

UNIVERSIDADE ESTADUAL PAULISTA “JÚLIO DE MESQUITA FILHO”

Programa de Pós-graduação em Ciência e Tecnologia de Materiais

Alexandre Fioravante de Siqueira

**ESTUDOS DE IMAGENS PROVENIENTES DE MEMBRANAS DE
BORRACHA NATURAL COM ADITIVOS METÁLICOS
UTILIZANDO TÉCNICAS DE FOURIER, GABOR E *WAVELETS***

Presidente Prudente

2011

Alexandre Fioravante de Siqueira

ESTUDOS DE IMAGENS PROVENIENTES DE MEMBRANAS DE BORRACHA
NATURAL COM ADITIVOS METÁLICOS UTILIZANDO TÉCNICAS DE FOURIER,
GABOR E *WAVELETS*

Dissertação apresentada como requisito para
obtenção do título de Mestre em Ciência e Tecnologia de
Materiais à Universidade Estadual Paulista “Júlio de
Mesquita Filho” – Programa de Pós-graduação em Ciência e
Tecnologia de Materiais, área de concentração
Caracterização de Materiais, sob a orientação do Professor
Doutor Aldo Eloizo Job e a coorientação do Professor
Doutor Messias Meneguette Junior.

Presidente Prudente

2011

Siqueira, Alexandre Fioravante de.

Estudos de imagens provenientes de membranas de borracha natural com aditivos metálicos utilizando técnicas de Fourier, Gabor e *wavelets* / Alexandre Fioravante de Siqueira, 2011.

128 f. : il.

Orientador: Aldo Eloizo Job

Dissertação (Mestrado) - Universidade Estadual Paulista. Faculdade de Ciências e Tecnologia, Presidente Prudente, 2011

1. Borracha natural. 2. Transformada de Fourier. 3. *Wavelets*. 4. Visão computacional. I. Universidade Estadual Paulista. Faculdade de Ciências e Tecnologia. II. Título.

ATA DE APROVAÇÃO



ATA DA DEFESA PÚBLICA DA DISSERTAÇÃO DE MESTRADO DE ALEXANDRE FIORAVANTE DE SIQUEIRA, DISCENTE DO PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA E TECNOLOGIA DE MATERIAIS, DO(A) FACULDADE DE CIÊNCIAS DE BAURU.

Aos 20 dias do mês de junho do ano de 2011, às 14:00 horas, no(a) Faculdade de Ciências e Tecnologia - UNESP - Presidente Prudente, reuniu-se a Comissão Examinadora da Defesa Pública, composta pelos seguintes membros: Prof. Dr. ALDO ELOIZO JOB do(a) Departamento de Física Química e Biologia / Faculdade de Ciências e Tecnologia de Presidente Prudente, Prof. Dr. MESSIAS MENEGUETTE JUNIOR do(a) Departamento de Matemática, Estatística e Computação / Faculdade de Ciências e Tecnologia de Presidente Prudente, Profa. Dra. MARGARETE OLIVEIRA DOMINGUES do(a) Instituto Nacional de Pesquisas Espaciais - INPE, sob a presidência do primeiro, a fim de proceder a arguição pública da DISSERTAÇÃO DE MESTRADO de ALEXANDRE FIORAVANTE DE SIQUEIRA, intitulado "ESTUDOS DE IMAGENS OBTIDAS POR MICROSCOPIAS DE VARREDURA, FORÇA ATÔMICA E ÓPTICA, UTILIZANDO AS TRANSFORMADAS WAVELET E DE FOURIER". Após a exposição, o discente foi argüido oralmente pelos membros da Comissão Examinadora, tendo recebido o conceito final: Aprovado _____. Nada mais havendo, foi lavrada a presente ata, que, após lida e aprovada, foi assinada pelos membros da Comissão Examinadora.

Prof. Dr. ALDO ELOIZO JOB

Prof. Dr. MESSIAS MENEGUETTE JUNIOR

Profa. Dra. MARGARETE OLIVEIRA DOMINGUES

*Aos meus pais Aparecido e Luciclei, que
já enfrentaram grandes desafios em
vossas vidas, e aos meus sobrinhos
Igor e Eduardo, que chegaram
para continuar a nossa história.*

AGRADECIMENTOS

Como disse certa vez minha amiga Aline Gobbi, se à presença de Deus em minha vida eu pudesse associar uma função, com certeza esta seria contínua e munida das melhores propriedades matemáticas. Senhor, Te agradeço sem medidas por me abençoar sempre com o melhor, e por me capacitar para transpor todos os desafios que o mundo sempre propôs. Agradeço também com medida infinita ao meu senhor Jesus, por nos fornecer o primeiro conhecimento que toda pessoa deveria portar: “*amai-vos uns aos outros, como eu vos ame!*”.

Aos meus pais, Aparecido e Luciclei, que amo de todo o meu coração. Em toda sua vida houve honestidade, trabalho, fidelidade e, acima de tudo, muito amor. Eles são exemplo para qualquer casal, por todo o relacionamento que construíram e pela doação constante aos filhos. Agradeço muito a vocês por todas as oportunidades que me deram para que eu pudesse trilhar meus caminhos, e a Deus por ser vosso filho.

À minha senhora, Dayane Fumiyo. Se ao lado de todo grande homem deve existir uma grande mulher, com certeza serei grande apenas pela vossa presença, meu doce amor. Com todo o seu jeito “oriental” de ser você entrou na minha vida pra tomar conta de tudo, e quão feliz sou eu desde esse dia... Agradeço-lhe muito por toda a paciência, compreensão, carinho, cuidado e amor, e também pelas leituras constantes e forçadas deste trabalho. Muito obrigado também à sua família, dona Mamie (*okaachan*), seu Zé Carlos (sogrão) e meu cunhado David.

Ao Professor Doutor Aldo Eloizo Job, meu orientador e grande aconselhador, pela confiança e por todo o apoio para que o trabalho pudesse ser realizado. Dizem que a relação entre matemático e físico é difícil, contudo aprendo muito com a experiência que o senhor tem em abundância. Muito obrigado pela amizade concedida, pelas risadas e cervejas, por todos os empurrões psicológicos (algumas vezes merecidos... Algumas!) e pelos valores que ajudaram

muito no meu crescimento profissional e pessoal. Agradeço também à sua senhora, dona Ângela, pelos conselhos.

Ao Professor Doutor Messias Meneguette Júnior, meu coorientador neste trabalho, mas meu patrono matemático. Alguém com quem eu tive o prazer de conviver por anos da minha vida e me ensinou muito sobre valores, paciência e amizade, tudo com muito bom humor. Muito obrigado pela amizade, por todos os “cafezes”, estudos em conjunto, conversas sobre a vida e o mundo, e várias outras coisas que aconteceram ao longo de muitos anos. Obrigado também à sua senhora, a Professora Doutora Arlete Meneguette.

Aos professores que compuseram a banca, Professor Doutor Aylton Pagamisse e Professor Doutor José Ricardo de Rezende Zeni e Professora Doutora Margarete Oliveira Domingues. Aqui também deixo um agradecimento especial a Professora Doutora Eniuce Menezes de Souza, pelas dicas e correções valiosas. Muito obrigado pela presteza dos senhores, por todo vosso apoio e respeito para com este trabalho, tão engrandecido por vossa ajuda.

À galera do DFQB/POSMAT, que me acolheu durante o período de realização do trabalho. Obrigado ao Flávio (Makoto), Diogo (Diogão), Pedro (Aoki. Obrigado pelo carro!), Sabrina, Deuber, Elton, Carlos Gomes (Carlão), Renivaldo (Pirapó), Felipe (El Felipón), Priscila, Rafael (Mister), Fernando (Fernandinho), Renato (Sunshine), Rosana, entre outros. Em especial agradeço à Leandra Salmazo (Lê), grande amiga com a qual formamos a dupla dinâmica, ao Flávio Cabrera (Le Pac Man), pela parceria e pelas imagens de microscopia contidas no trabalho, e também ao colega Diego Henrique (Di), que nos deixou no dia 03 de julho para viver com Deus. Obrigado também aos Professores Doutores Eduardo René Perez Gonzalez, Carlos José Leopoldo Constantino, Carlos Alberto Tello Saenz e ao Professor Mestre Eduardo Roque Budemberg, pelo apoio prestado, e às secretárias Andressa (Dessa), do POSMAT, e Juvanir (Ju),

do DFQB, por toda a atenção e empenho em seus serviços (na maioria das vezes, procurados de última hora).

À galera do DMEC/PósMAC, minha conexão com o mundo matemático. Agradeço muito ao Diego (Cabeça), Tamiris (Tah), Marilaine (irmã), Camila, Vanderléa (Léia), Marluce e Marluci por todo o apoio concedido.

Aos meus gurus matemáticos, Glauber Renato Colnago (Galbe), grande amigo de várias tardes de Mario Kart e Metallica que me iniciou no caminho matemático, e Rafael Marcel Asmat Uceda (Gonzaliño), exemplo de superação com o qual eu pude conviver. Sem sua ajuda e amizade, não conseguiria começar (graças ao Glauber e à ajuda concedida nas tardes de domingo) e terminar (graças ao Rafael e todos os conselhos e momentos de risada nos cursos de verão) a faculdade de matemática com bom aproveitamento. Também ao bom amigo do tempo de graduação Rogério Galante Negri (Gerin), que me ajudou nos estudos estatísticos e de reconhecimento de padrão.

Ao *rock 'n' roll*, ao *blues*, ao *jazz*, à ópera e à música erudita, companheiros inseparáveis nas horas de atribulação mental. Como disse o grande compositor Johann Sebastian Bach, “*onde há devoção à música, Deus está sempre por perto com sua presença generosa*”. Um agradecimento especial aos gigantes Chuck Berry, Nat King Cole, Freddie Mercury, Django Reinhardt, Scatman John, Ray Charles, Louis Armstrong, Chico Buarque, Ronnie James Dio, Luciano Pavarotti, John Coltrane, B. B. King, John Denver, Tom Jobim, Eric Clapton, às senhoras Ella Fitzgerald, Nara Leão, Bonnie Raitt, Karen Carpenter, Teresa Teng, Elis Regina e finalmente, ao meu compositor preferido, Frédéric Chopin.

A todos que de alguma forma contribuíram para o desenvolvimento deste projeto.

À FAPESP, por financiar este trabalho.

*“It's the terror of knowing, what this world is about!
Watching some good friends screaming 'Let me out'!
Pray tomorrow – gets me higher, high, high!
Pressure on people – people on streets!
Turned away from it all like a blind man,
Sat on a fence but it don't work...
Keep coming up with love but it's so
slashed and torn, why, why, why?
Love, love, love, love, love...
Insanity laughs under pressure we're cracking!
Can't we give ourselves one more chance?
Why can't we give love that one more chance?
Why can't we give love, give love, give love, give love...?
'Cause love's such an old fashioned word,
and love dares you to care for the people
on the edge of the night
and love dares you to change our way of
caring about ourselves!
This is our last dance!
This is ourselves... Under pressure!”*

(Queen – Under Pressure)

RESUMO

A pesquisa aqui apresentada visa o reconhecimento de padrões entre imagens de microscopias de diferentes tipos, adquiridas de membranas de borracha natural com aditivos metálicos. Estes estudos deram origem ao *software* “WaveFPR” (*Wavelet and Fourier transforms for Pattern Recognition*), criado para auxiliar na aplicação das ferramentas matemáticas envolvidas, a saber: as transformadas de Fourier e de Gabor, e também as *wavelets* de Haar e Daubechies. Para cada imagem processada pelo *software*, é gerado um conjunto de coeficientes correspondente àquela imagem. Estes coeficientes são interpretados como uma “assinatura digital” da membrana; cada transformada retorna uma assinatura única para cada imagem. Estas assinaturas podem ser comparadas entre si, e esta comparação retorna informações relativas às membranas de borracha natural. O *software* criado oferece ainda uma interface para a utilização da técnica de emparelhamento (*template matching*) entre uma imagem-modelo de uma partícula metálica e uma imagem-alvo de borracha natural com aditivo metálico, com o uso dos coeficientes gerados. Do processamento de várias imagens foi construído um banco de dados com os coeficientes retirados destas imagens analisadas. Com este banco de dados e o uso da técnica de emparelhamento, são especificados os materiais constituintes de uma amostra, com o processamento da imagem pelo programa e a comparação dos resultados obtidos com os dados armazenados previamente.

Palavras-chave: Borracha Natural, Transformada de Fourier, *Wavelets*, Emparelhamento, Visão Computacional.

ABSTRACT

The research presented here aims at recognizing patterns between images of different types of microscopy, acquired from natural rubber membranes with metallic additives. These studies gave rise to the software “WaveFPR” (*Wavelet* and Fourier transforms for Pattern Recognition), created to assist in the application of the mathematical tools involved, namely: the Fourier and Gabor transforms, and also Haar and Daubechies *wavelets*. For each image processed by the software, a set of coefficients corresponding to that image is generated. These coefficients are interpreted as a “digital signature” of the membrane; each transform returns a unique signature to each image. These signatures can be compared to each other, and this comparison returns information about the natural rubber membranes. The software also offers an interface to use the template matching technique between a template image of a metallic particle and a target image of natural rubber with metallic additive, using the generated coefficients. A database was built with the coefficients taken from the analyzed images. This database contains information from several images. With this information and the template matching technique, the constituent materials of a sample are specified, processing the image with the software and comparing the obtained results with the previously stored data.

Keywords: Natural Rubber, Fourier Transform, *Wavelets*, Template Matching, Computational Vision.

LISTA DE FIGURAS

| | |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| Figura 1.1 – Coleta do látex em seringueiras <i>hevea brasiliensis</i> | 5 |
| Figura 1.2 – A <i>jebong</i> , faca utilizada para coleta do látex. | 6 |
| Figura 2.1 – Convenção dos eixos para representação de imagens digitais. | 11 |
| Figura 2.2 – Passos fundamentais do processamento de imagens. | 12 |
| Figura 2.3 – (a) imagem em formato ‘jpg’. (b) imagem (a) em escala de cinza. (c) imagem (a) com o mapa de cores ‘winter’. | 22 |
| Figura 4.1 – A função escala φ de Haar. | 36 |
| Figura 4.2 – A <i>wavelet</i> -mãe de Haar, ψ | 38 |
| Figura 4.3 – Funções-escala φ e <i>wavelets</i> -mãe ψ de Daubechies, para (a) $r = 2$ (D4); (b) $r = 3$ (D6); (c) $r = 4$ (D8); (d) $r = 10$ (D20). | 42 |
| Figura 4.4 – O processo de decomposição em múltiplos níveis esquematizado (onde $S = cA2 + cD2 + cD1$). | 43 |
| Figura 5.1 – Os casos gerais de diagrama de dispersão para correlação positiva: (a) correlação fraca; (b) correlação forte; (c) correlação perfeita. | 46 |
| Figura 5.2 – Os casos gerais de diagrama de dispersão para correlação negativa: (a) correlação fraca; (b) correlação forte; (c) correlação perfeita. | 47 |
| Figura 6.1 – A interface principal do programa “WaveFPR”. | 50 |
| Figura 6.2 – A janela relativa à opção “Transformada de Fourier” e seus resultados. | 51 |
| Figura 6.3 – A janela relativa à opção “Transformada de Gabor” e seus resultados. | 52 |
| Figura 6.4 – A janela relativa à opção “ <i>Wavelet</i> de Haar” para o nível de decomposição 1 e seus resultados. | 53 |
| Figura 6.5 – A janela relativa à opção “ <i>Wavelet</i> de Daubechies” para o nível de decomposição 1 e seus resultados. | 54 |
| Figura 6.6 – A janela relativa à opção “Emparelhamento...” e seus resultados. | 55 |
| Figura 7.1 – Imagem de amostra de borracha natural com nanopartículas de ouro, ampliada em 100x, obtida por microscopia óptica aos 15 minutos após a deposição da amostra na solução de ouro. | 56 |
| Figura 7.2 – Imagem binária obtida da Figura 7.1, após o processamento pelo método de Otsu. | 57 |
| Figura 7.3 – Resultados da aplicação do algoritmo referente à transformada de Fourier, após (a) aplicação direta de <code>fft2</code> ; (b) após a aplicação de <code>fft2</code> , o emprego de <code>fftshift</code> e <code>gscale</code> | 59 |

| | |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| Figura 7.4 – Resultado da aplicação do algoritmo relativo à transformada de Fourier na Figura 7.1. (a) Espectro de Fourier da imagem de entrada; (b) projeção dos valores do espectro de Fourier no eixo xy | 60 |
| Figura 7.5 – Resultado da aplicação do algoritmo relativo à transformada de Gabor na Figura 7.1. (a) Espectro de Gabor da imagem de entrada; (b) projeção dos valores do espectro de Gabor no eixo xy | 62 |
| Figura 7.6 – Espectros resultantes da aplicação da <i>wavelet</i> de Haar na Figura 7.1. (a) Nível 1 de decomposição; (b) nível 5 de decomposição. | 64 |
| Figura 7.7 – Projeções resultantes da aplicação da <i>wavelet</i> de Haar na Figura 7.1 pelos algoritmos implementados. (a) Nível 1 de decomposição; (b) nível 2 de decomposição; (c) nível 3 de decomposição; (d) nível 4 de decomposição; (e) nível 5 de decomposição..... | 65 |
| Figura 7.8 – Espectros resultantes da aplicação da <i>wavelet</i> de Daubechies na Figura 7.1. (a) Nível 1 de decomposição; (b) nível 7 de decomposição. | 68 |
| Figura 7.9 – Projeções resultantes da aplicação da <i>wavelet</i> de Daubechies na Figura 7.1 pelos algoritmos implementados. (a) Nível 1 de decomposição; (b) nível 2 de decomposição; (c) nível 3 de decomposição; (d) nível 4 de decomposição; (e) nível 5 de decomposição; (f) nível 6 de decomposição; (g) nível 7 de decomposição..... | 68 |
| Figura 7.10 – Platôs gerados pela aplicação da <i>wavelet</i> de Haar em imagens de microscopia óptica de membranas de borracha natural com aditivos metálicos. Os espectros foram obtidos de membranas de borracha natural depositadas em solução de ouro por (a) 6 minutos; (b) 9 minutos; (c) 30 minutos; (d) 60 minutos..... | 71 |
| Figura 7.11 – Regiões em nível geradas pela aplicação da <i>wavelet</i> de Daubechies em imagens de microscopia óptica de membranas de borracha natural com aditivos metálicos. Os espectros foram obtidos de membranas de borracha natural depositadas em solução de ouro por (a) 6 minutos; (b) 9 minutos; (c) 30 minutos; (d) 60 minutos. | 72 |
| Figura 7.12 – (a), (c), (e) Imagens de membranas de borracha natural e (b), (d), (f) a respectiva projeção dos espectros de Fourier para cada imagem, indicando padrões regulares no segundo plano da figura original. | 74 |
| Figura 7.13 – Correlação entre imagens de borracha natural com ouro para Fourier, entre (a) 6 minutos e 9 minutos. (b) 9 minutos e 30 minutos. (c) 30 minutos e 60 minutos. (d) 60 minutos e 120 minutos. | 77 |
| Figura 7.14 – Correlação entre imagens de borracha natural com ouro para Gabor, entre (a) 6 minutos e 9 minutos. (b) 9 minutos e 30 minutos. (c) 30 minutos e 60 minutos. (d) 60 minutos e 120 minutos. | 79 |
| Figura 7.15 – Correlação das imagens de borracha natural com ouro para Haar, entre (a) 6 minutos e 9 minutos. (b) 9 minutos e 30 minutos. (c) 30 minutos e 60 minutos. (d) 60 minutos e 120 minutos. | 81 |

| | |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| Figura 7.16 – Correlação das imagens de borracha natural com ouro para Daubechies, entre (a) 6 minutos e 9 minutos. (b) 9 minutos e 30 minutos. (c) 30 minutos e 60 minutos. (d) 60 minutos e 120 minutos. | 83 |
| Figura 7.17 – Aplicação do algoritmo de emparelhamento em uma imagem de borracha natural obtida por microscopia de força atômica. (a) imagem modelo; (b) imagem alvo; (c) resultado. | 84 |
| Figura 7.18 – As regiões semelhantes à imagem modelo detectadas na imagem alvo da Figura 7.17. (a) Soma da diferença quadrática das imagens. (b) Correlação cruzada normalizada. | 86 |
| Figura 7.19 – Aplicação do algoritmo de emparelhamento em uma imagem de borracha natural com aditivo metálico, obtida por microscopia óptica. (a) Imagem modelo; (b) imagem alvo; (c) resultado..... | 87 |
| Figura 7.20 – As regiões semelhantes à imagem modelo detectadas na imagem alvo da Figura 7.19. (a) Soma da diferença quadrática das imagens. (b) Correlação cruzada normalizada. | 88 |

LISTA DE TABELAS

| | |
|----------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| Tabela 1 – Cores resultantes da correspondência entre os valores numéricos no mapa de cores RGB..... | 16 |
| Tabela 2 – Mapas de cores predefinidos pelo Matlab. | 17 |
| Tabela 3 – Extensões possíveis para os tipos de arquivo nas funções <i>imread</i> e <i>imwrite</i> | 21 |
| Tabela 4 – Tamanho das matrizes resultantes da aplicação dos níveis de decomposição <i>wavelet</i> para Haar. | 62 |
| Tabela 5 – Tamanho das matrizes resultantes da aplicação dos níveis de decomposição <i>wavelet</i> para Daubechies..... | 65 |
| Tabela 6 – Coeficientes de correlação obtidos da transformada de Fourier de imagens depositadas em solução de ouro por diferentes espaços de tempo. | 76 |
| Tabela 7 – Coeficientes de correlação obtidos da transformada de Gabor de imagens depositadas em solução de ouro por diferentes espaços de tempo. | 78 |

SUMÁRIO

| | |
|--------------------------------------------------------------------------------|----|
| INTRODUÇÃO | 1 |
| CAPÍTULO 1 – DO LÁTEX ÀS IMAGENS | 4 |
| 1.1. A borracha natural | 4 |
| 1.2. Obtenção do látex | 6 |
| 1.3. A preparação das membranas de borracha natural com nanopartículas de ouro | 7 |
| 1.4. As microscopias empregadas para obtenção das imagens | 7 |
| 1.4.1. Microscopia eletrônica de varredura (MEV) | 8 |
| 1.4.2. Microscopia de força atômica (<i>atomic force microscopy</i> , AFM) | 8 |
| 1.4.3. Microscopia óptica (MO) | 9 |
| CAPÍTULO 2 – IMAGENS DIGITAIS | 10 |
| 2.1. Representação de imagens digitais | 10 |
| 2.2. O processamento de imagens | 11 |
| 2.2.1 – Aquisição da imagem | 12 |
| 2.2.2 – O pré-processamento | 13 |
| 2.2.3 – Segmentação da imagem e representação | 13 |
| 2.2.4 – Descrição | 14 |
| 2.2.5 – Reconhecimento e interpretação | 14 |
| 2.3. Imagens no Matlab | 15 |
| 2.3.1 – Cores | 16 |
| 2.3.2 – Formatos de imagem | 19 |
| 2.3.3 – Trabalhando com arquivos de imagens digitais | 20 |
| CAPÍTULO 3 – FOURIER E GABOR | 23 |
| 3.1. A transformada de Fourier | 24 |
| 3.1.1. A transformada contínua de Fourier | 25 |
| 3.1.2. A transformada discreta de Fourier | 25 |
| 3.1.3. A transformada rápida de Fourier (FFT) | 26 |
| 3.2. A transformada de Gabor | 27 |
| 3.2.1. A transformada contínua de Gabor | 27 |
| 3.2.2. A transformada discreta de Gabor | 28 |

| | |
|----------------------------------------------------------------------------------------------------------|----|
| 3.2.3. Filtros de Gabor..... | 29 |
| CAPÍTULO 4 – <i>WAVELETS</i> | 31 |
| 4.1. A transformada <i>wavelet</i> contínua (CWT)..... | 32 |
| 4.2. A transformada <i>wavelet</i> discreta (DWT)..... | 33 |
| 4.3. A análise multirresolução | 34 |
| 4.3.1. Análise das <i>wavelets</i> de Haar | 35 |
| 4.3.2. A definição de análise multirresolução | 39 |
| 4.3.3. As <i>wavelets</i> de Daubechies | 40 |
| 4.4. O algoritmo piramidal de Mallat | 42 |
| CAPÍTULO 5 – ESTATÍSTICA E VISÃO COMPUTACIONAL..... | 45 |
| 5.1. Correlação entre imagens..... | 45 |
| 5.2. A técnica de emparelhamento (<i>Template matching</i>) | 47 |
| CAPÍTULO 6 – O <i>SOFTWARE</i> “WaveFPR” | 49 |
| 6.1. A janela relativa à transformada de Fourier..... | 51 |
| 6.2. A janela relativa à transformada de Gabor | 52 |
| 6.3. A janela relativa à <i>wavelet</i> de Haar | 52 |
| 6.4. A janela relativa à <i>wavelet</i> de Daubechies..... | 53 |
| 6.5. A janela que fornece a técnica de emparelhamento..... | 54 |
| CAPÍTULO 7 – RESULTADOS E DISCUSSÃO..... | 56 |
| 7.1. O pré-processamento da imagem de entrada | 57 |
| 7.2. Algoritmo referente à transformada de Fourier | 58 |
| 7.3. Algoritmos referentes à transformada de Gabor..... | 60 |
| 7.4. Algoritmos referentes à <i>wavelet</i> de Haar | 62 |
| 7.5. Algoritmos referentes à <i>wavelet</i> de Daubechies | 65 |
| 7.6. O conjunto de imagens de teste | 69 |
| 7.6.1. Aplicação das transformadas nas imagens de teste obtidas por microscopia óptica..... | 69 |
| 7.6.2. Aplicação das transformadas nas imagens de teste obtidas por microscopia de força atômica..... | 73 |
| 7.7. Relação entre imagens retiradas em diferentes tempos de deposição | 75 |
| 7.8. Algoritmos referentes à técnica de emparelhamento | 83 |
| CONCLUSÃO..... | 89 |
| REFERÊNCIAS | 92 |

| | |
|---------------------------------------------------------------------------------------------------------------------------------|-----|
| APÊNDICE A – O TEOREMA DE Balian-LOW | 97 |
| APÊNDICE B – A OPERAÇÃO DE CONVOLUÇÃO | 99 |
| APÊNDICE C – A CRIAÇÃO DOS GRÁFICOS REFERENTES ÀS <i>WAVELETS</i> DE DAUBECHIES | 101 |
| APÊNDICE D – FUNÇÕES PARA A APLICAÇÃO DA TRANSFORMADA DE GABOR, <code>transgabor.m</code> E <code>gaborfilter.m</code> | 103 |
| APÊNDICE E – FUNÇÃO PARA A APLICAÇÃO DA TÉCNICA DE EMPARELHAMENTO, <code>template_matching.m</code> | 107 |

INTRODUÇÃO

As primeiras informações sobre a descoberta da borracha natural datam de 1493, quando Cristóvão Colombo, em sua segunda viagem à América do Sul, viu haitianos brincando com bolas feitas com seiva de coloração branca extraída de árvores. Os nativos chamavam a árvore de *kawotchou*, ou “árvore que chora”, em crioulo haitiano. O termo “borracha” foi utilizado pela primeira vez por John Priestly em 1770, quando soube que aquele material podia apagar marcas feitas com lápis[1].

A borracha foi introduzida no mundo ocidental por Charles de La Condamine, que enviou amostras do Peru à França em 1736, e publicou os resultados de seus estudos em 1745. Problemas no manuseio da borracha foram solucionados por Thomas Hancock em 1820, quando criou a máquina chamada “mastigadora”, que permitia ao usuário misturar, amaciar e formatar borracha sólida. Além disso, no ano de 1839, Charles Goodyear descobriu que aquecendo uma mistura de borracha e enxofre, tinha como resultado produtos com propriedades muito melhores que utilizando a borracha bruta. Era criado o processo de vulcanização, e rapidamente a demanda de artigos publicados tendo como tema a borracha cresceu, sendo que os produtos eram produzidos em sua maioria a partir da borracha natural vinda da região amazônica [2].

A borracha natural obtida do látex continua a ser fonte em potencial em diversas áreas de pesquisa, embora nos últimos três anos tivessem sido encontrados poucos artigos na literatura que viessem a contribuir com estudos de incorporação coloidal de nanopartículas metálicas em látex de borracha natural [3]. Pesquisadores brasileiros têm contribuído para o desenvolvimento de novos materiais com propriedades elastoméricas utilizando borracha natural

e/ou sintética, visando diversas áreas de aplicação. Dentre estes materiais, se destacam os com propriedades condutivas. A borracha natural com aditivos metálicos, núcleo de numerosas pesquisas em novos materiais, é objeto importante de estudo devido às características condutoras apresentadas.

Neste estudo são utilizadas ferramentas matemáticas para estudo de imagens provenientes de membranas de borracha natural obtidas de diferentes clones, entre os disponíveis na região de Presidente Prudente, com nanopartículas metálicas. Tais ferramentas são as transformadas *wavelet*, de Fourier e de Gabor, que atribuem às imagens um vetor de características próprio. Para esta tarefa foi criado um *software* na plataforma Matlab, denominado “WaveFPR” (*Wavelet and Fourier transforms for Pattern Recognition*), que realiza o pré-processamento das imagens de entrada e aplica as transformadas citadas de acordo com a escolha do usuário. Após o processamento, o programa permite a gravação dos coeficientes gerados pelas transformadas e de espectros relativos às imagens inseridas.

O programa WaveFPR permite ainda o uso da técnica de emparelhamento (também conhecida como *template matching*) para o reconhecimento automático de partículas metálicas em uma imagem de amostra de borracha natural. Nesta técnica são utilizados os coeficientes gerados previamente pelo *software*, com a aplicação das transformadas na imagem de entrada.

O estudo aqui exposto segue esta ordem:

- No Capítulo 1 deste trabalho são apresentados os processos desde a coleta do látex da seringueira, passando pela preparação das membranas e adição das partículas, até a obtenção das imagens digitais processadas e estudadas.

- Uma introdução sobre imagens digitais e sua manipulação pelo Matlab é dada no Capítulo 2. São listadas as diferentes formas de obtenção das imagens, sua preparação e segmentação, além de sua interpretação.
- O Capítulo 3 aborda as transformadas de Fourier e Gabor em suas versões discreta e contínua. Também são descritos os filtros de Gabor e a transformada rápida de Fourier.
- A teoria relativa às *wavelets*, também detalhando suas versões discreta e contínua, é dada no Capítulo 4. As *wavelets* de Haar e Daubechies são apresentadas, assim como o algoritmo piramidal de Mallat e uma introdução à análise multirresolução.
- Um breve estudo sobre as técnicas estatísticas e visão computacional é exposto no Capítulo 5. Tal capítulo aborda as técnicas de emparelhamento e correlação.
- A utilização e as funções do *software* WaveFPR são dadas no Capítulo 6. É apresentada uma visão geral das janelas que compõem o programa, assim como as informações retornadas por cada janela.

Ao fim desta pesquisa são dados os resultados conseguidos utilizando as transformadas Fourier, de Gabor e *wavelet*, após sua aplicação em membranas de borracha natural obtidas em condições semelhantes. O processamento prévio recebido pelas imagens também é abordado. Os resultados gráficos apresentados pelas transformadas são avaliados com o uso da técnica estatística de correlação, que mostra como as imagens apresentadas por cada transformada estão relacionadas.

CAPÍTULO 1 – DO LÁTEX ÀS IMAGENS

Neste capítulo são abordadas as técnicas de extração do látex, assim como o processo de obtenção das membranas de borracha natural, a deposição de nanopartículas metálicas nestas membranas, e os passos necessários para a obtenção das imagens a serem trabalhadas. As imagens utilizadas no trabalho foram cedidas por Flávio Camargo Cabrera, que mantém um projeto sobre redução *in situ* de nanopartículas metálicas em amostras de borracha natural [4].

1.1. A borracha natural

A borracha natural, como produto bruto, é a seiva leitosa coagulada (látex) extraída de espécies pertencentes sobretudo à família das euforbiáceas (*euphorbiaceae*). A espécie que apresenta maior produtividade de látex é a seringueira (*hevea brasiliensis*) [5], porém outras espécies, como a mamona (*ricinus communis* L.) e a mandioca (*manihot esculenta*), podem apresentar produção de seiva, ainda que em menor escala. Há também outras famílias que contêm espécies produtoras de borracha, como as moráceas (*moraceae*, de onde é produzida a borracha-de-assam), as apocináceas (*apocynaceae*, de onde provém a borracha-de-seda) e as espécies dente-de-leão (*taraxacum*) [6]. Do ponto de vista químico, o látex é uma dispersão coloidal de uma substância polimérica em um meio aquoso, essencialmente composta pelo monômero *cis-1,4-isopreno* [7], que apresenta comportamento visco-elástico.

Para a extração do látex são feitas incisões na casca da árvore, processo chamado de sangria. Ao serem realizadas as incisões, os vasos laticíferos liberam um fluido viscoso de

coloração amarelo-esbranquiçada, que é armazenado em tigelas de plástico, vidro ou metal. A liberação desse fluido cessa em um intervalo de três a quatro horas [8]. Na Figura 1.1 é ilustrado o processo de coleta do látex em seringueiras.



Figura 1.1 – Coleta do látex em seringueiras *hevea brasiliensis*.

O látex apresenta estabilidade de curta duração, devido à ação de microorganismos e enzimas que reduzem seu pH. Com isso ocorre sua coagulação espontânea, e é necessária a adição de estabilizadores (por exemplo, a amônia) para mantê-lo na forma de dispersão aquosa.

A obtenção de seringueiras que apresentem alta produtividade de látex é alcançada através do processo de enxertia de borbulhas, que consiste em destacar de uma planta um fragmento de casca munido de um gomo, inserindo-o no caule de outra planta. As árvores utilizadas no processo, provindas de mudas obtidas a partir de sementes, são reconhecidas como de alta produção. Estas mudas, assim enxertadas, são denominadas clones [9].

1.2. Obtenção do látex

O látex utilizado para aquisição das imagens foi obtido em diferentes árvores do clone RRIM 600, desenvolvido pelo Rubber Research Institute of Malaysia – RRIM, pertencentes à Fazenda Indiana, localizada na região de Presidente Prudente. A alta produção é o destaque deste clone, apresentando uma das maiores produtividades de borracha seca [2].

As incisões exercidas na árvore, para coleta do látex, são feitas sobre o caule a uma altura de 1,2 m a partir do solo. Nesta técnica é utilizada a *jebong*, ou faca seringueira, uma ferramenta cortante em forma de U. A Figura 1.2 mostra a faca *jebong*.



Figura 1.2 – A *jebong*, faca utilizada para coleta do látex.

O corte é feito a um ângulo de 30° com a horizontal, atingindo cerca de metade da circunferência do tronco, e aproximadamente 1,5 mm do córtex é removido por cada incisão. Por fim, o látex obtido é armazenado em recipientes de vidro de coloração escura, evitando assim a degradação do látex pela ação da luz solar.

1.3. A preparação das membranas de borracha natural com nanopartículas de ouro

Na confecção das membranas de borracha natural, o látex é diluído em amônia a proporção de 2%, e depositado em placas de Petri de 9,5 cm de diâmetro interno e fundo reto pela técnica de *casting*. Após a deposição, a membrana é submetida ao aquecimento numa estufa de secagem com circulação de ar forçado a uma temperatura de 80°C, pelo período de 12 horas, numa proporção de 10 ml, de modo a formar membranas de borracha natural com uma espessura de 0,62 mm. A deposição das nanopartículas de ouro é dada pela técnica de redução *in situ*, utilizando a solução de cloreto de ouro (AuCl₃) [4].

1.4. As microscopias empregadas para obtenção das imagens

A microscopia é a ciência que estuda os métodos e as aplicações em que se usa o microscópio, aparelho utilizado para a observação de objetos com dimensões inferiores ao limite de resolução do olho humano (0,1 mm).

O microscópio é classificado em função do número de sistema de lentes. Suas classificações são:

- *Microscópio simples (ou lupa)*: possui apenas uma lente, ou um sistema de lentes centradas.
- *Microscópio composto*: possui dois sistemas de lentes centradas, ocular e objetiva, para produzir uma imagem ampliada.
- *Microscópio fotônico*: o responsável pela transmissão de imagem é um feixe de fótons.
- *Microscópio eletrônico*: o responsável pela transmissão da imagem é um feixe de elétrons.

Nas próximas seções serão abordados os microscópios utilizados para a obtenção das imagens trabalhadas nesta pesquisa: microscópio eletrônico de varredura (MEV), microscópio de força atômica (AFM) e microscópio óptico (MO).

1.4.1. Microscopia eletrônica de varredura (MEV)

O microscópio eletrônico de varredura (conhecido pela sigla MEV; em inglês, *scanning electron microscope*, SEM) é geralmente utilizado para o estudo de estruturas superficiais ou subsuperficiais de amostras com dimensões relativamente grandes. Possui alta profundidade de foco, podendo obter diferentes relevos da superfície da amostra simultaneamente. O MEV também produz imagens de alta resolução, o que garante obter alta ampliação de detalhes sem perder a nitidez.

O MEV funciona graças a elétrons gerados por um canhão de elétrons. Neste canhão os elétrons são gerados pelo aquecimento resistivo de um filamento de tungstênio em forma de V utilizando uma fonte de alta tensão. A tensão de operação pode variar de 1 kV até 40 kV, e quanto maior a tensão de aceleração, mais fino se torna o feixe de elétrons, o que resulta em uma maior resolução [10].

1.4.2. Microscopia de força atômica (*atomic force microscopy*, AFM)

A microscopia de força atômica (*atomic force microscopy*, AFM) é uma técnica que mede as interações entre uma ponteira e a superfície a ser analisada. Tais interações se dão a partir de forças de Van der Waals, forças eletrostáticas (coulombianas), magnéticas e de atrito, dependendo do tipo de material e da distância entre a ponteira e a amostra [2].

A técnica de AFM fornece, entre outras informações, a topografia da superfície do filme, com seus defeitos e ordenamentos moleculares. Tais informações possibilitam a medida direta de suas alturas e profundidades, assim como a rugosidade do filme. As medidas de AFM são feitas localmente, varrendo áreas de centenas de *ångströms* quadrados, e não requerem que a superfície estudada seja condutora, o que confere ao AFM numerosas aplicações.

1.4.3. Microscopia óptica (MO)

O microscópio óptico é comumente utilizado para a visualização e caracterização de texturas e superfícies. Este microscópio pertence ao grupo de microscópios compostos, contando com dois sistemas de lentes convergentes, objetiva e ocular, como citado anteriormente. A objetiva é um conjunto de lentes que apresenta pequena distância focal e que fornece uma imagem real e aumentada do objeto que é observado. A ocular, também formada por lentes convergentes, atua como uma lupa, que proporciona uma imagem virtual e aumentada da imagem real formada pela objetiva.

A potência do microscópio óptico é resultado do produto da ampliação linear da objetiva pela potência da ocular; seu valor será elevado quando as distâncias focais da objetiva e ocular forem pequenas.

Uma das limitações da microscopia óptica é a profundidade de foco, na situação em que são analisadas amostras com topografia formada por picos e vales. Outra limitação é o aumento máximo possível, limitado em cerca de 2000 vezes. Como consequência, não é possível detectar pequenos detalhes estruturais através desta técnica [11].

CAPÍTULO 2 – IMAGENS DIGITAIS

A palavra imagem, do latim *imago*, diz respeito à representação visual de um objeto. Matematicamente, o termo imagem refere-se à função bidimensional de intensidade de luz $f(x, y)$ [12], onde x e y denotam as coordenadas espaciais e o valor $f(x, y)$ é proporcional ao brilho, ou nível de cinza, da imagem no ponto (x, y) .

2.1. Representação de imagens digitais

Uma imagem digital é uma imagem $f(x, y)$ discretizada tanto nas coordenadas espaciais quanto no brilho, que pode ser entendida como uma matriz cujos índices de linhas e colunas identificam um ponto na imagem. O valor correspondente do elemento da matriz identifica o nível de cinza naquele ponto, e os elementos dessa matriz digital são chamados de elementos da imagem, os conhecidos *pixels* (abreviações de “*picture elements*”). Quanto mais *pixels* uma imagem tiver, melhores são a resolução e qualidade desta. Na Figura 2.1, tem-se uma representação dos eixos X e Y, assim como a origem, em uma imagem digital. A convenção dada é a usada no processamento de imagens [12].

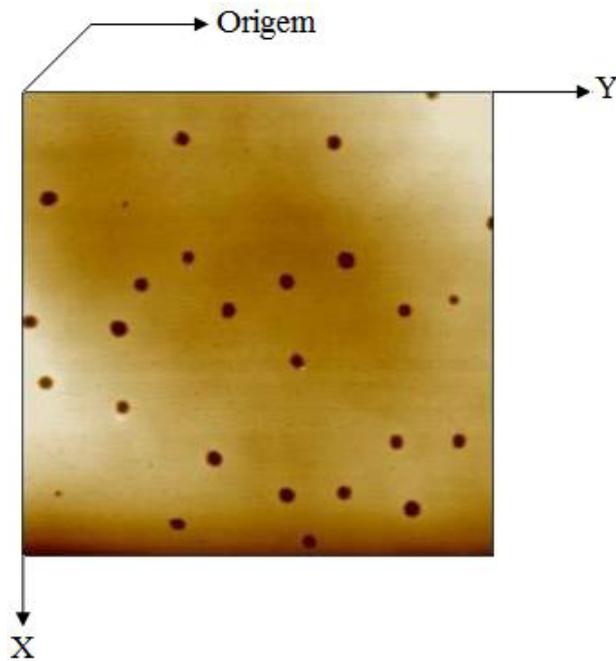


Figura 2.1 – Convenção dos eixos para representação de imagens digitais.

Pode ser útil visualizar a função da imagem em perspectiva, com um terceiro eixo representando os níveis de cinza. Usando esta convenção para atribuir proporcionalmente valores mais altos para áreas de maior brilho, é obtida a altura dos componentes da figura proporcional ao brilho correspondente na imagem.

2.2. O processamento de imagens

O interesse em métodos de processamento digital de imagens surgiu principalmente pela necessidade de melhorar a qualidade da informação pictórica para interpretação humana.

Uma das primeiras aplicações das técnicas do processamento de imagens foi a melhoria de ilustrações de jornais enviados por cabo submarino entre Londres e Nova Iorque, por

volta de 1920 [12]. Estas técnicas serão denominadas aqui como “passos fundamentais do processamento de imagens”, e são exibidas no diagrama abaixo [13].

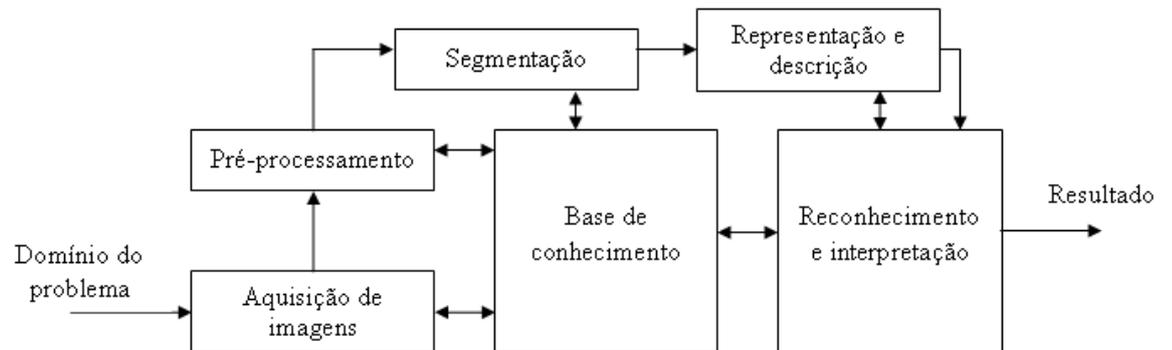


Figura 2.2 – Passos fundamentais do processamento de imagens.

Estes passos são discutidos em detalhes nas próximas seções. As descrições a seguir foram obtidas de Vanessa Bastos em [13], um sítio na internet com informações sobre processamento de imagens.

2.2.1 – Aquisição da imagem

O primeiro passo no processo é a aquisição da imagem digital. Neste passo é necessário um sensor para imageamento, assim como a capacidade de digitalizar o sinal produzido pelo sensor. O sensor para imageamento poderia ser uma câmera de TV, uma câmera digital, um microscópio digital, ou também uma câmera de varredura por linha. Se a saída da câmera ou outro sensor de imageamento não se encontrar na forma digital, um conversor analógico-digital realiza a digitalização (por exemplo, um *scanner*).

2.2.2 – O pré-processamento

A função chave do pré-processamento é melhorar a imagem, de tal forma que as chances de sucesso dos processos seguintes sejam aumentadas. O pré-processamento envolve tipicamente técnicas para o realce de contrastes, remoção de ruído e isolamento de regiões cuja textura indique a probabilidade de alguma informação relevante.

2.2.3 – Segmentação da imagem e representação

Após o pré-processamento, a imagem é dividida em partes (ou objetos) constituintes. A este estágio dá-se o nome de segmentação.

Em geral, a segmentação automática é uma das tarefas mais difíceis no processamento de imagens digitais: um procedimento de segmentação robusto favorece substancialmente a solução bem sucedida de um problema de imageamento, enquanto algoritmos de segmentação fracos ou erráticos garantem falhas no processamento, na maioria dos casos.

A saída do estágio de segmentação é constituída tipicamente por dados em forma de *pixels*, correspondendo tanto à fronteira de uma região como a todos os pontos dentro da mesma. É necessário converter os dados desta saída para uma forma adequada ao processamento computacional.

A primeira decisão que precisa ser tomada na segmentação de imagens é se os dados devem ser representados como fronteiras ou como regiões completas. A representação por fronteira é adequada quando o interesse se concentra nas características da forma externa, tais como cantos ou pontos de inflexão. A representação por região é adequada quando o interesse se concentra em propriedades internas, tais como textura ou esqueleto da imagem. Entretanto, em algumas aplicações, essas representações coexistem.

A escolha de uma representação é apenas parte da solução para representar os dados iniciais numa forma adequada para o subsequente processamento computacional. Um método para descrever os dados também deve ser especificado, de forma que as características de interesse sejam enfatizadas.

2.2.4 – Descrição

Depois da escolha da representação, começa o processo de descrição, também chamado seleção de características. Este processo procura extrair peculiaridades que resultem em alguma informação quantitativa de interesse, ou que sejam básicas para discriminação entre classes de objetos.

2.2.5 – Reconhecimento e interpretação

O último estágio do processamento de imagens envolve reconhecimento e interpretação. Reconhecimento é o processo que atribui um rótulo a um objeto, baseado na informação fornecida pelo seu descritor. Interpretação envolve a atribuição de significado a um conjunto de objetos reconhecidos, procurando atribuir algum significado a um conjunto de entidades rotuladas. Por exemplo, uma cadeia de cinco números seguidos por um hífen mais três números pode ser interpretada como um código de endereçamento postal.

Depois de sofrer os processos listados, a imagem pode ser trabalhada pelos algoritmos relativos às transformadas. Porém, o Matlab fornece ao usuário enorme flexibilidade no tratamento de imagens, através de comandos e ferramentas muito utilizadas no processamento de imagens. A manipulação básica de imagens no Matlab é tratada na seção a seguir; as

informações desta seção, assim como as tabelas, foram obtidas em sua maioria de Duane Hanselman e Bruce Littlefield em [14].

2.3. Imagens no Matlab

Além de ser um poderoso recurso para cálculos, o Matlab se distingue pelo modo interessante e informativo de apresentar os dados. Este *software* contém diversas ferramentas para exibir informações visualmente, em duas ou três dimensões.

O Matlab pode exibir diversos tipos de imagens, sendo criadas e armazenadas como números de ponto flutuante com precisão dupla (*double*) e também como inteiros sem sinal, de 8 bits (*uint8*) ou 16 bits (*uint16*). No Matlab os comandos `load` e `save` podem ser usados para, respectivamente, ler e gravar imagens em arquivos de extensão “`mat`”, assim como manipular arquivos de imagens de diversas extensões comerciais, como “`gif`” ou “`jpeg`”.

Imagens no Matlab são representadas por uma matriz de dados e, normalmente, uma matriz de mapa de cores associada à primeira. Existem três tipos de matrizes de dados de imagem, sendo que cada uma é interpretada de modo diferente: imagens indexadas, imagens de intensidade e imagens *true-color*, também chamadas RGB (*Red, Green, Blue*).

Uma imagem RGB é criada a partir de uma matriz m por n por 3, contendo “trios” RGB válidos. As dimensões linha e coluna especificam a localização do *pixel* e a página, ou terceira dimensão, especifica cada componente de cor. Por exemplo, o *pixel* P_{ij} é desenhado na cor especificada por $X(i, j, :)$. Não é necessário um mapa de cores, já que os dados de cores são armazenados dentro da própria matriz de dados de imagem. Por exemplo, o comando `image(X)` exibe a imagem contida na variável X , onde esta é uma imagem RGB, com dimensão m por n por 3, podendo conter dados *uint8*, *uint16* ou *double*.

2.3.1 – Cores

Muitas vezes um simples gráfico bi ou tridimensional não pode exibir todas as informações que se deseja apresentar de uma só vez. A cor pode fornecer uma “dimensão” adicional, apresentando mais dados sobre a imagem apresentada.

Um mapa de cores é um vetor contendo valores que variam de uma maneira especificada da primeira linha até a última. O Matlab utiliza um vetor numérico, intitulado mapa de cores, com três colunas para representar os valores referentes às cores. Cada linha na matriz representa uma cor individual, com números entre zero e um. Os números em cada linha especificam a intensidade das componentes vermelho verde e azul de uma cor, costumeiramente chamadas de valores RGB. A Tabela 1 ilustra a correspondência entre os valores numéricos no mapa de cores RGB e as cores resultantes.

Tabela 1 – Cores resultantes da correspondência entre os valores numéricos no mapa de cores RGB.

| Vermelho | Verde | Azul | Cor resultante |
|----------|-------|------|----------------|
| 1 | 0 | 0 | Vermelho |
| 0 | 1 | 0 | Verde |
| 0 | 0 | 1 | Azul |
| 1 | 1 | 0 | Amarelo |
| 1 | 0 | 1 | Magenta |
| 0 | 1 | 1 | Ciano |
| 0 | 0 | 0 | Preto |
| 1 | 1 | 1 | Branco |
| 0,5 | 0,5 | 0,5 | Cinza-médio |
| 0,67 | 0 | 1 | Violeta |

| Vermelho | Verde | Azul | Cor resultante |
|----------|-------|------|-----------------|
| 1 | 0,4 | 0 | Laranja |
| 0,5 | 0 | 0 | Vermelho escuro |
| 0 | 0,5 | 0 | Verde escuro |

Respectivamente, a primeira, segunda e terceira colunas neste mapa de cores são as intensidades de vermelho, verde e azul. O Matlab contém diversos mapas de cores predefinidos, como mostra a Tabela 2.

Tabela 2 – Mapas de cores predefinidos pelo Matlab.

| Mapa de cores | Descrição |
|---------------|-----------------------------------------------------------------|
| Hsv | Escala com cores saturadas, que começa e termina no vermelho. |
| Jet | Variante do mapa Hsv, que começa no azul e termina no vermelho. |
| Hot | Alterna entre preto, vermelho, amarelo e branco. |
| Cool | Tons de ciano e magenta. |
| Summer | Tons de verde e amarelo. |
| Autumn | Tons de vermelho e amarelo. |
| Winter | Tons de azul e verde. |
| Spring | Tons de magenta e amarelo. |
| White | Mapa de cores totalmente branco. |
| Gray | Escala linear de tons de cinza. |
| Bone | Escala de tons de cinza levemente azulados. |
| Pink | Tons pastéis de rosa. |
| Copper | Escala linear de tons acobreados. |
| Prism | Alterna entre as cores do arco-íris (vermelho, laranja, |

| Mapa de cores | Descrição |
|---------------|-------------------------------------------------|
| | amarelo, verde, azul, anil e violeta). |
| Flag | Alterna entre vermelho, branco, azul e preto. |
| Lines | Mapa que usa as mesmas cores do comando “plot”. |

Normalmente, cada mapa de cores gera uma matriz de 64 linhas e três colunas, que especifica os valores RGB de 64 cores. A cada uma dessas funções pode-se fornecer como parâmetro o número de linhas que devem ser gerados. Por exemplo, o comando `hot(m)` gerará uma matriz de dimensão m por 3, contendo os valores RGB de cores que variam desde o preto até o branco, passando por tons de vermelho, laranja e amarelo.

Na linha de comando do Matlab, tem-se:

```
>> hot (8)
```

O resultado do comando é a matriz dada abaixo.

```
ans =
    0.3333    0    0
    0.6667    0    0
    1.0000    0    0
    1.0000  0.3333    0
    1.0000  0.6667    0
    1.0000  1.0000    0
    1.0000  1.0000  0.5000
    1.0000  1.0000  1.0000
```

Os mapas de cores podem ser usados através do comando `colormap(M)`, que define a matriz M como o mapa de cores que será utilizado na janela atual.

2.3.2 – Formatos de imagem

O tipo de dado numérico padrão no Matlab é o *double*, um número de ponto flutuante de 64 *bits*, de precisão dupla. O Matlab aceita de forma limitada outros formatos para imagens, tais como caracteres de 16 *bits* (*uint16*) e inteiros de 8 *bits* sem sinal (*uint8*). Os comandos `image` e `imagesc` podem exibir imagens de 8 e 16 *bits* sem convertê-las primeiro ao formato *double*.

Para imagens indexadas, `image` atribui os valores zero e 255 ao primeiro e último registros, respectivamente, de um mapa de 256 cores, definindo automaticamente o espaçamento entre as cores.

Como o intervalo normal de dados *double* para imagens indexadas se inicia com o valor unitário, a conversão entre *uint8/uint16* e *double* requer a soma de uma unidade aos valores. Além disso, as operações matemáticas em vetores *uint8* não estão definidas e para executar operações matemáticas sobre inteiros sem sinal, é preciso convertê-los ao formato *double*. Os comandos abaixo convertem os dados do tipo *uint8* contidos em `Xuint8` para *double* e os convertem de volta para *uint8*.

```
>> Xdouble = double(Xuint8) + 1;  
>> Xuint8 = uint8(Xdouble - 1);
```

Conversões para *double* também podem ser normalizadas, por exemplo, com os comandos abaixo:

```
>> Xdouble = double(Xuint8)/255;  
>> Xuint8 = uint8(round(Xdouble * 255));
```

Os dados de cor de 8 *bits* contidos em uma imagem RGB são automaticamente escalados quando esta é exibida. Um exemplo é a cor branca, normalmente quando se usa

doubles. Se a mesma cor é armazenada como um dado de 8 *bits*, ela é representada pelo vetor [255 255 255].

2.3.3 – Trabalhando com arquivos de imagens digitais

Os dados de uma imagem podem ser salvos em um arquivo e recarregados no Matlab com o emprego de vários formatos de arquivos. As funções `save` e `load` do Matlab aceitam dados de imagem em formatos *double*, *uint8* ou *uint16*, da mesma maneira que aceitam qualquer outra variável ou tipo de dado do Matlab.

Quando se gravam imagens indexadas ou imagens de intensidade com mapas de cores diferentes do padrão, é preciso gravar o mapa de cores junto com os dados de imagem, como dado no comando abaixo:

```
>> save exemplo.mat X map
```

O Matlab também aceita diversos formatos de arquivos de imagem comerciais, por meio das funções `imread` e `imwrite`. A sintaxe de `imread` é:

```
>> A = imread(NOMEARQ, EXT)
```

A função lê a imagem do arquivo e a armazena na matriz *A*. *NOMEARQ* é a *string* que especifica o nome do arquivo, e *EXT* é a variável que especifica o formato do arquivo.

A sintaxe de chamada para `imwrite` varia conforme o tipo da imagem e o formato do arquivo.

```
>> A = imwrite (A, NOMEARQ, EXT)
```

Esta função escreve imagens da variável *A* para um arquivo, onde *NOMEARQ* é a *string* que especifica o nome do arquivo e *EXT* o seu formato.

Os possíveis valores para *EXT* nas funções `imread` e `imwrite` são dados abaixo, na Tabela 3, seguida por um exemplo da utilização destas duas funções.

Tabela 3 – Extensões possíveis para os tipos de arquivo nas funções `imread` e `imwrite`.

| Extensão (EXT) | Tipo de arquivo |
|-----------------------|----------------------------------|
| 'jpg' ou 'jpeg' | Joint Photographic Experts Group |
| 'tif' ou 'tiff' | Tagged Image File Format |
| 'gif'* | Graphics Interchange Format |
| 'bmp' | Windows Bitmap |
| 'png' | Portable Network Graphics |
| 'hdf' | Hierarchical Data Format |
| 'pcx' | Windows Paintbrush |
| 'xwd'* | X Window Dump |
| 'cur'* | Window Cursor Resources |
| 'ico'* | Window Icon Resources |

*valores permitidos apenas para `imread`.

```
>> A = imread('latex','jpg');
% Lê uma imagem de arquivo jpg.
>> figure(1); image(A);
% Abre a imagem A na janela figure(1).
>> I = rgb2gray(A);
% Converte a imagem em escala de cinza.
>> figure(2); imshow(I)
% Abre a imagem I na janela figure(2).
>> figure(3); imshow(I); colormap(winter)
% Abre a imagem I na janela figure(3) e atribui o mapa de cores
"winter".
```

As três imagens produzidas em *figure(1)*, *figure(2)* e *figure(3)*, respectivamente, são dadas na Figura 2.3, abaixo.

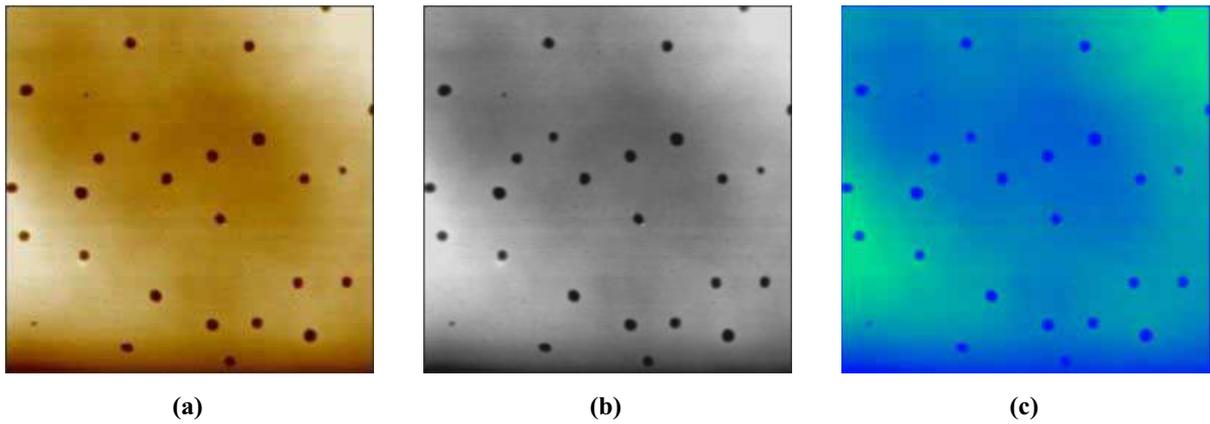


Figura 2.3 – (a) imagem em formato ‘jpg’. (b) imagem (a) em escala de cinza. (c) imagem (a) com o mapa de cores ‘winter’.

CAPÍTULO 3 – FOURIER E GABOR

Aqui será descrita a teoria referente às transformadas de Fourier e Gabor, utilizadas no estudo das imagens digitais constantes deste trabalho, e também presentes no programa WaveFPR. Tais transformadas permitem a análise de um sinal no domínio de frequência, facilitando o processamento deste sinal.

Jean-Baptiste Joseph Fourier foi um matemático e físico francês, mundialmente conhecido por iniciar os estudos sobre a decomposição de funções periódicas em séries trigonométricas convergentes (hoje, chamadas séries de Fourier), e a sua aplicação nos problemas da condução do calor.

Fourier foi motivado pelo estudo da equação do calor, que é gerida por uma equação diferencial linear. Entretanto, a transformada de Fourier diagonaliza todos os operadores lineares tempo-invariantes, que são as bases do processamento de sinais [15]. Tal transformada é uma das técnicas matemáticas com maior número de aplicações; por exemplo, as imagens em formato JPEG são comprimidas pela transformada de Fourier (formato substituído pelo JPEG2000, no qual a compressão é feita com o uso das *wavelets* [16]). Contudo, pela falta de uma análise no domínio do tempo, os esforços foram voltados para a criação de uma transformada que disponibilizasse informações nos domínios de tempo e frequência. A transformada de Gabor é uma dessas alternativas.

Nomeada após os estudos do engenheiro húngaro Dennis Gabor, a transformada de Gabor é um caso especial da transformada em janelas de Fourier (*windowed Fourier transform*),

sendo usada para determinar a frequência senoidal e o conteúdo de fase das seções locais de um sinal, conforme ele muda com o tempo.

Gabor foi premiado com o Nobel de Física pela invenção e aperfeiçoamento da holografia, referida naquela ocasião como um “sistema de fotografia tridimensional, com várias aplicações” [17].

3.1. A transformada de Fourier

A transformada de Fourier é uma transformada integral que expressa uma função como soma ou integral de funções senoidais, multiplicadas por coeficientes. Por ser uma ferramenta versátil, é usada em vários campos da ciência, a saber: física, química, teoria dos números, análise combinatória, processamento de sinais e imagens, teoria das probabilidades, entre outras áreas, como análise de sistemas lineares, óptica, modelagem de processos aleatórios e problemas de valores de contorno [18].

Essa transformada torna possível analisar um sinal no domínio do tempo pelo conteúdo de sua frequência, trabalhando primeiro com a tradução de uma função no domínio temporal para uma função no domínio frequencial. Assim, o sinal pode ser analisado pelo conteúdo de sua frequência, pois os coeficientes de Fourier da função transformada representam a contribuição de cada função seno e cosseno em cada frequência.

O uso amplo de tal transformada se dá pela possibilidade de tornar um problema de difícil resolução em um mais simples de ser resolvido. Em processamento de sinais, por exemplo, a transformada de Fourier é tipicamente utilizada para decompor um sinal nas suas componentes em frequência e suas amplitudes.

3.1.1. A transformada contínua de Fourier

A definição formal da transformada de Fourier de uma função $f(x)$ é dada por

$$F(\xi) = \int_{-\infty}^{\infty} f(x)e^{-i\xi x} dx, \quad (3.1)$$

onde ξ é comumente chamada de variável de frequência.

Aplicando a fórmula de Euler ($e^{i\theta} = \cos \theta + i \sin \theta$) em (3.1), segue que

$$F(\xi) = \int_{-\infty}^{\infty} f(x)(\cos \xi - \sin i\xi) dx,$$

de onde se vê que $F(\xi)$ é uma soma de senos e cossenos.

Tomando a função $F(\xi)$, pode-se obter $f(x)$ pela inversa da transformada de Fourier, que é dada pela equação

$$f(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\xi)e^{i\xi x} d\xi. \quad (3.2)$$

A variante bidimensional da transformada de Fourier é dada por (3.1):

$$F(\xi, \varphi) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y)e^{-i(\xi x + \varphi y)} dx dy,$$

em que ξ e φ são as variáveis de frequência [19].

3.1.2. A transformada discreta de Fourier

A transformada discreta de Fourier aproxima a transformada de Fourier de uma função, dado um número finito de pontos. O cálculo computacional da transformada de Fourier requer valores discretos de $f(x)$ para sua implementação, uma vez que um computador trabalha apenas com dados discretos. Então, supondo que uma função $f(x)$ foi discretizada tomando n parcelas Δk separadamente, tem-se a sequência resultante $\{f(k_0), f(k_0 + \Delta k), \dots, f(k_0 + (n-1)\Delta k)\}$. Esta sequência permite expressar $f(x)$ como

$$f(x) = f(k_0 + k\Delta k),$$

na qual k toma os valores discretos $0, 1, \dots, n-1$.

Usando a discretização para $f(x)$ desenvolvida acima, a transformada de Fourier é representada por

$$F(\xi) = \frac{1}{n} \sum_{k=0}^{n-1} f(k) e^{\frac{-i\xi k}{n}} \quad (3.3)$$

para $\xi = 0, \dots, n-1$.

No caso de duas variáveis, de (3.3) segue que a transformada discreta de Fourier é dada pelas equações

$$F(\xi, \varphi) = \frac{1}{n^2} \sum_{k=0}^{n-1} \sum_{l=0}^{n-1} f(k, l) e^{\frac{-i(\xi k + \varphi l)}{n}}, \quad (3.4)$$

para $\xi = 0, \dots, n-1$ e $\varphi = 0, \dots, n-1$.

3.1.3. A transformada rápida de Fourier (FFT)

Como a transformada discreta de Fourier é uma ferramenta valiosa na análise de sinais, é desejável que haja um algoritmo eficiente para calculá-la. Para aproximar uma função por pontos, assim como para aproximar a integral de Fourier pela transformada discreta de Fourier, é preciso aplicar uma matriz cuja ordem n é o número de pontos.

Há n^2 operações envolvidas na multiplicação de uma matriz por um vetor, e conforme o número de pontos aumenta, há uma rápida ampliação no número de operações, tornando o custo computacional elevado. Contudo, se a função possui 2^j pontos espaçados uniformemente, então a matriz de Fourier pode ser fatorada em um produto de apenas algumas matrizes esparsas, e os fatores resultantes podem ser aplicados em um vetor, num total de $n \cdot \log_2 n$ operações

aritméticas [20]. Esta é a chamada transformada rápida de Fourier, ou *Fast Fourier Transform* (FFT).

3.2. A transformada de Gabor

Com a transformada de Gabor, uma janela de observação é deslocada no domínio de tempo e a transformada de Fourier é aplicada na porção visível da função. A função a ser transformada é multiplicada por uma função gaussiana g_α qualquer, que pode ser vista como uma janela, e a função resultante é então processada pela transformada de Fourier. A função “janela” indica que o sinal próximo ao tempo analisado terá maior consideração [21].

3.2.1. A transformada contínua de Gabor

Para qualquer valor fixo de $\alpha > 0$, a transformada de Gabor de uma função $f \in L^2(\mathbb{R})$ é definida por

$$G_b^\alpha[f(\varphi)] = \int_{-\infty}^{\infty} (e^{-i\varphi t} f(x)) g_\alpha(x-b) dx, \quad (3.7)$$

com a gaussiana g_α dada por

$$g_\alpha = \frac{1}{\sqrt{2\pi\alpha}} e^{-\frac{x^2}{4\alpha}},$$

ou seja, (3.7) localiza a transformada de Fourier de f em torno de $x = b$. O comprimento da janela é determinado pela constante positiva α (fixada) [22].

3.2.2. A transformada discreta de Gabor

Gabor introduziu [21] uma aproximação que caracteriza uma função temporal em tempo e frequência simultaneamente. Mais tarde, esta aproximação passou a ser conhecida como expansão de Gabor. Para um sinal $s(t)$, a expansão de Gabor é definida como

$$s(t) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} C_{m,n} h_{m,n}(t),$$

com

$$h_{m,n}(t) = h(t - mT)e^{jn\Omega t}.$$

Nesta expansão T e Ω representam o tempo e a amostragem de frequência, respectivamente.

Ainda que a expansão de Gabor tenha se mostrado muito útil para o processamento de sinais, suas aplicações foram limitadas pelas dificuldades associadas com a implementação computacional dos coeficientes de Gabor $C_{m,n}$. De acordo com o teorema de Balian-Low (dado no Apêndice A), $h_{m,n}(t)$ não forma uma base ortogonal; portanto, a seleção dos coeficientes $C_{m,n}$ em geral não é única [23].

Uma solução para este problema foi proposta por Martin Bastiaans [24], e constitui em introduzir uma função auxiliar $\gamma(t)$ e então calcular os coeficientes de Gabor utilizando o produto interno usual, projetando $s(t)$ sobre $\gamma(t)$.

Com base nos estudos de Bastiaans e estudos mais recentes, Shie Qian e Dapang Chen [23] apresentaram um conjunto de algoritmos para a transformada discreta de Gabor. Por exemplo, a expansão de Gabor discreta para sequências infinitas é dada por

$$\bar{s}(i) = \sum_{m=-L/\Delta M}^{Ls/\Delta M-1} \sum_{n=0}^{N-1} C_{m,n} \bar{h}_{m,n}(t),$$

$$C_{m,n} = \sum_{i=0}^{Ls-1} s(i) \gamma_{m,n}^{-*}(i),$$

em que

$$\bar{h}_{m,n}(t) = \bar{h}(i - m\Delta M) W_N^{ni},$$

$$\bar{\gamma}_{m,n}(t) = \bar{\gamma}(i - m\Delta M) W_N^{ni}.$$

Com esta expansão, a solução para os coeficientes de Gabor é dada por um sistema linear.

Peter Søndergaard [25] criou um algoritmo que também possibilita a análise e síntese de Gabor em multisiniais. Neste trabalho, a transformada discreta de Gabor é dada por

$$c(m, n, w) = \sum_{l=0}^{L-1} f(k, w) g(l - an) e^{-2\pi i m l / M},$$

com g sendo uma janela que localiza o sinal no tempo e na frequência. Então, o operador C_g é considerado como uma matriz, e as somas da definição são manipuladas diretamente. A vantagem deste método é de trabalhar com transformadas rápidas e curtas de Fourier, não requerendo multiplicação por exponenciais complexas. Este algoritmo possui um custo computacional baixo, comparado a outros algoritmos. Contudo, a implementação de tal método não é trivial para sinais com mais de uma dimensão.

Uma das soluções para a implementação da transformada discreta de Gabor é dada pelos *filtros de Gabor*, implementados nesta pesquisa. Os algoritmos utilizados são apresentados na Seção 7.3 e no Apêndice D.

3.2.3. Filtros de Gabor

Filtros de Gabor estão diretamente relacionados às *wavelets* de Gabor. Usualmente, é criado um banco de filtros de Gabor com várias escalas e rotações; em geral, expansões não são

aplicadas nas *wavelets* de Gabor, pois tal operação resulta em um grande custo computacional.

Os filtros são então convolucionados com o sinal, gerando um *espaço de Gabor*.

CAPÍTULO 4 – *WAVELETS*

Wavelets (termo proveniente da palavra francesa *ondeletes*) são ondas pequenas com determinadas propriedades que as tornam adequadas a servirem de base para decomposição de outras funções, da mesma forma que pela teoria de Fourier um sinal pode ser expresso como uma soma, possivelmente infinita, de senos e cossenos.

A grande desvantagem da expansão de Fourier é que não existe resolução de tempo, apenas de frequência. Isto significa que embora possamos determinar todas as frequências presentes em um sinal, não se sabe onde elas estão presentes. Para solucionar este problema várias soluções foram desenvolvidas nas décadas passadas. As *wavelets* vieram como uma extensão natural na teoria de Fourier, representando um sinal conjuntamente nos domínios de tempo e frequência.

A teoria *wavelet* foi estruturada na década de 1980, ainda que suas ideias tivessem sido propostas por Alfred Haar, em 1909. As *wavelets* de Haar ficaram no anonimato por muitos anos e, por um período muito longo, foram a única base ortonormal de *wavelets* conhecida. Em 1985, Stéphane Mallat deu à teoria um grande impulso, através de seu trabalho em processamento digital de imagens. Inspirado neste trabalho, Yves Meyer construiu a primeira *wavelet* não-trivial (suave) [26].

Poucos anos mais tarde, Ingrid Daubechies [27] fez uso dos trabalhos de Mallat para construir um conjunto de bases ortonormais de *wavelets* suaves, com suportes compactos. Os trabalhos de Daubechies são os alicerces das aplicações atuais de *wavelets*[26].

A transformada *wavelet* pode ser vista como um mecanismo para decompor sinais, permitindo analisar os dados em diferentes domínios de frequências com a resolução de cada componente ligada à sua escala. As *wavelets* são lineares e invariantes sob transformações de translação e dilatação, ou seja, é possível construir *wavelets* usando translações e dilatações sobre funções [18].

Por causa de suas propriedades únicas, custo computacional relativamente baixo e também pela simplicidade desta ferramenta, as *wavelets* são preferidas em lugar de técnicas mais complicadas e onerosas, sendo utilizadas na Análise Funcional[28], no estudo de propriedades multifractais[29], singularidades[30] ou oscilações[31] locais de funções, em soluções de equações diferenciais[32], reconhecimento de padrões, compressão de imagens[33] e sons[34], processamento de geometria digital[35], na solução de vários problemas de Física, Biologia, Medicina, Astronomia, Engenharia Nuclear, entre várias outras áreas.

4.1. A transformada *wavelet* contínua (CWT)

A transformada *wavelet* contínua (*continuous wavelet transform*, CWT) de uma série temporal f é definida como a transformada

$$\Phi_f^\psi(a, b) = \int_{-\infty}^{\infty} f(\xi) \overline{\psi_{a,b}(\xi)} d\xi, \quad (4.1)$$

em que $a > 0$, e

$$\psi_{a,b}(\xi) = \frac{1}{\sqrt{a}} \psi\left(\frac{\xi - b}{a}\right) \quad (4.2)$$

representa uma família de funções *wavelet* escolhida, chamada *wavelet*-mãe.

O parâmetro a diz respeito à escala, b é um parâmetro de translação ou localização da *wavelet*-mãe ψ , e $\bar{\psi}_{a,b}(\xi)$ é o conjugado complexo de $\psi_{a,b}(\xi)$. A variação de a causa uma dilatação, quando $a > 1$, e uma contração, quando $a < 1$. Conforme b varia, a função f é localmente analisada nas vizinhanças deste ponto. Pode-se dizer que a CWT é equivalente a um microscópio matemático, cuja aproximação é dada pelo inverso do parâmetro de dilatação.

4.2. A transformada *wavelet* discreta (DWT)

Na derivação CWT, muito da informação sobre escalas ou tempos próximos é redundante, o que resulta em custo computacional elevado. Em alguns casos, este problema pode ser resolvido com o uso de uma transformada *wavelet* discreta (*discrete wavelet transform*, DWT), que utiliza valores discretos de escala e localização. Para evitar redundâncias, podem ser usadas funções *wavelet* que formam uma base ortogonal e definir a DWT como

$$\Phi_{f_k} = \int_{-\infty}^{+\infty} f(\xi) \bar{\psi}_k^{-j}(\xi) d\xi, \quad (4.3)$$

em que

$$\psi_k^j(\xi) = 2^{-\frac{j}{2}} \psi(2^{-j} \xi - k).$$

Na função ψ acima j e k são, respectivamente, valores discretos de escala e localização. Tais conjuntos de funções *wavelet* são ortogonais e suas respectivas funções são transladadas e dilatadas.

Os sinais $f(\xi)$ são representados por séries, como dado abaixo:

$$f(\xi) = \sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} d_k^j \psi_k^j(\xi), \quad (4.4)$$

em que $\psi_k^j(\xi) = \psi(2^j \xi - k)$ são funções *wavelet* e d_k^j são coeficientes *wavelet* dados por

$$d_k^j = \int f(u)\psi_k^j(u)du.$$

É possível mostrar que a amplitude dos coeficientes *wavelet* é associada com variações abruptas de sinal, ou “detalhes” de maior frequência. A transformada de Fourier não é capaz de reconhecer estas regiões de transição. Na análise de Fourier, somente a presença das frequências envolvidas é detectada, sem fornecer informações sobre a localização espacial destas.

4.3. A análise multirresolução

As primeiras construções de bases ortonormais das *wavelets* pareciam um pouco “miraculosas”. A situação mudou com o advento da análise multirresolução, criada por Mallat e Meyer, que pode ser vista como uma estrutura de dados que produz uma representação sucessivamente condensada da informação em uma imagem. A ideia de observar sinais e analisá-los em várias escalas foi muito bem recebida no campo da Visão Computacional, onde a vantagem mais óbvia desta abordagem é a possibilidade de reduzir o custo computacional de várias operações em imagens[36].

A análise multirresolução provê uma estrutura natural para o entendimento das bases *wavelet*, bem como permite a construção de novos exemplos [27]. Ela consiste de uma sequência de espaços sucessivos de aproximação V_j , que satisfazem

$$\dots \subset V_{-2} \subset V_{-1} \subset V_0 \subset V_1 \subset V_2 \subset \dots$$

com

$$\overline{\bigcup_{j \in \mathbb{Z}} V_j} = L^2(\mathfrak{R}),$$

$$\bigcap_{j \in \mathbb{Z}} V_j = \{0\}.$$

A construção dos espaços V_j , assim como sua relação com as *wavelets*, é dada abaixo. O desenvolvimento aqui é o mesmo construído por Alfred Boggess e Francis Narcowich [37], que começa com o estudo da *wavelet* de Haar e estende tal estudo para situações mais gerais.

Os gráficos das funções-escala e *wavelets*-mãe apresentados nas seções seguintes foram obtidos utilizando o algoritmo cascata de Mallat em conjunto com o programa Gnuplot. Um tutorial para a sua obtenção é dado no Apêndice C.

4.3.1. Análise das *wavelets* de Haar

As funções ψ (*wavelet*-mãe) e φ (função escala) geram uma família de funções que podem ser usadas para dividir ou reconstruir um sinal. A análise *wavelet* mais simples é baseada na função escala de Haar, onde os blocos são dilatações e translações da função escala básica, dada pela expressão

$$\varphi(x) = \begin{cases} 1, & x \in [0,1) \\ 0, & x \in (-\infty,0) \cup [1,\infty) \end{cases}. \quad (4.5)$$

O gráfico da expressão (4.5) é dado na Figura 4.1.

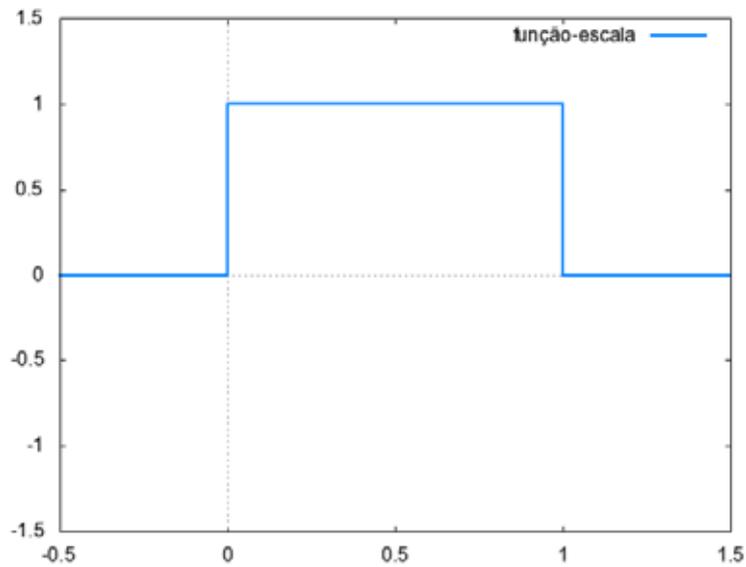


Figura 4.1 – A função escala φ de Haar.

Seja V_0 o espaço de todas as funções da forma

$$\sum_{k \in \mathbb{Z}} a_k \varphi(x-k), a_k \in \mathbb{R}.$$

Como k se estende por um conjunto finito, cada elemento de V_0 é zero fora de um conjunto limitado. Uma função dessa forma é dita de suporte compacto ou finito.

Suponha j um inteiro não-negativo. O espaço das funções no nível j , denotado por V_j , é definido como o espaço gerado pelo conjunto

$$\{\dots, \varphi(2^j x+1), \varphi(2^j x), \varphi(2^j x-1), \varphi(2^j x-2), \dots\} \text{ em } \mathbb{R}.$$

V_j é o espaço de funções seccionalmente constantes de suporte compacto cujas descontinuidades estão contidas no conjunto

$$\{\dots, -\frac{1}{2^j}, 0, \frac{1}{2^j}, \frac{2}{2^j}, \frac{3}{2^j}, \dots\}.$$

Uma função em V_0 é uma função seccionalmente constante de suporte finito cujas descontinuidades estão contidas no conjunto dos inteiros. Qualquer função em V_0 está também

em V_I , que consiste de funções seccionalmente contínuas cujas descontinuidades estão contidas no conjunto $\{\dots, -1/2, 0, 1/2, 1, 3/2, \dots\}$. O mesmo vale para $V_1 \subset V_2$, e assim por diante:

$$V_0 \subset V_1 \subset V_2 \subset \dots \subset V_j \subset V_{j+1} \subset \dots$$

É desejável ter um algoritmo eficiente para decompor um sinal em suas componentes relativas a V_j . Um meio de realizar eficientemente essa decomposição é construir uma base ortonormal para V_j , usando o produto interno do espaço das funções quadrado-integráveis, L^2 .

Para a construção de tal base, o espaço V_0 será analisado. Este espaço é gerado por φ e suas translações. As funções $\varphi(x-k)$ têm norma unitária em L^2 , isto é

$$\|\varphi(x-k)\|_{L^2}^2 = \int_{-\infty}^{\infty} \varphi(x-k)^2 dx = \int_k^{k+1} dx = 1.$$

Se $j \neq k$, $\varphi(x-j)$ e $\varphi(x-k)$ têm suportes disjuntos. Então,

$$\langle \varphi(x-j), \varphi(x-k) \rangle_{L^2} = \int_{-\infty}^{\infty} \varphi(x-j)\varphi(x-k) dx = 0, j \neq k,$$

e o conjunto $\{\varphi(x-k), k \in Z\}$ é uma base ortonormal para V_0 .

De uma forma mais geral, pode-se dizer que o conjunto de funções $\{2^{j/2}\varphi(2^j x-k); k \in Z\}$ é uma base ortonormal para V_j . O fator $2^{j/2}$ surge pelo fato de que

$$\int_{-\infty}^{\infty} (\varphi(2^j x))^2 dx = 1/2^j.$$

Além da base ortonormal para V_j , é necessário isolar os “saltos” que pertencem a V_j mas não a V_{j-1} . Para isso é utilizada a função ψ .

A ideia é decompor V_j como soma ortogonal de V_{j-1} e seu complemento. A construção se inicia com $i = I$.

Como V_0 é gerado por φ e suas translações, é razoável supor que o complemento ortogonal seja gerado pelas translações de alguma função ψ . Há dois fatores principais para a construção de tal função, a saber:

- ψ está em V_1 , podendo ser expressa como

$$\psi(x) = \sum_l a_l \varphi(2x-l),$$

para alguns escalares a_l , com um número finito de a_l não-nulos. Esta expressão é válida, pois $\varphi(2x) \in V_1$, e $\varphi(2x-1) \in V_1$.

- ψ é ortogonal a V_0 , ou seja,

$$\int \psi(x)\varphi(x-k)dx = 0,$$

para todo k inteiro.

A função ψ mais simples que atende ambas as condições é a função dada pelo gráfico da Figura 4.2.

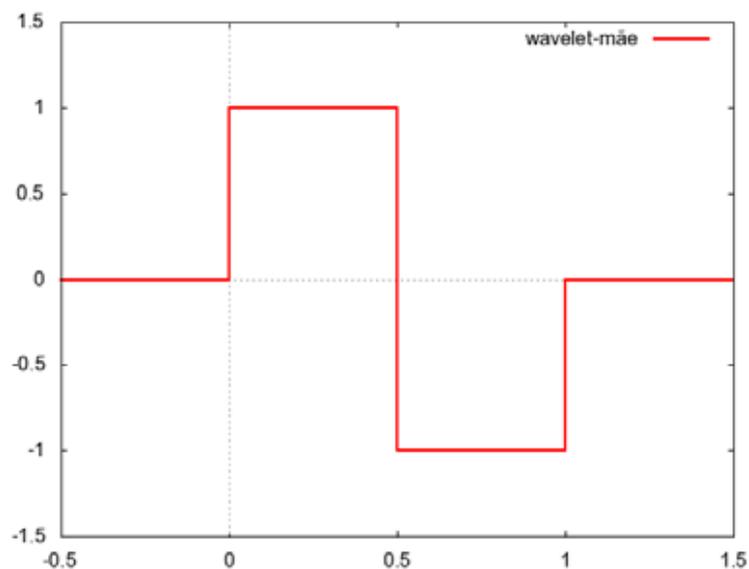


Figura 4.2 – A *wavelet*-mãe de Haar, ψ .

Esta função pode ser escrita por

$$\psi(x) = \varphi(2x) - \varphi\left(2\left(x - \frac{1}{2}\right)\right) = \varphi(2x) - \varphi(2x - 1),$$

satisfazendo a condição (1 ψ). Além disso, para $k = 0$, tem-se que $\int_{-\infty}^{\infty} \psi(x)\varphi(x) = 0$. Para $k \neq 0$,

tem-se que os suportes de ψ e $\varphi(x-k)$ não se sobrepõem; então, $\int_{-\infty}^{\infty} \psi(x)\varphi(x-k) = 0$. Portanto, ψ pertence a V_1 e é ortogonal a V_0 .

Finalmente, ψ é chamada *wavelet* de Haar, e é dada pela função

$$\psi(x) = \varphi(2x) - \varphi(2x - 1).$$

A DWT usando a *wavelet* de Haar detecta variações abruptas de sinal, isto é, ela consegue detectar características de localização no espaço físico. Todavia, existem limitações no uso desta *wavelet*. Como as funções base de Haar são funções de passos descontínuos, não são muito adequadas para a análise de funções estáveis com derivações contínuas. Visto que as imagens geralmente contêm regiões estáveis, a transformada *wavelet* de Haar não provê resultados satisfatórios em muitas aplicações com imagens [38].

4.3.2. A definição de análise multirresolução

O procedimento construído para a *wavelet* de Haar resulta em duas funções: a função escala φ e a *wavelet*-mãe de Haar, ψ . Ambas as funções são simples, e conduzem a um algoritmo de decomposição elementar.

A análise multirresolução possui construção similar, com versões contínuas de φ e ψ . Esta modificação generaliza a definição dada com as *wavelets* de Haar na seção anterior. A definição geral de análise multirresolução é dada a seguir.

Definição (análise multirresolução). Seja V_j , $j = \dots, -2, -1, 0, 1, 2, \dots$ uma sequência de subespaços de funções em L^2 . A coleção de espaços $\{V_j, j \in \mathbb{Z}\}$ é chamada de *análise multirresolução*, com φ sua função escala, se:

$$- V_j \subset V_{j+1};$$

$$- \overline{\cup V_j} = L^2(\mathbb{R});$$

$$- \cap V_j = \{0\};$$

$$- f(x) \in V_j \Leftrightarrow f(2^{-j}x) \in V_0;$$

- $\varphi \in V_0$ e $\{\varphi(x-k), k \in \mathbb{Z}\}$ é base ortonormal para V_0 , considerando o produto interno de L^2 .

Dizer que $V_j \subset V_{j+1}$ significa que informações, ou “detalhes”, são adicionadas ao passar do nível de resolução j (ou escala 2^j) para o nível $j+1$ (escala 2^{j+1}) [39]. Em suma, a análise multirresolução consiste de uma sequência de *espaços de aproximação* sucessivos V_j [27], e permite obter uma interpretação invariante de escala de uma imagem.

4.3.3. As *wavelets* de Daubechies

As *wavelets* de Haar possuem grandes desvantagens; por exemplo, sua descontinuidade, ainda que possuam suporte compacto. Esta *wavelet*, juntamente com a *wavelet* de Shannon (em referência ao matemático e engenheiro elétrico americano Claude Shannon), eram as únicas opções disponíveis antes de Daubechies descobrir a hierarquia de *wavelets* que leva seu nome.

Conforme se avança na hierarquia de Daubechies, suas *wavelets* vão se tornando mais suaves, ou seja, possuem um número maior de derivadas contínuas. A mais simples destas

wavelets é a de Haar, a única descontínua; as outras são contínuas e de suporte compacto. Tal propriedade pode ser analisada para satisfazer condições em uma aplicação particular.

As *wavelets*-mãe propostas por Daubechies [27] possuem, para $r \in Z$, a base ortonormal para $L^2(\mathfrak{R})$ definida por

$$\varphi_{r,j,k}(x) = 2^{j/2} \varphi_r(2^j x - k), j, k \in Z,$$

onde a função $\varphi_r(x)$ em $L^2(\mathfrak{R})$ satisfaz: $\varphi_r(x - k) \mid k \in Z$ é uma base sequencial ortonormal em $L^2(\mathfrak{R})$. Aqui, j é o índice da escala, k é o índice de translação e r é o número de momentos nulos.

Cada *wavelet* possui um número de momentos nulos igual à metade do número de coeficientes. Um momento nulo limita a habilidade da *wavelet* de representar informação em um sinal, expressa como comportamento polinomial. Por exemplo, D2 (a *wavelet* de Haar), com um momento nulo, facilmente codifica polinômios de um coeficiente (componentes de sinal constantes). Logo, D4 codifica polinômios com dois coeficientes (componentes de sinal constante e linear) e D6 codifica com facilidade polinômios com três coeficientes (componentes constante, linear e quadrática). Engenheiros e matemáticos se referem de forma diferente às *wavelets* de Daubechies; engenheiros consideram o número de coeficientes (D2, D4, D6), enquanto matemáticos consideram a ordem de aproximação local, ou momentos nulos (ordem 1, ordem 2, ordem 3).

Algumas das *wavelets*-mãe de Daubechies e suas funções-escala, geradas com diferentes coeficientes, são dadas na Figura 4.3.

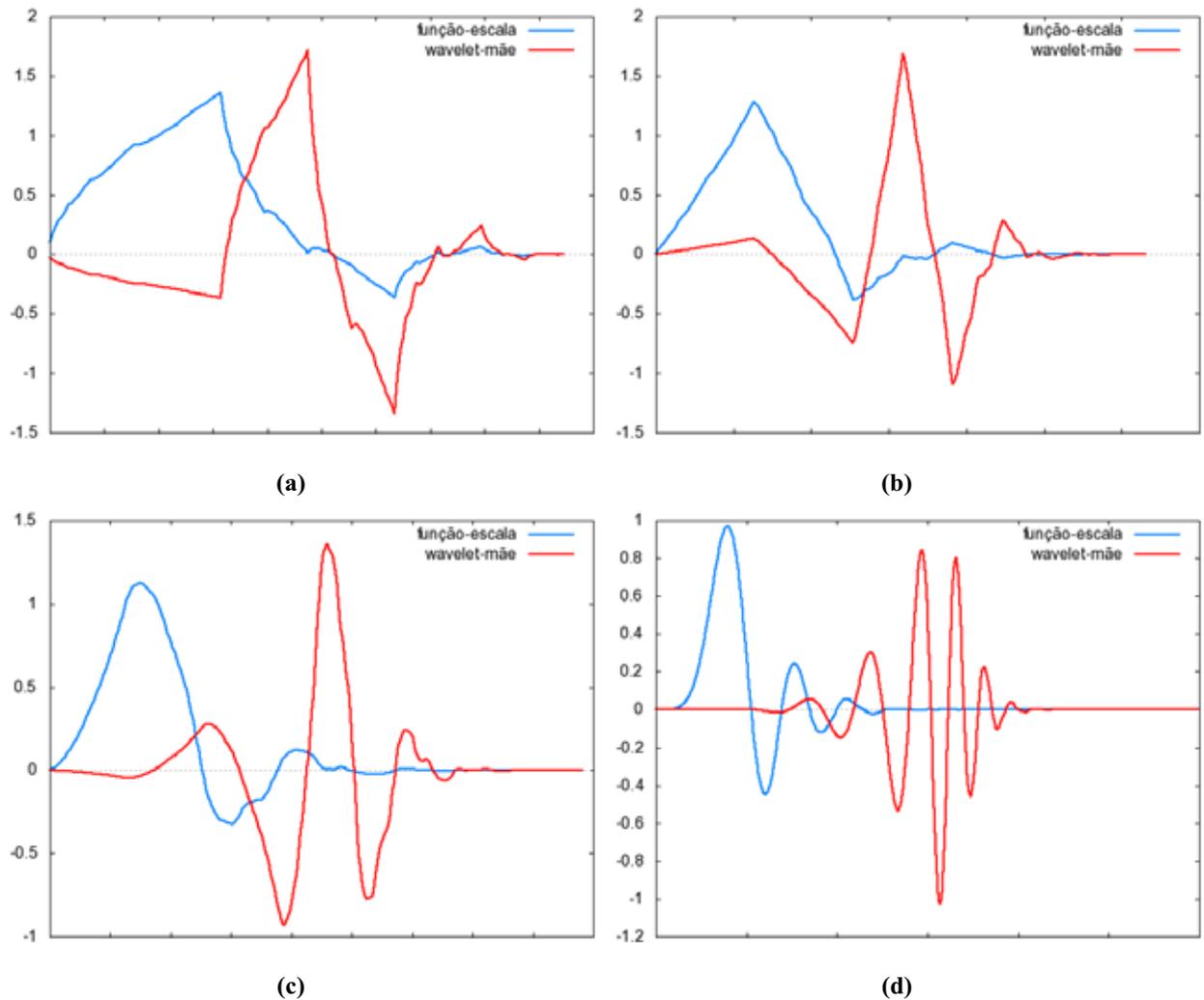


Figura 4.3 – Funções-escala φ e *wavelets*-mãe ψ de Daubechies, para (a) $r = 2$ (D4); (b) $r = 3$ (D6); (c) $r = 4$ (D8); (d) $r = 10$ (D20).

4.4. O algoritmo piramidal de Mallat

Um método rápido e confiável de implementação da transformada *wavelet* discreta foi desenvolvido por Mallat em 1989, e é conhecido como algoritmo piramidal de Mallat. A análise multirresolução com o uso deste algoritmo visa obter aproximações (representações de baixa frequência do sinal original) e detalhes (diferença entre duas aproximações sucessivas do

sinal original). Uma aproximação mantém a propensão do sinal original, enquanto um detalhe exhibe os componentes de alta frequência do mesmo [40].

O algoritmo piramidal se constitui dos cálculos da DWT e da transformada *wavelet* inversa. A multirresolução é obtida com a utilização de um banco de filtros composto por filtros passa-baixa e passa-alta de decomposição (L e H) e por filtros passa-baixa e passa-alta de reconstrução (L' e H').

A partir de um sinal S, dois conjuntos de coeficientes podem ser produzidos, com o auxílio da DWT: os coeficientes de aproximação, cA1, e os coeficientes de detalhe cD1. Tais coeficientes são obtidos através da convolução de S com os filtros passa-baixa L e passa-alta H, para aproximação e detalhe, respectivamente. Em seguida é aplicada a *decimação diádica*, ou seja, a cada duas saídas do filtro uma é descartada.

Um processo de decomposição de múltiplos níveis é conseguido como na Figura 4.4, onde as aproximações sucessivas são decompostas de modo que S é dividido em várias componentes de resolução mais baixa [40]. Uma função para a aplicação do algoritmo cascata de Mallat é exposta no Apêndice C.

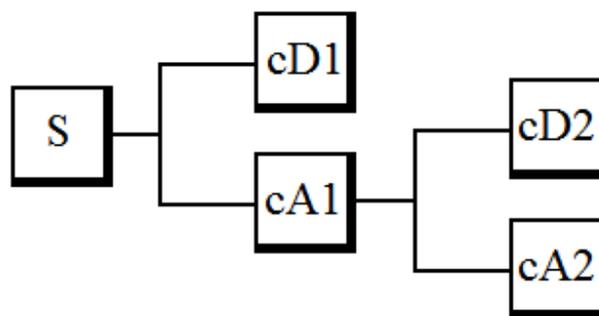


Figura 4.4 – O processo de decomposição em múltiplos níveis esquematizado (onde $S = cA2 + cD2 + cD1$).

O algoritmo piramidal produz os coeficientes *wavelet* em n operações. Como visto no Capítulo 3, a transformada rápida de Fourier calcula a transformada discreta de Fourier em $n \cdot \log_2 n$ operações. As *wavelets* têm sido promovidas como sendo “mais rápidas que a FFT”, mas deve-se ter em mente que as transformadas discretas Fourier e *wavelet* retornam informações muito diferentes [41].

CAPÍTULO 5 – ESTATÍSTICA E VISÃO COMPUTACIONAL

O princípio básico da estatística é auxiliar na avaliação de relações entre variáveis, se utilizando de teorias probabilísticas para explicar a frequência de fenômenos e para possibilitar a previsão desses fenômenos no futuro. Por outro lado, a visão computacional desenvolve teorias e tecnologias para a construção de sistemas que obtém informação de imagens ou quaisquer dados multidimensionais.

O problema clássico da visão computacional e do processamento de imagens é determinar se uma imagem contém ou não um objeto ou uma característica. Tal tarefa pode ser resolvida de forma robusta e sem esforço humano, mas ainda não foi resolvida satisfatoriamente para objetos arbitrários em situações arbitrárias.

Neste capítulo será abordada a operação de correlação, assim como a técnica de *template matching*, ou emparelhamento.

5.1. Correlação entre imagens

A correlação é a medida do grau de associação de duas variáveis, não necessariamente em seu valor, mas em um comportamento geral. Para se obter a medida do grau de associação entre duas variáveis, o coeficiente de correlação é utilizado. Este coeficiente é definido como

$$r = \frac{S_{xy}}{S_x S_y},$$

onde S_{xy} é a covariância entre x e y :

$$s_{xy} = \frac{\sum x_i y_i - n\bar{x}\bar{y}}{n-1}.$$

O valor da correlação está contido no intervalo $[-1, 1]$. Uma correlação positiva indica que as duas variáveis se movem juntas, e quanto mais a correlação se aproxima de um, maior é a relação entre elas; semelhantemente, uma correlação negativa indica que as duas variáveis movem-se em direções opostas, e que a relação é maior quanto mais próxima de -1. Uma correlação próxima a zero indica que as duas variáveis não estão relacionadas.

Duas variáveis que estão perfeitamente correlacionadas positivamente ($r = 1$) movem-se essencialmente em perfeita proporção na mesma direção, enquanto dois conjuntos que estão perfeitamente correlacionados negativamente movem-se em perfeita proporção em direções opostas.

A melhor forma de se estudar a correlação entre duas variáveis é o diagrama de dispersão, também chamado de *scatterplot*. Este diagrama é utilizado para correlacionar dados, como a influência de um fator sobre uma propriedade, ou dados obtidos de maneiras diversas. Na Figura 5.1 são dados os três casos para o diagrama de dispersão com correlação positiva.

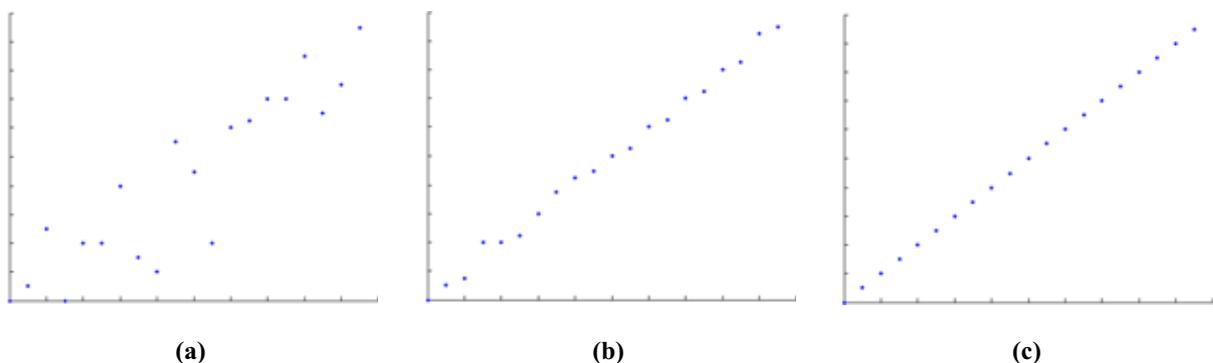


Figura 5.1 – Os casos gerais de diagrama de dispersão para correlação positiva: (a) correlação fraca; (b) correlação forte; (c) correlação perfeita.

De forma semelhante, na Figura 5.2 são dados os três casos para o diagrama de dispersão com correlação negativa.

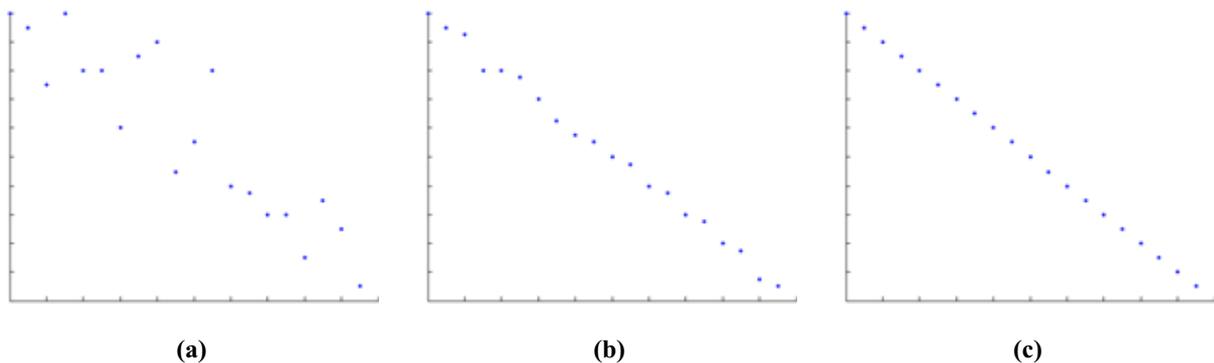


Figura 5.2 – Os casos gerais de diagrama de dispersão para correlação negativa: (a) correlação fraca; (b) correlação forte; (c) correlação perfeita.

5.2. A técnica de emparelhamento (*Template matching*)

Um dos problemas frequentes no processamento de imagens é determinar a posição de certo padrão dado em uma imagem, ou em parte dela. A porção de importância na imagem recebe o nome de *região de interesse*. Este problema possui seu correspondente no processamento de sinais, que consiste na determinação de um sinal digital recebido utilizando, por exemplo, um filtro adaptado[42].

A técnica de emparelhamento se refere ao processo de detectar um objeto de certo tamanho, forma e orientação em uma imagem, aplicando um operador em uma região semelhante do objeto[43]. Tal objeto recebe o nome de *modelo* (de *template*), e a imagem hospedeira é denominada *imagem-alvo* (*target*).

Uma das mais estudadas abordagens para o embasamento da técnica de emparelhamento é a *correlação cruzada* (*cross-correlation*), e seu uso nesta técnica foi motivado pela distância euclidiana quadrada:

$$d_{f,t}^2(u, v) = \sum_{x,y} [f(x, y) - t(x - u, y - v)]^2,$$

onde f é a imagem e a soma é sob x, y , na janela contendo o atributo t , posicionada em u, v .

Na expansão de d^2 ,

$$d_{f,t}^2(u, v) = \sum_{x,y} [f^2(x, y) - 2f(x, y)t(x - u, y - v) + t^2(x - u, y - v)],$$

o termo $\sum t^2(x - u, y - v)$ é aproximadamente constante. Logo, o termo de correlação cruzada restante,

$$c(u, v) = \sum_{x,y} f(x, y)t(x - u, y - v),$$

é uma medida da similaridade entre a imagem e o atributo[44].

CAPÍTULO 6 – O *SOFTWARE* “WaveFPR”

A fim de estudar as imagens de microscopia em questão, foi desenvolvido um *software* que reúne todas as ferramentas utilizadas (os algoritmos gerados para as transformadas de Fourier, Gabor e *wavelets* de Haar e Daubechies, para segmentação das imagens e também os relativos à técnica de emparelhamento), unindo praticidade e rapidez em uma ferramenta de fácil utilização.

O *software* em questão, denominado WaveFPR (*wavelets and Fourier transforms for pattern recognition*), foi desenvolvido na plataforma Matlab, e é o resultado do trabalho voltado à implementação dos algoritmos referentes às transformadas e técnicas de processamento de imagens citadas ao longo do texto. Com WaveFPR, o usuário pode inserir uma imagem de entrada e, após a segmentação automática desta, aplicar as transformadas disponíveis, a saber: Fourier, Gabor, e *wavelets* de Haar e Daubechies.

Após a aplicação das transformadas selecionadas, o programa retorna a matriz de coeficientes que diz respeito à transformada escolhida e uma visualização desta matriz em duas dimensões, que é a projeção dos valores do espectro relativo à transformada em questão no plano xy . Há também a opção para exibição do espectro referente à transformada escolhida pelo usuário.

O conjunto de coeficientes e gráficos é gerado na janela de sua transformada, onde o usuário também tem a opção de gravar em disco a matriz de coeficientes e os dois gráficos gerados, para posterior comparação.

Uma imagem da janela principal do programa é dada na Figura 6.1.

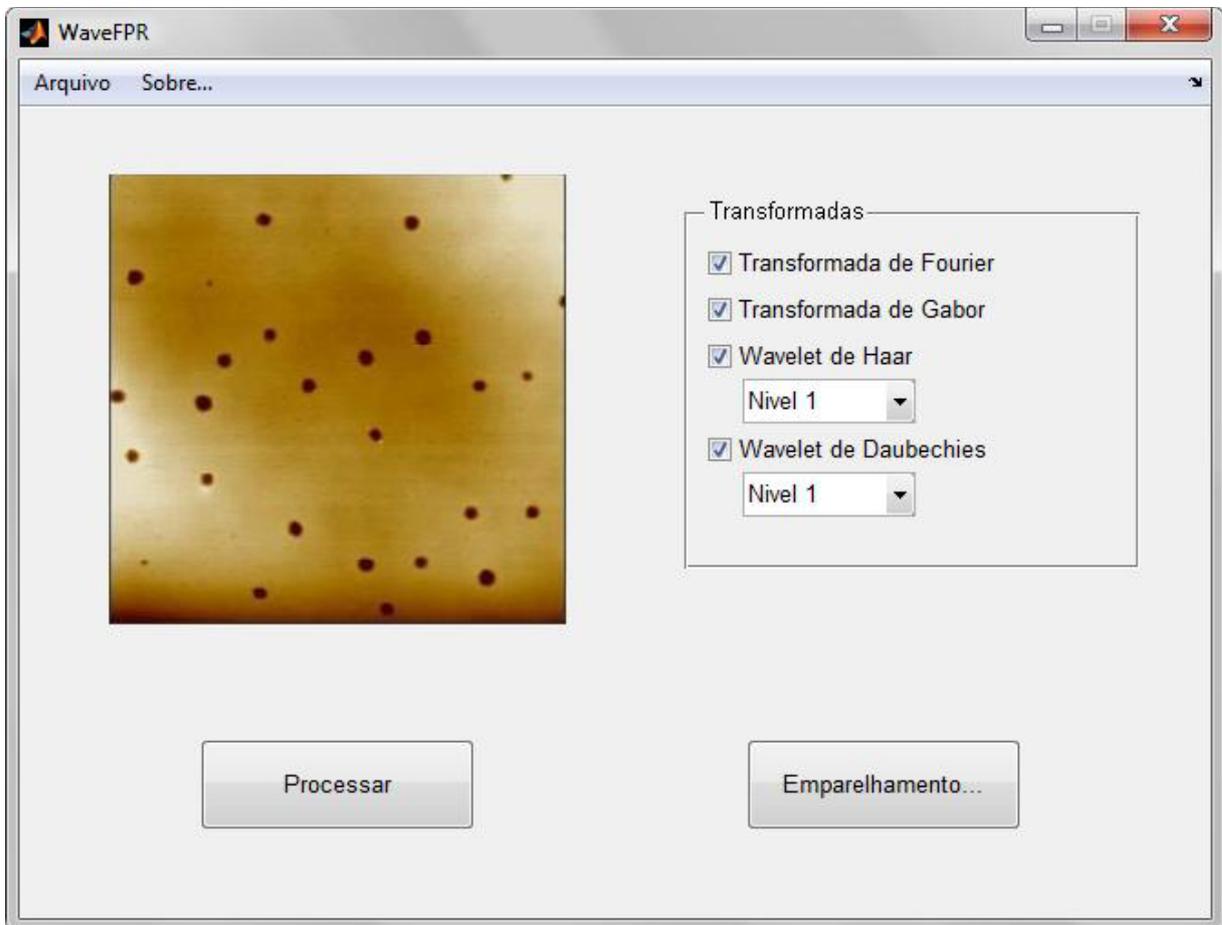


Figura 6.1 – A interface principal do programa “WaveFPR”.

Na janela principal o programa traz uma barra de menus (“Arquivo” e “Sobre...”), uma imagem previamente escolhida pelo usuário, caixas de seleção referentes às transformadas e dois botões, o botão “Processar” e o botão “Emparelhamento...”. O nível de decomposição das *wavelets* de Haar e Daubechies pode ser escolhido, utilizando-se o menu *pop-up* correspondente, habilitado quando a caixa de seleção da *wavelet* de interesse é marcada.

Marcando as caixas de seleção, o usuário pode escolher qual transformada será aplicada na imagem selecionada. O processamento é feito na imagem de entrada após o botão

“Processar” ser clicado. Depois do clique, o programa executa o pré-processamento, e abre as janelas relativas às transformadas selecionadas anteriormente.

O programa aplica as transformadas na imagem de entrada de acordo com as opções selecionadas pelo usuário na tela inicial. Para cada transformada é apresentado seu espectro, uma projeção dos valores do espectro em duas dimensões e uma matriz de pontos. As janelas abertas pelo usuário para cada transformada são discutidas nas seções seguintes.

6.1. A janela relativa à transformada de Fourier

Clicando no botão “Processar” com a caixa de seleção “Transformada de Fourier” marcada, na tela principal do WaveFPR (Figura 6.1), é aberta uma janela, de nome “Transformada de Fourier”. A janela relativa a esta opção exibe a projeção do espectro de Fourier no plano xy e a matriz de coeficientes que gera o espectro. Também há botões para a gravação da imagem bidimensional, da matriz de dados e para a exibição do espectro de Fourier relativo à imagem. Esta janela é exibida na Figura 6.2.

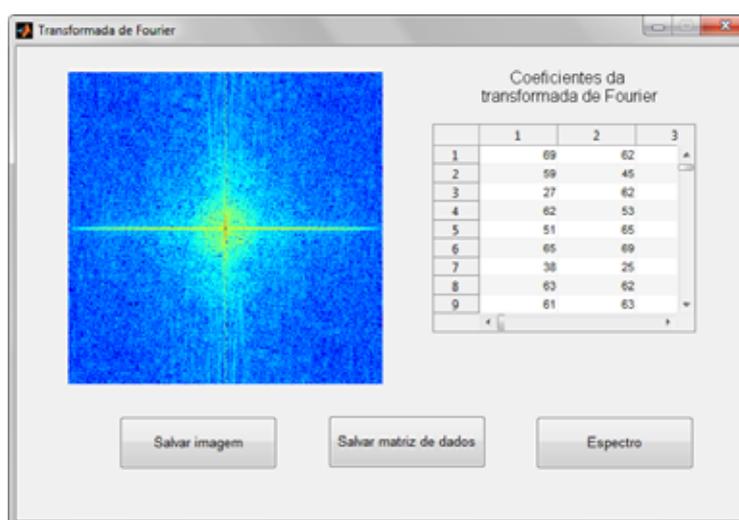


Figura 6.2 – A janela relativa à opção “Transformada de Fourier” e seus resultados.

6.2. A janela relativa à transformada de Gabor

Clicando no botão “Processar” com a caixa de seleção “Transformada de Gabor” marcada, é aberta a janela de nome “Transformada de Gabor”. A janela relativa a esta opção exibe a projeção do espectro da transformada de Gabor no plano xy e a matriz de coeficientes que gera o espectro. Também há botões para a gravação da imagem bidimensional, da matriz de dados e para a exibição do espectro da transformada de Gabor relativo à imagem. Esta janela é exibida na Figura 6.3.

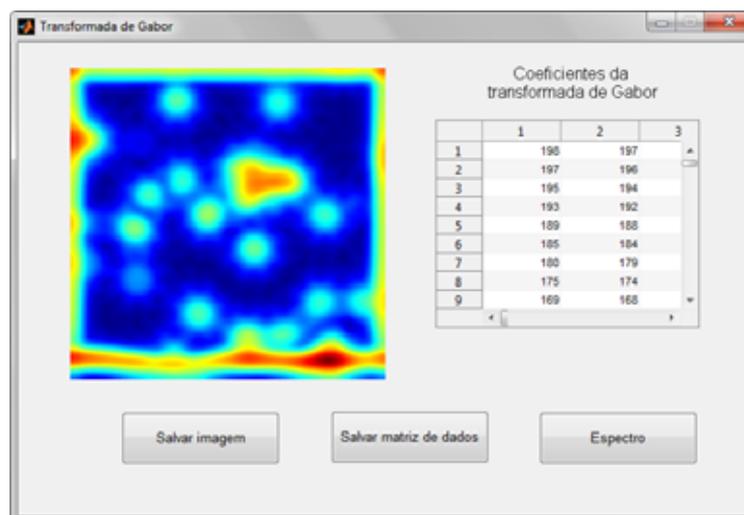


Figura 6.3 – A janela relativa à opção “Transformada de Gabor” e seus resultados.

6.3. A janela relativa à *wavelet* de Haar

Clicando no botão “Processar” com a caixa de seleção “*Wavelet* de Haar” marcada, é aberta a janela de nome “*Wavelet* de Haar”, com o nível de decomposição escolhido pelo usuário na tela inicial do *software*. A janela relativa a esta opção exibe a projeção do espectro da *wavelet*

de Haar no plano xy e a matriz de coeficientes que gera o espectro. Também há botões para a gravação da imagem bidimensional, da matriz de dados e para a exibição do espectro da *wavelet* de Haar relativo à imagem. Esta janela é exibida na Figura 6.4.

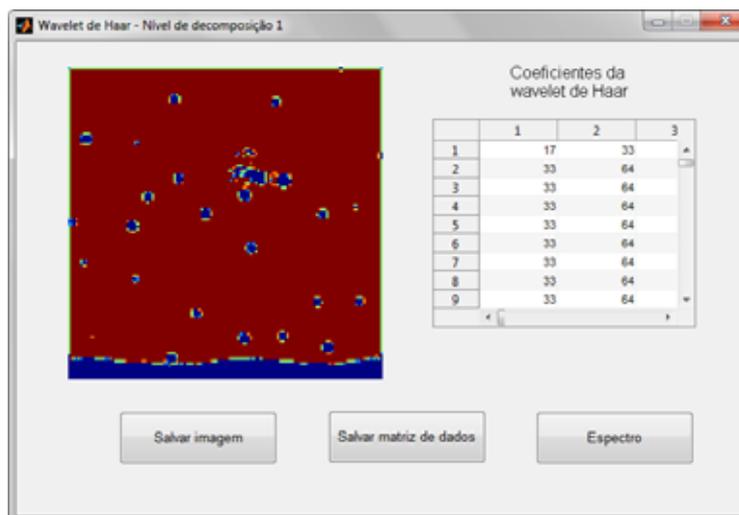


Figura 6.4 – A janela relativa à opção “*Wavelet* de Haar” para o nível de decomposição 1 e seus resultados.

6.4. A janela relativa à *wavelet* de Daubechies

Clicando no botão “Processar” com a caixa de seleção “*Wavelet* de Daubechies” marcada, é aberta a janela de nome “*Wavelet* de Daubechies”, com o nível de decomposição escolhido pelo usuário na tela inicial do *software*. A janela relativa a esta opção, da mesma forma que as outras janelas mostradas pelo programa, exibe a projeção do espectro da *wavelet* de Daubechies no plano xy e a matriz de coeficientes que gera o espectro. Também há botões para a gravação da imagem bidimensional, da matriz de dados e para a exibição do espectro da *wavelet* de Daubechies relativo à imagem. Esta janela é exibida na Figura 6.5.

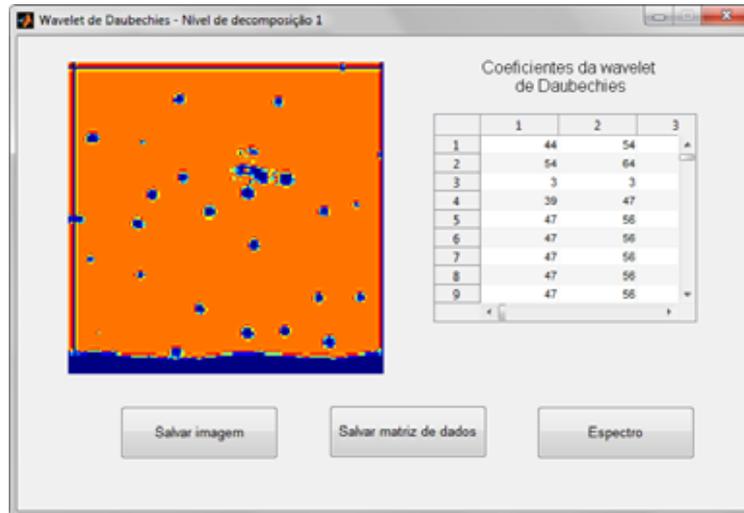


Figura 6.5 – A janela relativa à opção “*Wavelet* de Daubechies” para o nível de decomposição 1 e seus resultados.

6.5. A janela que fornece a técnica de emparelhamento

Clicando no botão “Emparelhamento...” na janela inicial do software, é aberta a janela de nome “Emparelhamento (Template Matching)”. Esta janela possui o botão “Processar”, que quando pressionado, solicita ao usuário as imagens modelo e alvo para a aplicação do algoritmo de emparelhamento.

Após a escolha das imagens modelo e alvo, é aplicado o algoritmo de emparelhamento, que retorna um sinal na imagem alvo como resultado. As imagens escolhidas pelo usuário são mostradas à esquerda na janela, enquanto o resultado é mostrado à direita. Esta janela é exibida na Figura 6.6.

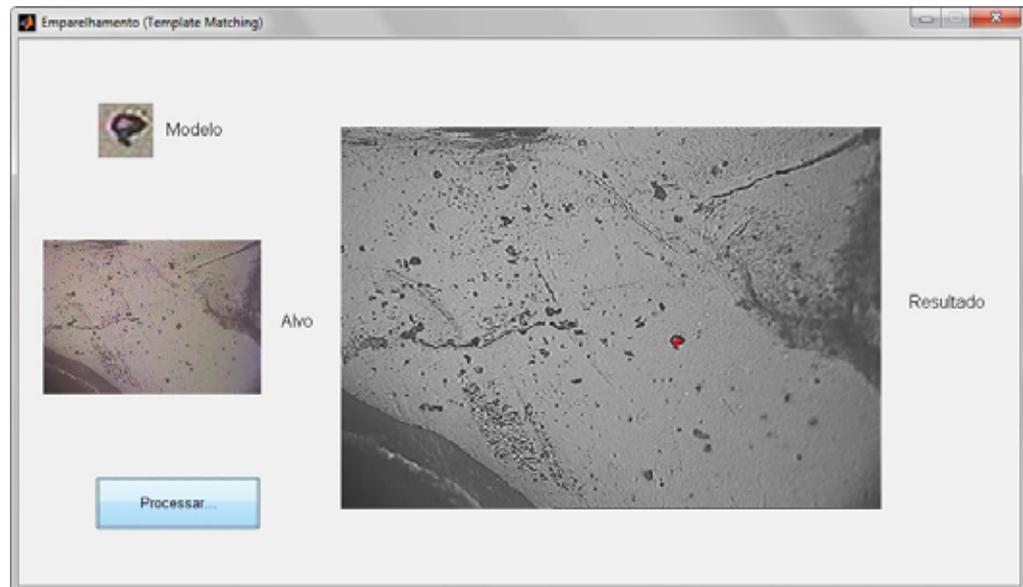


Figura 6.6 – A janela relativa à opção “Emparelhamento...” e seus resultados.

CAPÍTULO 7 – RESULTADOS E DISCUSSÃO

Os resultados apresentados mostram o funcionamento dos algoritmos implementados, assim como a análise de algumas imagens de teste. Estas imagens, em sua maioria, são provenientes de membranas de borracha natural com nanopartículas metálicas de ouro.

As seções deste capítulo tratam do pré-processamento da imagem de entrada, da implementação dos algoritmos referentes às transformadas e do algoritmo da técnica de emparelhamento. Finalizando, há os testes realizados com os algoritmos implementados, os resultados (em sua maioria dados de forma visual) obtidos com a aplicação destes, e o estudo da relação entre imagens de mesmo material, obtidas em situações semelhantes.

Para exemplificar o pré-processamento da imagem, assim como os resultados gráficos obtidos, será utilizada a Figura 7.1, dada abaixo.

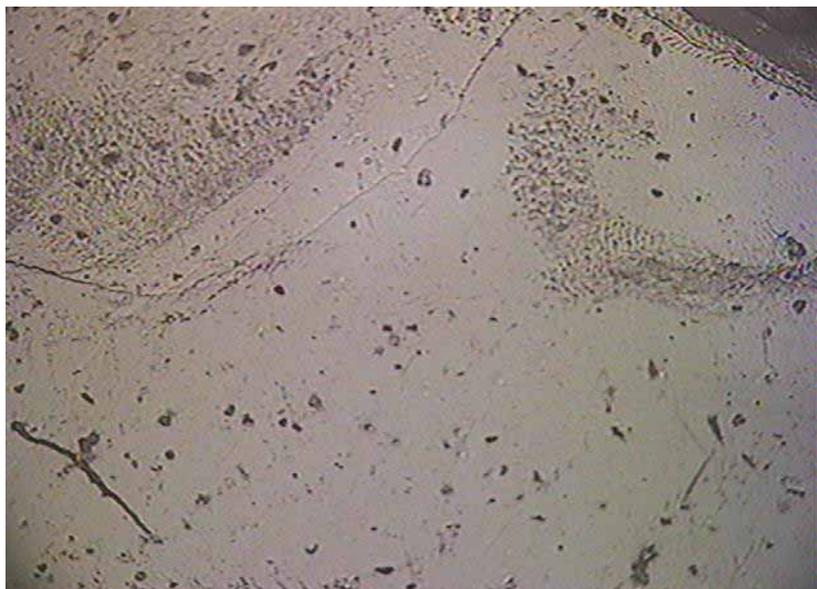


Figura 7.1 – Imagem de amostra de borracha natural com nanopartículas de ouro, ampliada em 100x, obtida por microscopia óptica aos 15 minutos após a deposição da amostra na solução de ouro.

7.1. O pré-processamento da imagem de entrada

O algoritmo de pré-processamento deve detectar e preencher as bordas e as fissuras contidas na imagem da amostra de látex. Se a imagem contiver aditivos ou impurezas, estes serão representados como regiões na imagem binária. Esta imagem é processada pelos algoritmos seguintes.

O pré-processamento inicia com a padronização das imagens, comprimindo ou expandindo suas dimensões para 256x256, dependendo do tamanho inicial da imagem. Após esta padronização, é aplicado o método de Otsu, que consiste na limiarização (*thresholding*) global de uma imagem baseada em seu histograma [45]. O nome do método é uma homenagem a Nobuyuki Otsu, seu criador.

Após o pré-processamento é obtida uma imagem binária proveniente da original, que apresenta um esboço dos contornos. O resultado do pré-processamento da Figura 7.1 é apresentado na Figura 7.2, dada abaixo.

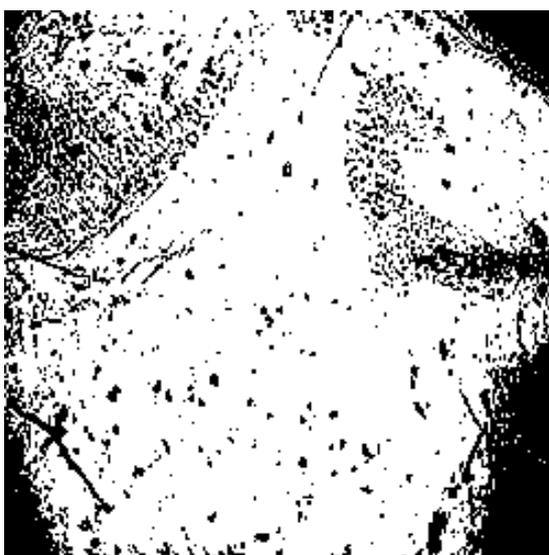


Figura 7.2 – Imagem binária obtida da Figura 7.1, após o processamento pelo método de Otsu.

O pré-processamento realizado pelo algoritmo é bastante satisfatório. A figura processada pelos algoritmos referentes às transformadas é a Figura 7.2, chamada de imagem segmentada.

7.2. Algoritmo referente à transformada de Fourier

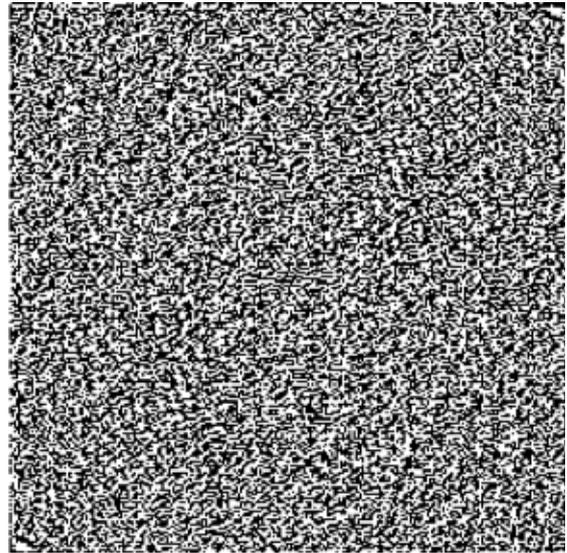
A transformada de Fourier é aplicada na imagem fornecida pelas instruções seguintes.

```
>> A1 = fft2 (img_name);  
>> img_log = fftshift(log(1+abs(A1)));  
>> img_scale = gscale(img_log);
```

A função `fft2` retorna a transformada rápida bidimensional de Fourier da imagem `img_name`, e `fftshift` troca os quadrantes ímpares e pares da imagem entre si. A função `gscale`, obtida de [12], converte uma matriz tridimensional em uma bidimensional, preservando as intensidades de cor. O deslocamento `fftshift`, em conjunto com a função `gscale`, enriquece os resultados gráficos apresentados, simplificando ainda os coeficientes resultantes.

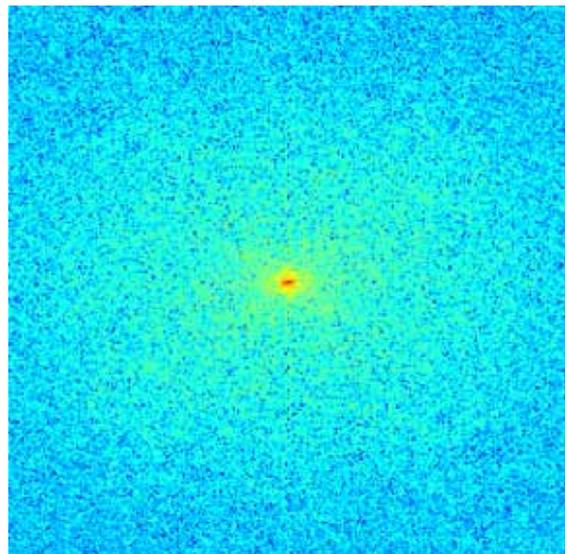
A Figura 7.3 mostra os resultados obtidos após a aplicação de `fft2(img_name)` e a associação `fftshift + gscale`. Os coeficientes apresentados são os três primeiros resultantes das instruções, enquanto que a imagem **(a)** apresentada corresponde à parte real da aplicação de `fft2`. Na imagem **(b)** o mapa de cores `jet` foi aplicado, colorindo o resultado.

$$\begin{bmatrix} 50193 & -8.524153e+003 & -3.503364e+003 \\ -2.440617e+003 & 5.681764e+000 & 9.305135e+002 \\ -8.306798e+002 & -9.415190e+001 & -4.015648e+002 \end{bmatrix}$$



(a)

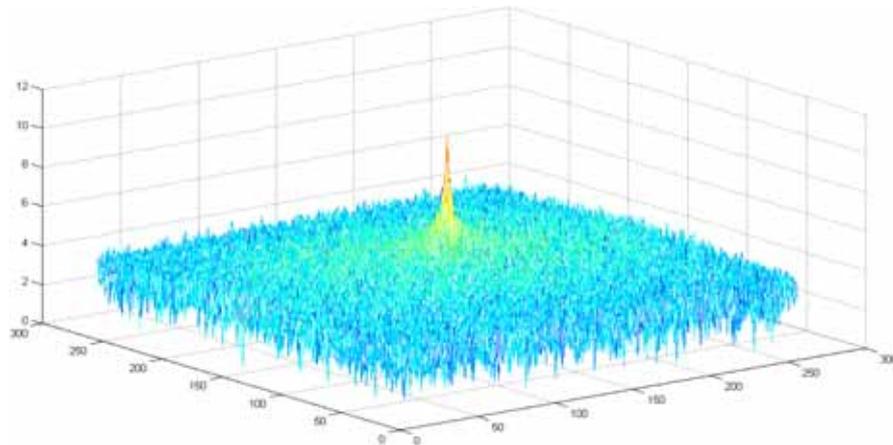
$$\begin{bmatrix} 104 & 71 & 72 \\ 94 & 69 & 93 \\ 94 & 71 & 62 \end{bmatrix}$$



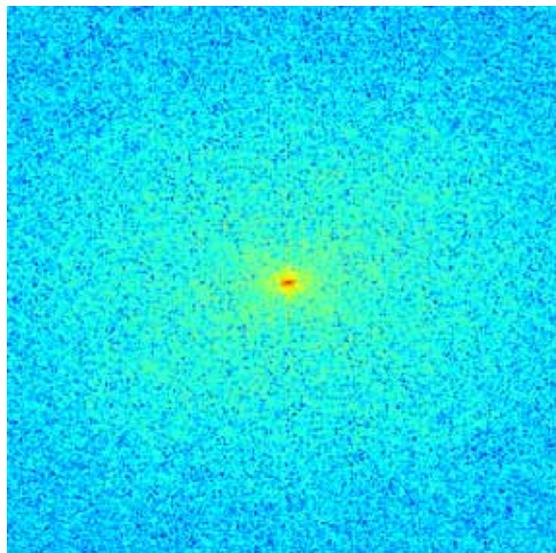
(b)

Figura 7.3 – Resultados da aplicação do algoritmo referente à transformada de Fourier, após (a) aplicação direta de `fft2`; (b) após a aplicação de `fft2`, o emprego de `fftshift` e `gscale`.

A transformada de Fourier na imagem segmentada corresponde diretamente à imagem de entrada, e o resultado é uma matriz de ordem 256. Tomando a Figura 7.1 como a imagem de entrada, têm-se os seguintes resultados, para a aplicação do algoritmo:



(a)



(b)

Figura 7.4 – Resultado da aplicação do algoritmo relativo à transformada de Fourier na Figura 7.1. **(a)** Espectro de Fourier da imagem de entrada; **(b)** projeção dos valores do espectro de Fourier no eixo xy .

7.3. Algoritmos referentes à transformada de Gabor

Neste estudo foi implementada uma versão modificada do filtro de Gabor para imagens apresentado por Gao Yang, professor da Nanyang Technological University, Cingapura[46]. A função de Gabor utilizada na construção se encontra na forma descrita por Khaled Hammouda e Ed Jernigan para segmentação de texturas[47],

$$\Lambda(x, y) = \frac{1}{2\pi\sigma_x\sigma_y} \exp\left[-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}\right)\right] \cos(2\pi u_0 x), \quad (3.10)$$

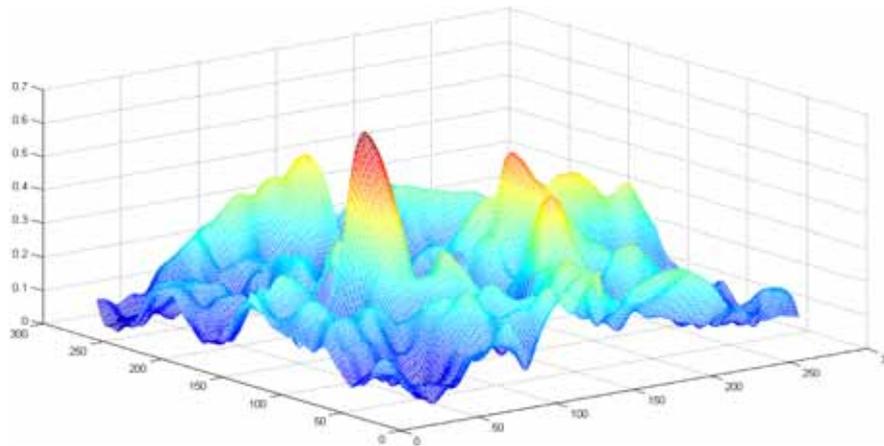
em que σ_x e σ_y são os desvios-padrão de $\Lambda(x, y)$ nos eixos x e y , respectivamente.

A transformada de Gabor é aplicada na imagem segmentada correspondente à imagem de entrada pelos algoritmos presentes nas funções `transgabor.m` e `gaborfilter.m` (Apêndice D, [46]), gerando uma matriz de ordem 256. Estes algoritmos são invocados pela instrução seguinte:

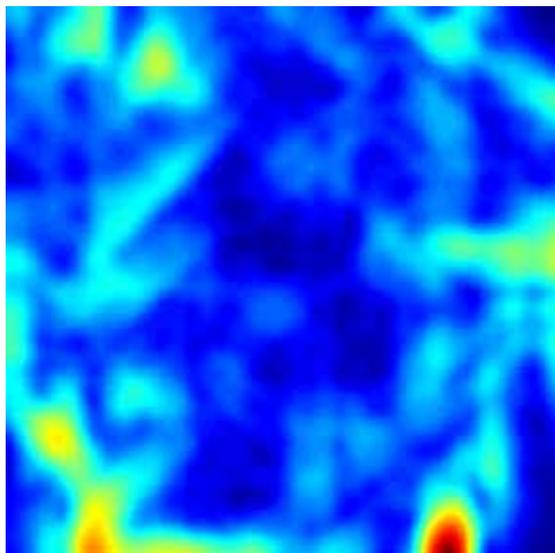
```
>> [td_gabor, mat_gabor] = transgabor(img_name);
```

em que `td_gabor` é o conjunto de coeficientes que dá origem ao espectro de Gabor e `mat_gabor` é a matriz de coeficientes gerada pela aplicação da transformada na imagem.

Aplicando a instrução acima em conjunto com as funções dadas no Apêndice D na Figura 7.1, têm-se os resultados:



(a)



(b)

Figura 7.5 – Resultado da aplicação do algoritmo relativo à transformada de Gabor na Figura 7.1. **(a)** Espectro de Gabor da imagem de entrada; **(b)** projeção dos valores do espectro de Gabor no eixo xy .

7.4. Algoritmos referentes à *wavelet* de Haar

A transformada *wavelet* de Haar é aplicada pelos algoritmos implementados em cinco níveis de decomposição para as imagens de entrada. O tamanho das imagens resultantes de cada decomposição é dado abaixo, na Tabela 4.

Tabela 4 – Tamanho das matrizes resultantes da aplicação dos níveis de decomposição *wavelet* para Haar.

| Nível de decomposição | Matriz resultante |
|-----------------------|-------------------|
| Imagem de entrada | 256x256 |
| Nível 1 | 128x128 |
| Nível 2 | 64x64 |
| Nível 3 | 32x32 |

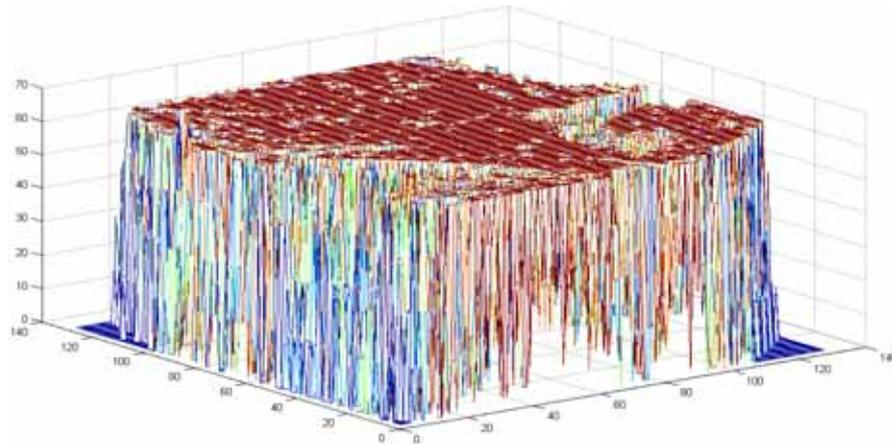
| Nível de decomposição | Matriz resultante |
|-----------------------|-------------------|
| Nível 4 | 16x16 |
| Nível 5 | 8x8 |

Os comandos para a aplicação da *wavelet* de Haar na imagem segmentada correspondente à imagem de entrada são iterativos, isto é, a cada processamento é utilizado o nível de decomposição *wavelet* anterior. O método iterativo para a obtenção dos níveis de decomposição *wavelet* é devido ao algoritmo piramidal de Mallat (Seção 4.4). As instruções são as seguintes, para os dois primeiros níveis de decomposição:

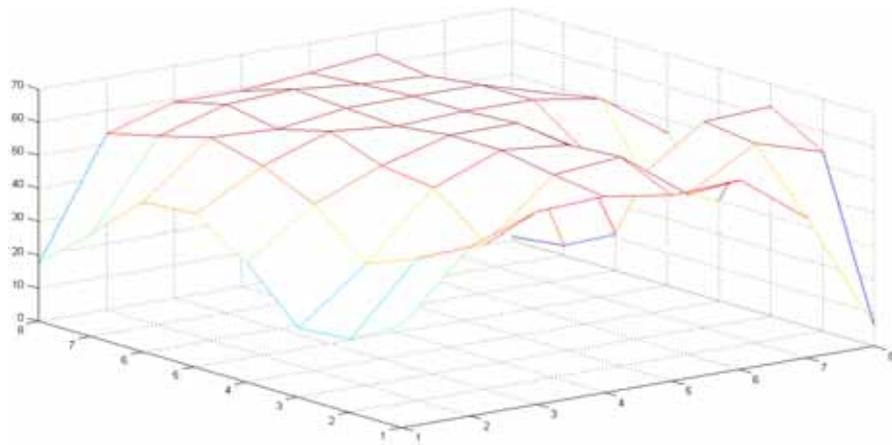
```
>> map = colormap;
>> nbc01 = size(map,1);
% Aplicação do nível 1 de decomposição para a wavelet de Haar
>> [cA1, ~, ~, ~] = dwt2(img_name, 'haar');
>> cod_cA1 = wcodemat(cA1, nbc01);
>> cA1_scale = gscale(cod_cA1);
% Aplicação do nível 2 de decomposição para a wavelet de Haar
>> [cA2, ~, ~, ~] = dwt2(cA1, 'haar');
>> cod_cA2 = wcodemat(cA2, nbc01);
>> cA2_scale = gscale(cod_cA2);
```

A função `dwt2`, no primeiro nível de decomposição, aplica a transformada *wavelet* bidimensional discreta na imagem `img_name`. No segundo nível de decomposição a DWT bidimensional é aplicada em `cA1`, matriz dos coeficientes *wavelet* de aproximação da primeira decomposição. O argumento `'haar'` indica que a *wavelet* de Haar é utilizada. O dimensionamento `wcodemat` retorna uma versão codificada da matriz argumento, baseando-se em seu mapa de cores. A implementação dos outros níveis de decomposição é semelhante.

A aplicação dos algoritmos para a *wavelet* de Haar na Figura 7.1 retorna os espectros exibidos na Figura 7.6, referentes ao primeiro e último níveis de decomposição. Por sua vez, as projeções dos níveis de decomposição para a *wavelet* de Haar são dadas na Figura 7.7.



(a)



(b)

Figura 7.6 – Espectros resultantes da aplicação da *wavelet* de Haar na Figura 7.1. **(a)** Nível 1 de decomposição; **(b)** nível 5 de decomposição.

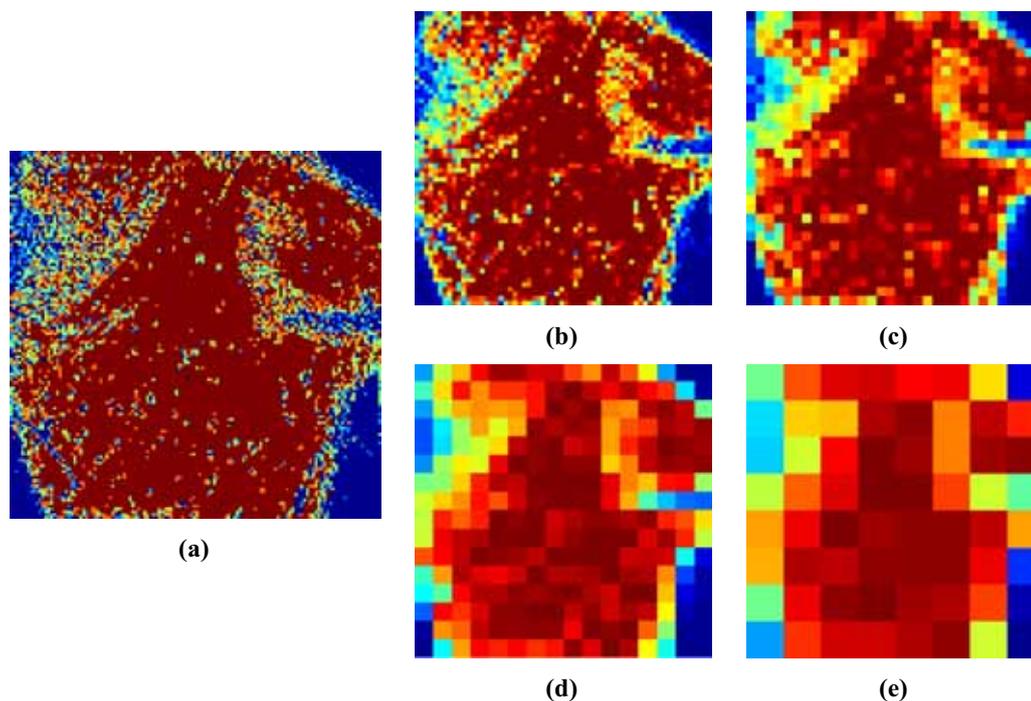


Figura 7.7 – Projeções resultantes da aplicação da *wavelet* de Haar na Figura 7.1 pelos algoritmos implementados. **(a)** Nível 1 de decomposição; **(b)** nível 2 de decomposição; **(c)** nível 3 de decomposição; **(d)** nível 4 de decomposição; **(e)** nível 5 de decomposição.

7.5. Algoritmos referentes à *wavelet* de Daubechies

A transformada *wavelet* de Daubechies é aplicada pelos algoritmos implementados em sete níveis de decomposição para as imagens de entrada. O tamanho das imagens resultantes de cada decomposição é dado na Tabela 5.

Tabela 5 – Tamanho das matrizes resultantes da aplicação dos níveis de decomposição *wavelet* para Daubechies.

| Nível de decomposição | Matriz resultante |
|-----------------------|-------------------|
| Imagem de entrada | 256x256 |
| Nível 1 | 131x131 |

| Nível de decomposição | Matriz resultante |
|-----------------------|-------------------|
| Nível 2 | 69x69 |
| Nível 3 | 38x38 |
| Nível 4 | 22x22 |
| Nível 5 | 14x14 |
| Nível 6 | 10x10 |
| Nível 7 | 8x8 |

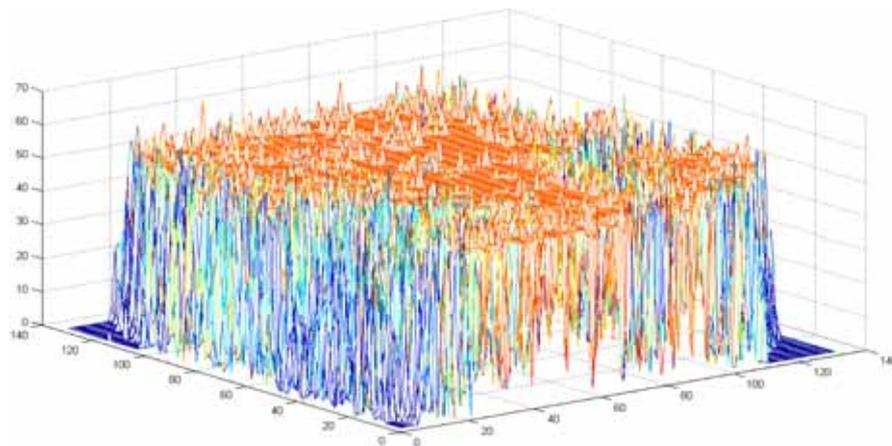
A *wavelet* de Daubechies implementada pelos algoritmos é a D4, com dois momentos nulos, dada na Figura 4.3. Da mesma forma que para as *wavelets* de Haar, as instruções para a aplicação da *wavelet* de Daubechies na imagem segmentada são iterativas. Os códigos são os seguintes, para os três primeiros níveis de decomposição:

```
>> map = colormap;
>> nbcou = size(map,1);
% Aplicação do nível 1 de decomposição para a wavelet de Daubechies
>> [cA1, ~, ~, ~] = dwt2(img_name, 'db4');
>> cod_cA1 = wcodemat(cA1, nbcou);
>> cA1_scale = gscale(cod_cA1);
% Aplicação do nível 2 de decomposição para a wavelet de Daubechies
>> [cA2, ~, ~, ~] = dwt2(cA1, 'db4');
>> cod_cA2 = wcodemat(cA2, nbcou);
>> cA2_scale = gscale(cod_cA2);
% Aplicação do nível 3 de decomposição para a wavelet de Daubechies
>> [cA3, ~, ~, ~] = dwt2(cA2, 'db4');
>> cod_cA3 = wcodemat(cA3, nbcou);
```

```
>> cA3_scale = gscale(cod_cA3);
```

A função `dwt2`, no primeiro nível de decomposição, aplica a transformada *wavelet* bidimensional discreta na imagem `img_name`, enquanto que no segundo nível de decomposição a DWT bidimensional é aplicada em `cA1`, matriz dos coeficientes *wavelet* de aproximação da primeira decomposição. Por fim, no terceiro nível de decomposição a DWT bidimensional é aplicada em `cA2`, matriz dos coeficientes *wavelet* de aproximação da segunda decomposição. O argumento `'db4'` indica que a *wavelet* D4 de Daubechies é utilizada. O dimensionamento `wcodemat` retorna uma versão codificada da matriz argumento, baseando-se em seu mapa de cores. A implementação dos outros níveis de decomposição é análoga.

A aplicação dos algoritmos para a *wavelet* de Daubechies na Figura 7.1 retorna os espectros exibidos na Figura 7.8, referentes ao primeiro e último níveis de decomposição. Por sua vez, as projeções dos níveis de decomposição para a *wavelet* de Daubechies são dadas na Figura 7.9.



(a)

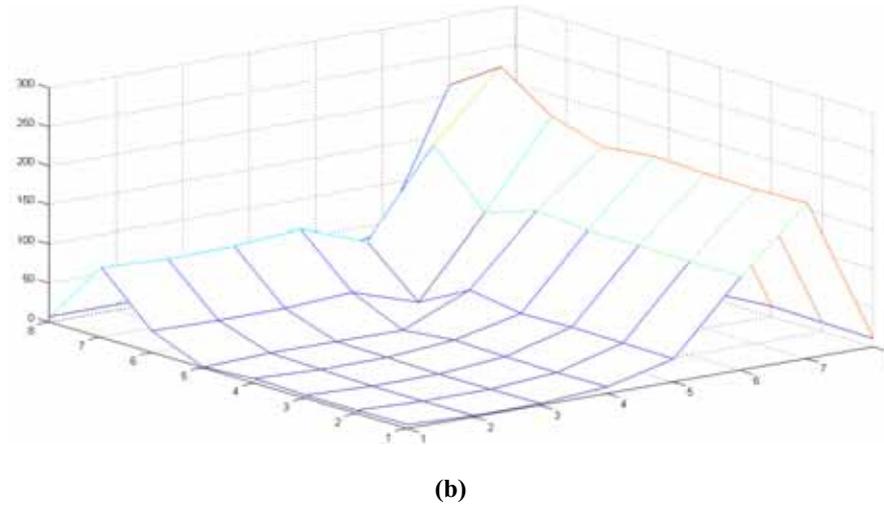


Figura 7.8 – Espectros resultantes da aplicação da *wavelet* de Daubechies na Figura 7.1. **(a)** Nível 1 de decomposição; **(b)** nível 7 de decomposição.

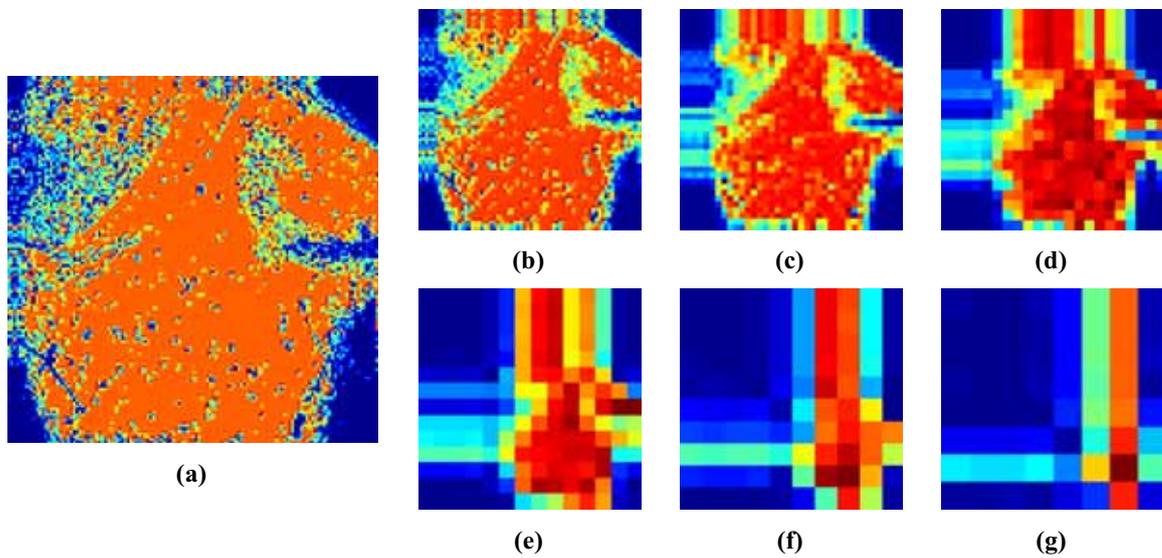


Figura 7.9 – Projeções resultantes da aplicação da *wavelet* de Daubechies na Figura 7.1 pelos algoritmos implementados. **(a)** Nível 1 de decomposição; **(b)** nível 2 de decomposição; **(c)** nível 3 de decomposição; **(d)** nível 4 de decomposição; **(e)** nível 5 de decomposição; **(f)** nível 6 de decomposição; **(g)** nível 7 de decomposição.

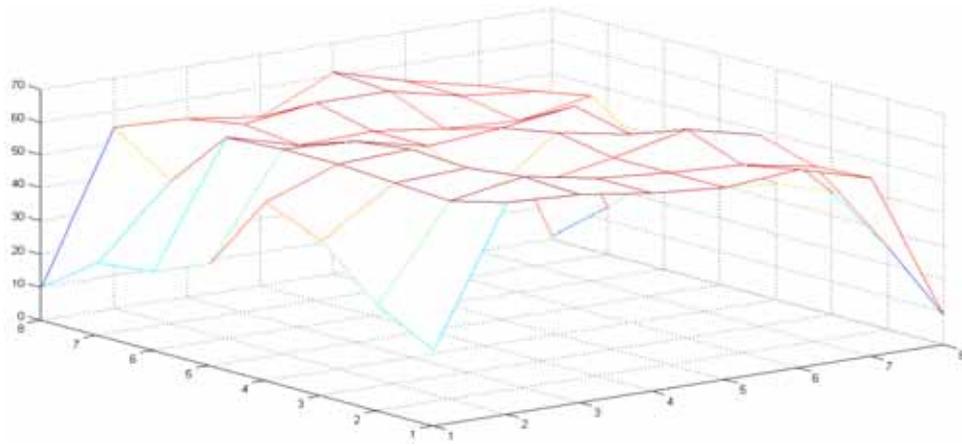
7.6. O conjunto de imagens de teste

O conjunto de imagens de teste possui vinte e cinco imagens, entre imagens de microscopia óptica, de varredura e de força atômica com borracha natural pura ou contendo aditivos metálicos. Os algoritmos para todas as transformadas foram aplicados nas imagens de teste. Os coeficientes relacionados a cada transformada foram gravados, assim como os resultados gráficos gerados, constituindo o banco de dados das amostras. Algumas particularidades da aplicação das transformadas em cada tipo de microscopia são dadas a seguir.

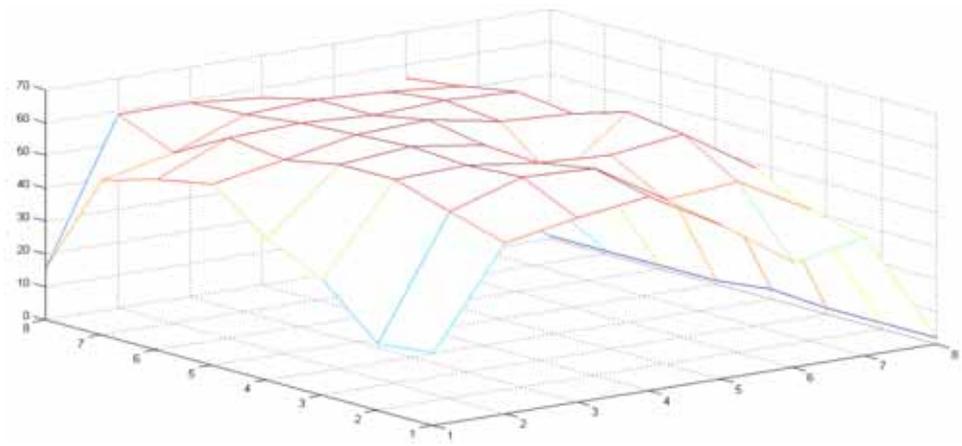
7.6.1. Aplicação das transformadas nas imagens de teste obtidas por microscopia óptica

A aplicação das transformadas *wavelet* nas imagens de membranas de borracha natural com aditivos metálicos constantes do grupo de imagens de teste gerou regiões semelhantes. A aplicação do último nível de decomposição das *wavelets* de Haar gerou ‘platôs’, enquanto a aplicação do último nível de decomposição das *wavelets* de Daubechies gerou regiões niveladas.

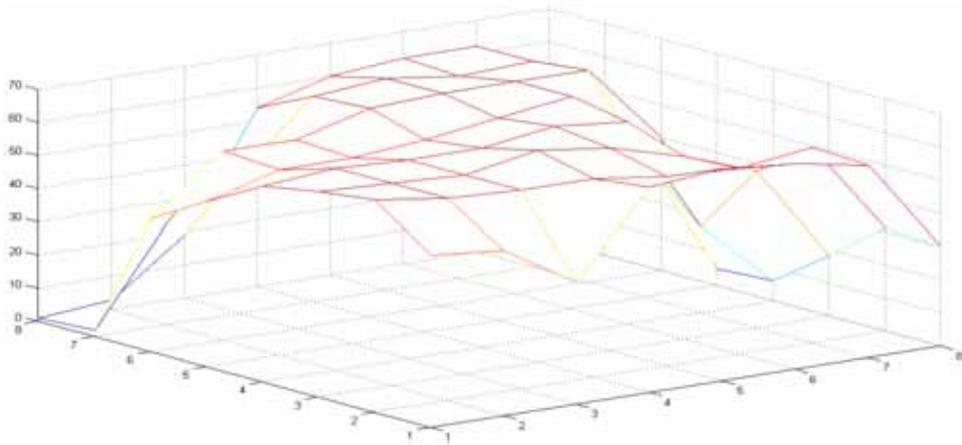
As Figuras 7.10 e 7.11 expõem exemplos de tais situações. Os espectros a seguir foram obtidos com a utilização das decomposições de último nível das *wavelets* de Haar e Daubechies. As imagens que deram origem a estes espectros foram retiradas por microscopia óptica, ampliação de 100x, de membranas de borracha natural com ouro depositado por redução *in situ*. Os espectros (a), (b), (c) e (d) de cada imagem foram obtidos respectivamente de membranas depositadas em ouro por 3, 6, 30 e 60 minutos.



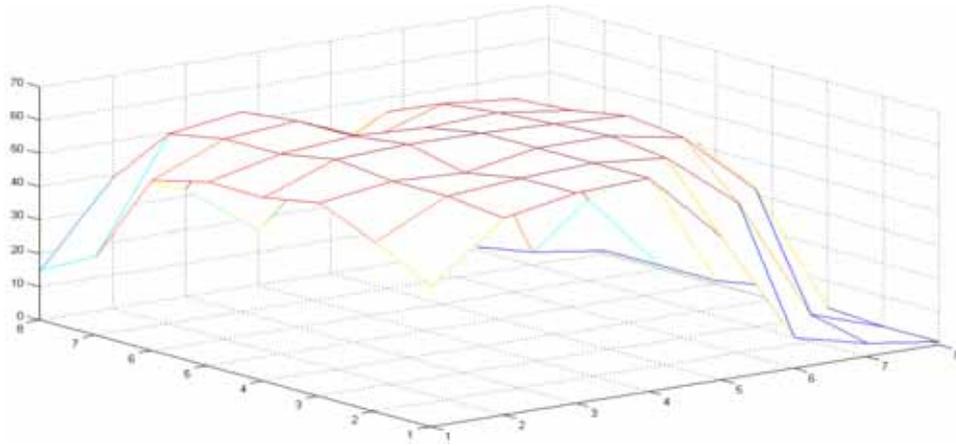
(a)



(b)

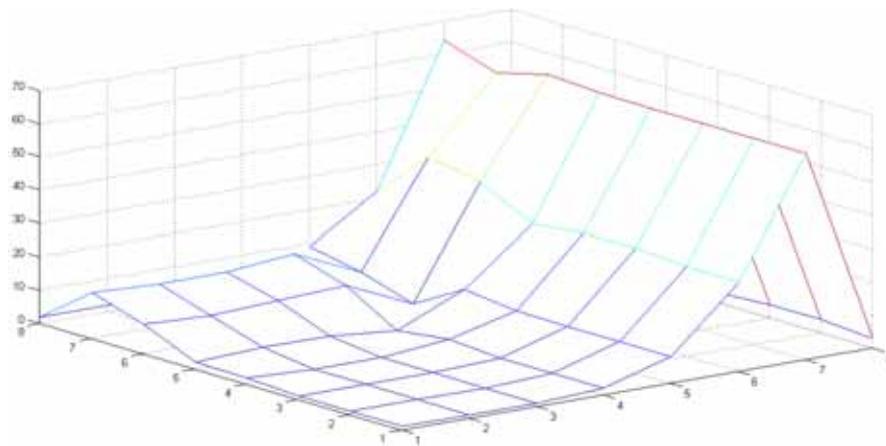


(c)

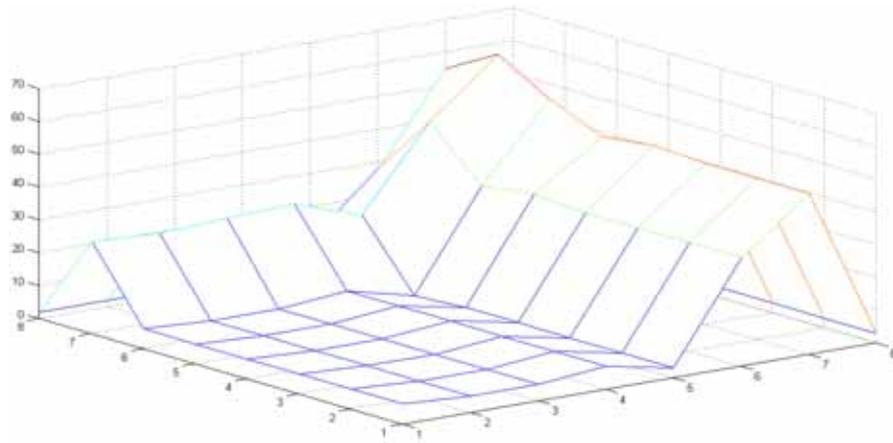


(d)

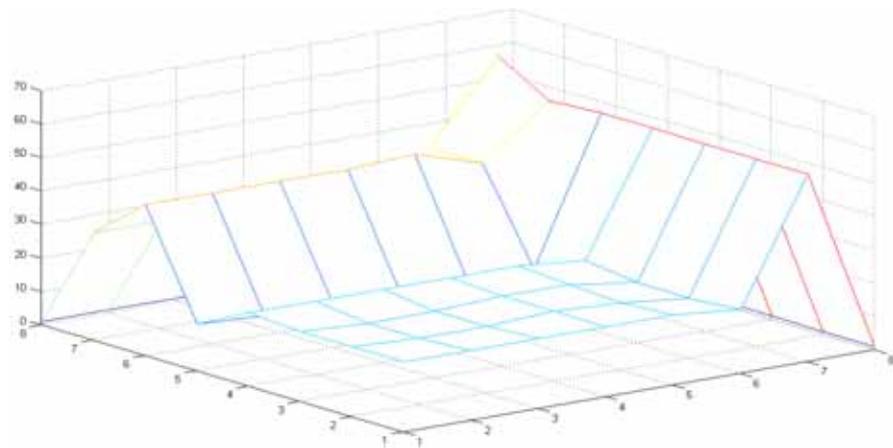
Figura 7.10 – Platôs gerados pela aplicação da *wavelet* de Haar em imagens de microscopia óptica de membranas de borracha natural com aditivos metálicos. Os espectros foram obtidos de membranas de borracha natural depositadas em solução de ouro por (a) 6 minutos; (b) 9 minutos; (c) 30 minutos; (d) 60 minutos.



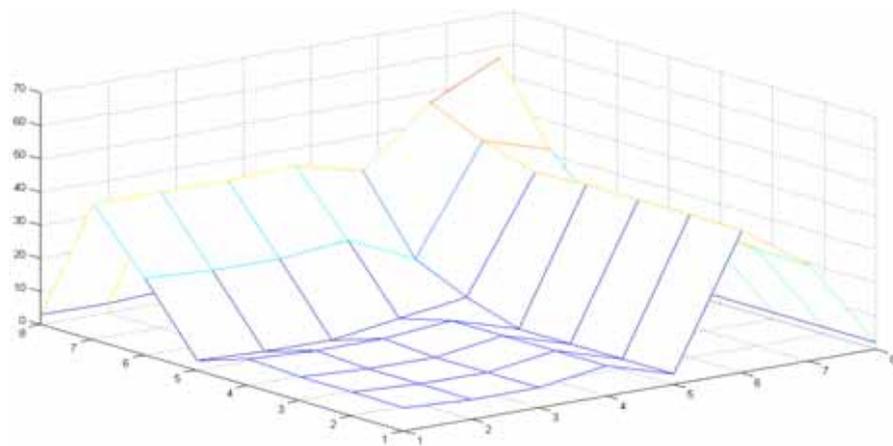
(a)



(b)



(c)



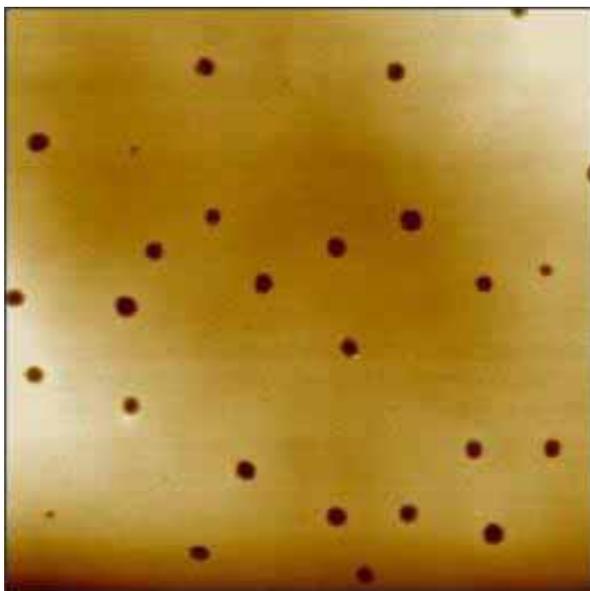
(d)

Figura 7.11 – Regiões em nível geradas pela aplicação da *wavelet* de Daubechies em imagens de microscopia óptica de membranas de borracha natural com aditivos metálicos. Os espectros foram obtidos de membranas de borracha natural depositadas em solução de ouro por (a) 6 minutos; (b) 9 minutos; (c) 30 minutos; (d) 60 minutos.

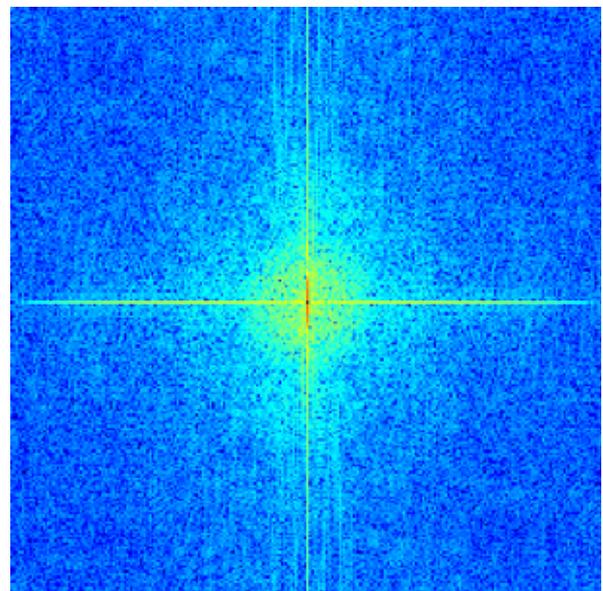
7.6.2. Aplicação das transformadas nas imagens de teste obtidas por microscopia de força atômica

A aplicação da transformada de Fourier em certas imagens de membranas de borracha natural contidas entre as imagens de teste mostrou regiões dominantes originadas dos padrões regulares no segundo plano da imagem original, geralmente passando verticalmente e horizontalmente através do centro.

A transformada de Fourier mostra que a imagem de entrada contém componentes para todas as frequências, mas sua magnitude se torna menor para frequências mais altas. Então, componentes de baixa frequência contém mais informação da imagem do que os componentes de alta frequência. Exemplos desta situação são dados na Figura 7.12, onde imagens de microscopia de força atômica são exibidas ao lado da projeção de suas respectivas transformadas.



(a)



(b)

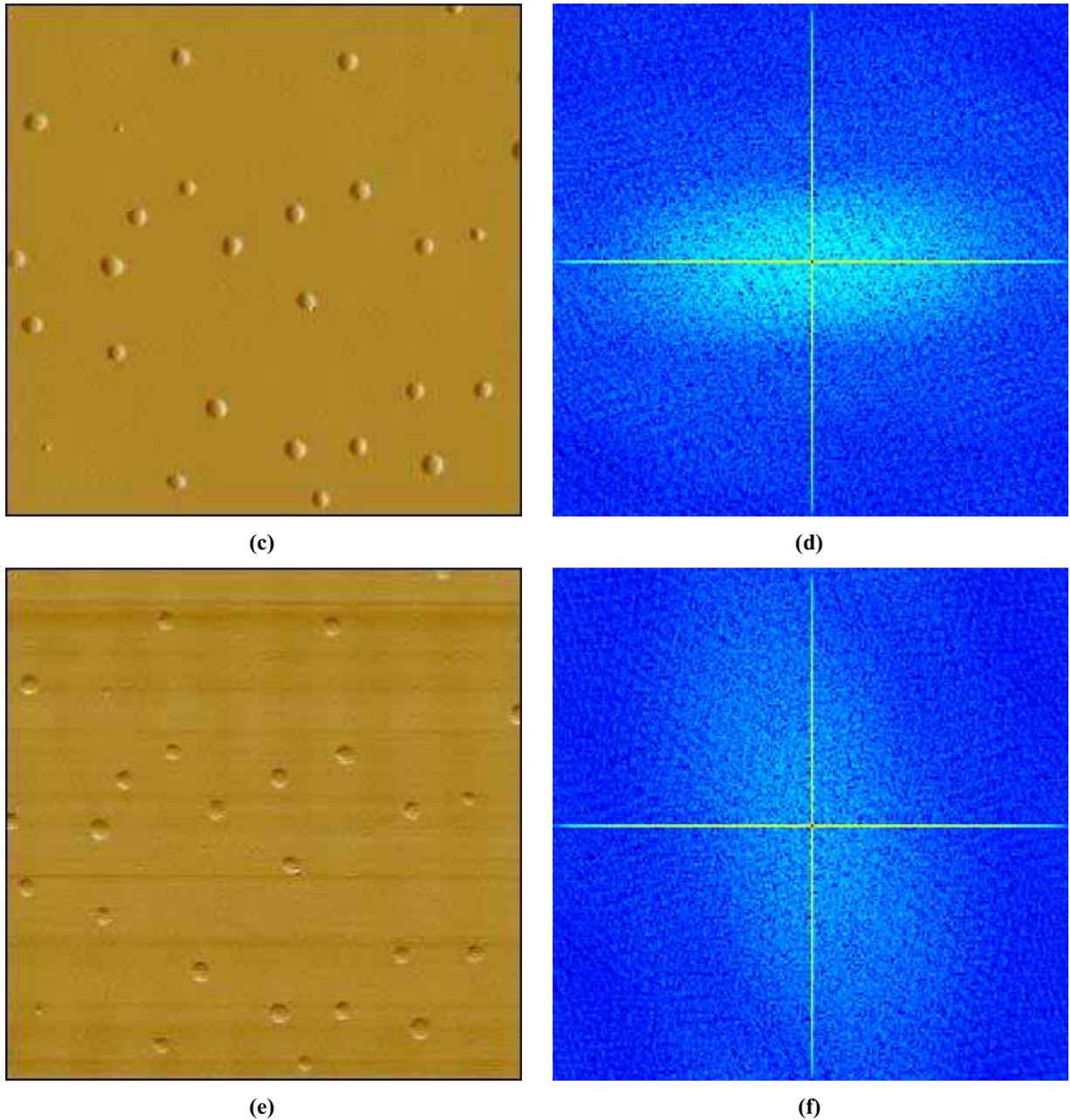


Figura 7.12 – (a), (c), (e) Imagens de membranas de borracha natural e (b), (d), (f) a respectiva projeção dos espectros de Fourier para cada imagem, indicando padrões regulares no segundo plano da figura original.

7.7. Relação entre imagens retiradas em diferentes tempos de deposição

Entre as imagens do conjunto de teste, há cinco imagens da mesma amostra de borracha natural com ouro, obtidas por microscopia óptica a uma ampliação de 100x, depositadas na solução de ouro por tempos diferentes: 6 minutos, 9 minutos, 30 minutos, 60 minutos e 120 minutos. Para o cálculo da correlação foi criado um pequeno programa no Matlab, sendo utilizadas as matrizes dos coeficientes relacionados às imagens de entrada. Estas matrizes foram previamente salvas utilizando o programa WaveFPR.

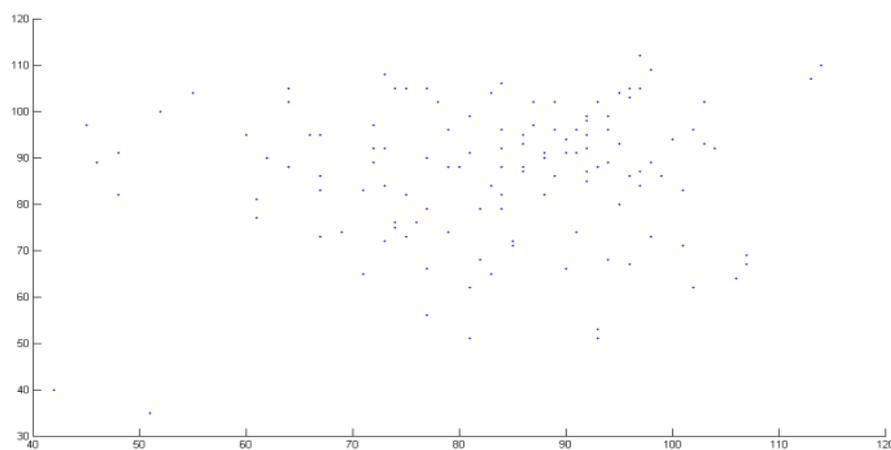
As imagens foram relacionadas duas a duas, do tempo menor para o maior; isto é, a correlação entre as imagens depositadas por seis e nove minutos foi estabelecida, depois a correlação foi tomada entre as imagens depositadas por nove e trinta minutos, e assim por diante. No caso das *wavelets* de Haar e Daubechies, a correlação foi tomada nível a nível para cada dupla de imagens: imagens de seis minutos e nove minutos, nível 1 de decomposição *wavelet*, seis minutos e nove minutos, nível 2 de decomposição, e assim sucessivamente.

As transformadas de Fourier e Gabor não apresentaram relação aparente entre as imagens de teste. O resultado é esperado, pois Pedro Morettin mostra que duas séries temporais que diferem em apenas um ponto possuem espectros de Fourier com coordenadas totalmente diferentes[48]. Para a transformada de Gabor espera-se o mesmo resultado, pois esta pode ser vista como um caso particular da transformada de Fourier. As *wavelets* de Haar e Daubechies mostraram forte correlação entre as imagens; tal relação cresce a cada nível de decomposição, como mostrado a seguir. Tal fato é alcançado pela diminuição de coeficientes no decorrer das decomposições, como visto nas Tabelas 4 e 5. A informação resultante contida nos últimos níveis de decomposição é bastante semelhante, pela comparação de amostras obtidas do mesmo material (borracha natural com nanopartículas de ouro).

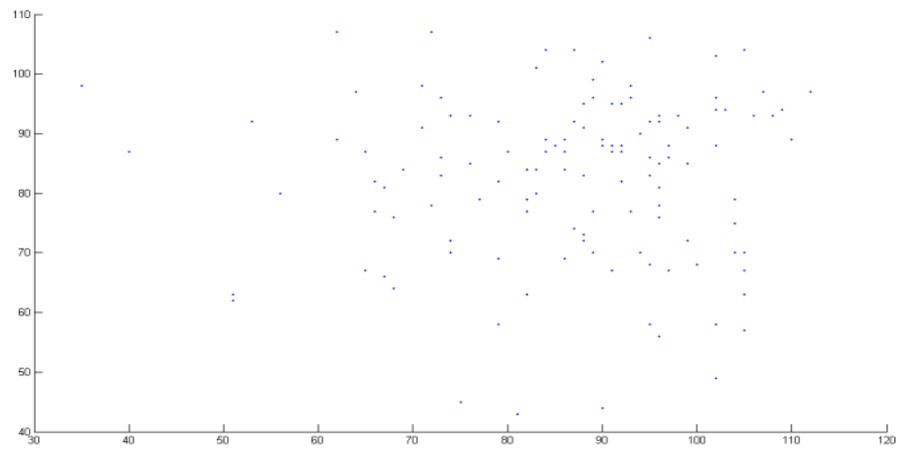
O valor dos coeficientes de correlação para a transformada de Fourier é dado na Tabela 6, enquanto que a Figura 7.13 exibe o diagrama de dispersão da primeira coluna dos coeficientes de Fourier obtidos das imagens de teste, nesta ordem: 6 minutos e 9 minutos; 9 minutos e 30 minutos; 30 minutos e 60 minutos; 60 minutos e 120 minutos.

Tabela 6 – Coeficientes de correlação obtidos da transformada de Fourier de imagens depositadas em solução de ouro por diferentes espaços de tempo.

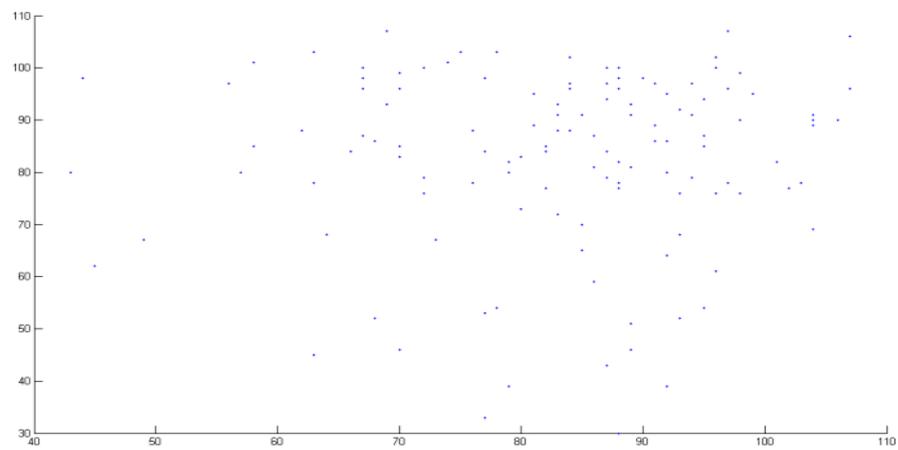
| Imagens relacionadas | Correlação |
|-----------------------------|-------------------|
| 6 min e 9 min | 27,72% |
| 9 min e 30 min | 26,10% |
| 30 min e 60 min | 26,62% |
| 60 min e 120 min | 24,92% |



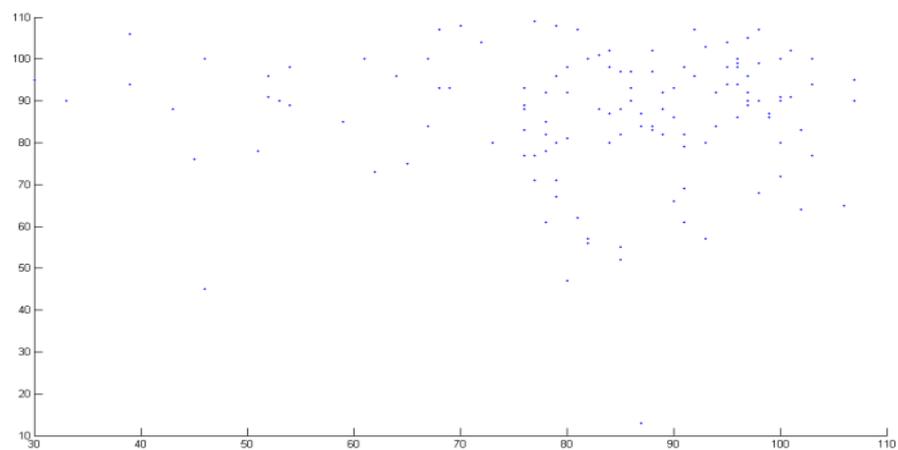
(a)



(b)



(c)



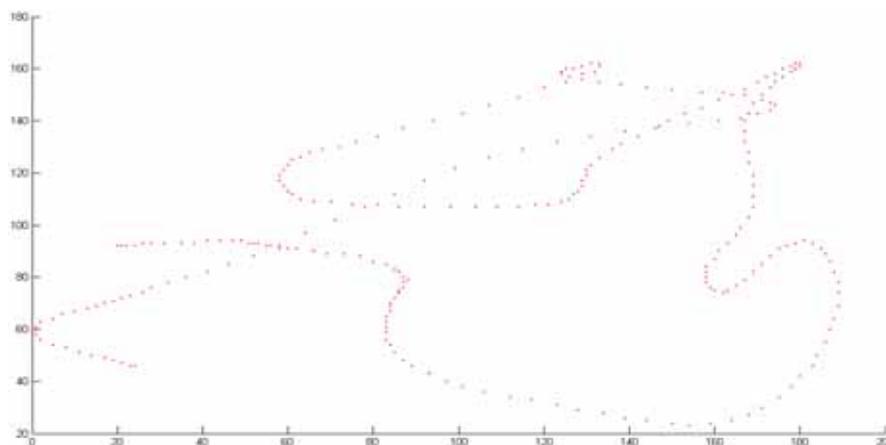
(d)

Figura 7.13 – Correlação entre imagens de borracha natural com ouro para Fourier, entre (a) 6 minutos e 9 minutos. (b) 9 minutos e 30 minutos. (c) 30 minutos e 60 minutos. (d) 60 minutos e 120 minutos.

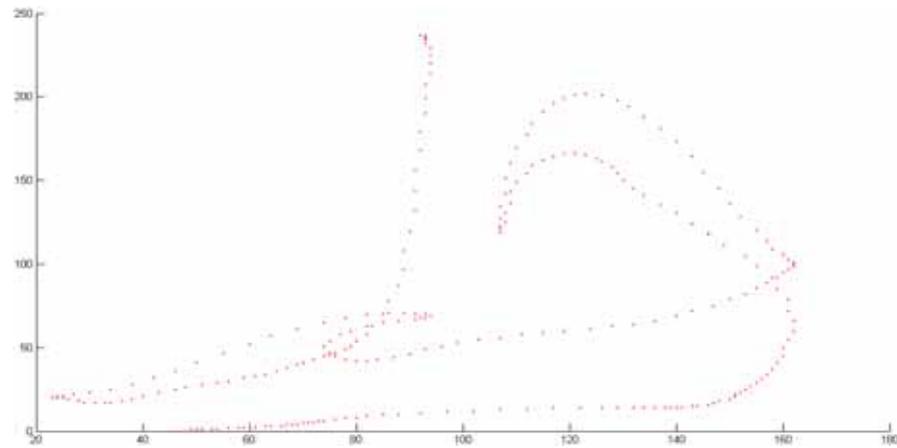
Da mesma forma, o valor dos coeficientes de correlação para a transformada de Gabor é dado na Tabela 7. A Figura 7.14 ilustra a correlação da primeira coluna dos coeficientes de Gabor para as imagens de teste, na ordem dada para a transformada de Fourier: 6 minutos e 9 minutos; 9 minutos e 30 minutos; 30 minutos e 60 minutos; 60 minutos e 120 minutos. Os *scatterplots* resultantes da correspondência entre os coeficientes de Gabor são intrigantes, por apresentarem formas contínuas. Ainda assim, a correlação entre as imagens de teste se conserva baixa, como na aplicação da transformada de Fourier.

Tabela 7 – Coeficientes de correlação obtidos da transformada de Gabor de imagens depositadas em solução de ouro por diferentes espaços de tempo.

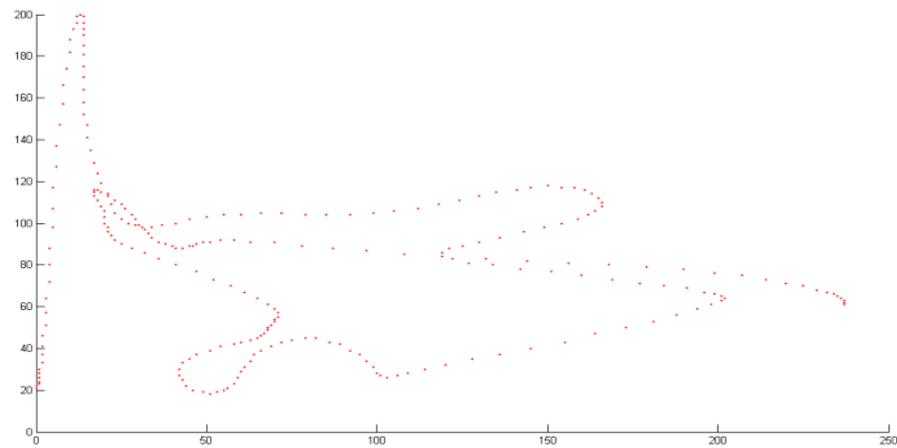
| Imagens relacionadas | Correlação |
|----------------------|------------|
| 6 min e 9 min | -2,23% |
| 9 min e 30 min | -2,40% |
| 30 min e 60 min | -8,38% |
| 60 min e 120 min | 26,59% |



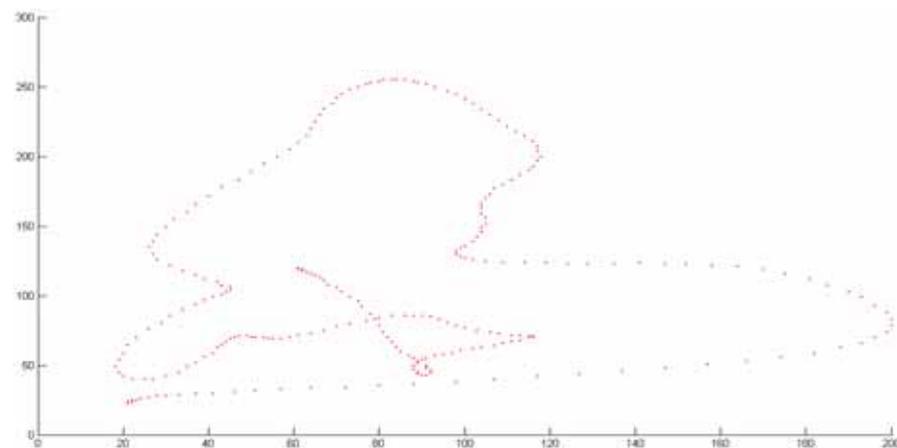
(a)



(a)



(b)



(c)

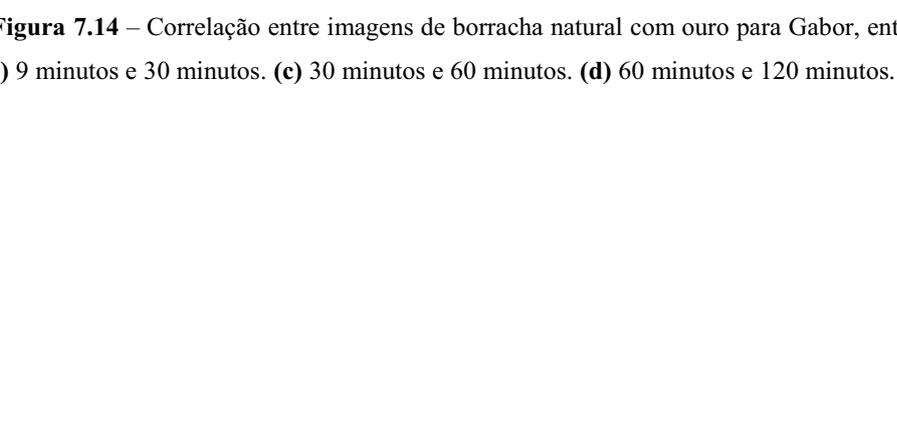
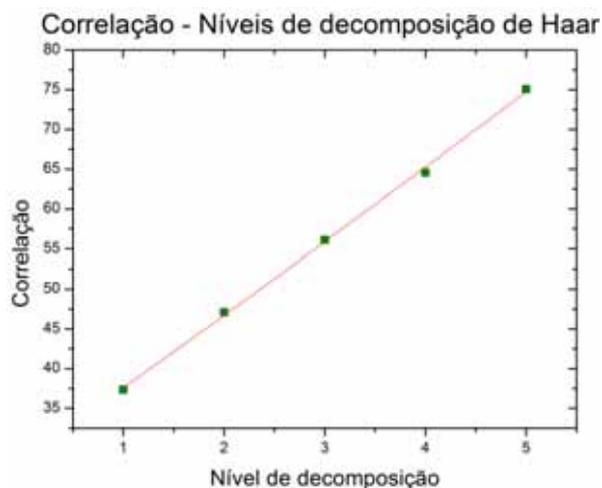


Figura 7.14 – Correlação entre imagens de borracha natural com ouro para Gabor, entre (a) 6 minutos e 9 minutos. (b) 9 minutos e 30 minutos. (c) 30 minutos e 60 minutos. (d) 60 minutos e 120 minutos.

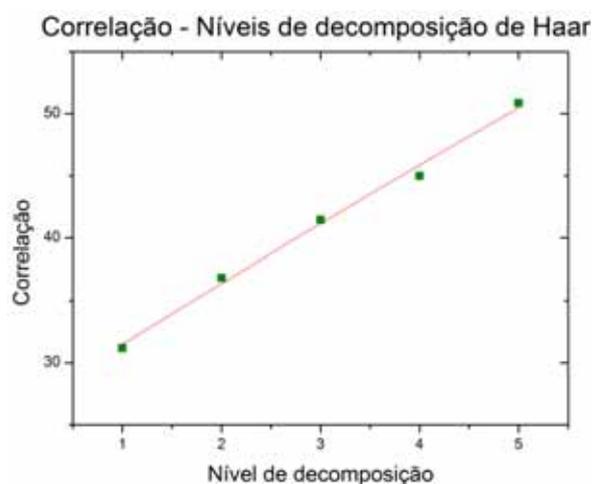
A Figura 7.15 ilustra o valor das correlações entre os coeficientes da *wavelet* de Haar para cada nível de decomposição, e também um gráfico do comportamento da correlação para as imagens de teste, na ordem: 6 minutos e 9 minutos; 9 minutos e 30 minutos; 30 minutos e 60 minutos; 60 minutos e 120 minutos. A alta relação entre as imagens é confirmada com o ajuste de uma curva polinomial aos valores correlacionais correspondentes aos níveis de decomposição.

| Haar | 100x, 6min – 9 min |
|---------|--------------------|
| Nível 1 | 37,36% |
| Nível 2 | 47,10% |
| Nível 3 | 56,11% |
| Nível 4 | 64,56% |
| Nível 5 | 75,07% |



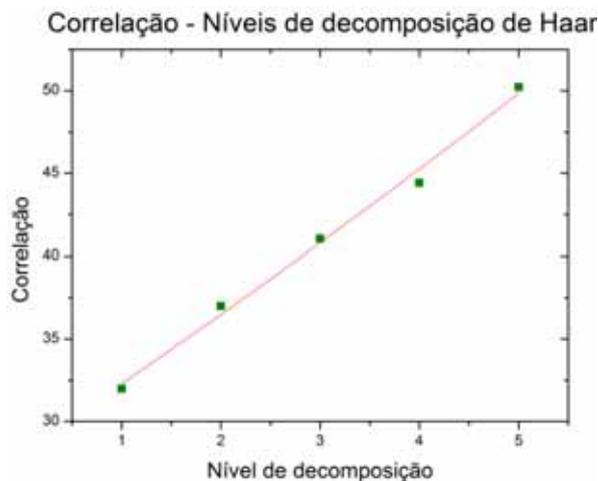
(a)

| Haar | 100x, 9min – 30min |
|---------|--------------------|
| Nível 1 | 31,20% |
| Nível 2 | 36,85% |
| Nível 3 | 41,45% |
| Nível 4 | 44,99% |
| Nível 5 | 50,86% |



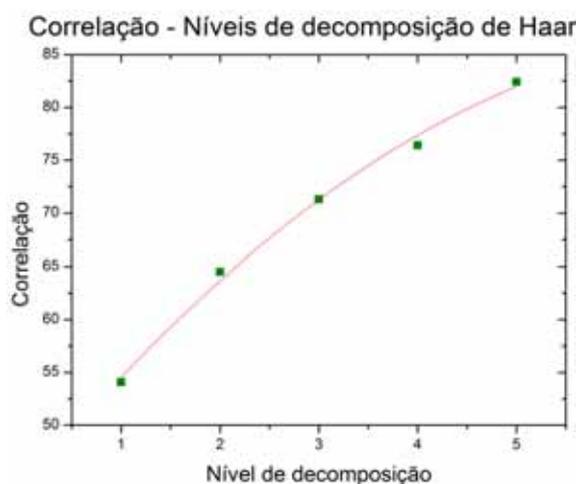
(b)

| Haar | 100x, 30min – 60min |
|---------|---------------------|
| Nível 1 | 31,99% |
| Nível 2 | 37,00% |
| Nível 3 | 41,04% |
| Nível 4 | 44,41% |
| Nível 5 | 50,22% |



(c)

| Haar | 100x, 60min – 120min |
|---------|----------------------|
| Nível 1 | 54,09% |
| Nível 2 | 64,52% |
| Nível 3 | 71,30% |
| Nível 4 | 76,43% |
| Nível 5 | 82,43% |



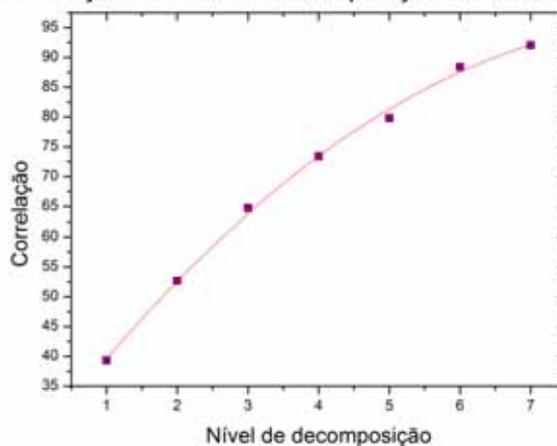
(d)

Figura 7.15 – Correlação das imagens de borracha natural com ouro para Haar, entre (a) 6 minutos e 9 minutos. (b) 9 minutos e 30 minutos. (c) 30 minutos e 60 minutos. (d) 60 minutos e 120 minutos.

Finalmente, na Figura 7.16 é dado o valor das correlações para os coeficientes de Daubechies para as imagens de teste obtidas aos: 6 minutos e 9 minutos; 9 minutos e 30 minutos; 30 minutos e 60 minutos; 60 minutos e 120 minutos. Da mesma forma, uma curva polinomial é ajustada aos valores correlacionais correspondentes aos níveis de decomposição para a *wavelet* de Daubechies.

| Daubechies | 100x, 6min – 9min |
|------------|-------------------|
| Nível 1 | 39,31% |
| Nível 2 | 52,68% |
| Nível 3 | 64,74% |
| Nível 4 | 73,40% |
| Nível 5 | 79,82% |
| Nível 6 | 88,42% |
| Nível 7 | 92,09% |

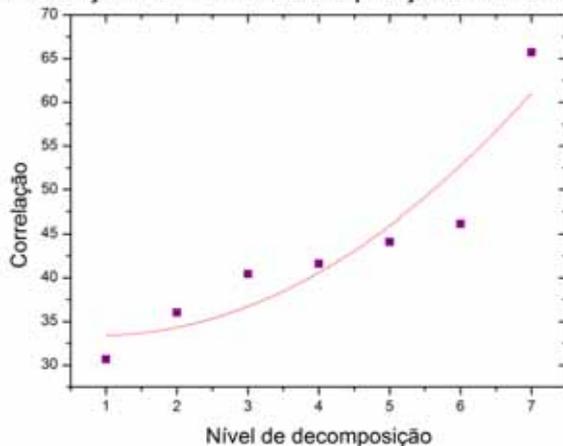
Correlação - Níveis de decomposição de Daubechies



(a)

| Daubechies | 100x, 9min – 30min |
|------------|--------------------|
| Nível 1 | 30,69% |
| Nível 2 | 36,03% |
| Nível 3 | 40,44% |
| Nível 4 | 41,62% |
| Nível 5 | 44,10% |
| Nível 6 | 46,19% |
| Nível 7 | 65,71% |

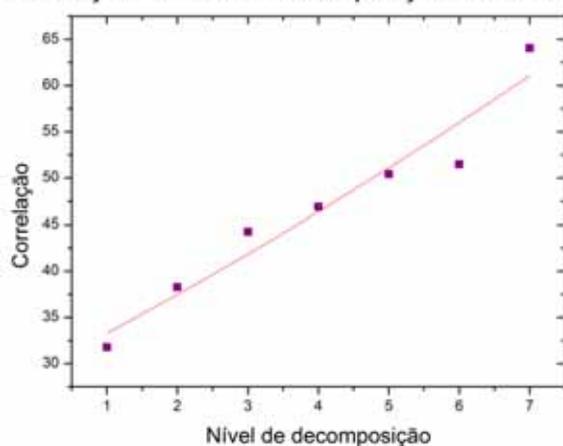
Correlação - Níveis de decomposição de Daubechies



(b)

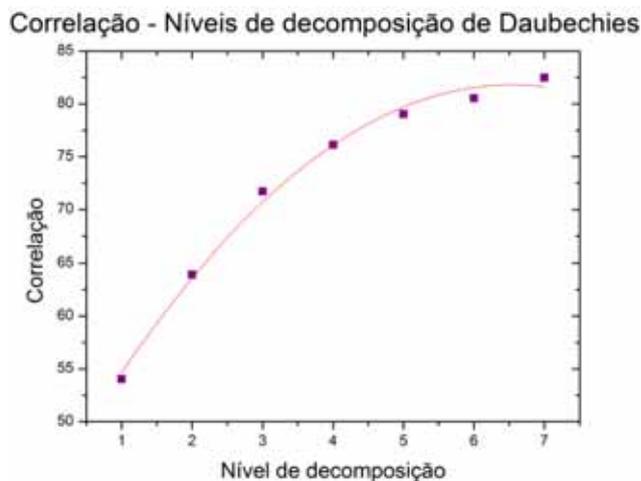
| Daubechies | 100x, 30min – 60min |
|------------|---------------------|
| Nível 1 | 31,77% |
| Nível 2 | 38,28% |
| Nível 3 | 44,26% |
| Nível 4 | 46,90% |
| Nível 5 | 50,41% |
| Nível 6 | 51,48% |
| Nível 7 | 64,06% |

Correlação - Níveis de decomposição de Daubechies



(c)

| Daubechies | 100x, 60min – 120min |
|------------|----------------------|
| Nível 1 | 54,05% |
| Nível 2 | 63,93% |
| Nível 3 | 71,71% |
| Nível 4 | 76,14% |
| Nível 5 | 79,05% |
| Nível 6 | 80,54% |
| Nível 7 | 82,49% |



(d)

Figura 7.16 – Correlação das imagens de borracha natural com ouro para Daubechies, entre (a) 6 minutos e 9 minutos. (b) 9 minutos e 30 minutos. (c) 30 minutos e 60 minutos. (d) 60 minutos e 120 minutos.

7.8. Algoritmos referentes à técnica de emparelhamento

A técnica de emparelhamento implementada neste estudo é uma versão modificada do algoritmo apresentado por Dirk-Jan Kroon, professor da Universidade de Twente, Países Baixos[49]. O algoritmo utiliza a correlação cruzada normalizada, como abordado por J. P. Lewis em [44].

A técnica de emparelhamento é aplicada utilizando-se uma imagem modelo e uma imagem alvo. O algoritmo responsável, presente na função `template_matching.m`, é dado no Apêndice E. Este algoritmo é invocado pelas instruções a seguir:

```
[I_SSD, I_NCC] = template_matching(temp_image, targ_image);
% Encontrando a correspondência máxima na imagem I_SDD
[x,y] = find(I_SSD == max(I_SSD(:)));
```

onde `temp_image` e `targ_image` são, respectivamente, as imagens modelo e alvo, `I_SSD` é a soma da diferença quadrática e `I_NCC` é a correlação cruzada normalizada entre as imagens.

O algoritmo foi testado em imagens de microscopia óptica e de força atômica, retirando-se uma região da imagem e a utilizando como modelo. Exemplos da aplicação deste algoritmo são dados a seguir.

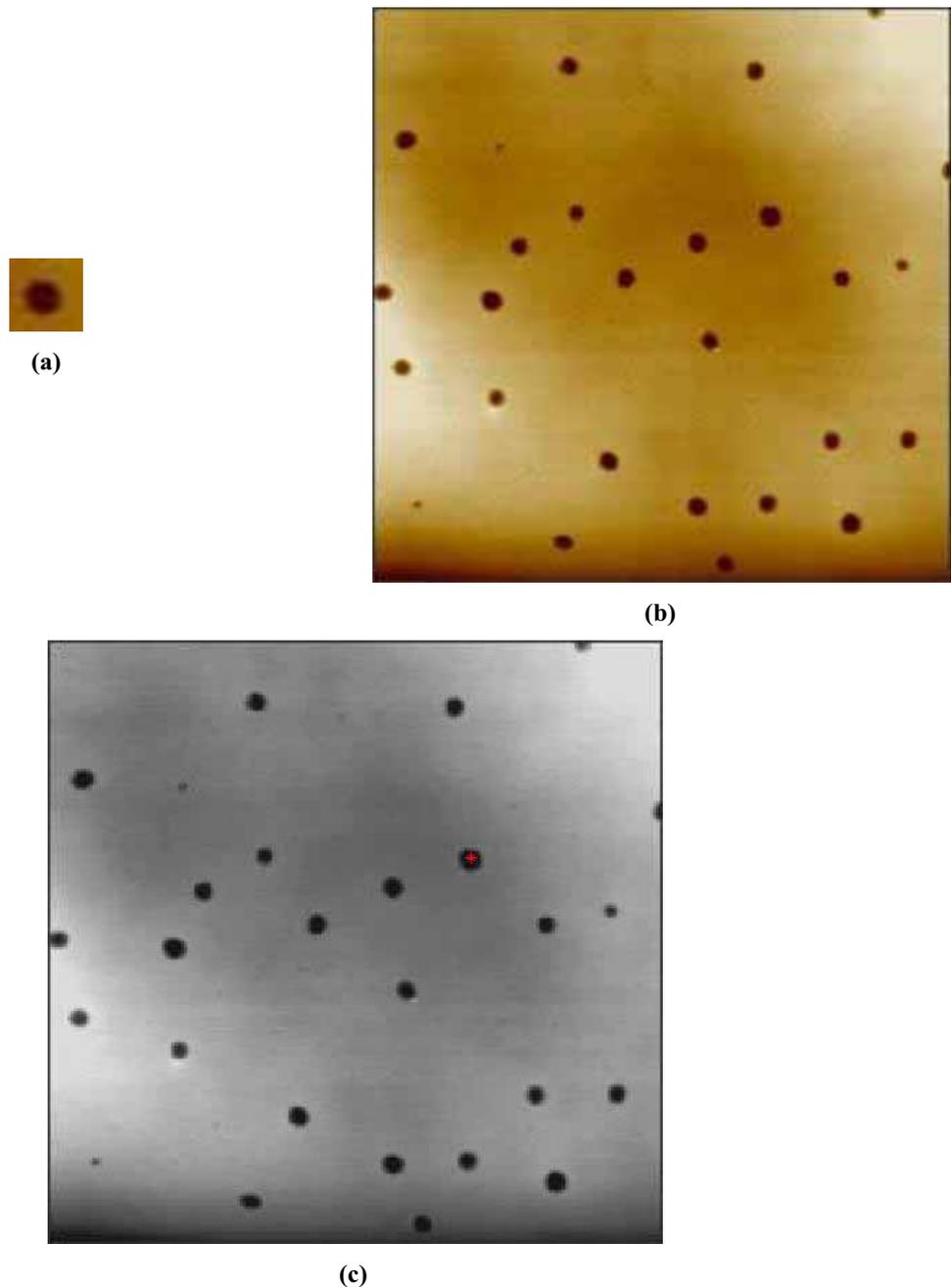
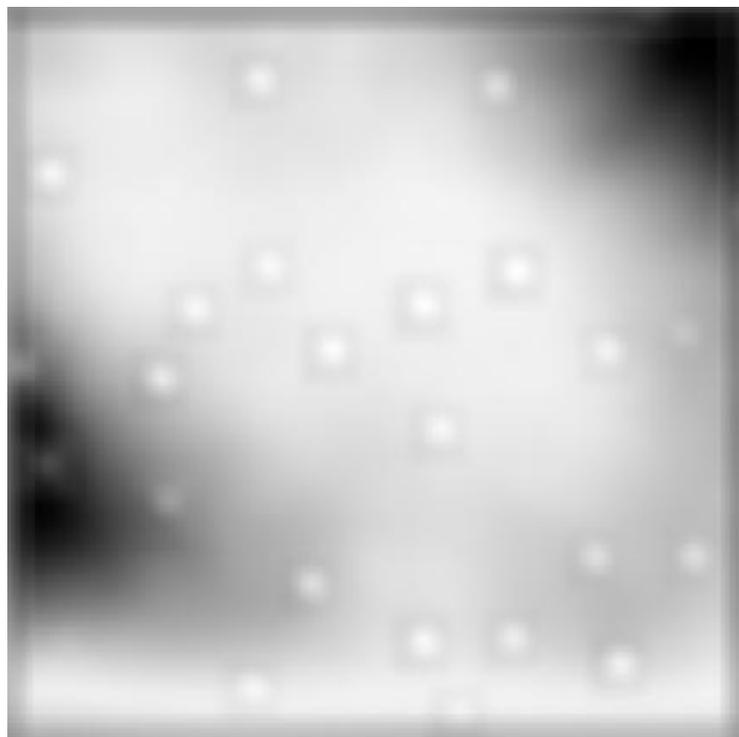
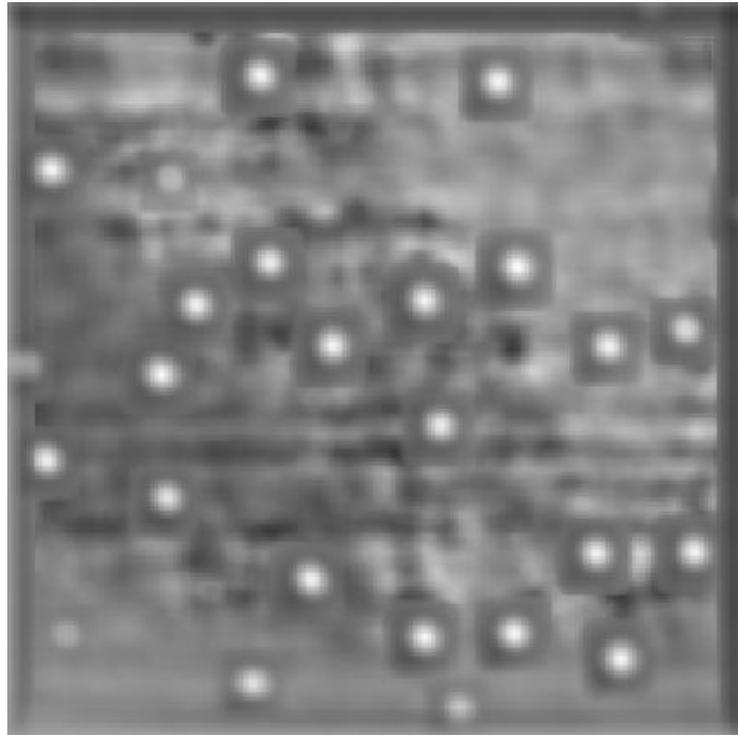


Figura 7.17 – Aplicação do algoritmo de emparelhamento em uma imagem de borracha natural obtida por microscopia de força atômica. **(a)** imagem modelo; **(b)** imagem alvo; **(c)** resultado.

A Figura 7.17 mostra o resultado obtido da aplicação do algoritmo em uma imagem de borracha natural proveniente de microscopia de força atômica. Pode-se notar que há várias regiões semelhantes na imagem alvo, porém os algoritmos implementados selecionam apenas um. A semelhança destas regiões com a imagem modelo pode ser afirmada a partir das imagens da soma da diferença quadrática e da correlação cruzada normalizada, dadas na Figura 7.18.



(a)

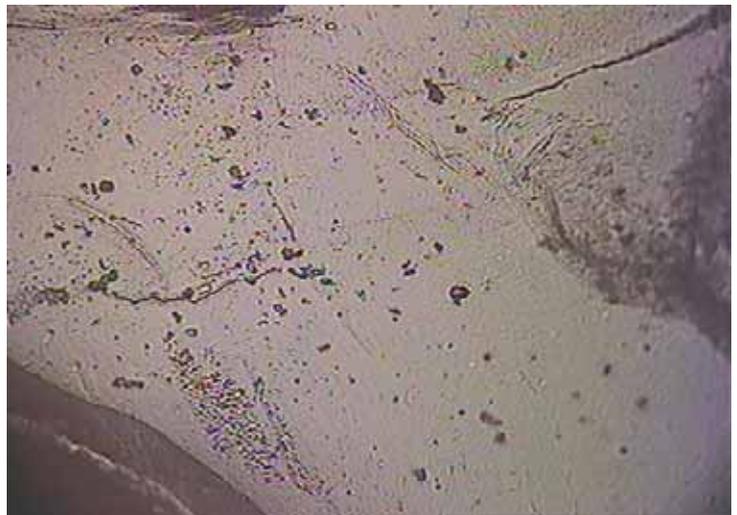


(b)

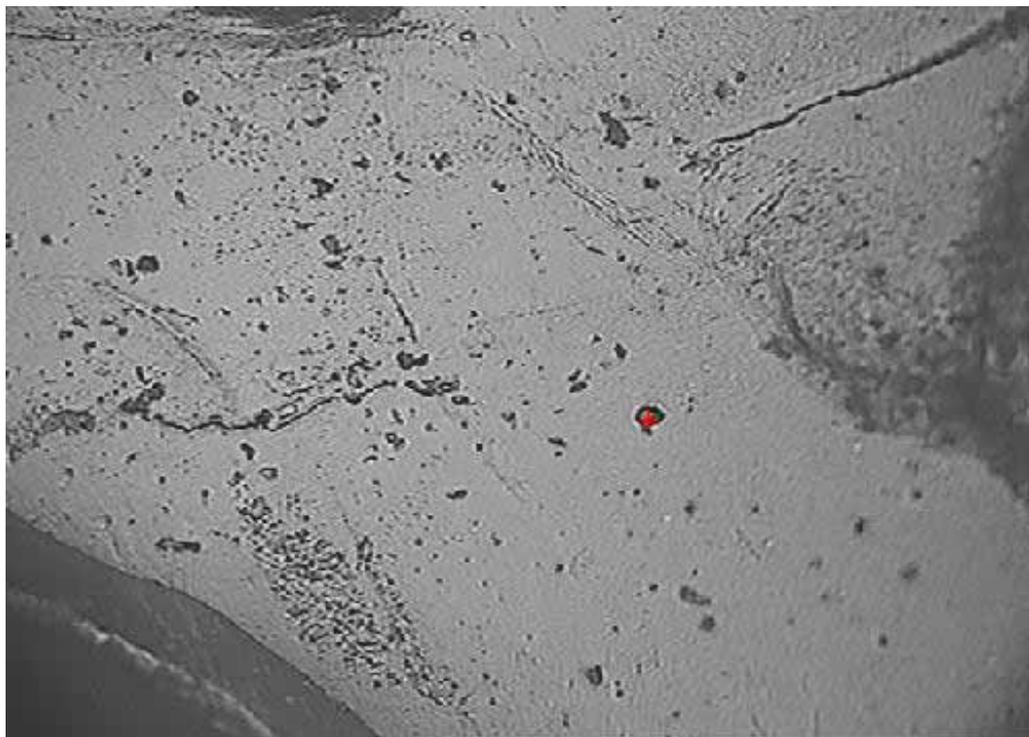
Figura 7.18 – As regiões semelhantes à imagem modelo detectadas na imagem alvo da Figura 7.17. **(a)** Soma da diferença quadrática das imagens. **(b)** Correlação cruzada normalizada.



(a)



(b)



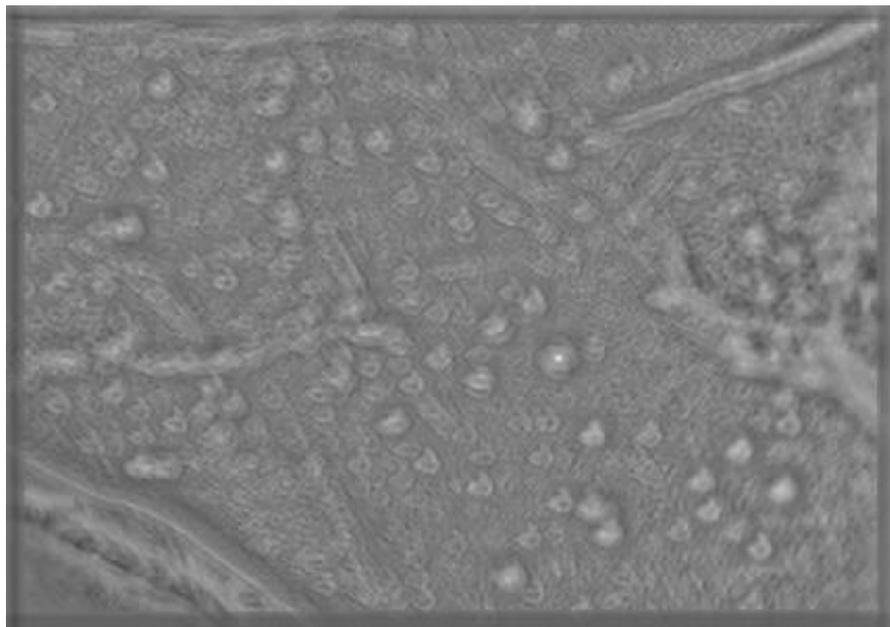
(c)

Figura 7.19 – Aplicação do algoritmo de emparelhamento em uma imagem de borracha natural com aditivo metálico, obtida por microscopia óptica. **(a)** Imagem modelo; **(b)** imagem alvo; **(c)** resultado.

A Figura 7.19 expõe o resultado obtido da aplicação do algoritmo em uma imagem de borracha natural com ouro, proveniente de microscopia óptica. Aqui, percebe-se que há apenas uma região que tenha alto grau de semelhança com a imagem modelo; a semelhança desta região com o modelo é vista nas imagens da soma da diferença quadrática e da correlação cruzada normalizada, dadas na Figura 7.20. É ilustrada uma região mais clara em cada imagem da Figura 7.20, que exibe a porção da imagem semelhante ao modelo. Esta região é apontada corretamente pelo algoritmo implementado.



(a)



(b)

Figura 7.20 – As regiões semelhantes à imagem modelo detectadas na imagem alvo da Figura 7.19. **(a)** Soma da diferença quadrática das imagens. **(b)** Correlação cruzada normalizada.

CONCLUSÃO

A proposta inicial do trabalho foi estudar imagens de amostras de borracha natural pura ou com aditivos metálicos por meio das transformadas de Fourier e Gabor, e das *wavelets* de Haar e Daubechies. Algoritmos foram criados para a implementação destas transformadas, sendo testados em imagens provenientes de microscopia eletrônica de varredura, óptica de força atômica.

Durante os estudos foi criado um *software* que realiza a aplicação das transformadas nas imagens, simplificando a utilização dos algoritmos designados para este fim. Como produto final tem-se um programa de fácil uso, que integra as ferramentas citadas no trabalho. O *software* WaveFPR pré-processa a imagem de entrada e aplica as transformadas citadas, de acordo com a escolha do usuário. Este *software* se encontra em processo de registro.

Os coeficientes relacionados às imagens, assim como os espectros e suas projeções, foram discriminados e salvos para posterior comparação. Tais informações integram um banco de dados em diretórios, que poderá ser disponibilizado para futuras consultas.

Os resultados obtidos com as transformadas para imagens de amostras obtidas de forma semelhante foram analisados pela técnica estatística de correlação, a fim de verificar a relação existente no processamento destas imagens por uma transformada. As *wavelets* de Daubechies e Haar mostraram melhor relação entre os coeficientes gerados a partir das imagens de teste. A transformada de Fourier não apresentou correlação aparente entre a maioria das imagens de teste, pois seu espectro varia drasticamente com a mudança de um único ponto na imagem.

As imagens de microscopia óptica analisadas apresentaram semelhanças em seus espectros, quando da aplicação das *wavelets* de Haar e Daubechies. A projeção da transformada de Fourier de algumas imagens obtidas por microscopia de força atômica apresentou “traços” horizontais e/ou verticais, indicando que a imagem relacionada perde magnitude nas frequências mais altas. Logo, os componentes de baixa frequência contém mais informação sobre essas imagens do que os componentes de alta frequência.

O método de pré-processamento utilizado se mostrou extremamente eficaz em todas as imagens testadas. As imagens foram bem representadas em suas versões binárias, graças ao método de segmentação de Otsu. Assim, os resultados obtidos pelas transformadas são confiáveis.

O algoritmo de emparelhamento implementado trouxe resultados muito satisfatórios, sinalizando regiões sem erros nas imagens-alvo testadas. Os bons resultados obtidos são confirmados pela correlação cruzada entre as imagens.

Além das imagens provindas das amostras de borracha natural com aditivos metálicos, foram estudados também outros tipos de amostras; por exemplo, as imagens de amostras de borracha natural com couro. Tal aplicação também obteve sucesso, mostrando que o programa, em sua atual composição, pode ser utilizado para aplicar as transformadas citadas em variadas imagens de amostras com eficácia.

Desta pesquisa surgiram várias idéias para trabalhos futuros:

- Estudar as imagens obtidas com outros tipos de transformadas, comparando assim os resultados obtidos. Tais transformadas poderiam ser outros tipos de *wavelets*, como as *shearlets*, *coiflets*, a transformada de Zak, entre outras. Da mesma forma, o autor espera estudar e conhecer melhor a análise multirresolução, abordando outras “pirâmides”, como a laplaciana e a gaussiana.

- Aprimorar o algoritmo para a técnica de emparelhamento, de forma que aponte mais de uma região em seu resultado. Assim, o algoritmo será mais completo para a caracterização das nanopartículas apresentadas nesta pesquisa.
- Estudar e utilizar a Análise de Séries Temporais para estimar a morfologia das amostras de borracha natural com aditivo metálico obtidas em vários tempos de deposição. Tomando, por exemplo, a *wavelet* de Daubechies de imagens obtidas de amostras de borracha natural com prata, depositadas por 6 minutos, 10 minutos, 30 minutos, e assim por diante, espera-se, com a Análise de Séries Temporais, determinar a aparência da estrutura desta amostra em um tempo dado.
- Segmentar a imagem de entrada tendo em vista sua textura, para conseguir uma análise morfológica diferenciada. O pré-processamento realizado nas imagens desta pesquisa visa a separação das regiões da imagem, e o autor acredita que a segmentação por textura traria informações diferentes sobre as imagens e seus materiais.
- Ampliar o banco de dados, aplicando as transformadas em imagens de diferentes materiais.
- Utilizar o conhecimento adquirido para desenvolver uma nova metodologia de Análise Morfológica de membranas. A correlação alta entre as imagens com a utilização das *wavelets* também se dá pela abordagem pesquisada aqui. O pré-processamento voltado às regiões e a análise multirresolução da imagem obtida relacionam as imagens semelhantes por superfície, possibilitando uma diferente metodologia de análise da estrutura do material.

REFERÊNCIAS

- [1] MORTON, M. **Rubber Technology**. 3. ed. Nova Iorque: Springer, 1987.
- [2] AGOSTINI, D. L. S. **Caracterização dos constituintes do látex e da borracha natural que estimulam a angiogênese**. Universidade Estadual Paulista. Presidente Prudente, p. 101. 2009. Dissertação (Mestrado em Ciência e Tecnologia de Materiais).
- [3] WEB OF SCIENCE. Web of Knowledge - Science - Thomson Reuters. **Web of Knowledge**. Disponível em: <http://www.isiwebofknowledge.com/>. Acesso em: 14 abr. 2010. Disponibiliza informações nas áreas das ciências, ciências sociais, artes e humanidade em mais de 9000 jornais de pesquisa.
- [4] CABRERA, F. C. **Preparação e caracterização de membranas de borracha natural com nanopartículas de ouro**. Universidade Estadual Paulista. Presidente Prudente, p. 69. 2009. Trabalho de conclusão de curso (Graduação em Licenciatura em Física).
- [5] SILVA, J. Q. E. A. Sistemas de exploração de seringueira utilizados em clones asiáticos Prang Besar no Oeste Paulista. **Pesquisa Agropecuária Brasileira**, Brasília, v. 42, n. 7, p. 949-955, jul. 2007.
- [6] PINTO, P. **Técnicas Laboratoriais de química – Obtenção de um elastômero e vulcanização do látex**. Escola Secundária do Padre António Martins Oliveira. Lagoa. 2004. Disponível em: http://pedropinto.com/files/secondary/tlq/tlqII_relatorio6.pdf.
- [7] AGOSTINI, D. L. S.; CONSTANTINO, C. J. L.; JOB, A. E. Thermal degradation of both latex and latex cast films forming membranes – Combined TG/FTIR investigation. **Journal of Thermal Analysis and Calorimetry**, v. 91, n. 3, p. 703-707, set. 2008.
- [8] PROLATEX COMÉRCIO DE LATEX LTDA., UNIVERSIDADE ESTADUAL DE CAMPINAS. **Desenvolvimento de metodologia aplicável à obtenção de látex de borracha natural catiônico em escala comercial**. Programa Inovação Tecnológica em Pequenas Empresas – PIPE. [S.l.]. 2004-2009. Processo FAPESP 04/02898-5.
- [9] MINISTÉRIO DA INDÚSTRIA E DO COMÉRCIO. SUPERINTENDÊNCIA DA BORRACHA. **Curso básico em tecnologia de elastômeros**. [S.l.]: [s.n.], v. 1-6, 1993.
- [10] BITTENCOURT, P. R. S. **Náilon 6 e 6,6 em Blendas com Lignina Kraft Modificada por Ácido Fórmico: Caracterização e Efeito de Radiação UV**. Universidade Estadual de Maringá. Maringá, p. 118. 2008. Tese (Doutorado em Química Aplicada).
- [11] NICOLA, G. L. **Análise de superfícies fresadas por diferentes estratégias em aço AISI H13 endurecido**. Universidade de Caxias do Sul. Caxias do Sul, p. 137. 2008. Dissertação (Mestrado em Materiais).
- [12] GONZALEZ, R. C.; WOODS, R. E. **Digital Image Processing**. 3. ed. New Jersey:

- Prentice Hall, 2007.
- [13] BASTOS, V. P. Técnicas de Segmentação de Imagens para Recuperação de Informações Visuais. Disponível em: <http://atlas.ucpel.tche.br/~vbastos/index.htm>. Acesso em: 22 jan. 2010.
- [14] HANSELMAN, D.; LITTLEFIELD, B. R. **Mastering Matlab 6**. New Jersey: Prentice Hall, 2007.
- [15] MALLAT, S. **A Wavelet tour of Signal Processing**. 2. ed. Londres: Academic Press, 1999.
- [16] ACHARYA, T.; TSAI, P. **JPEG2000 standard for image compression: concepts, algorithms and VLSI architectures**. New Jersey: John Wiley & Sons, 2005.
- [17] THE NOBEL PRIZE IN PHYSICS 1971. Disponível em: http://nobelprize.org/nobel_prizes/physics/laureates/1971/. Acesso em: 17 out 2010. Apresenta dados sobre Dennis Gabor, o laureado com o Prêmio Nobel de Física no ano de 1971.
- [18] SIQUEIRA, A. F.; MENEGUETTE, M.; JOB, A. E. Estudos de imagens digitais de borracha natural com aditivos metálicos utilizando a transformada de Fourier. **Anais do VI Workshop de Visão Computacional**, Presidente Prudente, jul 2010. Disponível em http://iris.sel.eesc.usp.br/wvc/anais_WVC2010/artigos/poster/72798.pdf.
- [19] BRACEWELL, R. N. **The Fourier Transform and its applications**. 3. ed. Singapore: McGraw-Hill, 2000.
- [20] CHEN, G. **Applications of Wavelet Transforms in Pattern Recognition and denoising**. Concordia University. Montreal, p. 82. 1999. Dissertação (Master of Computer Science).
- [21] GABOR, D. Theory of Communication. **J. Inst. Elec. Eng.**, v. 93, n. 3, p. 429-457, nov 1946.
- [22] CHUI, C. K. **An introduction to Wavelets**. Londres: Academic Press, 2003.
- [23] QIAN, S.; CHEN, D. Discrete Gabor Transform. **IEEE Transactions on Signal Processing**, v. 41, n. 7, p. 2429-2439, jul 1993.
- [24] BASTIAANS, M. J. Gabor's expansion of a signal into gaussian elementary signals. **Proc. IEEE**, v. 68, p. 594-598, abr 1980.
- [25] SONDERGAARD, P. L. An efficient algorithm for the Discrete Gabor Transform using full length windows. **Sampling Theory in Signal and Image Processing**, Marseilles, 2009.
- [26] LIMA, P. C. Wavelets: uma introdução. **Revista Matemática Universitária**, v. 33, p. 13-44, dez 2002. Disponível em http://rmu.sbm.org.br/Conteudo/n33/n33_Artigo02.pdf.
- [27] DAUBECHIES, I. **Ten Lectures on Wavelets**. Philadelphia: Society for Industrial and

- Applied Mathematics, v. 61, 1992.
- [28] SIDDIQI, A. H. **Applied Functional Analysis: Numerical Methods, Wavelet Methods, and Image Processing**. Nova Iorque: CRC Press, 2003.
- [29] ARNEODO, A.; BACRY, E. F.; MUZY, J. The thermodynamics of fractals revisited with wavelets. **Physica A: Statistical and Theoretical Physics**, v. 213, n. 1-2, p. 232-275, jan. 1995.
- [30] MALLAT, S.; HWANG, W. L. **Singularity detection and processing with wavelets**. New York University. Nova Iorque, p. 62. 1991.
- [31] CHAARI, O.; MEUNIER, M.; BROUAYE, F. Wavelets: a new tool for the resonant grounded power distribution systems relaying. **IEEE Transactions on Power Delivery**, v. 11, n. 3, p. 1301-1308, jul. 1996.
- [32] SHAKEN, F.; DEGHAN, M. Solution of delay differential equations via a homotopy perturbation method. **Mathematical and Computer Modelling**, v. 48, n. 3-4, p. 486-498, ago. 2008.
- [33] PITNER, S.; KAMARTHI, S. V. Feature extraction from wavelet coefficients for pattern recognition tasks. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 21, n. 1, p. 83-88, jan. 1999.
- [34] GUNASEKARAN, S.; REVATHY, K. Spectral Fluctuation Analysis for Audio Compression Using Adaptive Wavelet Decomposition. **Communications in Computer and Information Science**, 2010, Volume 70, 424-429, DOI: 10.1007/978-3-642-12214-9_71, v. 70, p. 424-429, 2010.
- [35] COIFMAN, R. R.; MAGGIONI, M. Geometry analysis and signal processing on digital data, emergent structures, and knowledge building. **SIAM News**, v. 41, n. 10, dec. 2010.
- [36] KOPPARAPU, S. K.; DESAI, U. B. **Bayesian approach to image interpretation**. Norwell: Kluwer Academic Publishers, 2001.
- [37] BOGGESE, A.; NARCOWICH, F. J. **First course in Wavelets with Fourier Analysis**. New Jersey: Prentice Hall, 2001.
- [38] WANG, J. Z. Methodological Review - Wavelets and Imaging Informatics: A Review of the Literature. **Journal of Biomedical Informatics**, v. 34, n. 2, p. 129-141, abr 2001.
- [39] SOUZA, E. M. **Análise de wavelets para detecção e correção do multicaminho no posicionamento relativo GNSS estático e cinemático**. Universidade Estadual Paulista. Presidente Prudente, p. 171. 2008. Tese (Doutorado em Ciências Cartográficas).
- [40] REIS, A. J. R.; SILVA, A. P. A. Aplicação da transformada wavelet discreta na previsão de carga a curto prazo via redes neurais. **SBA: Revista Controle & Automação**, v. 15, n. 1, p. 101-108, mar 2004. Disponível em http://www.scielo.br/scielo.php?script=sci_pdf&pid=S0103-17592004000100013&lng=pt&nrm=iso&tlng=pt.
- [41] NASON, G. P. **Wavelet Methods in Statistics with R**. Londres: Springer, 2008.

- [42] BRIECHLE, K.; HANEBECK, U. D. Template Matching using fast Normalized Cross Correlation, 2001. Disponível em http://i81pc23.itec.uni-karlsruhe.de/Publikationen/SPIE01_BriechleHanebeck_CrossCorr.pdf.
- [43] GOSE, E.; JOHNSONBAUGH, R.; JOST, S. **Pattern Recognition and Image analysis**. New Jersey: Prentice Hall, 1996.
- [44] LEWIS, J. P. Fast Template Matching. **Vision Interface**, p. 120-123, mai 1995.
- [45] OTSU, N. A. Threshold Selection Method from Gray-Level Histograms. **IEEE Transactions on Systems, Man and Cybernetics**, v. 9, n. 1, p. 62-66, 1979.
- [46] YANG, G. **MATLAB CENTRAL**. Disponível em: <http://www.mathworks.com/matlabcentral/fileexchange/23253>. Acesso em: 24 set 2010.
- [47] HAMMOUDA, K.; JERNIGAN, E. **Texture Segmentation using Gabor Filters**. Waterloo University. Waterloo. Disponível em <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.112.6669&rep=rep1&type=pdf>.
- [48] MORETTIN, P. A. **Ondas e ondaletas**. São Paulo: EdUSP, 1999.
- [49] KROON, D. J. **MATLAB CENTRAL**. Disponível em: <http://www.mathworks.com/matlabcentral/fileexchange/24925-fastrobust-template-matching>. Acesso em: 16 abr 2011.

APÊNDICES

APÊNDICE A – O TEOREMA DE BALIAN-LOW

O teorema de Balian-Low é dado neste apêndice. Este teorema dá uma condição para que bases de Gabor sejam ortonormais, e está relacionado com o conhecido princípio da incerteza de Heisenberg, um enunciado da mecânica quântica, formulado por Werner Heisenberg em 1927.

O enunciado do teorema de Balian-Low é precedido pela definição das bases de Gabor.

Definição (bases de Gabor). Seja $\mathfrak{R} = \bigcup_n [n, n+1]$, com n inteiro. Para cada um dos intervalos $[n, n+1]$ considera-se a base ortonormal trigonométrica, e cada elemento é multiplicado por uma função regular. O sistema resultante é denominado base de Gabor, dado por

$$\{g_{m,n}(x) = e^{2\pi imx} g(x-n) : m, n \in \mathbb{Z}\}.$$

A condição para que este sistema seja uma base ortonormal de $L^2(\mathfrak{R})$ é dada pelo teorema.

Teorema (de Balian-Low). Seja $g \in L^2(\mathfrak{R})$, e $g_{m,n}(x)$ os elementos de uma base de Gabor. Então,

$\{g_{m,n} : m, n \in \mathbb{Z}\}$ é uma base ortonormal de $L^2(\mathfrak{R})$ se, e só se $\int_{-\infty}^{\infty} x^2 |g(x)|^2 dx = \infty$ ou

$$\int_{-\infty}^{\infty} \xi^2 |G(\xi)|^2 d\xi = \infty.$$

Pode-se relacionar a primeira integral do enunciado com a velocidade de uma partícula, e a segunda integral com o momento desta partícula. O produto destas duas quantidades

tem como cota inferior a constante de Planck, muito importante em mecânica quântica. A constante de Planck tem o valor aproximado a $6,626 \times 10^{-34}$.

APÊNDICE B – A OPERAÇÃO DE CONVOLUÇÃO

Aqui são listados alguns teoremas e propriedades relativas à operação de convolução. Pode-se pensar na operação de convolução entre duas funções como uma média de uma função contra a outra. Tal operação se torna muito útil no trabalho com transformadas de Fourier, como mostrado.

A definição formal de convolução é dada abaixo.

Definição (convolução). Dadas duas funções f e g , define-se sua convolução $f * g$ por

$$f * g(x) = \int_{-\infty}^{\infty} f(x)g(t-x)dx.$$

O teorema a seguir mostra que a transformada de Fourier de uma convolução é dada pelo produto das transformadas das funções.

Teorema (convolução – tempo). Seja $g = f * h$. Então, $G(\xi) = F(\xi) H(\xi)$.

Demonstração.

Dados $F(g(x)) = G(\xi)$, $F(f(x)) = F(\xi)$ e $F(h(x)) = H(\xi)$, tem-se que

$$\begin{aligned} G(\xi) &= F(f(x)*h(x)) \\ &= F\left(\int_{-\infty}^{\infty} f(x)h(t-x)dx\right) \\ &= \int_{-\infty}^{\infty} \left(\int_{-\infty}^{\infty} f(u)h(x-u)du\right)e^{-i\xi x} dx \end{aligned}$$

$$\begin{aligned} &= \int_{-\infty}^{\infty} f(u) \left(\int_{-\infty}^{\infty} h(x-u) e^{-i\xi x} dx \right) du \\ &= H(\xi) \int_{-\infty}^{\infty} f(u) e^{-i\xi u} du \\ &= F(\xi) H(\xi). \quad \square \end{aligned}$$

Com uma construção similar, mostra-se que a transformada de Fourier de um produto é dada pela convolução das transformadas das funções, ou seja, $F(f(x)g(x)) = F(\xi) * G(\xi)$. Este teorema resultante é chamado *teorema da convolução na frequência*.

APÊNDICE C – A CRIAÇÃO DOS GRÁFICOS REFERENTES ÀS *WAVELETS* DE DAUBECHIES

Neste apêndice é listado o processo de obtenção dos gráficos das funções-escala e *wavelets*-mãe de Daubechies, como ilustradas na Figura 4.3. Nesta etapa foram utilizados os *softwares* Matlab, Gnuplot e o algoritmo cascata de Mallat, dado também neste apêndice, no arquivo ‘*cascade.m*’.

O processo de construção dos gráficos se dá pelos seguintes passos:

1) Obtenha os coeficientes da *wavelet* de sua escolha. Os coeficientes relativos às funções-escala de várias *wavelets* de Daubechies podem ser encontrados em <http://musicdsp.org/showone.php?id=20>. Esta fonte é disponibilizada por Kazuo Hatano e Olli Niemitalo.

2) Grave estes coeficientes em um arquivo de texto. No Matlab, insira os coeficientes em um vetor ‘*h*’ para aplicação.

3) Construa ‘*g*’ (vetor *wavelet*) a partir de *h*. Por exemplo, se a *wavelet* escolhida é a D20, proceda da seguinte forma:

```
>> g = [h(20) -h(19) h(18) -h(17) h(16) -h(15) h(14) -h(13) h(12) -h(11)
h(10) -h(9) h(8) -h(7) h(6) -h(5) h(4) -h(3) h(2) -h(1)];
```

4) Aplique o algoritmo ‘*cascade.m*’ nos vetores ‘*g*’ e ‘*h*’, gerando ‘*s*’ e ‘*w*’.

```
>> [s,w] = cascade(5,h,g);
```

5) Grave os resultados obtidos no Matlab em arquivos ‘*dat*’ de nome *daubscale.dat* e *daubwave.dat*, com os comandos:

```
>> save('daubscale.dat', '-ascii', '-double', 's');
```

```
>> save('daubwave.dat', '-ascii', '-double', 'w');
```

6) Utilize no Gnuplot os arquivos previamente salvos. Os gráficos presentes neste trabalho foram gerados com as seguintes opções:

```
set style data lines
set zeroa
set key top
set term png
set out "daubwave.png"
pl "daubscale.dat" ti "função-escala" w lines lt 3 lw 2, "daubwave.dat"
ti "wavelet-mãe" w lines lt 1 lw 2
set out
```

Consulte a ajuda dos programas Matlab e Gnuplot para esclarecimentos com relação aos comandos. Abaixo segue o algoritmo da função 'cascade.m'.

- cascade.m

```
function [s,w] = cascade(n,cs,cw)

s = cs;
w = cw;
x2(1:2:length(w)*2) = w;
x2(2:2:end)=0;
x(1:2:length(s)*2) = s;
x(2:2:end)=0;

for i = 1:n
    s = conv(x,cs);
    w = conv(x2,cs);

    x2(1:2:length(w)*2) = w;
    x2(2:2:end)=0;
    x(1:2:length(s)*2) = s;
    x(2:2:end)=0;
end
end
```

**APÊNDICE D – FUNÇÕES PARA A APLICAÇÃO DA TRANSFORMADA DE GABOR,
transgabor.m E gaborfilter.m**

Neste apêndice são listadas as funções contendo os algoritmos para a aplicação da transformada de Gabor em imagens, tendo suas fontes citadas ao longo do texto e modificadas pelo autor. Estas funções, em conjunto, aplicam a transformada de Gabor em uma imagem de entrada.

- transgabor.m

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% TRANSGABOR FUNCTION %%%%%%%%%%%
%
% Author: Alexandre Jaguar Fioravante de Siqueira
%
% Modified version of the function gabor_example.m, obtained from
% http://www.mathworks.com/matlabcentral/fileexchange/23253-gabor-filter
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Copyright (c) 2009, Gao Yang
%All rights reserved.

%Redistribution and use in source and binary forms, with or without
%modification, are permitted provided that the following conditions are
%met:

% * Redistributions of source code must retain the above copyright
% notice, this list of conditions and the following disclaimer.
% * Redistributions in binary form must reproduce the above copyright
% notice, this list of conditions and the following disclaimer in
% the documentation and/or other materials provided with the distribution

%THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
%AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
%IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
%ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
%LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
%CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF

```

GABOR, transgabor.m E gaborfilter.m

```
%SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS  
%INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN  
%CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)  
%ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE  
%POSSIBILITY OF SUCH DAMAGE.
```

```
function [img_log, img_scale] = transgabor(img_name)  
  
% an example to demonstrate the use of gabor filter.  
% the results mimic:  
% http://matlabserver.cs.rug.nl/edgedetectionweb/web/edgedetection\_examples  
% .html  
% using default settings (except for in radians instead of degrees)  
%  
% also, apparently the scaling of the numbers is different from the example  
% software at  
% http://matlabserver.cs.rug.nl  
% but are consistent with the formulae shown there  
  
lambda = 16;  
theta = 0;  
psi = [0 pi/2];  
gamma = 0.5;  
bw = 1;  
N = 8;  
  
img_in = im2double(img_name);  
%img_in(:, :, 2:3) = []; % discard redundant channels, it's gray anyway  
img_out = zeros(size(img_in,1), size(img_in,2), N);  
  
for n=1:N  
    gb = gaborfilter(bw, gamma, psi(1), lambda, theta)...  
        + 1i * gaborfilter(bw, gamma, psi(2), lambda, theta);  
    % gb is the n-th gabor filter  
    img_out(:, :, n) = imfilter(img_in, gb, 'symmetric');  
    % filter output to the n-th channel  
    theta = theta + 2*pi/N;  
    % next orientation  
end  
  
img_out_disp = sum(abs(img_out).^2, 3).^0.5;  
% default superposition method, L2-norm  
  
img_out_disp = img_out_disp./max(img_out_disp(:));  
% normalize  
  
imshow(img_out_disp);  
  
img_log = fftshift(log(1+abs(img_out_disp)));  
img_scale = gscale(img_out_disp);  
imshow(img_scale); colormap(jet);
```

GABOR, transgabor.m E gaborfilter.m

- gaborfilter.m

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% GABORFILTER FUNCTION %%%%%%%%%%
%
% Function obtained from the address
% http://www.mathworks.com/matlabcentral/fileexchange/23253-gabor-filter
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Copyright (c) 2009, Gao Yang
%All rights reserved.

%Redistribution and use in source and binary forms, with or without
%modification, are permitted provided that the following conditions are
%met:

% * Redistributions of source code must retain the above copyright
% notice, this list of conditions and the following disclaimer.
% * Redistributions in binary form must reproduce the above copyright
% notice, this list of conditions and the following disclaimer in
% the documentation and/or other materials provided with the distribution

%THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
%AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
%IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
%ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
%LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
%CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
%SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
%INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
%CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
%ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
%POSSIBILITY OF SUCH DAMAGE.

function gb = gaborfilter(bw,gamma,psi,lambda,theta)
% bw = bandwidth, (1)
% gamma = aspect ratio, (0.5)
% psi = phase shift, (0)
% lambda= wave length, (>=2)
% theta = angle in rad, [0 pi)

sigma = lambda/pi*sqrt(log(2)/2)*(2^bw+1)/(2^bw-1);
sigma_x = sigma;
sigma_y = sigma/gamma;

sz = fix(8*max(sigma_y,sigma_x));
if mod(sz,2) == 0, sz = sz+1;
end

% alternatively, use a fixed size
% sz = 60;

```

GABOR, transgabor.m E gaborfilter.m

```
[x y] = meshgrid(-fix(sz/2):fix(sz/2),fix(sz/2):-1:fix(-sz/2));  
% x (right +)  
% y (up +)  
  
% Rotation  
x_theta = x*cos(theta)+y*sin(theta);  
y_theta = -x*sin(theta)+y*cos(theta);  
  
gb = exp(-  
.5*(x_theta.^2/sigma_x^2+y_theta.^2/sigma_y^2)).*cos(2*pi/lambda*x_theta+psi);  
imshow(gb/2+0.5);
```

APÊNDICE E – FUNÇÃO PARA A APLICAÇÃO DA TÉCNICA DE EMPARELHAMENTO, `template_matching.m`

Neste apêndice é listada a função que contém os algoritmos para a aplicação da técnica de emparelhamento, tendo suas fontes citadas ao longo do texto. Esta função aplica a técnica de emparelhamento, fornecidas as imagens modelo e alvo.

```
function [I_SSD,I_NCC,Idata] = template_matching(T,I,IdataIn)
% TEMPLATE_MATCHING is a cpu efficient function which calculates matching
% score images between template and an (color) 2D or 3D image.
% It calculates:
% - The sum of squared difference (SSD Block Matching), robust template
%   matching.
% - The normalized cross correlation (NCC), independent of illumination,
%   only dependent on texture
% The user can combine the two images, to get template matching which
% works robust with his application.
% Both measures are implemented using FFT based correlation.
%
%   [I_SSD,I_NCC,Idata]=template_matching(T,I,Idata)
%
% inputs,
%   T : Image Template, can be grayscale or color 2D or 3D.
%   I : Color image, can be grayscale or color 2D or 3D.
%   (optional)
%   Idata : Storage of temporary variables from the image I, to allow
%           faster search for multiple templates in the same image.
%
% outputs,
%   I_SSD: The sum of squared difference 2D/3D image. The SSD sign is
%           reversed and normalized to range [0 1]
%   I_NCC: The normalized cross correlation 2D/3D image. The values
%           range between 0 and 1
%   Idata : Storage of temporary variables from the image I, to allow
%           faster search for multiple templates in the same image.
%
% Example 2D,
%   % Find maximum response
%   I = im2double(imread('lena.jpg'));
%   % Template of Eye Lena
%   T=I(124:140,124:140,:);
%   % Calculate SSD and NCC between Template and Image
%   [I_SSD,I_NCC]=template_matching(T,I);
```

EMPARELHAMENTO, template_matching.m

```

% % Find maximum correspondence in I_SDD image
% [x,y]=find(I_SSD==max(I_SSD(:)));
% % Show result
% figure,
% subplot(2,2,1), imshow(I); hold on; plot(y,x,'r*'); title('Result')
% subplot(2,2,2), imshow(T); title('The eye template');
% subplot(2,2,3), imshow(I_SSD); title('SSD Matching');
% subplot(2,2,4), imshow(I_NCC); title('Normalized-CC');
%
% Example 3D,
% % Make some random data
% I=rand(50,60,50);
% % Get a small volume as template
% T=I(20:30,20:30,20:30);
% % Calculate SSD between template and image
% I_SSD=template_matching(T,I);
% % Find maximum correspondence
% [x,y,z]=ind2sub(size(I_SSD),find(I_SSD==max(I_SSD(:))));
% disp(x);
% disp(y);
% disp(z);
%
% Function is written by D.Kroon University of Twente (February 2011)

if(nargin<3), IdataIn=[]; end

% Convert images to double
T=double(T); I=double(I);
if(size(T,3)==3)
    % Color Image detected
    [I_SSD,I_NCC,Idata]=template_matching_color(T,I,IdataIn);
else
    % Grayscale image or 3D volume
    [I_SSD,I_NCC,Idata]=template_matching_gray(T,I,IdataIn);
end

function [I_SSD,I_NCC,Idata]=template_matching_color(T,I,IdataIn)
if isempty(IdataIn), IdataIn.r=[]; IdataIn.g=[]; IdataIn.b=[]; end
% Split color image, and do template matching on R,G and B image
[I_SSD_R,I_NCC_R,Idata.r]=template_matching_gray(T(:,:,1),I(:,:,1),IdataIn.r);
[I_SSD_G,I_NCC_G,Idata.g]=template_matching_gray(T(:,:,2),I(:,:,2),IdataIn.g);
[I_SSD_B,I_NCC_B,Idata.b]=template_matching_gray(T(:,:,3),I(:,:,3),IdataIn.b);
% Combine the results
I_SSD=(I_SSD_R+I_SSD_G+I_SSD_B)/3;
I_NCC=(I_NCC_R+I_NCC_G+I_NCC_B)/3;

function [I_SSD,I_NCC,Idata]=template_matching_gray(T,I,IdataIn)
% Calculate correlation output size = input size + padding template
T_size = size(T); I_size = size(I);
outside = I_size + T_size-1;

% calculate correlation in frequency domain
if(length(T_size)==2)
    FT = fft2(rot90(T,2),outside(1),outside(2));

```

EMPARELHAMENTO, template_matching.m

```

    if(isempty(IdataIn))
        Idata.FI = fft2(I,outsized(1),outsized(2));
    else
        Idata.FI=IdataIn.FI;
    end
    Icorr = real(iff2(Idata.FI.* FT));
else
    FT = fftn(rot90_3D(T),outsized);
    FI = fftn(I,outsized);
    Icorr = real(iffn(FI.* FT));
end

% Calculate Local Quadratic sum of Image and Template
if(isempty(IdataIn))
    Idata.LocalQSumI= local_sum(I.*I,T_size);
else
    Idata.LocalQSumI=IdataIn.LocalQSumI;
end

QSumT = sum(T(:).^2);

% SSD between template and image
I_SSD=Idata.LocalQSumI+QSumT-2*Icorr;

% Normalize to range 0..1
I_SSD=I_SSD-min(I_SSD(:));
I_SSD=1-(I_SSD./max(I_SSD(:)));

% Remove padding
I_SSD=unpadarray(I_SSD,size(I));

if (nargout>1)
    % Normalized cross correlation STD
    if(isempty(IdataIn))
        Idata.LocalSumI= local_sum(I,T_size);
    else
        Idata.LocalSumI=IdataIn.LocalSumI;
    end

    % Standard deviation
    if(isempty(IdataIn))
        Idata.stdI=sqrt(max(Idata.LocalQSumI-
(Idata.LocalSumI.^2)/numel(T),0));
    else
        Idata.stdI=IdataIn.stdI;
    end
    stdT=sqrt(numel(T)-1)*std(T(:));
    % Mean compensation
    meanIT=Idata.LocalSumI*sum(T(:))/numel(T);
    I_NCC= 0.5+(Icorr-meanIT)./(2*stdT*max(Idata.stdI,stdT/1e5));

    % Remove padding
    I_NCC=unpadarray(I_NCC,size(I));
end

```

EMPARELHAMENTO, template_matching.m

```

function T=rot90_3D(T)
T=flipdim(flipdim(flipdim(T,1),2),3);

function B=unpadarray(A,Bsize)
Bstart=ceil((size(A)-Bsize)/2)+1;
Bend=Bstart+Bsize-1;
if(ndims(A)==2)
    B=A(Bstart(1):Bend(1),Bstart(2):Bend(2));
elseif(ndims(A)==3)
    B=A(Bstart(1):Bend(1),Bstart(2):Bend(2),Bstart(3):Bend(3));
end

function local_sum_I= local_sum(I,T_size)
% Add padding to the image
B = padarray(I,T_size);

% Calculate for each pixel the sum of the region around it,
% with the region the size of the template.
if(length(T_size)==2)
    % 2D localsum
    s = cumsum(B,1);
    c = s(1+T_size(1):end-1,:)-s(1:end-T_size(1)-1,:);
    s = cumsum(c,2);
    local_sum_I= s(:,1+T_size(2):end-1)-s(:,1:end-T_size(2)-1);
else
    % 3D Localsum
    s = cumsum(B,1);
    c = s(1+T_size(1):end-1,,:)-s(1:end-T_size(1)-1,,:);
    s = cumsum(c,2);
    c = s(:,1+T_size(2):end-1,:)-s(:,1:end-T_size(2)-1,:);
    s = cumsum(c,3);
    local_sum_I = s(:, :, 1+T_size(3):end-1)-s(:, :, 1:end-T_size(3)-1);
end

```