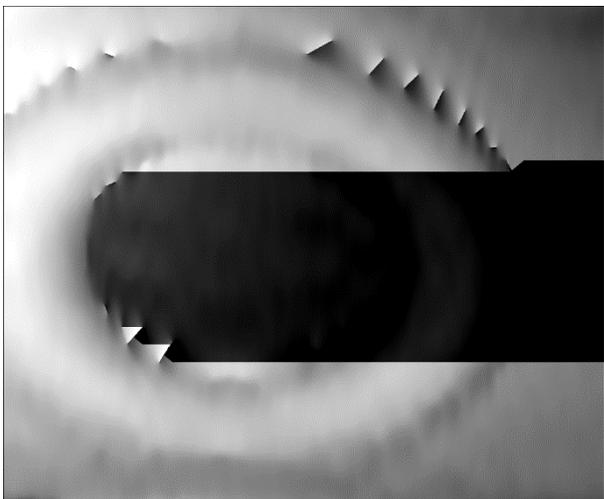


Fig. 6.17 Mapa de fases desempacotado obtido após a utilização do algoritmo de Goldstein sobre o mapa da Fig. 6.13.

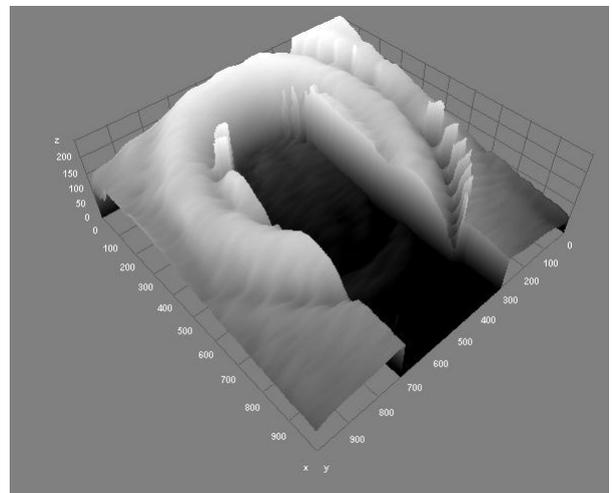


Fig. 6.18 Mapa de fases da Fig. 6.17 visto sob uma perspectiva tridimensional.

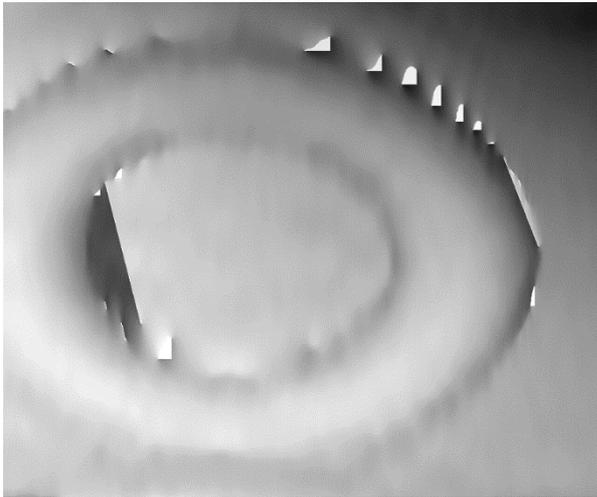


Fig. 6.19 Mapa de fases desempacotado obtido através da utilização do algoritmo proposto neste trabalho sobre o mapa da Fig. 6.13.

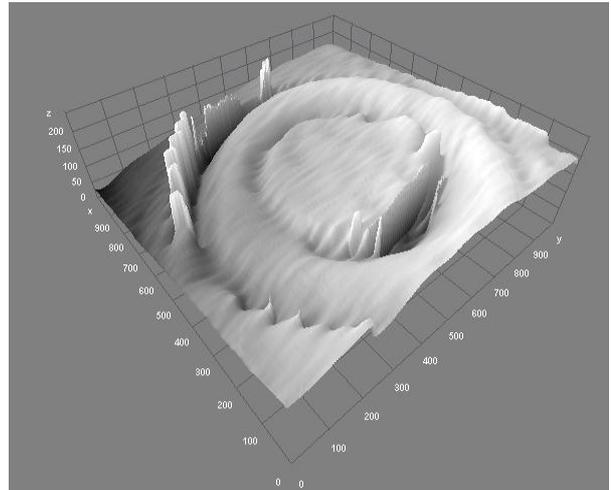


Fig. 6.20 Mapa de fases da Fig. 6.19, visto sob uma perspectiva tridimensional.

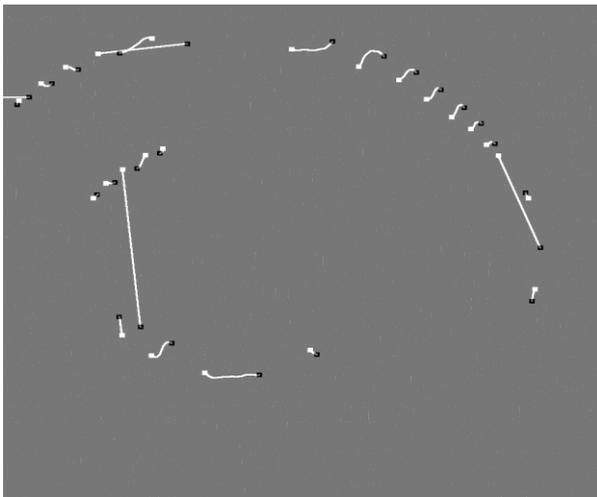


Fig. 6.21 Configuração de barreiras para balanceamento de cargas de resíduos efetuada pelo algoritmo proposto nesse programa.

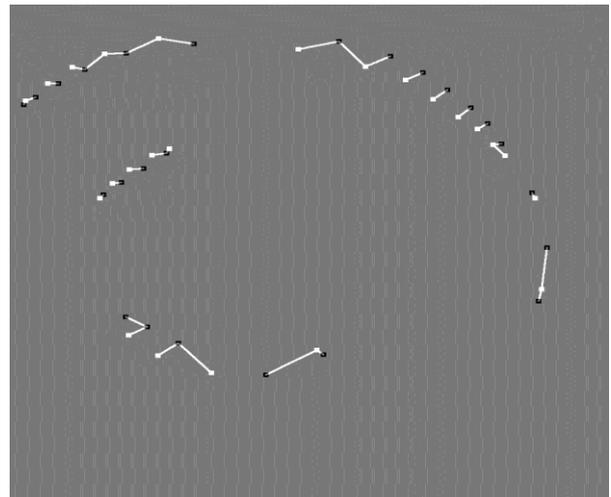


Fig. 6.22 Configuração de barreiras para balanceamento de cargas de resíduos efetuada pelo algoritmo de Goldstein.

A Fig. 6.21 e a Fig. 6.22 contêm a configuração de barreiras do nosso algoritmo e o de Goldstein. Novamente podemos perceber a diferença entre os resultados alcançados por cada técnica.

O desempacotamento linear não fez a verificação e isolamento das regiões problemáticas e propagou erros por todo o mapa, comprometendo severamente o resultado final alcançado.

O algoritmo de Goldstein apresentou qualidade superior ao de caminho não guiado, uma vez que identificou e isolou regiões problemáticas.

Mas ao comparar os resultados obtidos com o mapa gerado pelo algoritmo proposto nesse trabalho, podemos observar que o nosso apresentou maior eficiência na detecção e isolamento das regiões problemáticas, reduzindo a propagação de erros e conseqüentemente a quantidade de informação perdida.

Ao compararmos nosso algoritmo especificamente com o de Goldstein, que também utiliza a teoria de resíduos, só que com uma estratégia diferente para conexão das cargas, podemos observar que o de Goldstein propagou erros por grande parte do mapa (Fig. 6.18), comprometendo uma quantidade relativamente grande de informação enquanto que o nosso conseguiu conter o problema, não apresentando a mesma propagação (Fig. 6.20).

Por fim, apresentamos na Fig. 6.23 um padrão regular de franjas de interferência, gerado por um interferômetro de Michelson e projetado sobre a superfície de um semicilindro.

A Fig. 6.24 e a Fig. 6.25 contêm o respectivo mapa de fases gerado pela utilização da técnica de PTF sobre a Fig. 6.23.

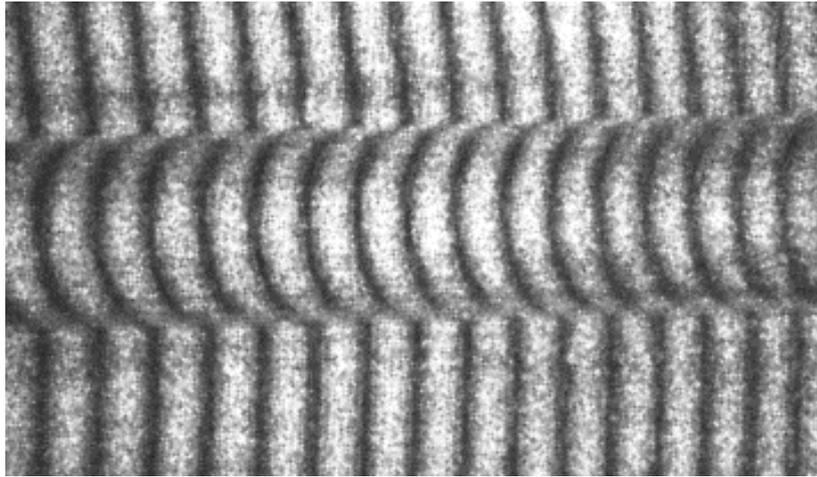


Fig. 6.23 Imagem com 200x300 píxeis, do padrão de franjas de interferência gerado por um interferômetro de Michelson e projetado sobre a superfície de um semicilindro.

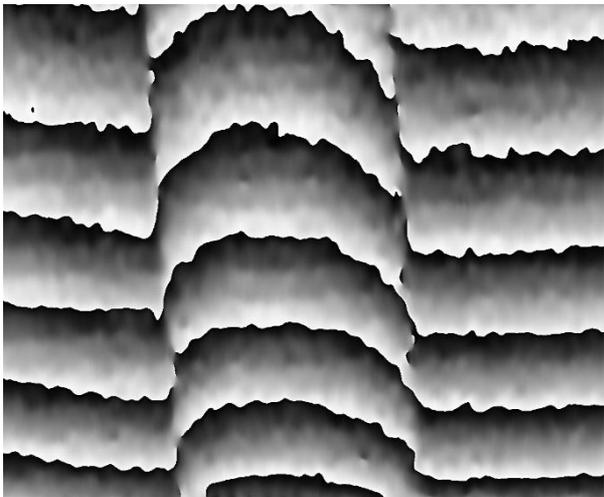


Fig. 6.24 Mapa de fases empacotado, obtido através da aplicação da técnica de PTF sobre o padrão de franjas da Fig. 6.23.

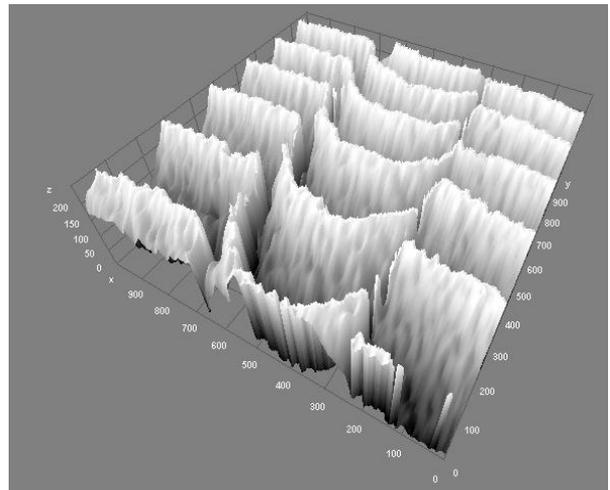


Fig. 6.25 Mapa de fases da Fig. 6.24 visto sob uma perspectiva tridimensional.



Fig. 6.26 Mapa de fases desempacotado após a utilização de um algoritmo simples que fez a verredura linear do mapa da Fig. 6.24.

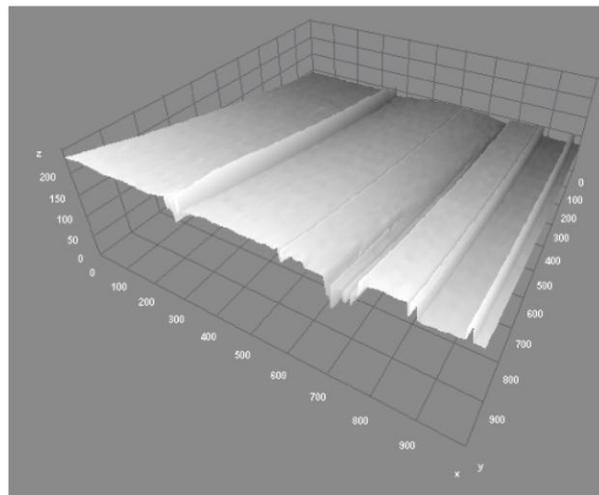


Fig. 6.27 Mapa de fases da Fig. 6.26 visto sob uma perspectiva tridimensional.

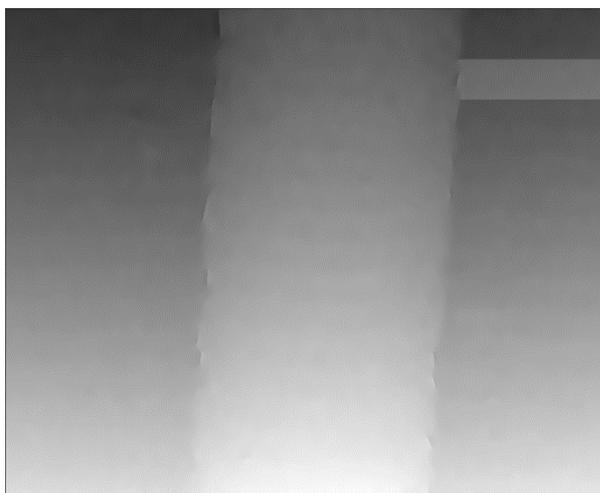


Fig. 6.28 Mapa de fases desempacotado obtido após a utilização do algoritmo de Goldstein sobre o mapa da Fig. 6.23.

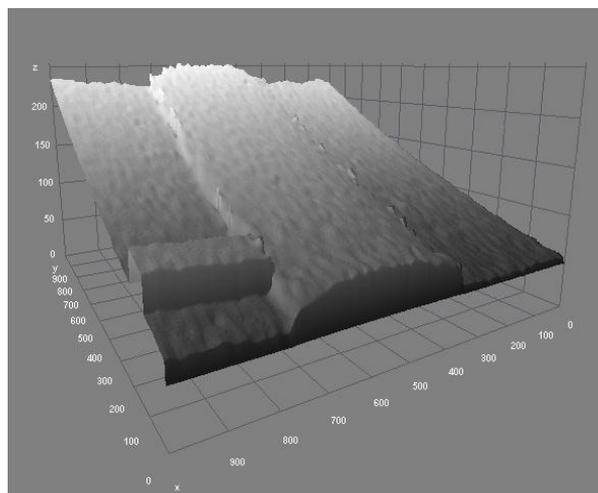


Fig. 6.29 Mapa de fases da Fig. 6.28 visto sob uma perspectiva tridimensional.

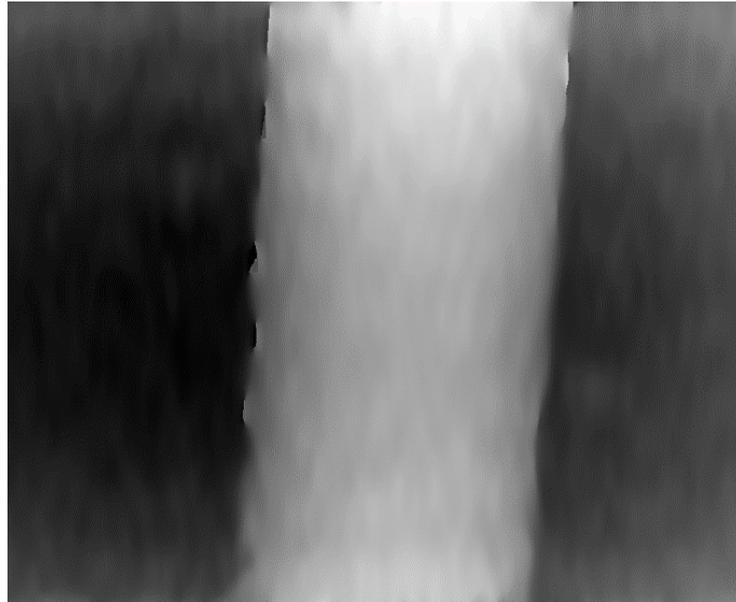


Fig. 6.30 Mapa de fases desempacotado obtido através da utilização do algoritmo proposto neste trabalho sobre o mapa da Fig. 6.24.

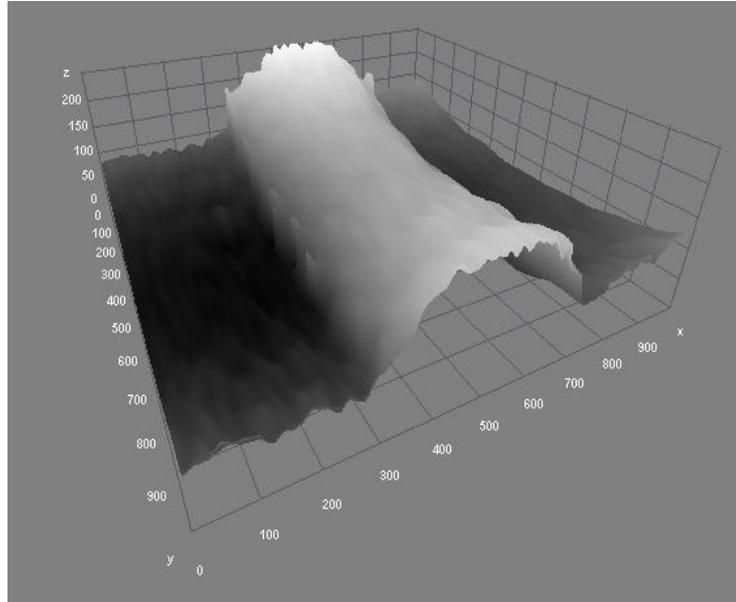


Fig. 6.31 Mapa de fases da Fig. 6.30, visto sob uma perspectiva tridimensional.

Na Fig. 6.26 e na Fig. 6.27 podemos observar o resultado do desempacotamento utilizando um algoritmo linear de caminho não guiado, na Fig. 6.28 e na Fig. 6.29, o resultado do algoritmo de Goldstein e na Fig. 6.30 e na Fig. 6.31, do nosso algoritmo.

Também apresentamos na Fig. 6.32 e na Fig. 6.33, para comparação, a configuração das barreiras definida por nosso algoritmo e por Goldstein. Se observarmos as regiões marcadas em vermelho, ampliadas na Fig. 6.34 (a) e (b), é possível perceber que apesar de ambas as configurações serem bastante similares, a diferença reside justamente na região onde houve propagação de erro no algoritmo de Goldstein. É possível observar, inclusive, que o algoritmo de Goldstein falha ao conectar alguns resíduos de mesma carga nesta região.

Comparando com os outros dois métodos, o desempacotamento efetuado com nosso algoritmo apresentou qualidade superior também para esse experimento. Neste ponto, ao atentarmos para o fato de que tanto a nossa proposta quanto a de Goldstein utilizam a teoria de resíduos para a identificação de regiões problemáticas, a diferença na estratégia de configuração das barreiras indica ser de grande impacto na qualidade do resultado final.

Além da análise da imunidade a ruídos e qualidade do mapa gerado, outro fator de grande importância para a utilização em aplicações práticas é o custo computacional, ou seja, a quantidade de processamento exigida do computador para a execução do algoritmo.

Como critério de avaliação, efetuamos a comparação do tempo de processamento de nosso algoritmo com o de Goldstein nos três mapas empacotados, anteriormente analisados.

O algoritmo linear foi descartado dessa comparação, pois apesar de terminar o desempacotamento em tempo muito reduzido quando comparado à maioria dos programas atuais, não foi capaz de gerar mapas com qualidade satisfatória durante os experimentos.

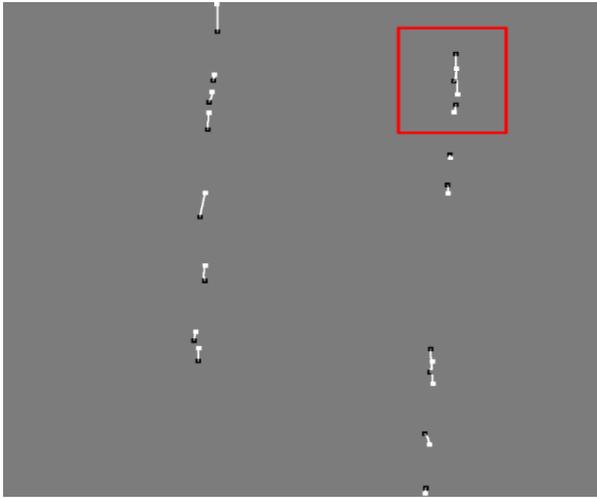


Fig. 6.32 Configuração de barreiras para balanceamento de cargas de resíduos efetuada pelo algoritmo proposto nesse programa.

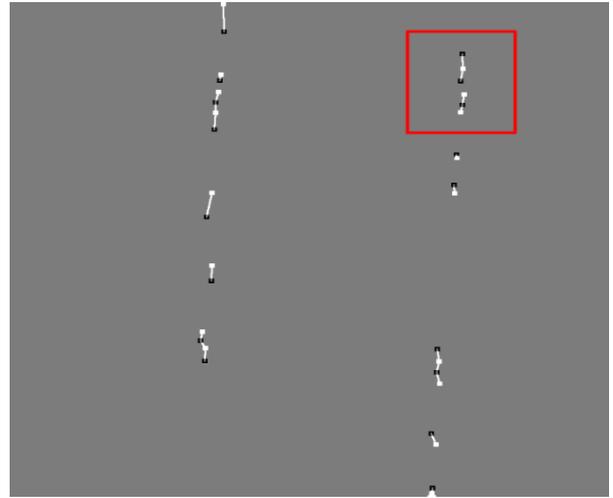
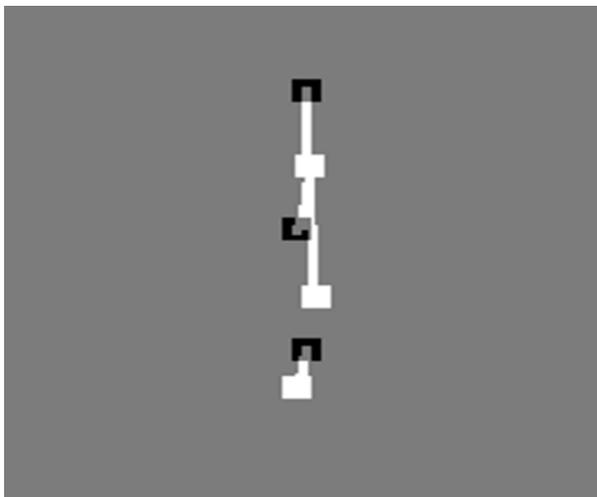
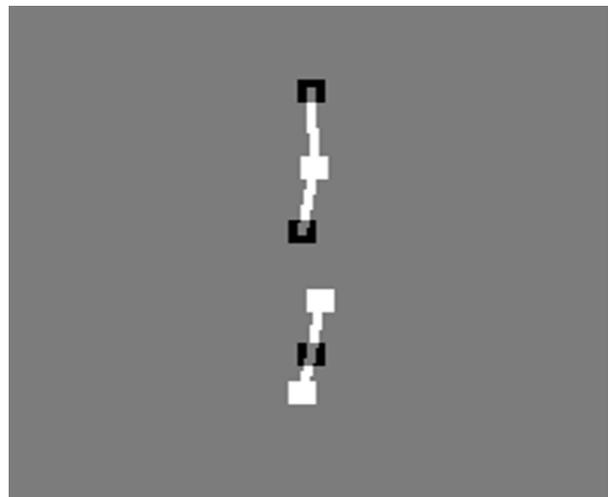


Fig. 6.33 Configuração de barreiras para balanceamento de cargas de resíduos efetuada pelo algoritmo de Goldstein.



(a)



(b)

Fig. 6.34 Ampliação de região com configuração de barreiras de balanceamento dos resíduos, em nosso algoritmo (a) e o algoritmo de Goldstein (b).

Para a medição do tempo, ambos os algoritmos foram executados pelo programa Matlab® versão R2012a da MathWorks® em um micro com Windows 7 de 64 bits. Apesar de lento quando comparado a outras linguagens de programação como o C++, entendemos não haver prejuízo para a análise uma vez que nosso objetivo é comparar o rendimento entre os dois algoritmos.

O hardware utilizado para os testes foi um microcomputador Desktop, com HD de 500GB, 8GB de memória DDR3 com frequência de 1333 Mhz e processador AMD Phenon II X6 1055T com seis núcleos trabalhando a 2.8 Ghz cada um.

O Quadro 2 contém a média dos tempos obtidos com os dois algoritmos para cada mapa, após cinco execuções seguidas.

### Quadro 2

Comparação entre os tempos de processamento do algoritmo de Goldstein e o proposto nesse trabalho

Experimento	Tempo de processamento do algoritmo (em minutos)	
	Goldstein	Algoritmo proposto
Projeção de franjas dinâmicas de Moiré (Fig. 6.2)	5:32	0:57
Projeção direta de franjas por projetor digital (Fig. 6.12)	3:16	0:53
Projeção de franjas obtidas em experimento com interferômetro de Michelson (Fig. 6.23)	4:55	0:55

O tempo de processamento de nosso algoritmo chegou a quase 1/6 do tempo do de Goldstein.

Nossa técnica, além de produzir resultados em tempo inferior ao algoritmo clássico da área, como visto anteriormente, possui qualidade superior, baixa complexidade, uma vez que necessita de apenas uma imagem do mapa de fases empacotado e não utiliza mapas de qualidade, simplificando e facilitando sua aplicação prática em situações dinâmicas em que exista dificuldade na obtenção de múltiplos mapas ou que o tempo de desempacotamento seja fator crítico para a análise.

Por fim, apresentamos no Quadro 3, um esquema comparativo das características de nosso algoritmo, que acreditamos ser responsáveis pelo bom desempenho apresentado.

### Quadro 3

Lista comparativa entre os fatores de desempenho do algoritmo de Goldstein e o proposto nesse trabalho

<b>Goldstein</b>	<b>Algoritmo proposto</b>
Efetua $(S/2)!$ ligações de resíduos, onde S é o número total de resíduos no mapa e calcula a área isolada para cada configuração de forma a escolher aquela que minimiza as regiões isoladas.	Dentre duas a quatro configurações já estabelecidas, verifica aquela que possui os menores caminhos de ligação entre resíduos de carga oposta.
Estabelece um caminho de conexão entre os resíduos, às vezes, cruzando regiões com informações úteis, que acabam sendo isoladas.	Utiliza as linhas de salto de fase dos mapas empacotados, como caminhos pré-estabelecidos, priorizando o isolamento de áreas com falsos saltos de fase.
Faz uma verificação da qualidade dos píxeis, para priorizá-los durante o processo de desempacotamento.	Não avalia a qualidade dos píxeis a serem empacotados

## Capítulo 7

# CONCLUSÕES E PERSPECTIVAS FUTURAS

### 7.1 CONCLUSÕES

O objetivo principal dessa dissertação foi explorar o problema do desempacotamento de fase em aplicações de imageamento e propor um algoritmo mais eficiente e versátil, que possa ser utilizado nas aplicações desenvolvidas no Lab. Óptica UFF e em outros tipos de mapas de fase, gerado por processos distintos.

Para isso, abordamos nesse trabalho a importância da fase como componente essencial na análise de imagens, uma vez que pode estar relacionada a diversas quantidades físicas de interesse e como a extração e análise de sua informação tem aplicação em várias áreas como Engenharia, Biologia, Astronomia, Medicina e até para fins militares (inSar).

Apresentamos o empacotamento como problema advindo da extração da informação de fase das imagens e o desempacotamento como necessidade para obtenção dos valores verdadeiros dessa fase.

Explicamos e demonstramos como a presença de ruídos diversos pode afetar o desempacotamento e até causar a falha global do processo, aumentando enormemente a complexidade e a dificuldade da tarefa e mencionamos que, em busca de uma solução para o problema, inúmeros pesquisadores tem desempenhado esforços no desenvolvimento de algoritmos mais robustos de desempacotamento.

Para apresentar o estado da arte das pesquisas na área, descrevemos alguns dos principais métodos desenvolvidos, passando por aqueles que necessitam de várias imagens, até aqueles que necessitam de apenas uma, com diferentes níveis de complexidade, tempo de processamento e imunidade a ruídos, além de comparar suas principais vantagens e desvantagens com a indicação dos cenários mais apropriados para sua utilização.

Introduzimos a teoria de resíduos como uma abordagem para localizar e tratar as regiões do mapa com potencial para propagação de erros durante o processo.

Demonstramos que fatores como descontinuidades de superfície e ruídos provenientes de falseamento (*aliasing*) por baixa taxa de amostragem, variações na refletância de superfícies, iluminação de fundo, entre outros, podem ser associados à presença de cargas de resíduos no mapa e que ao conectar cargas de valor oposto através de barreiras que delimitam regiões que não podem ser cruzadas durante o processo de desempacotamento, conseguimos diminuir e limitar a propagação de erros no mapa.

Nessa abordagem, o problema se resume em detectar e conectar os resíduos de cargas opostas e desde que a soma das cargas conectadas seja zero, não há restrição para a configuração de conexão a ser escolhida, porém, tem se buscado a minimização das barreiras para a diminuição da informação perdida durante o processo.

Os algoritmos que se baseiam nessa estratégia são chamados de balanceamento de resíduos ou de *branch-cuts*, que por sua facilidade de utilização, baixa complexidade e baixo custo computacional, quando comparado a outras técnicas como a de multifrequências, muitos pesquisadores têm buscado formas de otimização e melhora na qualidade do mapa gerado.

Com o objetivo de propor uma solução alternativa ao problema da minimização das barreiras, com vistas a uma melhora na qualidade e redução de tempo de

processamento, demonstramos que ao adicionar uma fase constante a todo o mapa empacotado, obtemos outro, com as regiões de salto deslocadas, porém, com a localização dos resíduos inalterada e propusemos um algoritmo que utiliza a análise dessas regiões de salto para estabelecer a conexão entre resíduos de carga oposta no mapa empacotado original.

Para avaliar sua imunidade a ruídos, o algoritmo proposto foi testado utilizando mapas de fase obtidos por métodos experimentais distintos, utilizados em algumas das aplicações do Lab. Óptica UFF e se mostrou robusto ao conseguir reconhecer as regiões problemáticas e conter a propagação de erros durante o desempacotamento.

Para avaliar sua eficiência, sobre cada mapa de fase empacotado, os resultados alcançados por nosso algoritmo foram comparados: inicialmente com um programa simples, que faz a varredura linear do mapa, para avaliar a efetividade da utilização dos resíduos na detecção de regiões problemáticas, e posteriormente a um algoritmo clássico de Goldstein, que também utiliza a teoria de resíduos mas difere do nosso quanto à estratégia de balanceamento de carga e forma de realização de varredura do mapa para remoção das descontinuidades.

Em todos os testes efetuados, nosso algoritmo se mostrou superior, conseguindo impedir a propagação de erros em regiões que os outros dois algoritmos falharam.

Outra análise efetuada foi quanto ao tempo de processamento, uma vez que este pode ser um fator crítico na utilização de nosso algoritmo em algumas aplicações, como por exemplo, a análise dinâmica de densidades de fluídos em um microscópio holográfico digital.

Novamente, quando comparado ao algoritmo de Goldstein, nossa proposta, além de conseguir minimizar a propagação de erros, ainda apresentou um tempo de processamento total de até 1/6 ou 17% do tempo consumido pelo outro algoritmo.

Ao que tudo indica, o ganho, tanto em termos de imunidade a ruídos quanto na apresentação de resultados mais rápidos, está fortemente relacionado à estratégia adotada para a escolha da configuração de conexão entre as cargas.

Frente aos resultados obtidos, concluímos que nossa técnica trouxe avanços em termos de qualidade, redução de tempo de processamento e praticidade, uma vez após iniciado o processamento, não é necessário nenhuma interação com o usuário.

Durante o desenvolvimento deste trabalho, inúmeros algoritmos, com hipóteses distintas foram explorados. Os testes efetuados e os resultados obtidos nos direcionaram para o programa atualmente utilizado, uma vez que nosso objetivo, além de contribuir com avanços nas pesquisas na área, foi o de criar uma ferramenta versátil, que possa, de forma rápida e dinâmica, analisar as informações obtidas nos diversos experimentos realizados no Lab. Óptica UFF e produzir resultados confiáveis, na análise de mapas de fase empacotados.

Por fim, em resumo, após os experimentos efetuados, podemos concluir que nossa técnica apresentou as seguintes vantagens:

- Redução de tempo de processamento, quando comparado a algoritmo similar;
- Versatilidade ao efetuar o desempacotamento de mapas oriundos de métodos experimentais distintos;
- A necessidade de apenas um mapa de fase para realização do desempacotamento;
- Boa imunidade a ruídos quando comparado a algoritmo similar;
- Performance estável ao analisar imagens de tamanhos ou resoluções diferentes;

- Não necessita de interação com o usuário após iniciado o programa e selecionado o mapa;
- Minimiza a região isolada no empacotamento e a consequente perda de informação, permitindo que sejam apresentados no mapa, detalhes que podem ser úteis para análises a serem efetuadas;

## **7.2 PERSPECTIVAS FUTURAS**

Com os resultados obtidos, planejamos estudar de forma mais aprofundada a relação das cargas de resíduo com problemas nos mapas de fase e explorar os limites e aplicabilidade de nossa técnica em mapas com diferentes níveis de intensidades e tipos de ruídos, provenientes de outras aplicações tecnológicas de imageamento como, por exemplo, topografia de superfície por radar de abertura sintética, e o microscópio holográfico digital por reflexão desenvolvido no Lab. Óptica UFF.

Na busca por um algoritmo definitivo de desempacotamento, desenvolveremos estudos comparativos de nossa técnica com outras propostas como as globais de minimização de norma e locais temporais de multifrequência.

Por fim, esperamos que nos testes comparativos a serem futuramente efetuados, nossa técnica seja efetiva na produção de resultados onde outros algoritmos falham, e naqueles em que forem insatisfatórios, a causa do problema possa ser identificada e sirva de indicativo para o aperfeiçoamento da teoria e técnica desenvolvidas nessa dissertação.

# APÊNDICE A

## A.1 – GERAÇÃO DE PADRÕES DE FRANJAS DE MOIRÉ

Os padrões de moiré podem ser obtidos da interação da luz difratada entre duas estruturas periódicas superpostas. Como será descrito nesta seção, a superposição de redes de Ronchi com espaçamentos entre as linhas levemente diferentes, por exemplo, produzem franjas de luz espaçadas regularmente (Fig. (A.1.1)).

A interpretação física da geração de um padrão de franjas de moiré é melhor explicada através da obstrução da luz transmitida pelas redes superpostas. Será considerada aqui a superposição de duas redes de Ronchi, que são redes quadradas compostas de linhas horizontais claras e escuras igualmente espaçadas e com alto contraste entre claro e escuro. As redes superpostas possuem espaçamentos entre as linhas levemente diferentes,  $p_1$  e  $p_2$  conforme Fig. (A.1.1). As regiões escuras de cada rede obstruem a luz transmitida (transmitância = 0). Enquanto que as linhas claras transmitem

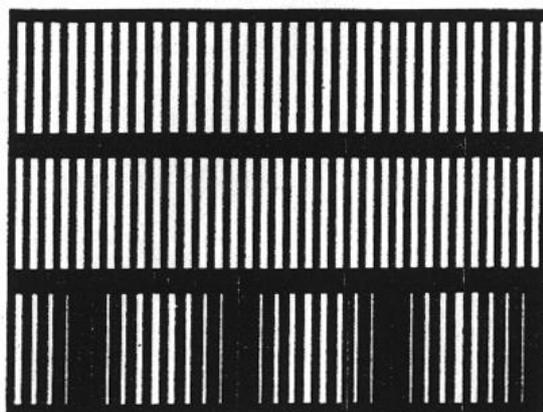


Fig. (A.1.1) Padrão de moiré resultante da superposição de redes de Ronchi com espaçamentos entre as linhas levemente diferentes  $p_1$  e  $p_2$  descrito pelo conceito de obstrução de transmitância de luz através de redes superpostas.

a luz (transmitância = 1). Quando as duas redes são superpostas, as transmitâncias da rede original são moduladas por um padrão de moiré de frequência mais baixa. Nas regiões onde as linhas claras de uma rede coincidem com as linhas claras da segunda rede a transmitância é 1. Esta é a franja de moiré clara. Por outro lado, quando a linha escura de uma rede coincide com a linha clara ou escura da segunda rede, a transmitância média é zero. Esta corresponde a franja de moiré escura. As transmitâncias variam linearmente entre estes limites.

Utilizando o formalismo matemático para indicar a posição das franjas, as redes são substituídas por um conjunto discreto de linhas descrito por duas equações, cada uma especificando a localização de uma linha no conjunto. Logo, as equações para duas redes de Ronchi com espaçamentos entre as linhas levemente diferentes,  $p_1$  e  $p_2$  ao longo do eixo  $y$  são

$$\begin{cases} y = np_2 \\ y = mp_1 \end{cases}, \quad (\text{A.1.1})$$

onde  $p_1 = kp_2$ . As Eq. (A.1.1) podem ser escritas como

$$\begin{cases} y = np_2 \\ y/k = mp_2 \end{cases}. \quad (\text{A.1.2})$$

Isto produz,

$$l = y \left( \frac{1}{p_2} - \frac{1}{p_1} \right), \quad l = n - m \quad (\text{A.1.3})$$

que é o fenômeno de batimento na técnica de moiré. Em outras palavras, um padrão na direção da rede original terá um espaçamento entre as franjas  $p$  dado por

$$\frac{1}{p} = \frac{1}{p_2} - \frac{1}{p_1}, \quad (\text{A.1.4})$$

que é análogo ao batimento no comprimento de onda na interferometria

$$\frac{1}{\lambda} = \frac{1}{\lambda_2} - \frac{1}{\lambda_1}. \quad (\text{A.1.5})$$

Além da descrição da formação de padrões de moiré através da superposição de duas redes de Ronchi, pode-se descrever a formação destes padrões através da superposição de quaisquer outras estruturas repetitivas. Entretanto isto não será descrito aqui, pois foge ao escopo do presente trabalho.

Os padrões de moiré podem ser produzidos de duas formas. A primeira é superpor duas redes com um pequeno ângulo de rotação entre elas [81], a segunda consiste em superpor duas redes de fase senoidais com uma pequena diferença de fase entre elas. Este processo será descrito na próxima seção.

## A.2 – GERAÇÃO DE PADRÕES DE MOIRÉ DINÂMICOS POR CRISTAIS FOTORREFRATIVOS

Redes de fase senoidais dinâmicas podem ser produzidas em materiais que exibem o efeito fotorrefrativo [82]. Tal efeito se deve a uma combinação bem controlada de fotocondutividade e o efeito eletroóptico [83]. A superposição de duas redes de fase senoidais dinâmicas com uma pequena diferença de fase entre elas e com alta frequência espacial (ao invés de redes de Ronchi com baixa frequência espacial e amplitude permanente), gera padrões de moiré em tempo real (dinâmicos).

Para descrever a geração dos padrões de moiré dinâmicos vamos levar em conta o processo de fotorrefração linear no cristal fotorrefrativo  $\text{Bi}_{12}\text{TiO}_{20}$  (BTO) utilizado nos experimentos desse trabalho.

Sabemos que qualquer estrutura periódica pode ser simplesmente representada por uma função do tipo [84],

$$f(x,y) = c n , \quad (\text{A.2.1})$$

onde  $c$  é uma constante e  $n$  é um inteiro que define uma ordem no padrão de franjas relacionado.

Como visto na seção anterior, padrões de moiré podem ser obtidos da interação de luz difratada entre duas estruturas periódicas regulares superpostas. A superposição de duas estruturas periódicas dadas pela Eq. (A.2.1) será

$$\frac{f_1(x,y)}{c_1} \pm \frac{f_2(x,y)}{c_2} = m , \quad (\text{A.2.2})$$

onde  $c_1$  e  $c_2$  são constantes e  $m$  é o inteiro que define cada ordem do padrão de franjas gerado equivalente. A Eq.(A.2.2) representa a estrutura de superposição do padrão de moiré.

Uma rede de fase senoidal dinâmica é gerada após uma amostra de cristal BTO ser iluminada por um padrão de interferência dado pela Eq.(A.2.1)

$$I(x) = I_o \left[ 1 + m \cos(k_g x) \right]. \quad (\text{A.2.3})$$

Após a geração desta rede de fase senoidal dinâmica no volume da amostra, os feixes incidentes são interrompidos. A direção de um dos feixes é levemente modificada e a amostra é iluminada por um novo padrão de interferência. Uma nova rede será gerada no volume da amostra. A geração da nova rede deve acontecer em um tempo suficiente para que a antiga rede não seja removida do sistema.

Duas redes de fase senoidais com frequências espaciais levemente diferentes foram geradas. Isto implica na modulação de dois índices de refração  $\Delta n_1(x)$  e  $\Delta n_2(x)$ , ambas definidas pela Eq. (A.2.4)

$$\begin{cases} \Delta n_1(x) = \Delta n_0 \text{sen}(k_{g1} x) \\ \Delta n_2(x) = \Delta n_0 \text{sen}(k_{g2} x) \end{cases} \quad (\text{A.2.4})$$

Então, estas duas redes, com duas diferentes frequências espaciais,  $k_{g1}$  e  $k_{g2}$ , são dinamicamente superpostas. De acordo com os argumentos discutidos anteriormente, Eq. (A.2.2), a modulação do índice de refração, no caso fotorrefrativo, agora resulta em

$$\Delta n(x) = \Delta n_1(x) + \Delta n_2(x) = \Delta n_o \left( \text{sen}(k_{g1}x) + \text{sen}(k_{g2}x) \right). \quad (\text{A.2.5})$$

A Eq.(A.2.5) pode ser reescrita da forma

$$\Delta n(x) = 2 \Delta n_o \left[ \text{sen} \left( \frac{k_{g1}x}{2} \right) \cos \left( \frac{k_{g1}x}{2} \right) + \text{sen} \left( \frac{k_{g2}x}{2} \right) \cos \left( \frac{k_{g2}x}{2} \right) \right], \quad (\text{A.2.6})$$

onde foi utilizada a relação trigonométrica  $\text{sen}(A+B) = \text{sen}(A)\cos(B) + \text{sen}(B)\cos(A)$ .

Com o auxílio de alguns artifícios matemáticos pode-se chegar a relação:

$$\Delta n(x) = 2 \Delta n_o \left[ \cos \left( \frac{1}{2}(k_{g1} - k_{g2})x \right) \text{sen} \left( \frac{1}{2}(k_{g2} + k_{g1})x \right) \right]. \quad (\text{A.2.7})$$

Chamando de  $\bar{k}_g = \frac{1}{2}(k_{g1} + k_{g2})$  a frequência espacial média e  $k_g^m = \frac{1}{2}(k_{g1} - k_{g2})$  a modulação

da frequência espacial, chega-se ao resultado final

$$\Delta n(x) = 2 \Delta n_o \cos(k_g^m x) \text{sen}(\bar{k}_g x). \quad (\text{A.2.8})$$

Um interessante caso ocorre quando as frequências espaciais,  $k_{g1}$  e  $k_{g2}$ , têm valores altos e, além disso, seus valores são comparáveis, isto é  $k_{g1} \sim k_{g2}$ ; Neste caso,  $\bar{k}_g \gg k_g^m$ , e o índice de refração resultante no volume da amostra será espacialmente modulado.

Isto significa que o padrão de franjas de moiré observado é produzido como um fenômeno de **batimento**, tratado exatamente como a interferência de duas ondas que se propagam com frequências próximas [19]. Mas neste caso, o fenômeno é devido a superposição dinâmica de duas redes de fase senoidais no volume da amostra do cristal BTO.

A modulação do índice de refração na Eq. (A.2.8) produz padrões de franjas de moiré dinâmicos através da eficiência de difração

$$\eta(x) = \eta \operatorname{sen}^2(k_g x), \quad (\text{A.2.9})$$

onde

$$\eta = 4 \eta_0 \cos^2(k_g^m x) \quad (\text{A.2.10})$$

é a amplitude da eficiência de difração modulada. A luz difratada do feixe de escrita produz padrões de moiré com amplitude de eficiência de difração expressa pela Eq.(A.2.9), e que pode ser reescrita como:

$$\eta = 2 \eta_0 [1 + \cos(2 k_g^m x)]. \quad (\text{A.2.9})$$

Está implícita uma dependência com  $2 k_g^m$ , ou simplesmente  $(k_{g1} - k_{g2})$ , que é a frequência espacial do batimento.

Concluimos assim, que o processo de formação de padrões dinâmicos de moiré devido à superposição de redes de fase senoidais com frequências espaciais levemente diferentes em um cristal fotorrefrativo BTO, aparecem devido à modulação no índice de refração fotorrefrativo resultante previsto pelo modelo físico desenvolvido.

### A.3 – MONTAGEM EXPERIMENTAL

A montagem experimental utilizada neste trabalho está esquematicamente representada na Fig.(A.3.1).

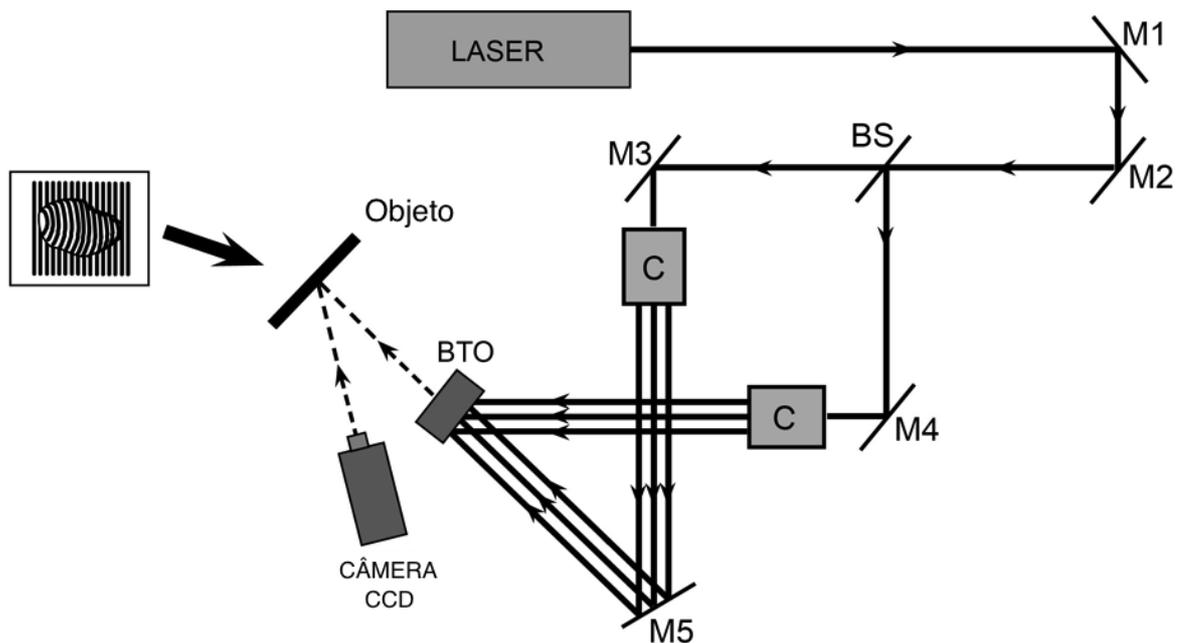


Fig.(A.3.1) Montagem experimental básica de mistura de duas ondas (2WM) utilizada neste trabalho.

O aparato experimental utilizou uma montagem básica de mistura de duas ondas (2WM) [85] com mecanismo de gravação apenas por difusão. O esquema mostra dois feixes oriundos de uma fonte de luz de laser de He-Ne ( $\lambda = 0,633 \mu m$ ) com potencia nominal de  $35 mW$ . BS é um divisor de feixes (“*beam-splitter*”) variável; os C’s são colimadores e M1, M2, M3, M4 e M5 são espelhos.

Os feixes de laser, um sendo o feixe “objeto” e o outro sendo o feixe “referência”, são projetados na face de entrada da amostra de um cristal  $Bi_{12}TiO_{20}$  (BTO) de dimensões  $8,0 \times 8,0 \times 8,0 mm^3$ . Esta amostra é de um cristal paraelétrico fotorrefrativo da família dos selenetos de simetria de grupo 23, apresentando uma forte atividade óptica natural, e índice de refração efetivo  $n = 2,58$ .

A Tabela A.3.1 lista alguns parâmetros físicos da amostra do cristal. O BTO é o único cristal com alta sensibilidade à linha vermelha do laser He-Ne em difusão apenas como mecanismo de gravação. Ambos os feixes são projetados na face de entrada do cristal sob um ângulo de Bragg de aproximadamente  $23^\circ$ . Isto significa, de acordo com a Lei de Bragg clássica, que uma rede de fase senoidal com frequência espacial da ordem de  $1000 \text{ linhas/mm}$  será produzida.

O procedimento experimental é realizado em uma sequência de etapas. Primeiramente, uma rede de fase senoidal em tempo real é gerada. Então, um obturador mecânico (não mostrado na figura) isola a amostra fotorrefrativa cobrindo ambos os feixes. Logo após, o espelho M5 é girado de um ângulo de aproximadamente  $1^\circ$ . O espelho M5 foi conectado a um suporte mecânico controlado por computador para fazer as rotações desejadas (calibrado por um experimento auxiliar). Deste modo, quando o obturador é aberto uma rede de fase senoidal adicional é produzida com uma pequena diferença na frequência espacial. Após a leitura dos dois feixes, o padrão de moiré é produzido e projetado sobre a superfície do objeto. O padrão de franjas de moiré dinâmico

observado possuía frequência espacial da ordem de  $2 - 5 \text{ linhas /mm}$  . Por fim, todo o processo de monitoramento e posteriormente de registro do padrão de franjas projetado sobre a superfície do objeto foi realizado por uma câmera CCD interfaceada a um computador IBM-PC.

**Tabela A.3.1**

Parâmetros físicos	Valores	Referência
Simetria de grupo	23	fabricante
Absorção óptica ( $\text{cm}^{-1}$ )	0.4 - 0.6 <sup>b</sup>	fabricante
Índice de refração	2.58 <sup>b</sup>	fabricante
Voltagem de meia ( kV)	3.2-3.3 <sup>b</sup>	fabricante
Permeabilidade dielétrica estática	47	fabricante
Atividade óptica ( $\text{graus mm}^{-1}$ )	6.32 <sup>b</sup>	[51]
Comprimento de transporte (difusão) $L_D$ ( $\mu\text{m}$ )	0.23 <sup>a,b</sup>	[52]
Comprimento de transporte (drift) $L_E/E$ [ $\mu\text{m}/(\text{kV/cm})$ ]	0.22 <sup>a,b</sup>	[52]
Eficiência quântica $\phi$	0.41 <sup>a,b</sup>	[52]
Coefficiente electroóptico ( $r_{41} = r_{52} = r_{63}$ ) $\text{cmV}^{-1}$	$5.17 \times 10^{-10}$ <sup>b</sup>	[53]

<sup>a</sup> Para rede principal de elétrons,

<sup>b</sup> Para o comprimento de onda  $0.633 \mu\text{m}$

Tabela A.3.1 - Alguns parâmetros físicos do cristal  $\text{Bi}_{12}\text{TiO}_{20}$ .

# APÊNDICE B

## B.1 – CÓDIGO FONTE DO ALGORITMO DESENVOLVIDO.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% DESEMPACOTAMENTO POR SALTOS DE FASE DESLOCADOS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Programa para desempacotamento bidimensional de fase por balanceamento
% de resíduos que utiliza mapas de salto de fase deslocados como parâmetro
% para a geração de máscara de conexão entre os resíduos.
%
%
% Este algoritmo foi desenvolvido por Marcos Eduardo de Oliveira e Júlio
% Cesar Dias de Souza, sob a orientação do professor Paulo Acioly Marques
% dos Santos da Universidade Federal Fluminense (UFF) e descrito na
% dissertação de mestrado "Desempacotamento de fase aplicado à
% perfilometria óptica" de Júlio Cesar Dias de Souza sob a orientação do
% professor Paulo Acioly Marques dos Santos.
%
%
% Entrada: 1. Imagem real no formato .mat, do tipo double, contendo o
% mapa de fases empacotado.
%
% Saídas: 1. Máscara de localização e conexão entre os resíduos.
%         2. Mapa de fases desempacotado.
%
% Criado em 01/10/2014.
% Última modificação em 05/01/2015.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear;                                %Limpeza das variáveis do programa.
load map.mat;                          %Leitura da imagem real do tipo double no
                                        %formato .mat, contendo o mapa de fases
                                        %empacotado.

par=0.5;                                %Determinação da margem de erro ao avaliar
                                        %se a região contém saltos de fase com
                                        %módulo igual a 2*pi.

s=size(map);                            %Leitura das dimensões do mapa de fases
                                        %empacotado.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Geração dos mapas de fase empacotados alfa2, alfa3 e alfa4.
% As regiões de salto de fase desses mapas são deslocadas em relação ao
% mapa empacotado inicial (map) respectivamente em pi/2, pi e 3*(pi/2).
% Após a geração dos mapas empacotados, suas regiões com salto de fase
% são analisadas para a geração de quatro máscaras binárias: im1, im2, im3
% e im4, que demarcam as regiões com salto de fase nos mapas empacotados
% map, alfa2, alfa3 e alfa4 respectivamente.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Adição de fase constante pi/2 a todos os pontos do mapa
%empacotado e realização de novo empacotamento para a geração do
%segundo mapa de fases deslocadas.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    alfa2=(pi/2)+map;
    alfa2(alfa2<-pi)= alfa2(alfa2<-pi)+2*pi;
    alfa2(alfa2>pi)= alfa2(alfa2>pi)-2*pi;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Adição de fase constante pi a todos os pontos do mapa
%empacotado e realização de novo empacotamento para a geração do
%segundo mapa de fases deslocadas.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    alfa3=(pi)+map;
    alfa3(alfa3<-pi)= alfa3(alfa3<-pi)+2*pi;
    alfa3(alfa3>pi)= alfa3(alfa3>pi)-2*pi;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Adição de fase constante 3*pi/2 a todos os pontos do mapa
%empacotado e realização de novo empacotamento para a geração do
%segundo mapa de fases deslocadas.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    alfa4=((3)*pi/2)+map;
    alfa4(alfa4<-pi)= alfa4(alfa4<-pi)+2*pi;
    alfa4(alfa4>pi)= alfa4(alfa4>pi)-2*pi;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Cálculo da diferença dos valores de fase entre os píxeis
%vizinhos nos 4 mapas de fase empacotados.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    m=(1:s(1,1));
    n=(1:s(1,2)-1);
    difeh1(m,n)=abs(map(m,n+1)-map(m,n));
    difeh2(m,n)=abs(alfa2(m,n+1)-alfa2(m,n));
    difeh3(m,n)=abs(alfa3(m,n+1)-alfa3(m,n));
    difeh4(m,n)=abs(alfa4(m,n+1)-alfa4(m,n));
    difeh1(1:end,s(1,2))=0;
    difeh2(1:end,s(1,2))=0;
    difeh3(1:end,s(1,2))=0;
    difeh4(1:end,s(1,2))=0;

    o=(1:s(1,1)-1);
    p=(1:s(1,2));

    difev1(o,p)=abs(map(o+1,p)-map(o,p));
    difev2(o,p)=abs(alfa2(o+1,p)-alfa2(o,p));
    difev3(o,p)=abs(alfa3(o+1,p)-alfa3(o,p));
    difev4(o,p)=abs(alfa4(o+1,p)-alfa4(o,p));
    difev1(s(1,1),1:end)=0;
    difev2(s(1,1),1:end)=0;
    difev3(s(1,1),1:end)=0;
    difev4(s(1,1),1:end)=0;

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Geração das máscaras im1, im2, im3 e im4 demarcando os pontos
% com salto de fase nos mapas map, alfa2, alfa3 e alfa4
% respectivamente.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

im1=double((difeh1+difev1)>(2*pi*(1-par)));
im2=double((difeh2+difev2)>(2*pi*(1-par)));
im3=double((difeh3+difev3)>(2*pi*(1-par)));
im4=double((difeh4+difev4)>(2*pi*(1-par)));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Localização e marcação dos resíduos.
%Como a localização é a mesma para os quatro mapas de fases, é necessário
% analisar apenas o mapa inicial (map).
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

i=(1:s(1,1)-1); j=(1:s(1,2)-1);

dif1(i,j)=map(i,j+1)-map(i,j);          dif2(i,j)= map(i,j)-map(i+1,j);
dif3(i,j)=map(i+1,j+1)-map(i,j+1);      dif4(i,j)=map(i+1,j)-
map(i+1,j+1);

dif1(dif1>pi)=dif1(dif1>pi)-2*pi;        dif2(dif2>pi)=dif2(dif2>pi)-
2*pi;                                     dif2(dif2<-pi)=dif2(dif2<-
pi)+2*pi;

dif3(dif3>pi)=dif3(dif3>pi)-2*pi;        dif4(dif4>pi)=dif4(dif4>pi)-
2*pi;                                     dif4(dif4<-pi)=dif4(dif4<-
pi)+2*pi;

%Matriz contendo a localização dos resíduos.
dif=dif1+dif2+dif3+dif4;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Os resíduos encontrados, quanto positivos, têm sua localização
% marcada, são nomeados com valores entre 10 e 100 e armazenados
% na matriz "qua" e quando negativos, têm sua localização marcada,
% são nomeados com valores maiores do que 100 e armazenados
% na matriz "qual".
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

cont=10;                                %valor inicial da marcação dos resíduos
%positivos.
cont1=100;                               %valor inicial da marcação dos resíduos
%negativos.
qua=zeros(s);                             %matriz para armazenar a localização
%dos resíduos positivos.
qual=zeros(s);                             %matriz para armazenar a localização
%dos resíduos negativos.

%Para cada ponto da mascara "dif" que contem a localização dos
%resíduos.
for i=(1:s(1,1)-1);

```

```

for j=(1:s(1,2)-1);
    if dif(i,j)>=pi % Se o ponto contém um resíduo
                    %positivo.
        qua(i,j)=cont; %O resíduo é marcado com um valor
                    %entre 10 e 100 e armazenado
                    %na matriz qua.
        cont=cont+1;
    elseif dif(i,j)<=-pi %Se o ponto contém um
                        %resíduo negativo.
        qual(i,j)=cont1; %O resíduo é marcado
                        %com um valor maior que
                        %100 e armazenado na
                        %matriz qual.
        cont1=cont1+1;
    end;
end;
end;
end;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Geração de 4 máscaras: ima1, ima2, ima3 e ima4, contendo
% respectivamente os saltos de fase das máscaras im1, im2, im3 e
% im4, sobrepostos à marcação da localização dos resíduos
% positivos e negativos mapeados nas matrizes qua e qual
% respectivamente.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

ima1=im1; ima1(qua~=0)=qua(qua~=0);
ima2=im2; ima2(qua~=0)=qua(qua~=0);
ima3=im3; ima3(qua~=0)=qua(qua~=0);
ima4=im4; ima4(qua~=0)=qua(qua~=0);

respos=bwmorph(qua,'dilate',4); %Máscara indicando a
                                %localização dos resíduos
                                %positivos.

resnegt=bwmorph(qual,'dilate',4);
resnegt=double(resnegt);
resnegt(resnegt==1)=-1; %Máscara indicando a
                        %localização dos resíduos
                        %negativos.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Neste ponto, tendo mapeado a localização dos resíduos e as regiões de
% salto de fase para os quatro mapas de fase gerados e nomeados os resíduos
% de forma a poder diferenciá-los, é verificado se entre dois resíduos há
% uma linha de salto de fase. Caso positivo, esta linha é marcada como
% uma barreira a ser comparada com outras barreiras para a minimização
% das conexões entre os resíduos.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

voltas=28; %Número de vezes em que a imagem será girada para
            %verificação das linhas de conexão entre os resíduos.

ft=2; %fator de comparação para avaliar se um resíduo será
      %conectado a outro ou à borda da imagem.

s1=size(ima1); %Variável para armazenar o tamanho da matriz ima1.

```

```

[x,y]=size(map); %Obtenção do tamanho da matriz map.

wind=1; %Espessura em pixels da borda a ser adicionada nas matrizes
        %zero1, zero2, zero3 e zero4.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        %Centralização das matrizes ima1, ima2, ima3 e ima4 nas matrizes
        %com borda zero.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

        zero1=zeros(s1+2*wind); zero1(wind+1:end-wind,wind+1:end-
wind)=ima1;
        zero2=zeros(s1+2*wind); zero2(wind+1:end-wind,wind+1:end-
wind)=ima2;
        zero3=zeros(s1+2*wind); zero3(wind+1:end-wind,wind+1:end-
wind)=ima3;
        zero4=zeros(s1+2*wind); zero4(wind+1:end-wind,wind+1:end-
wind)=ima4;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        % Neste ponto as linhas de conexão entre os resíduos são varridas
        %para para mapear sua extensão e localização.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

        for l=(1:voltas)          %Para l variando de 1 a "voltas".
            s2=size(zero1);      %Obtem o tamanho da matriz zero1.
            for i=(wind+1:s2(1,1)-wind) %Para cada linha, iniciando na
                %segunda e encerrando na penúltima
                    %linha.
                        for j=(wind+1:s2(1,2)-wind) %Para cada coluna, iniciando
                            %na segunda e terminando na
                                %penúltima.

                                    if zero1(i,j)>9 % Se o ponto da matriz contém um
                                        %resíduo.
                                            k=zero1(i,j);% Armazena o valor do resíduo.
                                            janela=zero1(i-wind:i+wind,j-wind:j+wind); % Cria
%uma janela de tamanho (1+2*wind x 1.2*wind)
                                            janela(janela==1)=k; % Atribui o valor do resíduo
%aos pixels da janela diferentes de zero.
                                            zero1(i-wind:i+wind,j-wind:j+wind)=janela;
%Atribui os valores dos pixels da janela à matriz zero1 original de
forma a demarcar a linha conectada ao resíduo, com o valor em que o
resíduo foi nomeado nas matrizes ima1, ima2, ima3 e ima4.
                                        end;

                                            %No loop abaixo, repete o mesmo procedimento efetuado
%no loop anterior.
                                            if zero2(i,j)>9
                                                k=zero2(i,j);
                                                janela=zero2(i-wind:i+wind,j-wind:j+wind);
                                                janela(janela==1)=k;
                                                zero2(i-wind:i+wind,j-wind:j+wind)=janela;
                                            end;

                                            %No loop abaixo, repete o mesmo procedimento efetuado
%no loop anterior.
                                            if zero3(i,j)>9

```

```

        k=zero3(i,j);
        janela=zero3(i-wind:i+wind,j-wind:j+wind);
        janela(janela==1)=k;
        zero3(i-wind:i+wind,j-wind:j+wind)=janela;
    end;

    %No loop abaixo, repete o mesmo procedimento efetuado
    %no loop anterior.
    if zero4(i,j)>9
        k=zero4(i,j);
        janela=zero4(i-wind:i+wind,j-wind:j+wind);
        janela(janela==1)=k;
        zero4(i-wind:i+wind,j-wind:j+wind)=janela;
    end;
end;
end;
end;

%Gira cada uma das matrizes em 90 graus para repetir o
%procedimento.
%até que todas as linhas que conectam os resíduos tenham sido
%marcadas.

zero1=imrotate(zero1,90);
zero2=imrotate(zero2,90);
zero3=imrotate(zero3,90);
zero4=imrotate(zero4,90);
end;

%Cria 4 matrizes de zeros com tamanho igual à matriz zero1.
imaneg1=zeros(s);
imaneg2=zeros(s);
imaneg3=zeros(s);
imaneg4=zeros(s);

%Armazena os valores das matrizes zero1, zero2, zero3 e zero4 nas
%matrizes ima1, ima2, ima3 e ima4 respectivamente.
ima1=zero1(wind+1:end-wind,wind+1:end-wind);
ima2=zero2(wind+1:end-wind,wind+1:end-wind);
ima3=zero3(wind+1:end-wind,wind+1:end-wind);
ima4=zero4(wind+1:end-wind,wind+1:end-wind);

mascara=zeros(s);

%Analisa as linhas de conexão entre os resíduos e calcula o tamanho
%de cada uma para escolher as conexões de menor comprimento.

for i=(10:max(max(qua))) %Para cada valor armazenado na matriz
                        %qua, verifica nas quatro matrizes ima, o
                        %tamanho das conexões entre os resíduos.

    N1=find(ima1==i); %Armazena a linha i de salto de fase da
                    %imagem ima1.
    N2=find(ima2==i); %Armazena a linha i de salto de fase da
                    %imagem ima2.
    N3=find(ima3==i); %Armazena a linha i de salto de fase da
                    %imagem ima3.
    N4=find(ima4==i); %Armazena a linha i de salto de fase da
                    %imagem ima4.

```

```

l1=length(N1); %Verifica o tamanho da linha i da imagem
                %ima1.
l2=length(N2); %Verifica o tamanho da linha i da imagem
                %ima2.
l3=length(N3); %Verifica o tamanho da linha i da imagem
                %ima3.
l4=length(N4); %Verifica o tamanho da linha i da imagem
                %ima4.

l=[l1 l2 l3 l4];

A1=find(l==min(l)); %Seleciona a barreira com menor
                   %comprimento.
A1=A1(1);
matcomp{i}=l(A1);

if l(A1)<(3.*s(1,2)/4) %Caso o comprimento da barreira
                    %selecionada seja inferior a 3/4 da
                    %largura da imagem, esta é
                    %marcada como barreira válida e
                    %adicionada à mascara.

    if A1==1
        mascara(ima1==i)=i;
    elseif A1==2
        mascara(ima2==i)=i;
    elseif A1==3
        mascara(ima3==i)=i;
    elseif A1==4
        mascara(ima4==i)=i;
    end;
end;

end;

mask=bwmorph(mascara,'dilate',4);
resneg=qual;
resneg(mask==1)=0; % Ao adicionar a barreira à matriz mascara, esta é
                  %excluída da matriz resneg.

imaneg1(resneg~=0)=resneg(resneg~=0);
imaneg2(resneg~=0)=resneg(resneg~=0);
imaneg3(resneg~=0)=resneg(resneg~=0);
imaneg4(resneg~=0)=resneg(resneg~=0);

% Verifica se há algum resíduo negativo remanescente, que não tenha
% sido conectado a um resíduo positivo.
for i=(3:s(1,1)-2)
    for j=(3:s(1,2)-2)
        if imaneg1(i,j)>9 % Se há um resíduo negativo remanescente no
                        %ponto verificado.
            janela=imaneg1(i-2:i+2,j-2:j+2); %Os dois pontos acima,
                                                %abaixo, à direita e
                                                %à esquerda são
                                                %armazenados para
                                                %verificação.

            if (sum(sum(janela))-imaneg1(i,j))==0 %se não houver
            %nenhuma barreira conectada ao resíduo, este é excluído da matriz
            %imaneg1.

```

```

        imaneg1(i,j)=0;
        end;
    end;

    % Os passos da verificação anterior são repetidos para a matriz
    %imaneg2.
    if imaneg2(i,j)>9
        janela=imaneg2(i-2:i+2,j-2:j+2);
        if (sum(sum(janela))-imaneg2(i,j))==0
            imaneg2(i,j)=0;
        end;
    end;

    % Os passos da verificação anterior são repetidos para a matriz
    %imaneg3.
    if imaneg3(i,j)>9
        janela=imaneg3(i-2:i+2,j-2:j+2);
        if (sum(sum(janela))-imaneg3(i,j))==0
            imaneg3(i,j)=0;
        end;
    end;

    % Os passos da verificação anterior são repetidos para a matriz
    %imaneg4.
    if imaneg4(i,j)>9
        janela=imaneg4(i-2:i+2,j-2:j+2);
        if (sum(sum(janela))-imaneg4(i,j))==0
            imaneg4(i,j)=0;
        end;
    end;
end;
end;

zero1=zeros(s1+2*wind); zero1(wind+1:end-wind,wind+1:end-wind)=imaneg1;
zero2=zeros(s1+2*wind); zero2(wind+1:end-wind,wind+1:end-wind)=imaneg2;
zero3=zeros(s1+2*wind); zero3(wind+1:end-wind,wind+1:end-wind)=imaneg3;
zero4=zeros(s1+2*wind); zero4(wind+1:end-wind,wind+1:end-wind)=imaneg4;

for i=(100:max(max(qual))) % Para cada valor armazenad na matriz qual,
    %verifica nas quatro matrizes ima o tamanho
    %das conexões entre os resíduos.

    N1=find(imaneg1==i); % Armazena a linha i de salto de fase da
    %imagem ima1.
    N2=find(imaneg2==i); % Armazena a linha i de salto de fase da
    %imagem ima2.
    N3=find(imaneg3==i); % Armazena a linha i de salto de fase da
    %imagem ima3.
    N4=find(imaneg4==i); % Armazena a linha i de salto de fase da
    %imagem ima4.

    l1=length(N1); % Verifica o tamanho da linha i da imagem
    %ima1.
    l2=length(N2); % Verifica o tamanho da linha i da imagem
    %ima2.
    l3=length(N3); % Verifica o tamanho da linha i da imagem
    %ima3.
    l4=length(N4); % Verifica o tamanho da linha i da imagem
    %ima4.

```

```

l=[l1 l2 l3 l4]; l(l==0)=100000;

A1=find(l==min(l)); % Seleciona a barreira com menor
                    %comprimento entre as três matrizes
                    %imaneg1, imaneg2, imaneg3 e imaneg4.
A1=A1(1);

if l(A1)<3*(s(1,2)/4) %Caso o comprimento da barreira
                    %selecionada seja inferior a 3/4 da
                    %largura da imagem, esta é
                    %marcada como barreira válida e
                    %adicionada à máscara.

    if A1==1
        mascara(imaneg1==i)=i;
    elseif A1==2
        mascara(imaneg2==i)=i;
    elseif A1==3
        mascara(imaneg3==i)=i;
    elseif A1==4
        mascara(imaneg4==i)=i;
    end;
end;

end;

qual(qual~=0)=1;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Verifica se há alguma linha conectando mais de um resíduo positivo ou
%negativo e desfaz a conexão.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for i=(3:s(1,1)-2)
    for j=(3:s(1,2)-2)

        if qual(i,j)==1 %Se há algum resíduo
                        %negativo no ponto
                        %verificado.

            janela=mascara(i-2:i+2,j-2:j+2); %Cria uma janela em torno
                                                %desse ponto.

            %Verifica quais valores diferentes constam na janela.
            janel=unique(janela); janel(janel==0)=[];

            comp=length(janel); %Conta quantos valores diferentes foram
                                %encontrados na janela.

            if comp>1 %Se mais de um valor foi encontrado, ou
                    %seja, se mais de uma linha
                    %está conectada a um mesmo resíduo.

                clear A2
                for z=1:comp % Verifica quantas linhas foram
                            %encontradas
                    A2(z)=matcomp{janel(z)};
                end
                A3=find(A2==min(A2)); % Seleciona a linha que contém o
                                        %menor valor.
                janel(janel==janel(A3))=[];
            end
        end
    end
end

```

```

        for Y=1:length(janel) %Elimina a linha selecionada da
                               %matriz mascara.
            mascara(mascara==janel(Y))=0;
        end;
    end;
end;
end;
end;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Remoção das conexões de resíduos ligados às bordas do mapa empacotado.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for i=1:s(1,1) % Caso haja alguma barreira conectada à borda esquerda,
                %esta é eliminada.
    if mascara(i,1)~=0;
        mascara(mascara==mascara(i,1))=0;
    end;
end;

for i=1:s(1,1) % Caso haja alguma barreira conectada à borda direita,
                %esta é eliminada.
    if mascara(i,s(1,2))~=0;
        mascara(mascara==mascara(i,end))=0;
    end;
end;

for j=1:s(1,2) % Caso haja alguma barreira conectada à borda superior,
                %esta é eliminada.
    if mascara(1,j)~=0;
        mascara(mascara==mascara(1,j))=0;
    end;
end;

for j=1:s(1,2) % Caso haja alguma barreira conectada à borda inferior,
                %esta é eliminada.
    if mascara(s(1,1),j)~=0;
        mascara(mascara==mascara(end,j))=0;
    end;
end;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Caso algum resíduo positivo não tenha sido conectado a outro negativo,
%os remanescentes são conectados entre si, desde que seja um resíduo
%positivo conectado a um negativo.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

mask=bwmorph(mascara,'dilate',4);
resneg=qua+qual; %qua = resíduos positivos; qual = resíduos negativos.
QUA=(qua~=0);
QUA1=-(qual~=0);
resneg=QUA+QUA1;
resneg(mask==1)=0; %resneg contém a posição de todos os resíduos
                    %remanescentes.

[lins, cols]=find(resneg~=0);

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Armazena em uma matriz na matriz posicoes, a localização de todos os
% resíduos que não tenham sido conectados a outro de carga oposta.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

posicoes=zeros(length(lins),2);
posicoes(:,1)=lins;
posicoes(:,2)=cols;
[pl, pc]=size(posicoes);
clear lins cols;
RETA=zeros(size(mascara));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Nos passos abaixo, para cada resíduo positivo remanescente encontrado,
%calcula-se a distância deste a todos os outros resíduos negativos
%remanescentes. O resíduo negativo que estiver mais perto, é conectado
%por uma linha reta ao resíduo positivo.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

menor_dist=zeros(5,3);
while pl>1; %Enquanto ainda sobraem elementos na matriz de resíduos
    %remanescentes.

    %Calcula a distância do resíduo remanescente às bordas da imagem.
    dist_borda_esq=posicoes(1,2);
    dist_borda_dir=s(1,2)-posicoes(1,2);
    dist_borda_sup=posicoes(1,1);
    dist_borda_inf=s(1,1)-posicoes(1,1);

    menor_dist(1,1)=ft*dist_borda_esq;
    menor_dist(1,2)=posicoes(1,1);
    menor_dist(1,3)=1;

    menor_dist(2,1)=ft*dist_borda_dir;
    menor_dist(2,2)=posicoes(1,1);
    menor_dist(2,3)=s(1,2);

    menor_dist(3,1)=ft*dist_borda_sup;
    menor_dist(3,2)=1;
    menor_dist(3,3)=posicoes(1,2);

    menor_dist(4,1)=ft*dist_borda_inf;
    menor_dist(4,2)=s(1,1);
    menor_dist(4,3)=posicoes(1,2);

    dist_res=zeros(pl-1,3);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Calcula a distância do resíduo remanescente a todos os outros
    %resíduos remanescentes de carga oposta e armazena na matriz
    %dist_res.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for I=1:pl-1
    dist_res(I,1)=sqrt((posicoes(1,1)-
posicoes(I+1,1))^2+(posicoes(1,2)-posicoes(I+1,2))^2);
    dist_res(I,2)=posicoes(I+1,1);

```

```

        dist_res(I,3)=posicoes(I+1,2);
        if
resneg(posicoes(1,1),posicoes(1,2))==resneg(posicoes(I+1,1),posicoes(I+
1,2))
            dist_res(I,1)=dist_res(I,1)*1000; %Descarta a distância
                                                %entre resíduos de mesmo
                                                %valor.
        end;
    end;

% Armazena na matriz menor_dist, a menor distância da matriz
%dist_res e a localização do ponto.
menor_dist(5,1)=min(dist_res(:,1));
f=find(dist_res(:,1)==min(dist_res(:,1)));
menor_dist(5,2)=dist_res(f,2);
menor_dist(5,3)=dist_res(f,3);

%Verifica qual o menor valor de distância da matriz menor_dist e armazena
%a posição do ponto (de resíduo ou borda) na variável menor_pos.
menor_pos=find(menor_dist(:,1)==min(menor_dist(:,1)));

%Variável que armazena a coordenada da linha do resíduo
%remanescente.
yi=posicoes(1,2);

%Variável que armazena a coordenada da coluna do resíduo
%remanescente.
xi=posicoes(1,1);

%Variável que armazena a coordenada da linha do resíduo ou borda
%mais próxima a ser conectada.
yf=menor_dist(menor_pos,3);

%Variável que armazena a coordenada da coluna do resíduo ou borda a
%ser conectada.
xf=menor_dist(menor_pos,2);

%Traça a reta entre os pontos iniciais e finais.
if xi==xf
    b=xi;
    A=0;
else
    b=(-yi*xf+yf*xi)/(xi-xf); % Coeficiente linear da reta que
                                %liga os pontos.
    A=((yi-yf)/(xi-xf));      %Coeficiente angular da reta.
end;

deltax=abs(xf-xi); %Variação em x, dos pontos da curva.
deltay=abs(yf-yi); %Variação em y, dos pontos da curva.

if deltax >= deltax % Se a variacao em y for maior do que a variacão em
                    %x.

    if yi>yf
        yt=yf;
        yf=yi;
    end;
end;

```

```

        yi=yt;
    end;

    for i=1:x
        for j=yi:yf;
            if A==0; linha=b; else linha=fix((j-b)/A); end;
            if i==linha
                RETA(i,j)=1;
            end;
        end;
    end;
end;

else %Se a variacao em x for maior do que a variacao em y.

    if xi>xf
        xt=xf;
        xf=xi;
        xi=xt;
    end;

    for i=xi:xf
        for j=1:y
            linha=fix((A.*i)+b);
            if j==linha
                RETA(i,j)=1;
            end;
        end;
    end;
end;

end;

    posicoes(1,:)=[];

    if menor_pos==5 %Se a menor distância for entre dois resíduos,
                    % armazeno na variável
                    % "extremos" e deleteo os dois resíduos da matriz
                    %inicial.
        posicoes(f,:)=[];
    end;
    clear f menor_pos dist_res menor_dist; %remove alguma variáveis
                                           %não mais utilizadas pelo
                                           %algoritmo.

    [pl, pc]=size(posicoes);

end;

%Quando houver apenas um resíduo remanescente, verifica a distância do
%ponto às bordas da imagem para conectar, através de uma linha reta,
%aquela que estiver mais próxima.
if pl==1
    menor_dist=zeros(4,3);
    dist_borda_esq=posicoes(1,2);
    dist_borda_dir=s(1,2)-posicoes(1,2);
    dist_borda_sup=posicoes(1,1);
    dist_borda_inf=s(1,1)-posicoes(1,1);

    menor_dist(1,1)=ft*dist_borda_esq;
    menor_dist(1,2)=posicoes(1,1);

```

```

menor_dist(1,3)=1;
menor_dist(2,1)=ft*dist_borda_dir;
menor_dist(2,2)=posicoes(1,1);
menor_dist(2,3)=s(1,2);

menor_dist(3,1)=ft*dist_borda_sup;
menor_dist(3,2)=1;
menor_dist(3,3)=posicoes(1,2);

menor_dist(4,1)=ft*dist_borda_inf;
menor_dist(4,2)=s(1,1);
menor_dist(4,3)=posicoes(1,2);

menor_pos=find(menor_dist(:,1)==min(menor_dist(:,1)));

yi=posicoes(1,2);
xi=posicoes(1,1);
yf=menor_dist(menor_pos,3);
xf=menor_dist(menor_pos,2);

%Traça a reta entre os pontos iniciais e finais.
if xi==xf
    b=xi;
    A=0;
else
    b=(-yi*xf+yf*xi)/(xi-xf); %Coeficiente linear da reta que
    %liga os pontos.
    A=((yi-yf)/(xi-xf)); %Coeficiente angular da reta.
end;

deltax=abs(xf-xi); %Variação em x, dos pontos da curva.
deltay=abs(yf-yi); %Variação em y, dos pontos da curva.

if deltax >= deltax % Se a variação em y for maior do que a
    %variação em x.

    if yi>yf
        yt=yf;
        yf=yi;
        yi=yt;
    end;

    for i=1:x
        for j=yi:yf;
            if A==0; linha=b; else linha=fix((j-b)/A); end;
            if i==linha
                RETA(i,j)=1;
            end;
        end;
    end;

else %Se a variação em x for maior do que a variação em y.

    if xi>xf
        xt=xf;
        xf=xi;
        xi=xt;
    end;

```

```

        for i=xi:xf
            for j=1:y
                linha=fix((A.*i)+b);
                if j==linha
                    RETA(i,j)=1;
                end;
            end;
        end;
    end;
end;

%Finalizando a geração da máscara com as barreiras a serem contornadas
%no processo de desempacotamento.

mascara=mascara+RETA;
mascara2=zeros(size(mascara));
resnegt2=zeros(size(mascara));
respos2=zeros(size(mascara));
mascara2(mascara~=0)=1;
resnegt2(resnegt~=0)=1;
respos2(respos~=0)=1;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Primeira saída: Máscara de localização e conexão entre os resíduos
% a ser utilizada pelo algoritmo para o desempacotamento.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

imtool(mascara2+resnegt2+respos2);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Tendo gerado a máscara, nos passos abaixo, é efetuado o
%desempacotamento, contornando as barreiras de conexão entre os
%resíduos positivos e negativos.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Determinação das regiões a serem contornadas no processo de
%desempacotamento.
s=size(mascara);
mask=zeros(s);
mask(mascara~=0)=1;
temp1=zeros(s);
temp2=zeros(s);
gam=zeros(s);

%Localiza uma linha e uma coluna que não tenham saltos de fase para
%iniciar o desempacotamento.
for i=(1:s(1,2))
    if sum(mask(:,i))==0
        temp1(:,i)=1;
        break;
    end;
end;

for i=(1:s(1,1))
    if sum(mask(i,:))==0
        temp2(i,:)=1;
        break;
    end;
end;

```

```

end;

%Localiza o ponto de interseção entre a linha e a coluna sem salto de
%fase.
temp=temp1+temp2;
[lin,col] = find(temp==2);

% Efetua o desempacotamento da linha e coluna selecionadas, a partir do
%ponto de interseção entre elas.
gam(lin:s(1,1),col)=unwrap(map(lin:s(1,1),col));
gam(lin:-1:1,col)=unwrap(map(lin:-1:1,col));
gam(lin,col:s(1,2))=unwrap(map(lin,col:s(1,2)));
gam(lin,col:-1:1)=unwrap(map(lin,col:-1:1));

%Adiciona os valores desempacotados à matriz mope.
temp=gam;
mask(temp~=0)=temp(temp~=0);
mope=mask;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Efetua o desempacotamento dos demais pontos a partir da linha e coluna
%inicialmente desempacotados, excluindo do processo as barreiras de
%conexão entre os resíduos, marcadas na matriz mascara.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%A partir do ponto de interseção entre a linha e coluna sem salto de
%fase marcado nos passos anteriores, até a última linha e coluna do
%mapa empacotado.
for i=(lin+1:s(1,1))
    for j=(col+1:s(1,2))
        if mope(i,j)==0;

            %Se o ponto pode ser desempacotado, calcula a diferença do
            %ponto atual para o ponto da coluna à esquerda
            %nas matrizes map, alfa1, alfa2 e alfa3, que contém os
            %mapas de fase empacotados deslocados.
            if mope(i,j-1)~=0 && mope(i,j-1)~=1

                dif1=(map(i,j)-map(i,j-1));
                dif2=(alfa2(i,j)-alfa2(i,j-1));
                dif3=(alfa3(i,j)-alfa3(i,j-1));
                dif4=(alfa4(i,j)-alfa4(i,j-1));
                dif=[abs(dif1) abs(dif2) abs(dif3) abs(dif4)];

                B=find(dif==min(dif)); %Seleciona a menor diferença
                %calculada sobre os mapas
                %empacotados.
                A=B(1);

            %Efetua o desempacotamento do ponto, a partir do mapa
            %cuja menor diferença tenha sido calculada.
            if A==1
                mope(i,j)=mope(i,j-1)+dif1;
            elseif A==2
                mope(i,j)=mope(i,j-1)+dif2;
            elseif A==3
                mope(i,j)=mope(i,j-1)+dif3;
            elseif A==4
                mope(i,j)=mope(i,j-1)+dif4;
            end;
        end;
    end;
end;

```

```

%Se o ponto pode ser desempacotado, calcula a diferença
%do ponto atual para o ponto da linha acima nas
%matrizes map, alfa1, alfa2 e alfa3, que contém os
%mapas de fase empacotados deslocados.
elseif mope(i-1,j)~=0 && mope(i-1,j)~=1

dif1=(map(i,j)-map(i-1,j));
dif2=(alfa2(i,j)-alfa2(i-1,j));
dif3=(alfa3(i,j)-alfa3(i-1,j));
dif4=(alfa4(i,j)-alfa4(i-1,j));
dif=[abs(dif1) abs(dif2) abs(dif3) abs(dif4)];

%Seleciona a menor diferença calculada sobre os mapas
%empacotados.
B=find(dif==min(dif));
A=B(1);

%Efetua o desempacotamento do ponto, a partir do mapa
%cuja menor diferença tenha sido calculada.
if A==1
    mope(i,j)=mope(i-1,j)+dif1;
elseif A==2
    mope(i,j)=mope(i-1,j)+dif2;
elseif A==3
    mope(i,j)=mope(i-1,j)+dif3;
elseif A==4
    mope(i,j)=mope(i-1,j)+dif4;
end;
end;
end;
end;
end;

%A partir do ponto de interseção entre a linha e coluna sem salto de
%fase marcado nos passos anteriores, até a primeira linha e última
%coluna do mapa empacotado.
for i=(lin-1:-1:1)
    for j=(col+1:s(1,2))
        if mope(i,j)==0;

            %Se o ponto pode ser desempacotado, calcula a diferença
            %do ponto atual para o ponto da coluna à esquerda
            %nas matrizes map, alfa1, alfa2 e alfa3, que contém os
            %mapas de fase empacotados deslocados.
            if mope(i,j-1)~=0 && mope(i,j-1)~=1

                dif1=(map(i,j)-map(i,j-1));
                dif2=(alfa2(i,j)-alfa2(i,j-1));
                dif3=(alfa3(i,j)-alfa3(i,j-1));
                dif4=(alfa4(i,j)-alfa4(i,j-1));
                dif=[abs(dif1) abs(dif2) abs(dif3) abs(dif4)];

                B=find(dif==min(dif)); %Seleciona a menor diferença
                %calculada sobre os mapas
                %empacotados.
                A=B(1);
            end;
        end;
    end;
end;

```

```

%Efetua o desempacotamento do ponto, a partir do mapa
%cuja menor diferença tenha sido calculada.
if A==1
    mope(i,j)=mope(i,j-1)+dif1;
elseif A==2
    mope(i,j)=mope(i,j-1)+dif2;
elseif A==3
    mope(i,j)=mope(i,j-1)+dif3;
elseif A==4
    mope(i,j)=mope(i,j-1)+dif4;
end;

%Se o ponto pode ser desempacotado, calcula a diferença
%do ponto atual para o ponto da linha abaixo nas
%matrizes map, alfa1, alfa2 e alfa3, que contém os
%mapas de fase empacotados deslocados.
elseif mope(i+1,j)~=0 && mope(i+1,j)~=1

dif1=(map(i,j)-map(i+1,j));
dif2=(alfa2(i,j)-alfa2(i+1,j));
dif3=(alfa3(i,j)-alfa3(i+1,j));
dif4=(alfa4(i,j)-alfa4(i+1,j));
dif=[abs(dif1) abs(dif2) abs(dif3) abs(dif4)];

%Seleciona a menor diferença calculada sobre
%os mapas empacotados.
B=find(dif==min(dif));
A=B(1);

%Efetua o desempacotamento do ponto, a partir do mapa
%cuja menor diferença tenha sido calculada.
if A==1
    mope(i,j)=mope(i+1,j)+dif1;
elseif A==2
    mope(i,j)=mope(i+1,j)+dif2;
elseif A==3
    mope(i,j)=mope(i+1,j)+dif3;
elseif A==4
    mope(i,j)=mope(i+1,j)+dif4;
end;
end;
end;
end;
end;

%A partir do ponto de interseção entre a linha e coluna sem salto de
%fase marcado nos passos anteriores, até a última linha e primeira
%coluna do mapa empacotado.
for i=(lin+1:s(1,1))
    for j=(col-1:-1:1)
        if mope(i,j)==0;
            %Se o ponto pode ser desempacotado, calcula a diferença do
            %ponto atual para o ponto da coluna à direita nas matrizes
            %map, alfa1, alfa2 e alfa3, que contém os mapas de fase
            %empacotados deslocados.

```

```

if mope(i,j+1)~=0 && mope(i,j+1)~=1

    dif1=(map(i,j)-map(i,j+1));
    dif2=(alfa2(i,j)-alfa2(i,j+1));
    dif3=(alfa3(i,j)-alfa3(i,j+1));
    dif4=(alfa4(i,j)-alfa4(i,j+1));
    dif=[abs(dif1) abs(dif2) abs(dif3) abs(dif4)];

    %Seleciona a menor diferena calculada sobre os mapas
    %empacotados.
    B=find(dif==min(dif));
    A=B(1);

    %Efetua o desempacotamento do ponto, a partir do mapa
    %cuja menor diferena tenha sido calculada.
    if A==1
        mope(i,j)=mope(i,j+1)+dif1;
    elseif A==2
        mope(i,j)=mope(i,j+1)+dif2;
    elseif A==3
        mope(i,j)=mope(i,j+1)+dif3;
    elseif A==4
        mope(i,j)=mope(i,j+1)+dif4;
    end;

    %Se o ponto pode ser desempacotado, calcula a diferena
    %do ponto atual para o ponto da linha acima nas atrizes
    %map, alfa1, alfa2 e alfa3, que contem os mapas de fase
    %empacotados deslocados.
    elseif mope(i-1,j)~=0 && mope(i-1,j)~=1

        dif1=(map(i,j)-map(i-1,j));
        dif2=(alfa2(i,j)-alfa2(i-1,j));
        dif3=(alfa3(i,j)-alfa3(i-1,j));
        dif4=(alfa4(i,j)-alfa4(i-1,j));
        dif=[abs(dif1) abs(dif2) abs(dif3) abs(dif4)];

        % Seleciona a menor diferena calculada sobre os mapas
        %empacotados.
        B=find(dif==min(dif));
        A=B(1);

        %Efetua o desempacotamento do ponto, a partir do mapa
        %cuja menor diferena tenha sido calculada.
        if A==1
            mope(i,j)=mope(i-1,j)+dif1;
        elseif A==2
            mope(i,j)=mope(i-1,j)+dif2;
        elseif A==3
            mope(i,j)=mope(i-1,j)+dif3;
        elseif A==4
            mope(i,j)=mope(i-1,j)+dif4;
        end;
    end;
end;
end;
end;

```

```

end;

% A partir do ponto de interseção entre a linha e coluna sem salto de
% fase marcado nos passos anteriores, até a primeira linha e primeira
% coluna do mapa empacotado.
for i=(lin-1:-1:1)
    for j=(col-1:-1:1)
        if mope(i,j)==0;

            % Se o ponto pode ser desempacotado, calcula a diferença do
            % ponto atual para o ponto da coluna à direita nas matrizes
            % map, alfa1, alfa2 e alfa3, que contém os mapas de fase
            % empacotados deslocados.
            if mope(i,j+1)~=0 && mope(i,j+1)~=1

                dif1=(map(i,j)-map(i,j+1));
                dif2=(alfa2(i,j)-alfa2(i,j+1));
                dif3=(alfa3(i,j)-alfa3(i,j+1));
                dif4=(alfa4(i,j)-alfa4(i,j+1));
                dif=[abs(dif1) abs(dif2) abs(dif3) abs(dif4)];

                %Seleciona a menor diferença calculada sobre os mapas
                %empacotados.
                B=find(dif==min(dif));
                A=B(1);

                %Efetua o desempacotamento do ponto, a partir do mapa
                %cuja menor diferença tenha sido calculada.
                if A==1
                    mope(i,j)=mope(i,j+1)+dif1;
                elseif A==2
                    mope(i,j)=mope(i,j+1)+dif2;
                elseif A==3
                    mope(i,j)=mope(i,j+1)+dif3;
                elseif A==4
                    mope(i,j)=mope(i,j+1)+dif4;
                end;

            %Se o ponto pode ser desempacotado, calcula a diferença
            %do ponto atual para o ponto da linha abaixo nas
            %matrizes map, alfa1, alfa2 e alfa3, que contém os
            %mapas de fase empacotados deslocados.
            elseif mope(i+1,j)~=0 && mope(i+1,j)~=1

                dif1=(map(i,j)-map(i+1,j));
                dif2=(alfa2(i,j)-alfa2(i+1,j));
                dif3=(alfa3(i,j)-alfa3(i+1,j));
                dif4=(alfa4(i,j)-alfa4(i+1,j));
                dif=[abs(dif1) abs(dif2) abs(dif3) abs(dif4)];

                %Seleciona a menor diferença calculada sobre os mapas
                %empacotados.
                B=find(dif==min(dif));
                A=B(1);

                %Efetua o desempacotamento do ponto, a partir do mapa
                %cuja menor diferença tenha sido calculada.

```

```

        if A==1
            mope(i,j)=mope(i+1,j)+dif1;
        elseif A==2
            mope(i,j)=mope(i+1,j)+dif2;
        elseif A==3
            mope(i,j)=mope(i+1,j)+dif3;
        elseif A==4
            mope(i,j)=mope(i+1,j)+dif4;
        end;
    end;
end;
end;
end;

mope(mask==1)=0; % Elimina a máscara inicialmente adicionada à matriz
                %mope.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Após o desempacotamento, calcula e remove a fase linear.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

x=(1:s(1,1));

Y=[mope(1,1) mope(s(1,1),1)]; %Determinação do ponto inicial e final do
                              %eixo Y.

X=[1 s(1,1); 1 1];          %Determinação do ponto inicial e final do
                              %eixo X.

A=Y/X;                       %Cálculo do coeficiente angular da reta
                              %paralela ao plano de projeção do mapa
                              %desempacotado.

ret= A(1,1).*x+A(1,2);      % Equação da reta paralela ao plano de
                              %projeção do mapa.

f=zeros(size(mope));
for i=(1:s(1,1))            %Geração do plano de projeção com base na
                              %reta calculada.
    f(i,:)=ret(i);
end
Mope=mope-f;                %Remoção da fase linear do mapa
                              %desempacotado e armazenamento na matriz
                              %Mope.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Segunda saída de processamento do algoritmo: o mapa de fase
desempacotado
%com a fase linear removida.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

imtool(Mope);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Encerramento do algoritmo %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

# REFERÊNCIAS

- [1] K., Creath, 1988. *Phase-measuring interferometry techniques*, Progress in Optics XXVI. Amsterdam: Elsevier, 349–93.
- [2] Chiang, P., Cai, Y., Mak, K. H. and Zheng, J., 2013. *A B-spline approach to phase unwrapping in tagged cardiac MRI for motion tracking.*, Magnetic Resonance in Medicine **69**, 1297–1309.
- [3] Starck, J.L., Murtagh, F., 2001. *Astronomical image and signal processing*, IEEE Signal Processing Magazine **2**, 30 -40.
- [4] Cuenca, MC., 2011. *A New Method for Temporal Phase Unwrapping of Persistent Scatterers InSAR Time Series.*, IEEE Transactions on Geoscience and Remote Sensing **11**, 4606 -4615.
- [5] Mitra, R., 1984. *Theory of Wave-Front Distortion Correction by Phase Conjugation.*, Journal of the Optical Society of America. A, Optics and Image Science **11**, 1103-1109.
- [6] henggen Quan, Cho Jui Tay, Lujie Chen, and Yu Fu, 2003. *Spatial-fringe-modulation-based quality map for phase unwrapping*, Applied Optics **35**, 7060-7065.
- [7] Quiroga, J. A., 1994. *Phase-unwrapping algorithm for noisy phase-map processing*, Applied optics **29**, 6725 -31
- [8] Irie, K, 2008. *A model for measurement of noise in CCD digital-video cameras*, Measurement science & technology **4**, 0957-0233.
- [9] Cancelo, Gustavo, 2012. *Achieving sub electron noise in CCD systems by means of digital filtering techniques that lower 1/f pixel correlated noise* Experimental astronomy **1**, 13 -29
- [10] Ghiglia, D.C. and Pritt, M.D., , 1998, *Two-Dimensional Phase Unwrapping: Theory, Algorithms and Software*, (John-Wiley & Sons).
- [11] Judge, T.R., Bryanston-Cross, P.J., 1994. *A review of phase unwrapping techniques in fringe analysis*, Optics and Lasers in Engineering **4**, 199–239.
- [12] Fornaro, G., Franceschetti, G., Lanari, R., Sansosti, E., Tesauro. M., 1997. *Global and local phase-unwrapping techniques: a comparison*, Journal of the Optical Society of America. A, Optics and image science **10**, 2702-2708.

- [13] Jeffrey, A., 2002, *Advanced Engeneering Mathematics*, (San Diego, San Francisco, New York, Boston, London, Toronto, Sydney, Tokyo, University of Newcastle-upon-Tyne), 625-861.
- [14] Kovesi, P., 1999. *Image features from phase congruency*, *Videre* **3**, 1-14.
- [15] Braithwaite, R. N., Bhanu, B., 1995. *Composite phase and phase-based Gabor element aggregation*, Proceedings. International Conference on Image Processing, 538-41.
- [16] Senneville, B., Mougnot, C., Quesson, B. et al, 2007. *MR thermometry for monitoring tumor ablation*, *European Radiology* **9**, 2401-2410.
- [17] Goldstein, R., Zebker, H., Werner, C., 1988. *Satellite radar interferometry: two-dimensional phase unwrapping*, in Symposium on the Ionospheric Effects on Communication and Related Systems, 713–720.
- [18] Oppenheim, A. V., Lim, J. S., 1981. *The importance of phase in signals*, Proceedings of the IEEE **5**, 529-541.
- [19] Hecht, E., 1998, *Optics*, (Addison Wesley), 3rd ed. Reading.
- [20] Tankam, P., 2011. *Use of digital color holography for crack investigation in electronic components*, *Optics and lasers in engineering* **11**, 1335 -1342.
- [21] Fried, David, 2001. *Adaptive optics wave function reconstruction and phase unwrapping when branch points are present*, *Optics communications* **1**, 43-72.
- [22] Brown, Gordon M., 1993. *Fringe analysis for automotive applications*, *Optics and Lasers in Engineering* **4-5**, 203–220.
- [23] Takeda, M., 1985. *Subfringe holographic interferometry by computer-based spatial-carrier fringe-pattern analysis* *ournal of optics* **3**, 127-131.
- [24] Roddier, F., 1988. *Interferometric imaging in optical astronomy*, *Physics reports* **2**, 297-166
- [25] Ruchel, Orlando T., 1996, *Princípios da comunicação digital*, (EDIPUCRS), 19-26.
- [26] Itoh, K., 1982. *Analysis of the phase unwrapping problem*, *Applied Optics* **14**.
- [27] Fu, Yanjun, 2013. *Three-dimensional profile measurement based on modified temporal phase unwrapping algorithm*, *Optik - International Journal for Light and Electron Optics* **6**, 557.
- [28] Huang, L., 2011. *Phase invalidity identification framework with the temporal phase unwrapping method*, *Measurement science & technology* **3**, 0957-0233.

- [29] Davila, A., Huntley, J. M., Pallikarakis, C., 2012. *Simultaneous wavenumber measurement and coherence detection using temporal phase unwrapping*, Applied Optics **5**, 558-567.
- [30] Liu, Y., Huang, D., Jiang, Y., 2012. *Flexible error-reduction method for shape measurement by temporal phase unwrapping: Phase averaging method*, Applied Optics **21**, 4945-4953
- [31] Asundi, A., 1998. *Fast phase-unwrapping algorithm based on a gray-scale mask and flood fill*, Applied optics **23**, 5416-5420
- [32] Huntley, J. M., 1989. *Noise-immune phase unwrapping algorithm*, Applied Optics **16**, 3268-3270.
- [33] K. Chen, J. Xi, Y. Yu & J. F. Chicharo, 2010. *Fast quality-guided flood-fill phase unwrapping algorithm for three-dimensional fringe pattern profilometry*, Optical Metrology and Inspection for Industrial Applications, 1-9.
- [34] Lo, C. F., 2002. *Surface normal guided method for two-dimensional phase unwrapping*, Optik **10**, 439-447.
- [35] Zhang, S., Li, X., Yau, S. T., 2007. *Multilevel quality-guided phase unwrapping algorithm for real-time three-dimensional shape reconstruction*, Applied Optics **1**, 50-57.
- [36] E. Li, X. Peng, J. Xi, J. Chicharo, J. Yao, D. Zhang, 2005. *Multi-frequency and multiple phase-shift sinusoidal fringe projection for 3D profilometry*, Optics Express **13**, 1561-1569.
- [37] Xu, Y., Jia, S., Luo, X., Yang, J., Zhang, Y., 2013. *Multi-frequency projected fringe profilometry for measuring objects with large depth discontinuities*, Optics Communications **288**, 0030-4018.
- [38] Takashi, K., Yonesaka, R., et. al., 2011. *High-speed phase imaging by parallel phase-shifting digital holography*, Optics Letters **21**, 4131-4133.
- [39] Cruz Meneses-Fabian, Uriel Rivera-Ortega, 2011. *Phase-shifting interferometry by wave amplitude modulation*, Optics Letters **36**, 2417-2419.
- [40] Zhongwei Li, Kai Zhong, Y. F. Li, Xiaohui Zhou, Yusheng Shi, 2013. *Multiview phase shifting: a full-resolution and high-speed 3D measurement framework for arbitrary shape dynamic objects*, Optics Letters **38**, 1389-1391.
- [41] Laura Ekstrand, Song Zhang, 2011. *Three-dimensional profilometry with nearly focused binary phase-shifting algorithms*, Optics Letters **23**, 4518-4520.

- [42] Chen, K., Jiangtao, X., Yanguang, Y., 2013. *Quality-guided spatial phase unwrapping algorithm for fast three-dimensional measurement*, Optics Communications **294**, 139–147.
- [43] Ming Zhao, Lei Huang, Qican Zhang, Xianyu Su, Anand Asundi, Qian Kemao, 2011. *Quality-guided phase unwrapping technique: comparison of quality maps and guiding strategies*, Applied Optics **33**, 6214-6224.
- [44] Suping Fang, Lei Meng, Leijie Wang, Pengcheng Yang, Masaharu Komori, 2011. *Quality-guided phase unwrapping algorithm based on reliability evaluation*, Applied Optics **28**, 5446-5452.
- [45] Lu, YG, 2012. *Weighted-phase-gradient-based quality maps for two-dimensional quality-guided phase unwrapping*, Optics and lasers in engineering **10**, 1397 -1404.
- [46] Zhong, HP, 2014. *Quality-Guided and Local Minimum Discontinuity*, IEEE Geoscience and Remote Sensing Letters [1545-598X] **1**, 215-219
- [47] Batuhan Osmanoglu, Timothy H. Dixon, Shimon Wdowinski, Enrique Cabral-Cano, 2011. *On the importance of path for phase unwrapping in synthetic aperture radar interferometry*, Applied Optics **19**, 3205-3220.
- [48] Zhu, Y., Zhu, L., Yang, Q., Lu, W., Liu, L., 2006. *Novel method to construct a quality map for phase unwrapping based on modulation and the phase gradient*, Optical engineering **10**, 105601.
- [49] Goldstein RM, Zebker HA, Werner CL., 1988. *Satellite radar interferometry: twodimensional phase unwrapping*, Radio Science **4**, 713-720.
- [50] Salah A. Karout, \* Munther A. Gdeisat, David R. Burton, and Michael J. Lalor, 2007. *Residue vector, an approach to branch-cut placement in phase unwrapping: theoretical study*, Applied Optics **21**, 4712-4727.
- [51] Haihua Cui, Wenhe Liao, Ning Dai, Xiaosheng Cheng, 2011. *Reliability-guided phase-unwrapping algorithm for the measurement of discontinuous three-dimensional objects*, Optical Engineering **6**, 63602.
- [52] Dapeng Gao, Fuliang Yin, 2012. *Mask Cut Optimization in Two-Dimensional Phase Unwrapping*, IEEE GEOSCIENCE AND REMOTE SENSING LETTERS **3**, 338-342.
- [53] Ghiglia, D. C., Romero, L. A., 1996. *Minimum Lp-norm two-dimensional phase unwrapping*, Journal of Optical Society of America **10**, 1999-2012.
- [54] Nico, G., Palubinskas, G., Datcu. M., 2000. *Bayesian Approaches to Phase Unwrapping: Theoretical Study*, IEEE TRANSACTIONS ON SIGNAL PROCESSING **9**, 2545-2556.

- [55] Chunsheng Yu, Qingjin Peng, 2007. *A correlation-based phase unwrapping method for Fourier-transform profilometry*, Optics and Lasers in Engineering **45**, 730-736.
- [56] Wenguo Li, A., B., Fang, S., Asundi, A., 2012. *Reliability-guided phase unwrapping algorithm following noncontinuous path based on color fringe projection*, Optik, 537-546.
- [57] Judge, T.R. , Bryanston-Cross, P.J., 1994. *A review of phase unwrapping techniques in fringe analysis*, Optics and Lasers in Engineering **4**, 199-239.
- [58] Donald, J. B., 1991. *Fourier fringe analysis: the two-dimensional phase unwrapping problem*, Appl. Opt. **30**, 3627-3632.
- [59] Srinivasan, V., Liu, H. C., Halioua, M., 1984. *Automated phase-measuring profilometry of 3-D diffuse objects*, Appl. Opt. **23**, 3105-3108.
- [60] Dongliang, Z., Feipeng, D., 2011. *A novel algorithm for branch cut phase unwrapping*, Optics and Lasers in Engineering **5**, 609-617.
- [61] Dongliang Zheng, Feipeng Da, 2012. *Self-correction phase unwrapping method based on Gray-code light*, Optics and Lasers in Engineering **8**, 1130-1139.
- [62] Wenguo Li, Suping Fang, 2012. *Reliability-guided phase unwrapping algorithm following noncontinuous path based on color fringe projection*, Optik - International Journal for Light and Electron Optics **6**, 537-546.
- [63] Sergio F., Munther A. G., Joaquim S., David B., 2011. *Automatic window size selection in Windowed Fourier Transform for 3D reconstruction using adapted mother wavelets*, Optics Communications **12**, 2797-2807.
- [64] Weijie Shi, Yongjian Zhu, Yong Yao, 2010. *Discussion about the DCT/FFT phase-unwrapping algorithm for interferometric applications*, Optik - International Journal for Light and Electron Optics **16**, 1443-1449.
- [65] Marvin A. S., Yimei Z., 2003. *Fast phase unwrapping algorithm for interferometric applications*, Opt. Lett. **28**, 1194-1196.
- [66] Gorthi, S. S., Rastogi, P., 2010. *Fringe projection techniques: Whither we are?*, Optics and Lasers in Engineering **2**, 133-140.
- [67] Huang P.S., Zhang S, Chiang F.P., 2004. *Trapezoidal phase-shifting method for 3D shape measurement*. , Proceedings of SPIE, vol. 5606, p. 142-152. in: K. G. Harding, editor Two- and Threedimensional vision systems for inspection, control, and metrology vol. II.

- [68] Zappa, E., Busca, G., 2012. *Static and dynamic features of Fourier transform profilometry: A review*, Optics and Lasers in Engineering **8**, 1140–1151.
- [69] Greivenkamp, J.E., 1992, *Phase shifting interferometry*, in: Optical Shop Testing (New York, Wiley), 2nd ed, 501–598.
- [70] Asundi, A., 1991. *Fringe analysis in moiré interferometry*, Proceedings of SPIE 1554B moiré technique, holographic interferometry, optical NDT, and applications to fluid mechanics, 472–480.
- [71] Phillion, D.W., 1997. Appl Opt. **36**, 809–811.
- [72] Tang, Shouhong., 1996. *Generalized algorithm for phase-shifting interferometry*, Proceedings of SPIE 2860 Laser Interferometry VIII: techniques and analysis, 34–44.
- [73] Kujawinska, M., 1987. Appl Opt. **26**, 4712.
- [74] Huang, P.S., Hu, Q., Chiang, F.P., 2002. Appl Opt. **41**, 4503.
- [75] Farrell, C.T., Player, M.A., 1994. Meas Sci Technol **5**, 648–54.
- [76] Takeda, M., Mutoh, K., 1983. Appl Opt. **22**, 3977.
- [77] Massig, J.H., Heppner, J., 2001. Appl. Opt. **40**, 2081-8.
- [78] Takeda, M., Gu Q, Kinoshita M, Takai H, Takahashi Y. , 1996. *Spatial frequency-multiplexed number-theoretic phase unwrapping technique for the Fourier-transform profilometry of objects with height discontinuities and/or spatial isolations*, Proceedings of SPIE 2860 laser interferometry, 46–52.
- [79] Takeda, M., Ina, H., Kobayashi, S., 1982. J. Opt. Soc. Am. **72**, 156-160.
- [80] See the website at <http://www.mathworks.com/matlabcentral/fileexchange/22504-2d-phase-unwrapping-algorithms>.
- [81] dos Santos, P.A.M, Nunes, L.C. da Silva, 2001. *Photorefractive holographic grating spacing estimated by a dynamica Moire fringe pattern generation*, Opt. Comm **194**, 97-101.
- [82] Herriau, J.P., Delboulbe, A., Huignard, J.P, 1983. *Spatial switching using a photoinduced holographic grating*, SPIE Proc **398**, 123.
- [83] Yeh, P., 1993, *Introduction to photorefractive nonlinear optics*, (New York, Wiley).
- [84] Malacara, D., 1989, *Óptica Básica, SEP*, (México, Fondo de Cultura).

[85] Mollenauer, L.F., Tomlinson, N.J., 1977. *Appl. Opt.* **16**, 555.