

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL  
FACULDADE DE INFORMÁTICA  
CIÊNCIA DA COMPUTAÇÃO

JOSUÉ DEMARTINI  
VINICIUS ALMEIDA RODRIGUES

**SOFTWARE PARA ESTUDO DAS CARACTERÍSTICAS TOPOGRÁFICAS DE  
IMAGENS GERADAS COM A TÉCNICA DE MOIRÉ**

Porto Alegre

2010

JOSUÉ DEMARTINI  
VINICIUS ALMEIDA RODRIGUES

**SOFTWARE PARA ESTUDO DAS CARACTERÍSTICAS TOPOGRÁFICAS DE  
IMAGENS GERADAS COM A TÉCNICA DE MOIRÉ**

Trabalho de conclusão de curso de graduação apresentado à Faculdade de Informática da Pontifícia Universidade Católica do Rio Grande do Sul, como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação.

**Orientador: Márcio Sarroglia Pinho**

Porto Alegre

2010

## RESUMO

Este trabalho apresenta uma ferramenta para o processamento de imagens geradas pelo fenômeno de Moiré. Este fenômeno é um padrão visual que surge quando as sombras de duas grades iguais são sobrepostas sobre um objeto. Este fenômeno é ideal para estudar a topografia de superfícies, pois gera linhas de contornos semelhantes às usadas em mapas topográficos, em que cada linha corresponde a uma altura. Estas informações tem sido usadas, por exemplo, para análise de postura do corpo de um ser humano, inspeção do formato de frutas, digitalização tridimensional de um objeto. Esta utilização, entretanto tem sido realizada por processos manuais, cansativos e repetitivos. A partir de algoritmos de processamento de imagens como suavização, segmentação, afinamento e vetorização foi desenvolvido um sistema computacional capaz de identificar características relevantes destas imagens, permitindo um uso mais simples, rápido e produtivo das imagens do fenômeno de Moiré.

## **ABSTRACT**

This work presents a software tool for processing images generated from the Moiré's phenomena. This phenomena appears on images when the shadows of two grids are cast over an object. This phenomena is interesting to study surface topography because it produces contours very similar to level curves used on topographical maps, in which every curve corresponds to a different height. This information has been used, for example, to analyze human poses, to inspect fruit's shape and to scan tridimensional objects. This applications, however, are usually manual and repetitive processes. Using image processing algorithms this work developed a computational system able to identify the main structures of these images, allowing an easier, fast and productive use of the Moiré's phenomena

## LISTA DE FIGURAS

Figura 1 Efeito Moiré gerado pela sobreposição de duas grades.....	13
Figura 2 “Franjas de Moiré” gerado pelo movimento da seda .....	14
Figura 3 Formação das franjas de Moiré pela interferência da luz ao passar por dois retículos .....	15
Figura 4 Franjas claras e escuras formadas pela intensidade da luz transmitida.....	15
Figura 5 Esquema proposto por Turner-Smith para obtenção da TMS .....	16
Figura 6 Configuração de Técnica de Moiré de Projeção.....	17
Figura 7 Representação da linha média da coluna .....	18
Figura 8 Moiré de Projeção aplicado sobre a fruta (A). Representação tridimensional em tons de cinza (B) e a sua representação tridimensional por malhas de contorno (C). .....	18
Figura 9 Técnica de Moiré aplicada na sola do pé .....	19
Figura 10 Franja de Moiré destacada.....	20
Figura 11 Pontos de inflexão formados pelo efeito de Moiré .....	21
Figura 12 Variação angular entre os pontos destacados a partir dos vértices formados pelo efeito de Moiré. ....	21
Figura 13 Período P entre duas franjas .....	22
Figura 14 Alinhamento entre as escápulas do indivíduo .....	23
Figura 15 Representação da inclinação de uma franja. ....	23
Figura 16 Aplicação da técnica de Moiré de sombra em uma calota.....	24
Figura 17 Seqüência de algoritmos aplicados sobre a imagem original para chegar ao resultado desejado armazenado na imagem resultante. ....	26
Figura 18 (a) Imagem Original, (b) Imagem Segmentada .....	27
Figura 19 Máscara 3x3 genérica .....	28
Figura 20 Demonstração do uso de uma máscara de convolução sobre um ponto da imagem.....	28
Figura 21 Matriz de convolução para detecção de pontos isolados .....	29
Figura 22 Máscaras para linhas .....	30
Figura 23 Máscara de Roberts .....	30
Figura 24 Máscara de Laplaciano .....	31
Figura 25 Exemplo de Histograma .....	32
Figura 26 Imagem e seu Histograma gerado pelo <i>software Gimp 2.6</i> .....	33

Figura 27 (a) matriz com os valores de nível de cinza de cada <i>pixel</i> ; (b) formação de duas regiões distintas de acordo com os valores dos <i>pixels</i> , azul (1, 2 e 3) e vermelho (4, 5 e 6). .....	34
Figura 28 Exemplo da estrutura de um Quadtree.....	35
Figura 29 Imagem original (esquerda), e seu esqueleto (direita) .....	36
Figura 30 Máscara para análise do <i>pixel P1</i> .....	37
Figura 31 Imagem e seu esqueleto gerado pelo algoritmo de Zhang-Suen .....	38
Figura 32 Máscara para visualização do método de afinamento de Holt .....	39
Figura 33 representação de conectividade.....	39
Figura 34 (a) código de encadeamento (b) imagem (c) vetorização .....	41
Figura 35 (a) primeira passagem (b) segunda passagem .....	41
Figura 36 Primeiro caso em que há falha. A imagem é percorrida de “A” até “C” .....	42
Figura 37 (a) caminho (“A B C D”) que deveria ser feito (b) caminhamento quando há erro .....	43
Figura 38 Aplicação da Técnica de Moiré de Sombra nas costas do paciente.....	44
Figura 39 Variação angular entre pontos de inflexão .....	46
Figura 40 Imagem normal (esq.) e Imagem com Smooth (dir.) .....	49
Figura 41 Resultado da segmentação .....	50
Figura 42 Imagem afinada pelo algoritmo implementado no trabalho (e) e pelo algoritmo de Skeletonization do software ImageJ (d).....	51
Figura 43 Imagem vetorizada - pixels roxos pontos gerados - pixels pretos são a imagem original aumentada .....	52
Figura 44 Franjas com intersecção. ....	53
Figura 45 Franjas destacadas com várias cores. ....	54
Figura 46 Imagem com franjas destacadas e com algumas partes de franjas selecionadas.....	56
Figura 47 Pontos máximo (verde) e mínimo (vermelho).....	58
Figura 48 Pontos de Inflexão: Vermelhos - soma dos pontos vizinhos são menores que 0. Azul - soma dos pontos vizinhos são maiores que 0. ....	59
Figura 49 Imagens originais. ....	61
Figura 50 Calota com aplicação da suavização .....	62
Figura 51 Suavização aplicada as Costas de uma pessoa .....	62
Figura 52 Exemplo de suavização aplicada as costas. ....	63
Figura 53 Suavização na face de um manequim.....	63

Figura 54 Segmentação aplicada a Figura 50.....	64
Figura 55 Segmentação aplicada a Figura 51.....	64
Figura 56 Segmentação aplicada a Figura 52.....	65
Figura 57 Segmentação aplicada a Figura 53.....	65
Figura 58 Imagem de manequim afinada.....	66
Figura 59 Imagem das costas afinada.....	66
Figura 60 Cálculo do Período.....	67
Figura 61 Cálculo do Período.....	67
Figura 62 Pontos de máximo e mínimo.....	68
Figura 63 Ponto de Máximo e Mínimo.....	69
Figura 64 Pontos de Inflexão.....	69
Figura 65 Pontos de Inflexão.....	70
Figura 66 Franjas das costas destacadas.....	70
Figura 67 Franjas do rosto destacadas.....	71
Figura 68 Franjas selecionadas a partir de uma imagem com as franjas das costas destacadas.....	71
Figura 69 Franjas selecionadas a partir de uma imagem com as franjas do rosto destacadas.....	72
Figura 70 Tela principal do <i>software</i> .....	78
Figura 71 Opção <i>Arquivo</i> da barra.....	79
Figura 72 Opção <i>Processos</i> da barra.....	79
Figura 73 Opção <i>Copia</i> da barra.....	80
Figura 74 Opção <i>Moiré da barra</i> .....	80

## LISTA DE ALGORITMOS

Algoritmo 1 - Detecção de descontinuidades .....	29
Algoritmo 2 - Vetorização.....	41
Algoritmo 3 - Suavização de Imagens .....	49
Algoritmo 4 - Destaca Franjas .....	54
Algoritmo 5 - Seleciona Franjas.....	55
Algoritmo 6 - Distância entre Duas Franjas .....	56
Algoritmo 7 - Acha Ponto máximo e mínimo.....	57
Algoritmo 8 - Acha ponto de inflexão .....	59
Algoritmo 9 - Ângulo .....	60

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b> .....	<b>10</b>
<b>2</b>	<b>MOTIVAÇÃO</b> .....	<b>12</b>
<b>3</b>	<b>CARACTERÍSTICAS E APLICAÇÕES DA TOPOGRAFIA DE MOIRÉ</b> .....	<b>13</b>
3.1	HISTÓRICO .....	13
3.2	FENÔMENO DE MOIRÉ.....	14
3.3	APLICAÇÕES.....	17
3.4	PRINCIPAIS CARACTERÍSTICAS DAS SOMBRAS DE MOIRÉ.....	20
<b>4</b>	<b>ALGORITMOS DE PROCESSAMENTO DE IMAGEM E VISÃO COMPUTACIONAL</b> .....	<b>25</b>
4.1	SEGMENTAÇÃO .....	26
4.1.1	<b>Detecção de descontinuidades</b> .....	<b>27</b>
4.1.2	<b>Detecção de similaridades</b> .....	<b>31</b>
4.2	AFINAMENTO.....	35
4.3	VETORIZAÇÃO .....	40
<b>5</b>	<b>SISTEMA DESENVOLVIDO</b> .....	<b>44</b>
<b>6</b>	<b>DETALHES DE IMPLEMENTAÇÃO DO PROCESSAMENTO DAS FRANJAS</b> 48	
6.1	PRÉ-PROCESSAMENTO DA IMAGEM .....	49
6.2	EXTRAÇÃO DAS CARACTERÍSTICAS DAS CURVAS DE MOIRÉ.....	52
6.2.1	<b>Destacar franjas</b> .....	<b>52</b>
	O RESULTADO DESTE ALGORITMO É MOSTRADO NA FIGURA 45. ....	54
6.2.2	<b>Selecionar franjas</b> .....	<b>54</b>
	O RESULTADO DESTE ALGORITMO PODE SER VISTO NA FIGURA 46. ....	55
6.2.3	<b>Cálculo do Período entre duas Franjas</b> .....	<b>56</b>
6.2.4	<b>Pontos de máximo e mínimo</b> .....	<b>56</b>
6.2.5	<b>Pontos de inflexão</b> .....	<b>58</b>
6.2.6	<b>Métodos auxiliares</b> .....	<b>59</b>
<b>7</b>	<b>RESULTADOS DE TESTES</b> .....	<b>61</b>
7.1	IMAGEM ORIGINAL .....	61
7.2	APLICAÇÃO DA SUAVIZAÇÃO (SMOOTH).....	61

7.3	APLICAÇÃO DA <i>SEGMENTAÇÃO</i> .....	63
7.4	APLICAÇÃO DO <i>AFINAMENTO</i> .....	65
7.5	APLICAÇÃO DO <i>CÁLCULO DO PERÍODO</i> .....	66
7.6	APLICAÇÃO DO <i>PONTO DE MÁXIMO E MÍNIMO</i> .....	68
7.7	APLICAÇÃO DO <i>PONTOS DE INFLEXÃO</i> .....	69
7.8	APLICAÇÃO DO <i>DESTACAR FRANJAS</i> .....	70
7.9	APLICAÇÃO DO <i>SELECIONAR FRANJAS</i> .....	71
<b>8</b>	<b>CONCLUSÃO</b> .....	<b>73</b>
<b>9</b>	<b>BIBLIOGRAFIA</b> .....	<b>74</b>
<b>10</b>	<b>APÊNDICE</b> .....	<b>78</b>

## 1 INTRODUÇÃO

A área de processamento de imagens (PI) vem permitindo a automatização de tarefas importantes, possíveis antes somente por métodos manuais. Pode-se fazer reconhecimento de áreas e objetos, trabalhar com filtros e efeitos e fazer modificações em imagens.

Processar uma imagem consiste em transformá-la, utilizando técnicas que permitam manipulá-la até alcançar o efeito desejado, seja destacando pontos desta figura, ou até aplicando filtros que a alterem. Quando se fala em transformação, busca-se a aplicação dos diversos algoritmos disponíveis na área da computação gráfica, como os algoritmos de segmentação, vetorização, detecção de bordas, entre outros, para poder trabalhar sobre uma imagem.

Uma das áreas em que o PI pode ser usado é na identificação das estruturas de sombra formadas pelo efeito de Moiré. O Efeito de Moiré é um padrão visual de interferência que pode ser gerado pela sobreposição de duas grades, posicionadas em ângulo de até 45° entre elas. Este fenômeno é ideal para estudar a topografia de superfícies, pois gera linhas de contornos semelhantes às usadas em mapas topográficos, em que cada linha corresponde a uma profundidade.

O objetivo é trabalhar sobre imagens, neste caso fotos que contenham a projeção do Efeito de Moiré de Sombra, e sobre estas facilitar a identificação das características topográficas da superfície sobre a qual está projetada a sombra de Moiré. A captura destas estruturas gera informações que podem ser usadas em diversas áreas, como análise de imagens médicas, reconstrução de objetos em 3D, análise de produtos em uma linha de produção, entre outros.

As imagens a serem usadas neste trabalho foram obtidas a partir da projeção das sombras de Moiré sobre corpos de seres humanos e objetos, pois conforme pode ser observado no trabalho de Ferreira (Ferreira, 2008), estas podem ser usadas para o diagnóstico de doenças tais como escoliose ou lordose. Cabe ressaltar, porém, que no presente trabalho são realizados esforços no sentido de identificar as principais características das curvas de Moiré, ficando tarefas como o diagnóstico ou a reconstrução 3D de objetos, fora do escopo deste trabalho.

Este documento está dividido da seguinte forma. No capítulo 2 apresenta-se a motivação para o projeto. Já o capítulo 3 descreve as características das curvas de

Moiré e como podem ser exploradas para obterem-se informações que possam ser usadas na prática. Na seção 4, é feita uma introdução sobre processamentos de imagens usados neste trabalho e também algoritmos utilizados na fase de pré-processamento. Os capítulos seguintes mostram como o sistema foi desenvolvido. O capítulo 5 faz uma apresentação do *software* e suas funcionalidades, enquanto o 6 mostra detalhes de como foram desenvolvidos os algoritmos que permitem ao usuário usufruir do sistema. No capítulo 7 encontram-se os resultados obtidos nos testes, bem como a explicação para cada um deles. As conclusões são apresentadas no capítulo 8. O capítulo 9 é a bibliografia utilizada para a construção do trabalho. No Apêndice, capítulo 10, é descrito como utilizar o sistema desenvolvido.

## 2 MOTIVAÇÃO

A técnica de Moiré vem sendo estudada desde 1874, quando duas grades iguais foram sobrepostas gerando uma sombra sobre um objeto (Rayleigh, 1874). Desde então muitos estudos foram realizados sobre possíveis aplicações desta técnica. Porém, a maioria destas aplicações ainda depende de métodos manuais, repetitivos e demorados para obter características importantes, o que incentivou o desenvolvimento de uma ferramenta que torne isso automático.

Sabe-se que é importante investir no desenvolvimento de ferramentas que aperfeiçoem os processos aplicados nas mais diversas áreas. Neste caso tem-se o exemplo da técnica de Moiré para o estudo de topografia, que por se tratar de um método barato, rápido e de fácil manipulação, foi aplicado em escolas no Japão para auxiliar na detecção de escoliose em crianças. Porém, apesar de agilizar os diagnósticos dos médicos, ainda havia muito trabalho manual de medição nas características que eles deveriam observar, o que os levava a exaustão. Devido à grande utilidade, a técnica de Moiré vem chamando a atenção de pesquisadores, e diversos trabalhos nas áreas da medicina, agricultura e estudo de superfícies, vêm sendo desenvolvidos nas últimas quatro décadas.

Este crescente interesse pela aplicação da técnica de Moiré, serve como motivação para implementar uma ferramenta que trabalhe no reconhecimento das características destas imagens topográficas.

Esta ferramenta poderá ser ampliada posteriormente em algum trabalho de conclusão, ou até mesmo servir como base para desenvolvedores que tenham interesse em implementar algo na área.

### 3 CARACTERÍSTICAS E APLICAÇÕES DA TOPOGRAFIA DE MOIRÉ

A Técnica de Moiré (TM) pode ser definida como um conjunto de métodos usados para medição de deformações em um plano, contornos topográficos, inclinações, curvaturas e formas dos objetos. A Técnica de Moiré mostra-se útil para coleta e manipulação de dados, pois de maneira ágil pode-se obter resultados tão confiáveis quanto os gerados por métodos mais complexos (Hertz H, 2005).

#### 3.1 Histórico

O estudo científico sobre a Técnica de Moiré iniciou em 1874, com Rayleigh (Rayleigh, 1874), quando ele sobrepôs duas grades iguais e observou o fenômeno (Figura 1).

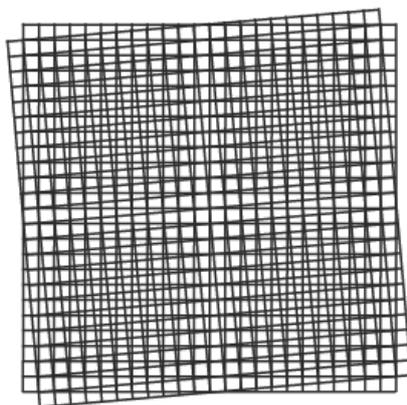


Figura 1 Efeito Moiré gerado pela sobreposição de duas grades

Fonte: (CreativePro, 2008)

O nome Moiré é de origem francesa e significa molhado ou ondulado. Refere-se também a um tecido de seda composto de duas camadas importado da China (Sciammarella, 1982). Quando em movimento, essas camadas geram padrões semelhantes a ondas, denominadas franjas de Moiré (Figura 2).



Figura 2 “Franjas de Moiré” gerado pelo movimento da seda

Fonte: (Libre, 2009)

Na década de 70, Takasaki (Takasaki, 1970) escreveu um artigo dizendo como é possível utilizar esta técnica para fazer topografias em corpos humanos, como alternativa para facilitar o diagnóstico de escoliose em homens. Neste artigo, Takasaki propôs que através das sombras (franjas) projetadas às costas, era possível ver algumas deformidades na coluna vertebral. A partir deste estudo muitos outros pesquisadores, de diversas áreas, passaram a estudar outras maneiras de obter as imagens e utilidades para esta técnica.

### 3.2 Fenômeno de Moiré

O fenômeno de Moiré é um efeito gerado quando duas estruturas formadas por curvas são sobrepostas, formando uma nova estrutura com ângulo de até  $45^\circ$ . Quando as duas estruturas têm o mesmo, ou ligeiramente diferentes, espaçamento e espessura de linha, geram uma interferência chamada franja.

As franjas de Moiré são sombras resultantes da interferência de uma grade, também chamada de retículo de referência ( $R_r$ ), posicionada entre o objeto que pretende-se analisar e uma fonte de luz. Quando o feixe de luz atravessa a tela de interferência, neste caso o Retículo de referência, produz um efeito de sombras em função da profundidade do alvo (Túlio Gomes, 2009). Segundo Lino (Lino, 2002), a intensidade da luz observada é na verdade a média da luz transmitida através das grades. Assim, onde a luz transmitida é máxima, tem-se o centro das franjas claras e, onde a luz transmitida é mínima, tem-se o centro das franjas escuras, como mostra a Figura 3, que demonstra uma luz incidindo sobre duas grades (Retículo 1 e

Retículo 2), explicando que a quantidade de luz que passa pelos intervalos maiores entre as duas grades formam as franjas claras, e os locais em que o intervalo é menor ou que não há intervalo algum para a luz passar, formam-se as franjas escuras. Na Figura 4, pode-se observar as franjas claras e escuras.

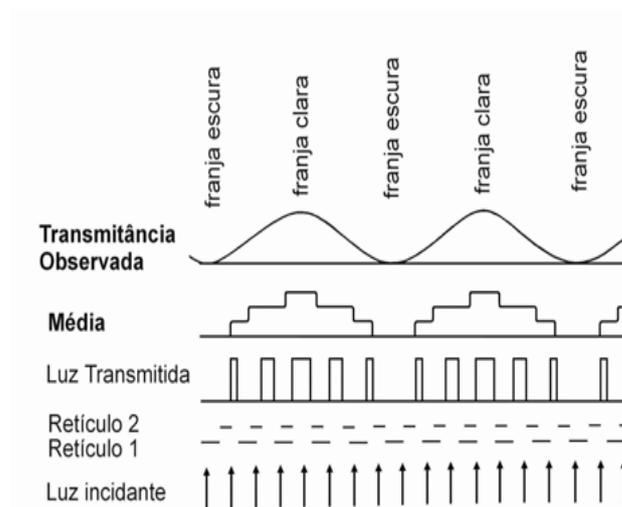


Figura 3 Formação das franjas de Moiré pela interferência da luz ao passar por dois retículos

Fonte: (Cloud, 1998)

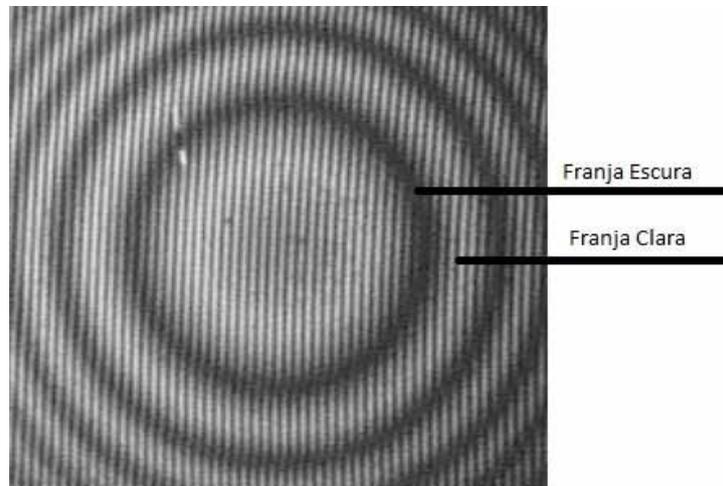


Figura 4 Franjas claras e escuras formadas pela intensidade da luz transmitida.

Fonte: O autor (2009)

As duas técnicas mais utilizadas para obter o efeito de Moiré são a Técnica de Moiré de Sombra (TMS) e a Técnica de Moiré de Projeção (TMP). Estas serão brevemente explicadas a seguir.

Na Técnica de Moiré de Sombra uma grade é colocada à frente do objeto que se deseja analisar. Quando a grade é iluminada por uma fonte de luz, a sombra dela

é projetada sobre a superfície do objeto, gerando o Retículo modelo ( $R_m$ ). O observador (ou câmera fotográfica) fica posicionado atrás da mesma grade, porém em posição e ângulo diferente da fonte de luz, a interferência gerada por essa diferença de ângulos, dará origem às franjas de Moiré.

O modelo proposto por Turner-Smith (Turner-Smith, 1997), mostra a configuração necessária para obter o efeito através da Técnica de Moiré de Sombra. Neste modelo, apresentado na Figura 5, pode-se observar uma fonte de luz  $P$  que gera um feixe de luz, que incide por uma grade  $G$ , gerando o fenômeno de Moiré observado na superfície  $S$ . No ponto  $O$  fica posicionado o observador.

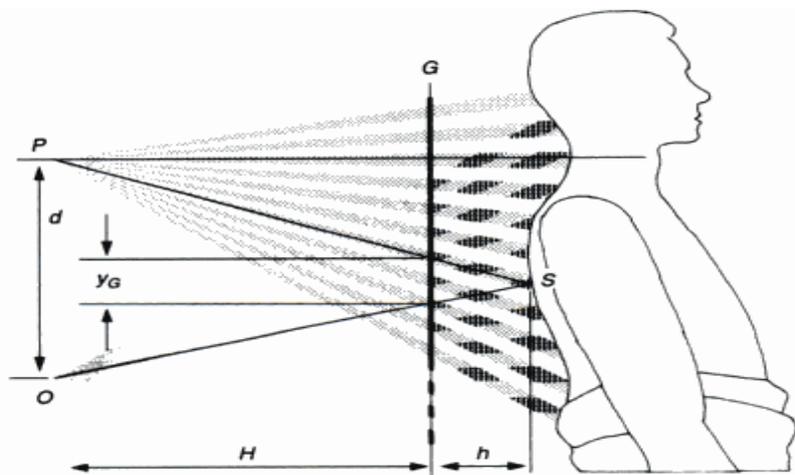


Figura 5 Esquema proposto por Turner-Smith para obtenção da TMS

Fonte: (Turner-Smith, 1997)

Outra possibilidade é o uso da Técnica de Moiré de Projeção (TMP), em que o mapa de franja é projetado sobre a superfície do objeto através da grade de um projetor. A câmera capta a interferência das franjas deformadas com uma segunda grade (Figura 6).

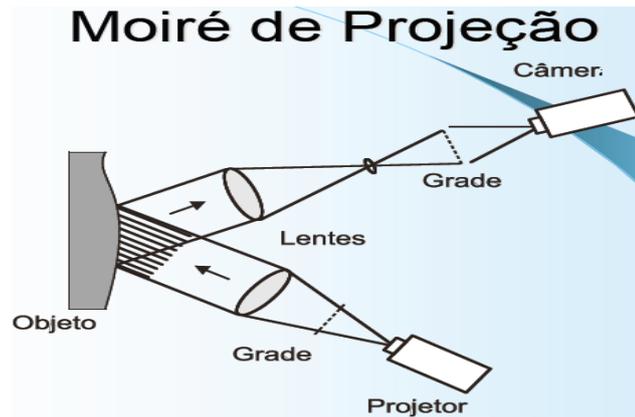


Figura 6 Configuração de Técnica de Moiré de Projeção

Fonte: (Cavaco, 2006)

Quanto à técnica de projeção, o retículo é projetado sobre a superfície do objeto, gerando o Retículo modelo. O retículo projetado sobre o plano de referência dá origem ao Retículo de referência e as franjas são observadas quando se sobrepõe as duas imagens.

### 3.3 Aplicações

Quando a técnica de Moiré é aplicada em objetos ou pessoas, gera uma série de padrões que podem ser utilizados para vários fins, como reconstrução 3D do objeto, medição de superfícies e análise de simetria. Para cada área de aplicação, um determinado grupo de características é formado, e devem ser reconhecidos e estudados para possibilitar a obtenção das informações desejadas.

Oster e Nishijima (Oster, et al., 1964) fizeram os primeiros estudos para estabelecer as relações entre grades superpostas e as franjas de Moiré resultantes. Cloud (Cloud, 1998) mostra que D. Tollenaar, em 1945, estudando o fenômeno de Moiré, descobriu que as franjas de Moiré comportam-se como amplificadoras de movimento, ou seja, podem dar uma alta sensibilidade a medições de movimentos. O mesmo Cloud cita vários autores que a partir daí empregaram o fenômeno para estudar deslocamento, deformação e tensão, entre eles Weller & Shepard em 1948 e Dantu em 1954. Após isto, o método foi refinado e aplicado a uma grande variedade de circunstâncias por Durelli & Parks (Parks, et al., 1970) que focaram seu trabalho no estudo das grades usadas para gerar o efeito de Moiré.

A partir destes estudos, Takasaki (Takasaki, 1970) e logo após Hertz (Hertz, et al., 2005) e Ferreira (Ferreira, 2008), utilizaram a técnica de Moiré de Sombra como alternativa de baixo custo para realização de análise postural em pessoas.

Ferreira (Ferreira, 2008) mostrou que se fossem marcados cada ponto de inflexão, representados pelos pontos pretos na Figura 7 e logo após estes fossem ligados através de uma linha, poderia então ser obtida uma representação visual de uma estrutura chamada de linha média da coluna vertebral, utilizado para diagnosticar problemas posturais como Escoliose.



Figura 7 Representação da linha média da coluna  
Fonte: Imagem modificada de (Ferreira, 2008)

Outra área de aplicação é geração de modelos digitais de objetos. Lino (Lino, 2008), por exemplo, propõe a reconstrução 3D de frutas e hortaliças a partir da topografia formada pela aplicação da técnica de Moiré (Figura 8).

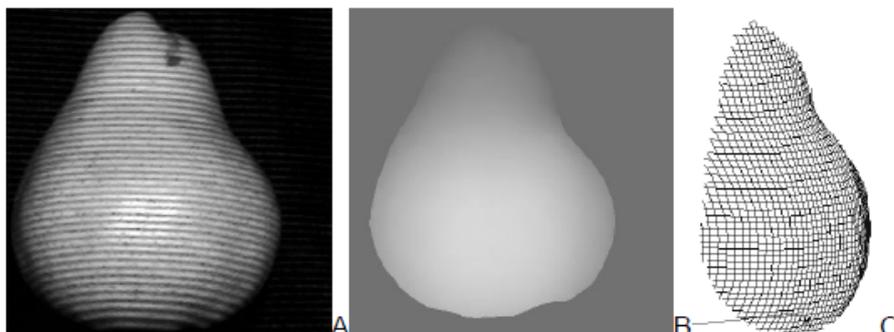


Figura 8 Moiré de Projeção aplicado sobre a fruta (A). Representação tridimensional em tons de cinza (B) e a sua representação tridimensional por malhas de contorno (C).

Fonte: (Lino, 2008)

Neste caso, as franjas de Moiré são destacadas por técnicas de segmentação, levando em conta o tom da cor, pois as franjas apresentam um tom mais escuro. Logo após, o resultante da segmentação deve ser armazenado em uma matriz de *pixels*. Nesta matriz deve-se aplicar um algoritmo de afinamento, formando franjas mais bem definidas. O ângulo, distância em *pixels* e comprimento destas franjas são usados para o cálculo das dimensões da fruta.

Em 1982, Katsuki analisou a simetria do nariz humano usando a topografia de Moiré. Com as franjas de Moiré geradas no rosto da pessoa, foram calculados os índices de simetria antes e depois de uma cirurgia facial.

No ano 2000, a National Cooperative Highway Reserch Program desenvolveu um protótipo para avaliação da superfície de pavimentos usando a técnica de Moiré de Sombra. Esse protótipo gravava as imagens de Moiré em alta resolução à medida que um veículo transita por uma estrada a uma velocidade de até 65 km/h. As imagens eram levadas a uma central onde eram analisadas para determinar o estado do pavimento e os tipos de danos que podiam ter sofrido.

A Técnica de Moiré também pode ser usada para diagnosticar lesões na sola do pé (HAMRA, et al., 1995). Na Figura 9, pode-se observar um exemplo da técnica aplicada ao pé de um indivíduo.

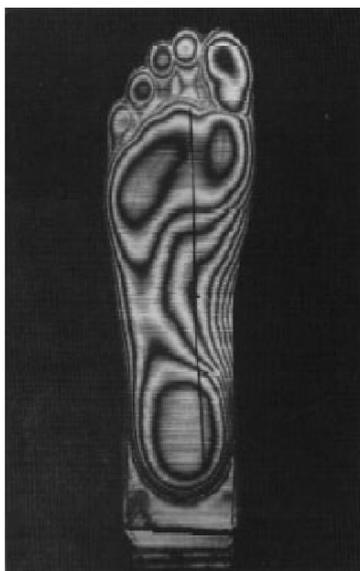


Figura 9 Técnica de Moiré aplicada na sola do pé

Fonte: (HAMRA, et al., 1995)

### 3.4 Principais Características das Sombras de Moiré

Cada uma das aplicações apresentadas na seção anterior utiliza uma ou mais características das franjas das imagens de Moiré a fim de obter o resultado desejado, dentre estas características estão a identificação da franja de Moiré, pontos de inflexão, período, inclinação de uma franja, variação angular e profundidade da franja.

Analisando as imagens do trabalho de Ferreira (Ferreira, 2008) pode-se observar algumas características que estão presentes em diversas das aplicações já citadas.

Em primeiro lugar, deve-se identificar a estrutura mais importante, a franja. É a partir desta informação que ficará possível fazer a verificação de qualquer outra característica presente na imagem. Na (Figura 10) esta destacada uma das franjas, como foi explicado no capítulo 3, as franjas são curvas de nível e auxiliam na identificação de deformidades.

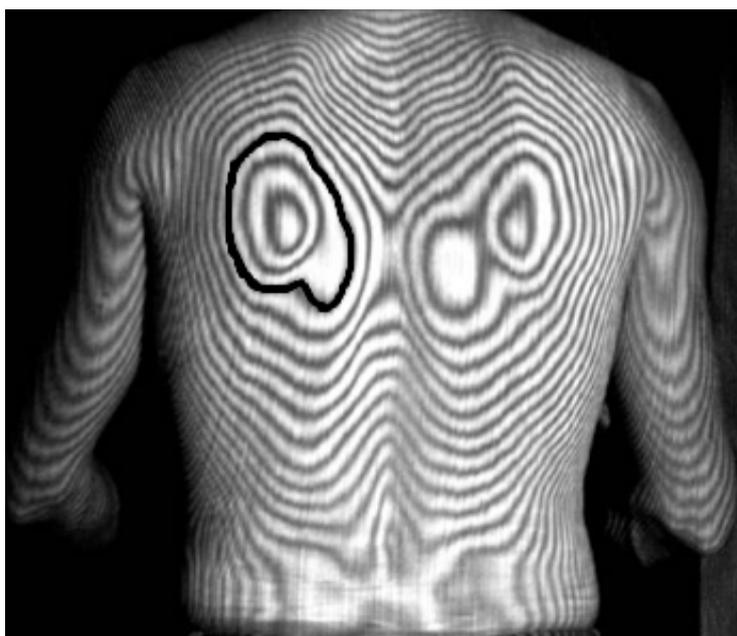


Figura 10 Franja de Moiré destacada

Fonte: Imagem modificada de (Ferreira, 2008)

Nesta imagem (Figura 11), percebe-se que na região central as franjas de Moiré possuem pontos de inflexão, que são úteis para encontrar a parte central de cada franja, e no caso da aplicação indicam a posição da coluna vertebral.



Figura 11 Pontos de inflexão formados pelo efeito de Moiré

Fonte: Imagem modificada de (Ferreira, 2008)

Com esse resultado é possível calcular o desvio angular entre dois pontos quaisquer. Para este cálculo é traçada uma linha horizontal na altura do primeiro ponto (Figura 12) da linha média da coluna e a partir dele obtêm-se os desvios (Ferreira, 2008).

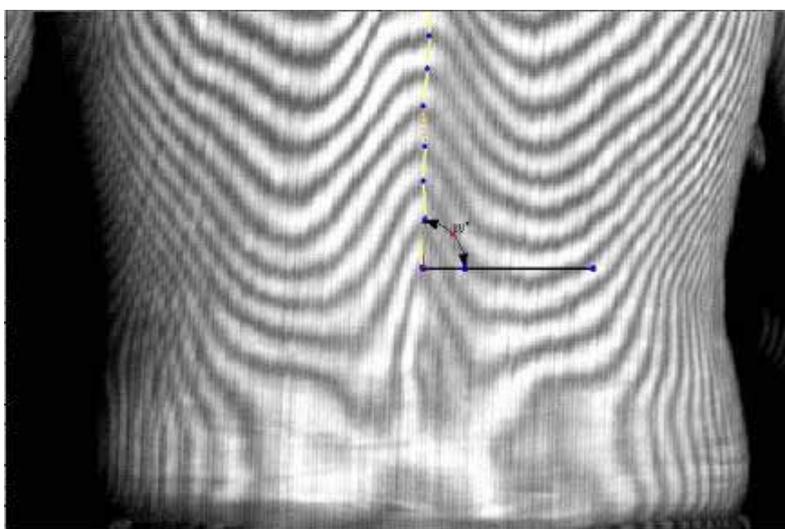


Figura 12 **Varição angular** entre os pontos destacados a partir dos vértices formados pelo efeito de Moiré.

Fonte: (Ferreira, 2008)

Outro ponto que deve ser levado em consideração é a sensibilidade das Técnicas de Moiré quando utilizada em uma aplicação. Esta sensibilidade depende principalmente do período do retículo (Figura 13), caracterizado pela distância entre

duas faixas claras ou escuras contíguas. Quanto menor o período maior a sensibilidade.

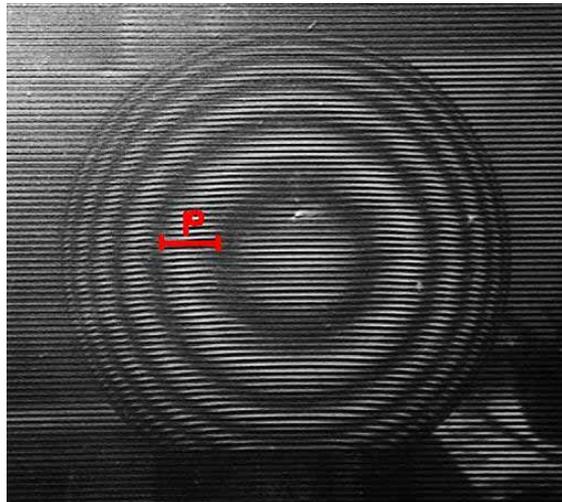


Figura 13 Período P entre duas franjas

Fonte: Imagem modificada de (Oliveira, 2006)

A sensibilidade da Técnica de Moiré é calculada usando a Equação 1, na qual  $P_m$  é o período da franja de Moiré,  $P_1$  é o período do retículo (grade) 1,  $P_2$  é o período do retículo (grade) 2.

$$P_m = \frac{P_1 \times P_2}{\sqrt{P_1^2 + P_2^2 - 2 \times P_1 \times P_2 \times \cos \theta}}$$

Equação 1 Cálculo do período

Outra característica importante das curvas de Moiré é o centróide, ou franja, mais interno de dentro para fora. Na aplicação de Ferreira (Ferreira, 2008), após detectar as centróides dos lados esquerdo e direito, pode-se obter pontos que permitem calcular o alinhamento entre a região das escápulas do indivíduo, como mostra a Figura 14. Com estes dois novos pontos é possível fazer o cálculo de desvio angular da região das escápulas.

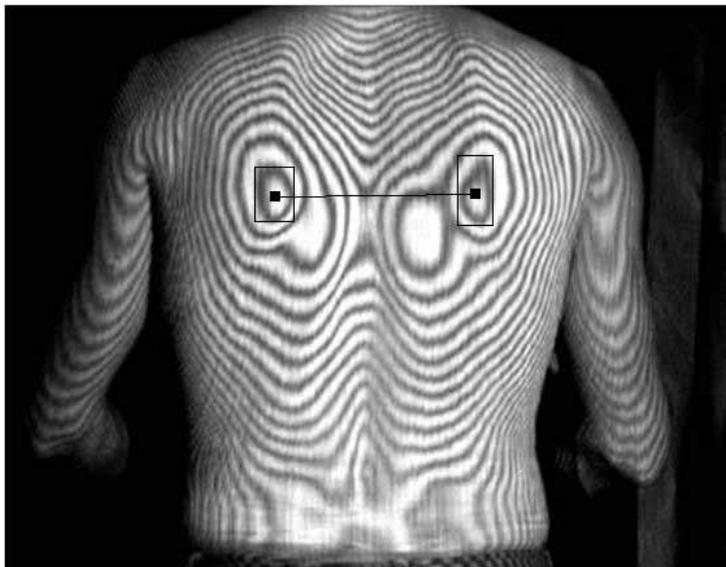


Figura 14 Alinhamento entre as escápulas do indivíduo  
 Fonte: Imagem de (Ferreira, 2008) modificada

Ainda com a Técnica de Moiré aplicada nas costas de uma pessoa é útil identificar a inclinação de uma franja. Para medir esta inclinação, seleciona-se o ponto central máximo e mínimo de uma franja e traça-se uma linha entre estes dois pontos (Figura 15).

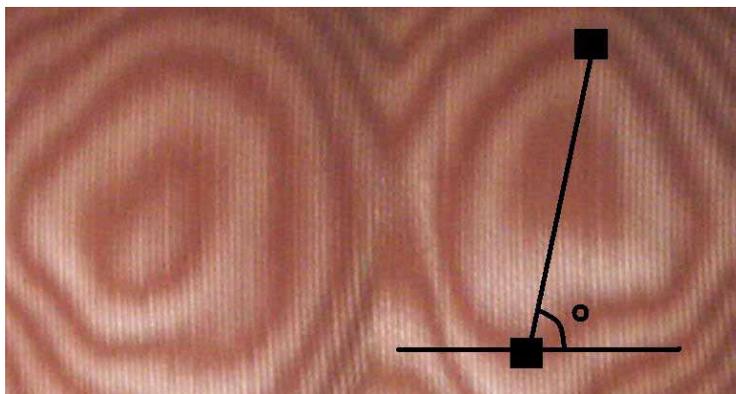


Figura 15 Representação da inclinação de uma franja.  
 Fonte: O Autor (2009)

Há também a possibilidade de calcular a profundidade de uma franja, este pode ser feito pela Equação 2 proposta por Takasaki (Takasaki, 1970), onde  $h$  é a profundidade (em relação as demais franjas, usada para fazer comparações e ver o ponto mais saliente no objeto em que a sombra é projetada) da franja em relação a grade,  $n$  é a distância da câmera até a grade,  $d$  é a distância da fonte de luz até a câmera,  $s$  é a distância entre os fios da tela. O cálculo de profundidade foi usado por

HAMRA (HAMRA, et al., 1995), para comparar o método manual com a Técnica de Moiré para ser usado como alternativa nos diagnósticos de lesões na sola do pé.

$$h = \frac{n \times l}{(d / s) - n}$$

Equação 2 Fórmula para calcular a profundidade de uma franja

Outro ponto que se deve levar em consideração, é que segundo estudos realizados por Batouche (Batouche, et al., 1996), pontos pertencentes a uma mesma franja, quando estas não apresentarem deformidades como pontos de inflexão (Figura 16), possuem mesma altitude, e a diferença de altitude entre franjas vizinhas é relativamente constante.

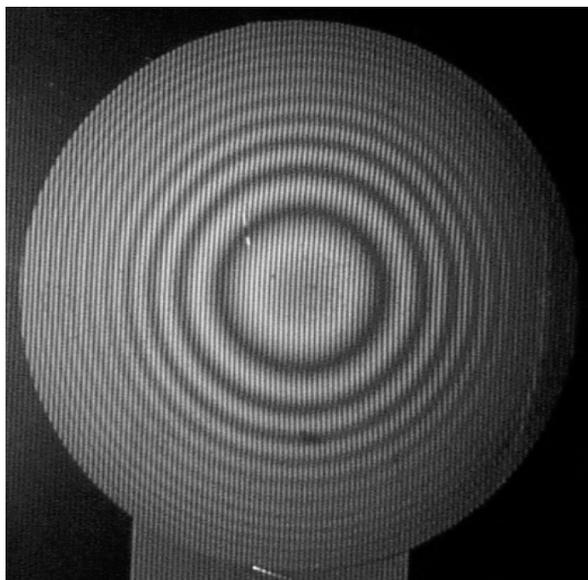


Figura 16 Aplicação da técnica de Moiré de sombra em uma calota

Fonte: (Lino, 2002)

## 4 ALGORITMOS DE PROCESSAMENTO DE IMAGEM E VISÃO COMPUTACIONAL

Imagem é uma forma de representar cenas através de um conjunto de dados discretos e finitos, chamados de pixels, colocados em um arranjo bidimensional. Essas imagens podem ser fotos e desenhos feitos por humanos ou até mesmo por um computador.

O Processamento de Imagens é uma área que busca fornecer ferramentas que permitem a transformação de qualquer tipo de imagem (Gonzales, et al., 2000). A idéia básica desta transformação é permitir a geração de produtos que realcem as informações contidas nas imagens para facilitar a interpretação e a análise humana. A área de processamento digital de imagens tem atraído grande interesse nos últimos anos. A evolução da computação digital, bem como o desenvolvimento de novos algoritmos e equipamentos de *hardware* que permitem o uso de *softwares* mais complexos, está possibilitando uma gama de aplicações cada vez maior.

Visão Computacional é o conjunto de métodos e técnicas que permitem que sistemas computacionais sejam capazes de realizar a interpretação de imagens (Wangenheim, 2008). Esta interpretação pode ser definida em termos computacionais como a transformação de um conjunto de dados digitais representando uma imagem, em uma estrutura de dados descrevendo a semântica deste conjunto de dados em um contexto qualquer.

Para alcançar o objetivo deste trabalho, que é destacar as principais estruturas de uma imagem gerada pelo efeito de Moiré é necessário aplicar uma seqüência de algoritmos que transformarão a imagem original em uma estrutura de dados que contenha as informações desejadas. O primeiro passo será segmentar a imagem, separando a área de interesse, no caso as franjas de Moiré do resto da imagem. Após este passo, sobre as regiões obtidas pela segmentação, deve-se aplicar um algoritmo de afinamento para poder simplificar a imagem e poder então aplicar a vetorização para obter os pontos importantes para a aplicação. Esta seqüência pode ser representada com a Figura 17.

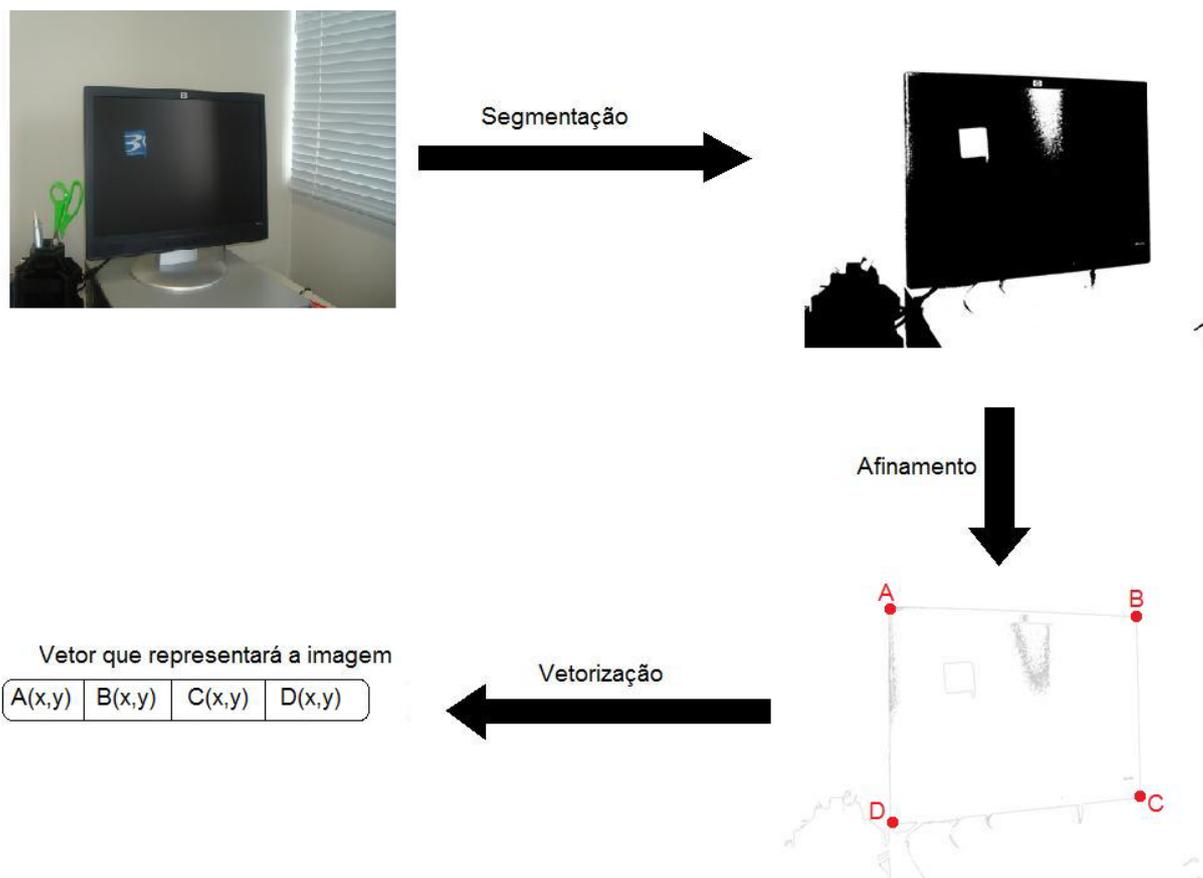


Figura 17 Seqüência de algoritmos aplicados sobre a imagem original para chegar ao resultado desejado armazenado na imagem resultante.

Fonte: O Autor (2009)

A seguir serão detalhas as técnicas para realizar estas transformações.

#### 4.1 Segmentação

O algoritmo de segmentação serve para subdividir uma imagem em suas partes ou objetos constituintes, para poder selecionar somente o que se deseja estudar, retirando tudo o que for desnecessário e evidenciando o conteúdo alvo. Uma prática geral dos diversos métodos de segmentação é agrupar, de alguma maneira, os pixels de mesma propriedade, como textura, cor ou intensidade, como mostram os itens (a) e (b) da Figura 18.

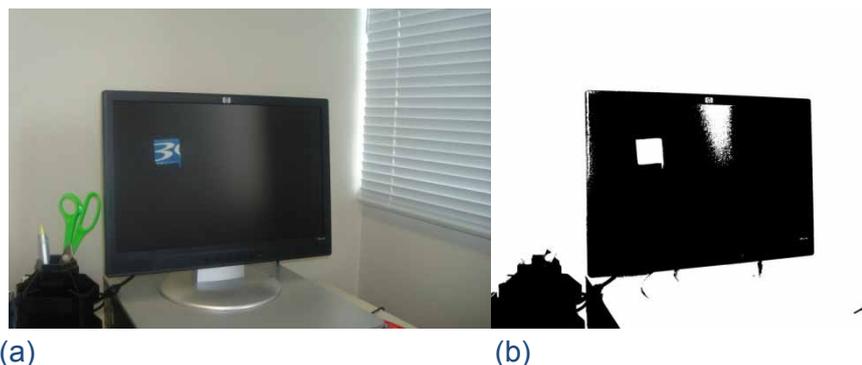


Figura 18 (a) Imagem Original, (b) Imagem Segmentada

Fonte: O Autor (2009)

Para imagens em tons de cinza, os algoritmos de segmentação baseiam-se geralmente nas técnicas de detecção de descontinuidade e similaridade que são aplicadas sobre os valores de níveis de cinza presentes na imagem (Gonzales, et al., 2000).

No caso da detecção de descontinuidade, a abordagem central é particionar uma imagem baseado em mudanças bruscas nos tons de cinza, buscando por pontos isolados e linhas e bordas de uma imagem. O outro caso, a similaridade, baseia-se em técnicas como limiarização, crescimento de regiões e divisão e fusão de regiões.

#### 4.1.1 Detecção de descontinuidades

Segundo Gonzales (Gonzales, et al., 2000), a maneira mais simples de encontrar descontinuidades é através de varredura da imagem por uma máscara (Figura 19), neste caso uma máscara 3x3, este processo que recebe o nome de **convolução** e envolve o cálculo, para cada pixel da imagem, da soma dos produtos dos coeficientes pelos níveis de cinza contidos na região englobados pela máscara, como mostra a Equação 3, na qual  $z_i$  é o nível de cinza do pixel associado ao coeficiente da máscara, no caso da Figura 19, está representado por  $w_5$ .

$$\sum_{i=1}^9 w_i z_i$$

Equação 3 Somatório dos produtos

$W1$	$W2$	$W3$
$W4$	$W5$	$W6$
$W7$	$W8$	$W9$

Figura 19 Máscara 3x3 genérica

Fonte: (Gonzales, et al., 2000)

A forma de aplicação da matriz de convolução sobre os pixels de uma imagem pode ser representada pela Figura 20, na qual o ponto central da máscara de convolução, neste caso o  $w_5$  é colocado sobre um ponto da imagem a ser transformada.

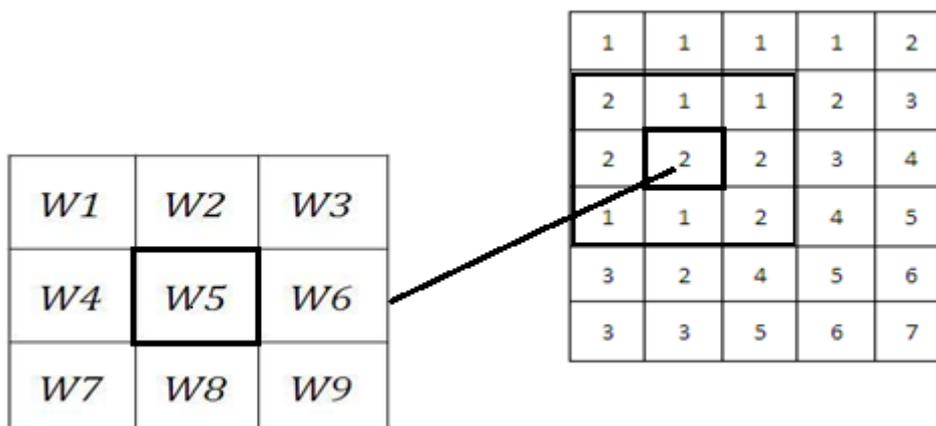


Figura 20 Demonstração do uso de uma máscara de convolução sobre um ponto da imagem

Fonte: O Autor (2009)

Deve-se aplicar esta máscara em todos os *pixels* da imagem, com exceção dos *pixels* das extremidades da imagem, pois não seria possível aplicar a máscara sobre eles. O algoritmo segue os seguintes passos:

1. O *pixel* analisado ficará posicionado junto ao pixel central da máscara.
2. Todos os valores da máscara serão multiplicados pelo seu *pixel* correspondente, acumulando um total de nove multiplicações a cada vez que a máscara é aplicada
3. Os produtos destas multiplicações devem ser somados
4. A soma representa o valor do operador aplicado ao *pixel* central

5. Repete-se os quatro passos anteriores para todos os *pixels* da imagem, com exceção dos *pixels* das extremidades.
6. Com este conjunto de informações cria-se um histograma com limites mais bem definidos, e então obter o limiar, que permite decidir qual cor (preto ou branco) cada pixel deve receber.

A seguir será desenvolvido um pseudocódigo que representa a execução dos passos descritos acima.

#### Algoritmo 1 - Detecção de descontinuidades

```

Para cada pixel p da Imagem I{
  variável somatório = 0;
  se (p[x,y] == extremidade){
    não faz nada
  }
  senão{
    somatório += p[x,y].valor * -4;
    somatório += p[x-1, y+1].valor * 0;
    somatório += p[x, y+1].valor * 1;
    somatório += p[x+1, y+1].valor * 0;
    somatório += p[x-1, y].valor * 1;
    somatório += p[x+1, y].valor * 1;
    somatório += p[x-1, y-1].valor * 0;
    somatório += p[x, y-1].valor * 1;
    somatório += p[x+1, y-1].valor * 0;
  }
  armazena em uma matriz[x,y] o valor do somatório
}

```

Para a detecção de pontos isolados, busca-se a condição em que  $|R| > T$  no qual  $T$  é um valor de tom de cinza, e  $|R|$  é o valor do somatório da Equação 3, considerando a matriz de convolução da Figura 21. Basicamente, tudo que essa formulação faz é medir as diferenças de pesos entre o ponto central e seus vizinhos. A idéia é que o nível de cinza de um ponto será totalmente diferente dos níveis de cinza de seus vizinhos.

-1	-1	-1
-1	8	-1
-1	-1	-1

Figura 21 Matriz de convolução para detecção de pontos isolados

Fonte: O Autor (2009)

Para a segmentação de linhas podem ser usadas matrizes de convolução como as representadas pela Figura 22. Se o valor resultante da aplicação de uma dessas máscaras for maior do que o valor que resulta da aplicação de qualquer outra delas, diz-se que foi detectada uma linha no sentido da respectiva máscara.

-1	-1	-1	-1	-1	2	-1	2	-1	2	-1	-1
2	2	2	-1	2	-1	-1	2	-1	-1	2	-1
-1	-1	-1	2	-1	-1	-1	2	-1	-1	-1	2
Horizontal	45° Positivos	Vertical	45° Negativos								

Figura 22 Máscaras para linhas

Fonte: (Gonzales, et al., 2000)

Para a detecção de bordas é necessário encontrar regiões onde ocorram variações abruptas nos valores dos tons das cores, uma vez que a borda caracteriza-se pelo limite entre duas regiões com propriedades distintas.

Existem diversos algoritmos para detecção de bordas, o mais simples é o operador de Roberts (Seara, 1998). Neste utiliza-se uma máscara 2x2 (Figura 23) para encontrar mudanças de direção em x e y.

0	1	1	0
-1	0	0	-1
Gx		Gy	

Figura 23 Máscara de Roberts

Fonte: (Seara, 1998)

A partir destas máscaras é feito o cálculo para determinar onde o pixel avaliado é ou não um pixel de borda. Para tanto calcula-se uma métrica chamada gradiente, obtido a partir da Equação 4.

$$|G| = \sqrt{G_x^2 + G_y^2}$$

Equação 4: Fórmula para cálculo do gradiente

Se o valor resultante for maior do que o menor valor de entrada da equação ( $G_x$  ou  $G_y$ ), o pixel é considerado ser parte de uma borda. Se for necessário determinar a direção da borda, pode-se utilizar a  $\alpha = \left(\frac{G_y}{G_x}\right)$

Equação 5.

$$\alpha = \text{atan}\left(\frac{G_y}{G_x}\right)$$

Equação 5: Fórmula para cálculo de direção da borda

Outro algoritmo para bordas é o Laplaciano, este é um operador que amplifica as transições das regiões em bordas dos objetos de interesse e de seus fundos respectivos. Assim, o histograma terá dois picos melhor definidos, e um limiar global poderá ser encontrado com mais precisão para posteriormente ser aplicado sobre a imagem. O operador Laplaciano usa uma máscara 3 x 3, representada na Figura 24.

0	1	0
1	-4	1
0	1	0

Figura 24 Máscara de Laplaciano

Fonte: O Autor (2009)

#### 4.1.2 Detecção de similaridades

A detecção de similaridades tem como objetivo a observação do interior dos objetos e não as fronteiras (bordas) que os delimitam. Logo, parte-se do princípio que os pixels que compõe um objeto têm propriedades similares, enquanto pixels de objetos distintos têm propriedades distintas. A seguir são descritas as principais estratégias para a detecção de similaridades.

A limiarização (ou binarização) é um processo que visa, a partir de uma imagem em tons de cinza, obter uma imagem binária. Os pixels com tons mais próximos a preto e os que são totalmente pretos na imagem original vão gerar pixels pretos na imagem binária. Por outro lado, os pixels cuja intensidade for branco ou próximo a ele, irão originar pixels brancos na imagem transformada.

Nesta técnica, a idéia central é a de que um objeto pode ser entendido como uma região formada por pixels contíguos que tenham em comum uma faixa de intensidades. Dessa forma, a limiarização usa a intensidade dos pixels para distinguir as regiões.

O processo se baseia na análise do histograma de freqüências da imagem (Figura 25), onde as faixas de intensidade correspondem às bandas do histograma. Um caso simples de ser entendido é o de um histograma bimodal. Nessa situação o histograma apresenta dois picos separados por um vale entre eles. Esses picos podem representar duas regiões distintas, como por exemplo, um fundo e um objeto.

Neste método é necessário determinar um limiar. O limiar consiste em um valor entre “zero” e o nível máximo da escala de cinza, para o qual *pixels* com intensidade acima ou igual a esse valor tornam-se brancos na imagem final, e *pixels* com intensidade menor, tornam-se pretos.

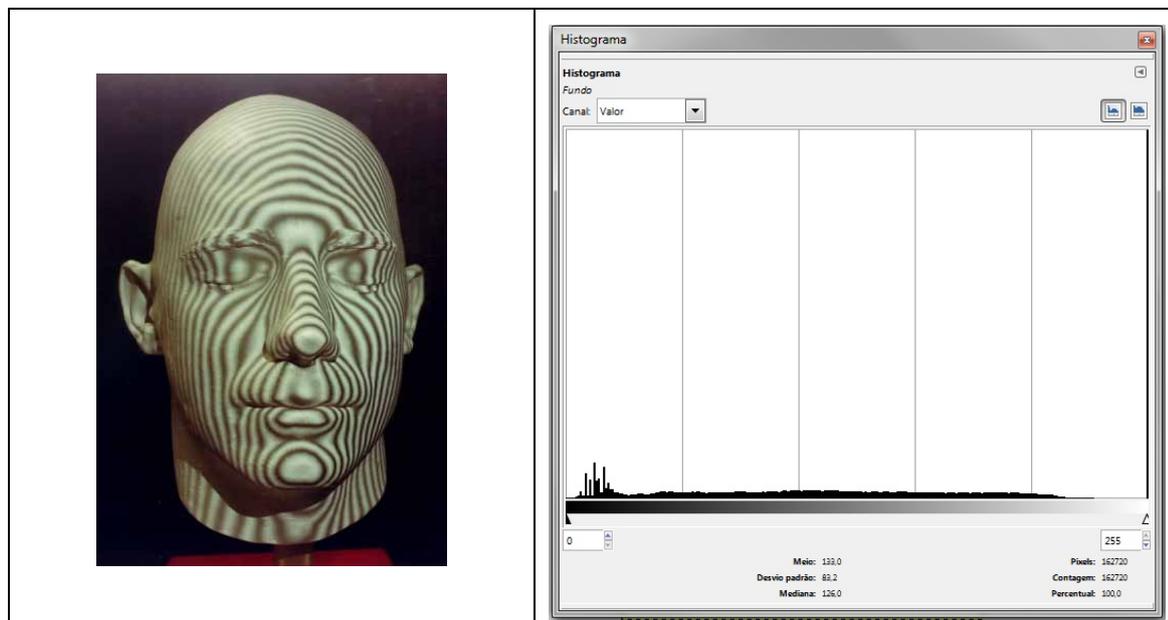


Figura 25 Exemplo de Histograma

Fonte: O Autor (2009)

A parte crucial para utilizar esta técnica com sucesso, é determinar um limiar que possa separar os *pixels* que fazem parte do fundo da imagem daqueles que definem o objeto de interesse.

Há diversas maneiras de resolver esta questão, dentre elas pode-se citar um dos mais utilizados por autores como Gonzales (Gonzales, et al., 2000), e Bastos (Bastos, 2008), chamado “método do vale” no qual se analisa um histograma gerado a partir da imagem que se deseja modificar, procurando por uma ou mais regiões semelhantes a um “vale”, representado na Figura 26 pela letra “T”.

Outro algoritmo para a escolha do limiar e também o mais simples, é dividir a escala de cinza exatamente ao meio. Por exemplo, para uma escala de cinza que vai de 0 a 255, o limiar utilizado é de 128. Porém, nem sempre o uso desta técnica permite chegar a resultados precisos.

Há também a escolha do limiar por iteração seletiva, no qual é calculada a estimativa inicial do valor do limiar. Este cálculo pode ser feito por qualquer método, incluindo a média de intensidade dos *pixels* da imagem. A partir deste valor calcula-se o valor médio das intensidades dos pontos com tonalidades maiores ou iguais ao limiar atual ( $T_1$ ) e o valor médio das intensidades dos pontos com tonalidades menores que o limiar atual ( $T_2$ ). Com estes valores, calcula-se o novo limiar ( $T = \frac{T_1 + T_2}{2}$

Equação 6) através da média entre os dois valores recém calculados Equação 6.

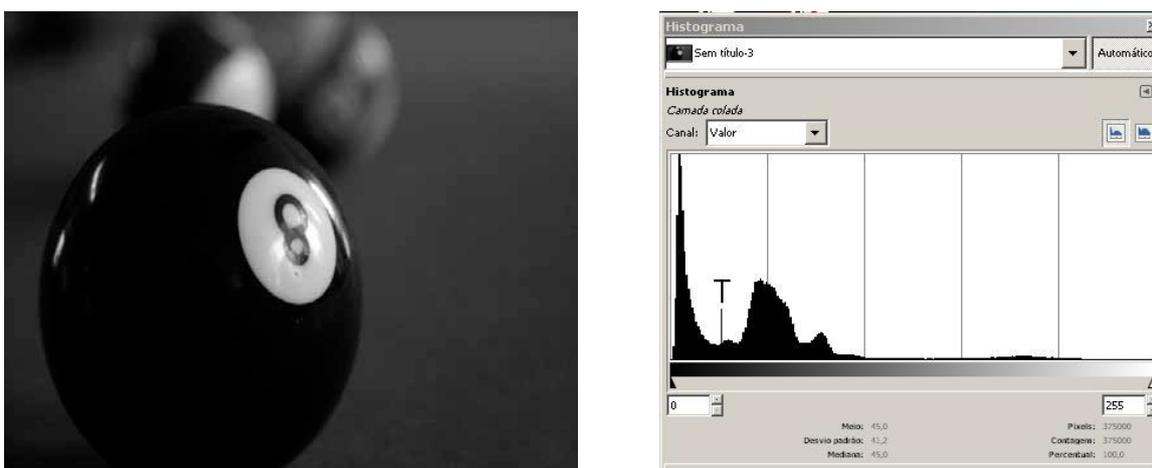


Figura 26 Imagem e seu Histograma gerado pelo software Gimp 2.6

Fonte: O Autor (2009)

$$T = \frac{T_1 + T_2}{2}$$

Equação 6: Cálculo do limiar por iteração seletiva

Em uma abordagem diferente de segmentação, a técnica de crescimento de regiões agrupa pixels ou sub-regiões em regiões maiores. A mais simples dessas abordagens é a agregação de pixels, que começa com um conjunto de pontos “semente” e, a partir deles, faz as regiões crescerem anexando a cada ponto semente aqueles pixels que possuam propriedades similares como, por exemplo, nível de cinza, cor ou textura. Na Figura 27, por exemplo, na figura (a) o *pixel* destacado indica a semente, e o parâmetro usado neste exemplo para agregar os *pixels* será o valor 3. A divisão deverá conter uma região com valores menor ou igual ao valor parametrizado e outra com valores maiores.

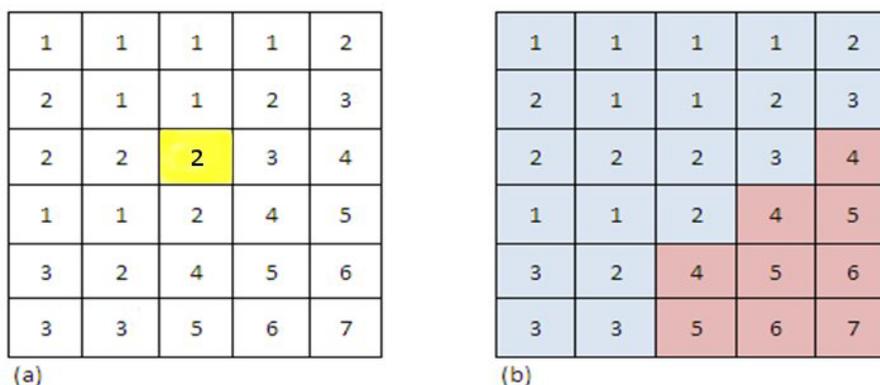


Figura 27 (a) matriz com os valores de nível de cinza de cada *pixel*; (b) formação de duas regiões distintas de acordo com os valores dos *pixels*, azul (1, 2 e 3) e vermelho (4, 5 e 6).

Fonte: O Autor (2009)

Uma imagem pode conter vários objetos, e cada objeto pode conter várias regiões correspondendo a diferentes partes do objeto, para segmentar uma imagem com esta característica pode-se usar a técnica de Divisão e Fusão. Assim como a de crescimento de regiões, possibilita agrupar *pixels* para formar regiões sobre as quais um padrão de homogeneidade deve ser verificado. Esse método, porém, não necessita o uso de sementes para começar o processo de segmentação. e pode ser associado a uma estrutura *Quadtree* (Figura 28) que permite, decompor e agrupar partes de uma imagem.

Para selecionar determinada área ou objeto da figura pode-se dividir esta imagem recursivamente, no entanto, o problema está em decidir qual propriedade

utilizar na subdivisão. Em algumas aplicações, a variação dos valores de intensidade de tons de cinza é usado como critério de parada da recursão que subdivide a imagem.

Ao invés de dividir a imagem de forma regular, pode-se optar pelo uso de uma *Quadtree* (Figura 28) que divide a imagem de forma adaptativa, com base em seu conteúdo. Neste caso, o critério para dividir ou não um área deve ser o mesmo que irá fundir as áreas no passo seguinte do algoritmo.

Para verificar as regiões formadas pela divisão e fundi-las com outras regiões que façam parte do mesmo objeto, faz-se a verificação do padrão de homogeneidade e então se decide se estas regiões devem ou não ser fundidas.

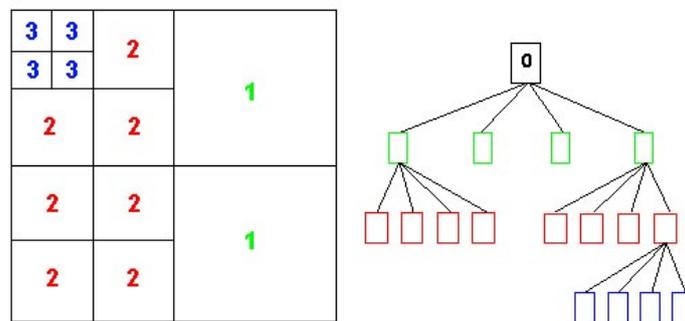


Figura 28 Exemplo da estrutura de um Quadtree

Fonte: (Morton, 2009)

Após o estudo das técnicas de segmentação, escolheu-se a técnica de Limiarização para trabalhar sobre a imagem com o efeito de Moiré, pois esta apresenta as características necessárias para separar as franjas geradas do resto da imagem. Isso se dá pelo fato desta técnica possibilitar a geração de uma imagem binária, separando as franjas (região de interesse) do resto da imagem, o que facilitará o processo de encontrar as estruturas formadas pelo efeito de Moiré.

## 4.2 Afinamento

Após obter-se uma imagem binária deve-se afiná-la permitindo chegar a uma imagem mais simplificada com largura de um *pixel*. Para isso usam-se algoritmos de afinamento.

O afinamento é caracterizado por sucessivas remoções de *pixels* redundantes (desnecessários) e é um método bastante utilizado na prática por proporcionar resultados bem satisfatórios. Para verificar se o algoritmo foi aplicado com sucesso algumas características devem ser observadas, são elas:

- os *pixels* do esqueleto devem formar o mesmo número de regiões que a imagem original possuía;
- as regiões precisam ter um *pixel* de largura;
- os *pixels* do esqueleto precisam ficar próximos de onde se encontrava o centro das regiões da imagem original.

Na (Figura 29) pode-se observar um exemplo da aplicação do algoritmo de afinamento, na qual a imagem inicial (esquerda) é afinada e gera um esqueleto, representado pela linha mais escura da segunda imagem (direita).



Figura 29 Imagem original (esquerda), e seu esqueleto (direita)

Fonte: (Schouten, 2004)

O afinamento pode ser obtido a partir de alguns algoritmos, dentre eles estão o algoritmo de Zhang-Suen (Suen, 1984) e de Holt (Holt, et al., 1987).

No algoritmo de Zhang-Suen toma-se por base sempre a comparação do *pixel* que se está tentando eliminar sobre seus oito vizinhos. Neste método, quatro regras deverão ser aplicadas e uma vez que as quatro sejam satisfeitas, o *pixel* poderá ser eliminado. Estas regras asseguram que se o pixel em questão for eliminado não fará com que diferentes regiões ligadas por ele passem a ficar separadas. Também assegura que a eliminação de pixels sempre ocorrerá nas bordas do objeto.

O algoritmo é dividido em duas iterações e levará em conta as seguintes convenções:

- a cor de um *pixel*  $(x, y)$  de uma imagem é representado pela notação  $\text{Imagem}(i, j)$ ;

- o fundo de uma imagem qualquer é preenchido pela cor branca.
- a conectividade de um ponto  $(x, y)$  é uma propriedade de ligação entre dois pixels que faz com que sejam considerados como fazendo parte de uma mesma região em uma imagem.

Na primeira interação, o pixel Imagem  $(i, j)$  é eliminado dependendo do valor verdade (se satisfaz a condição) das seguintes condições:

1. O número de conectividade é 1. Este número corresponde ao número de transições de branco para preto nos pixels que circundam o pixel central, representado pelo  $p1$  na Figura 30.

p9	p2	p3
p8	p1	p4
p7	p6	p5

Figura 30 Máscara para análise do *pixel P1*

Fonte: O Autor (2009)

2. Utilizando a mesma máscara da condição anterior, deve-se constatar que o número de *pixels* vizinhos pretos está entre dois e seis.
3. Ao menos um dos *pixels* Imagem  $(i, j+1)$ , Imagem  $(i-1, j)$  e Imagem  $(i, j-1)$  são fundo da imagem (branco). No caso da máscara utilizada, estes se referem aos *pixels*  $p2$ ,  $p4$  e  $p8$ ;
4. Ao menos um dos *pixels* Imagem  $(i-1, j)$ , Imagem  $(i+1, j)$  e Imagem  $(i, j-1)$  são fundo da imagem (branco). No caso da máscara utilizada, estes se referem aos *pixels*  $p2$ ,  $p6$  e  $p8$ .
3. Ao final desta interação os pixels marcados são eliminados. A próxima interação é a mesma exceto para os passos 3 e 4, então: Ao menos um dos *pixels* Imagem  $(i-1, j)$ , Imagem  $(i, j+1)$  e Imagem  $(i+1, j)$  são fundo da imagem (branco). No caso da máscara utilizada, estes se referem aos *pixels*  $p2$ ,  $p4$  e  $p6$ ;

4. Ao menos um dos *pixels* Imagem (i, j+1), Imagem (i+1, j) e Imagem (i, j-1) são fundo da imagem (branco). No caso da máscara utilizada, estes se referem aos *pixels* p4, p6 e p8.

Mais uma vez, os *pixels* marcados são eliminados, resultando em um esqueleto.

Na Figura 31, é apresentada uma imagem resultante da aplicação do algoritmo de Zhang-Suen na qual o objeto original aparece em cinza e em sua região central, em branco, está representado seu esqueleto.

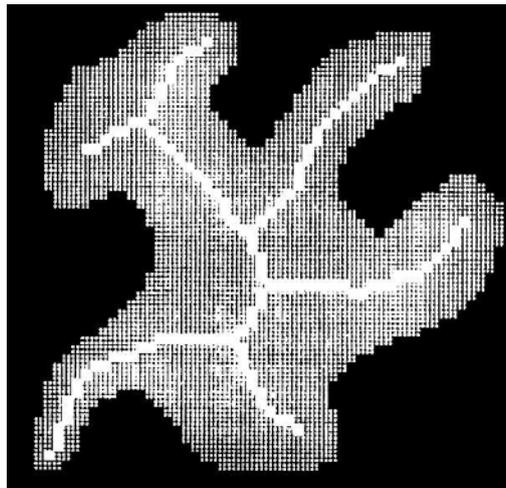


Figura 31 Imagem e seu esqueleto gerado pelo algoritmo de Zhang-Suen

Fonte: (Wangenheim, 2008)

Holt, por sua vez, sugere um algoritmo mais rápido. Primeiramente, as duas iterações são escritas como expressões lógicas usando a vizinhança 3 x 3 sobre um ponto corrente. Se o resultado das expressões lógicas for falso, o ponto é apagado, caso contrário ele permanece.

A primeira iteração pode ser escrita como:

$$V(C) \& \left( \sim edge(C) \mid \left( v(L) \& v(S) \& (v(N) \mid v(O)) \right) \right)$$

E a segunda como:

$$V(C) \& \left( \sim edge(C) \mid \left( v(O) \& v(N) \& (v(S) \mid v(L)) \right) \right)$$

Em que:

Levando em consideração a máscara da Figura 32, sabe-se que:

- N, S, L e O: representam os vizinhos do *pixel* analisado “C”, presentes em Norte, Sul, Leste e Oeste.
- $v()$ : representa o valor do ponto. O resultado será verdadeiro se o ponto for um ponto do objeto (preto), e falso se o ponto for de fundo (branco).
- $edge()$ : O resultado será verdadeiro se o ponto estiver na borda do objeto, e falso, caso contrário.

NO	N	NE
O	C	L
SO	S	SE

Figura 32 Máscara para visualização do método de afinamento de Holt

Fonte: O Autor (2009)

Um ponto de borda (função  $edge()$ ) é um ponto que atende as seguintes condições:

- Conectividade igual a 1, tal que a conectividade de um ponto representa o número de transições de ponto branco para ponto preto existentes no percurso efetuado no sentido horário a partir do ponto superior ao ponto corrente (Figura 33).

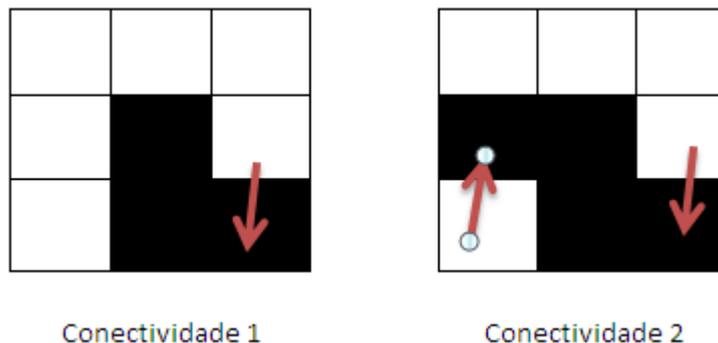


Figura 33 representação de conectividade

Fonte: O Autor (2009)

- Possui de dois a seis vizinhos pretos, onde vizinhos pretos de um ponto são todos os pontos pretos presentes nas 8 direções possíveis a partir do ponto corrente.

Com a combinação das duas iterações, Holt chegou a seguinte expressão lógica:

$$v(C) \& \left( \begin{array}{c} \sim edge(C) \mid (edge(L) \& v(N) \& v(S)) \mid (edge(S) \& v(O) \& v(L)) \mid \\ (edge(L) \& edge(SE) \& edge(S)) \end{array} \right)$$

### 4.3 Vetorização

O processo de vetorização é bastante usado na computação gráfica em aplicações que vão desde a análise de impressões até a análise de imagens de neurônios, porém o mais comum é para reconhecimento de caracteres. Neste trabalho, a necessidade de representar *rasters* em vetores é devido à maior facilidade de manuseio para o reconhecimento das características de Moiré. Neste capítulo será mostrado como será feito.

Atualmente, imagens digitais são armazenadas em *arrays* bi-dimensionais, conhecidos como *rasters*, porém nestes tipos de imagens são difíceis de aplicar rotações e escalas, além de ocuparem um grande espaço de armazenamento (Parker, 1988).

Para que a vetorização seja aplicada é necessário identificar e destacar os pontos de interesse, que neste caso será feito através dos algoritmos de segmentação e afinamento. Após estes processos será aplicado o algoritmo para vetorizar a imagem.

Para representar uma imagem em vetores, será utilizada uma codificação de números inteiros no intervalo de 0 a 7, chamada de código de encadeamento, o qual indicará a direção de um *pixel* como mostra na Figura 34.

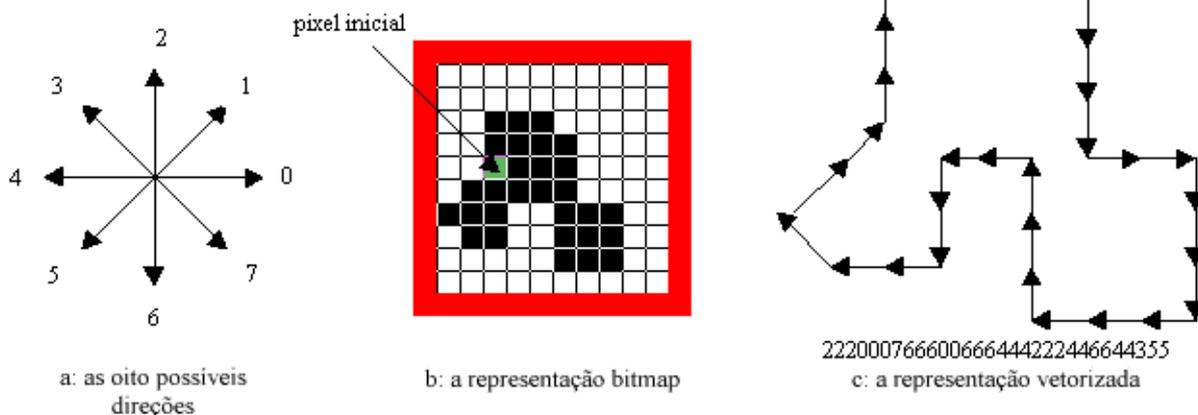


Figura 34 (a) código de encadeamento (b) imagem (c) vetorização

Fonte: (Pinho, 1999)

Neste trabalho será utilizado o algoritmo de Doué e Rubio (Doué, et al., 1995) percorrendo a imagem de baixo para cima e depois da direita para a esquerda adicionando uma camada de pixels à figura, cada vez que passa sobre um limite como mostram na Figura 35, os pontos marcados em verde.

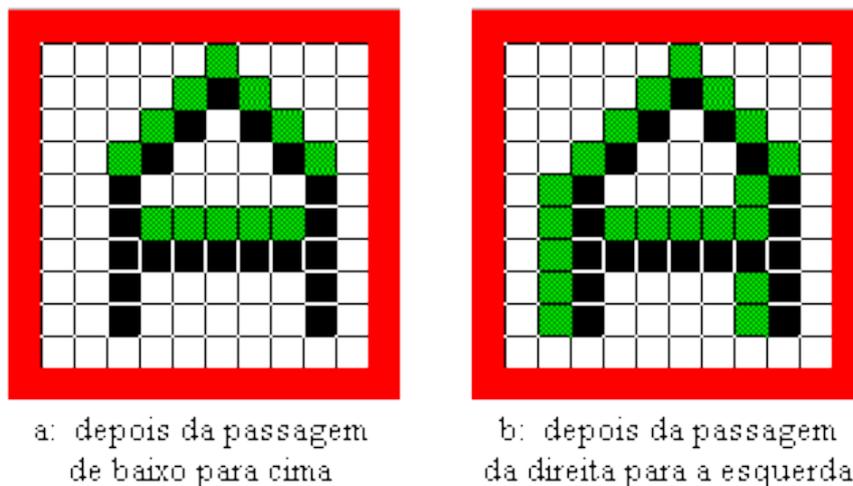


Figura 35 (a) primeira passagem (b) segunda passagem

Fonte: (Pinho, 1999)

Uma maneira de implementar a vetorização é mostrado no algoritmo a seguir.

### Algoritmo 2 - Vetorização

Fazer

```

marcar(p) //marcar a posição atual
for idelta = 0 to 7

```

```

delta = vetor[idelta]//obtem vetor de deslocamento
se borda(p+delta)//achou limite
    encadeia(delta) //adiciona à cadeia
p = p + delta //atualiza posição
enquanto HouverMovimento()

```

Porém, este algoritmo tem alguns problemas quando a imagem tem detalhes muito estreitos ou quando há linhas finas com ângulo de 45 graus.

No primeiro caso, o erro acontece quando o algoritmo que percorre os pixels na figura se aproxima de um buraco com 1 pixel de largura, como mostra a Figura 36. A falha ocorre visto que começando em “A” a verificação das direções é feita em sentido anti-horário e ele acaba entrando no buraco (“C”).

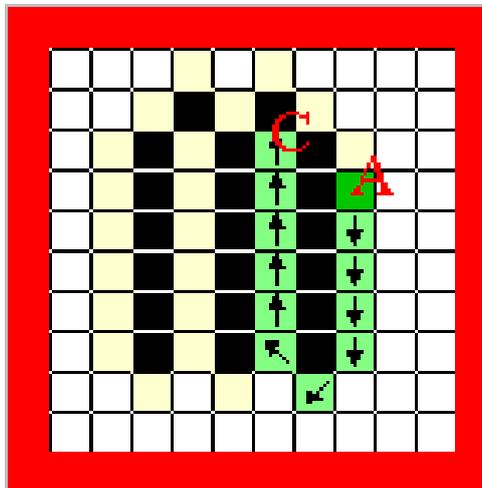


Figura 36 Primeiro caso em que há falha. A imagem é percorrida de “A” até “C”

Fonte: (Pinho, 1999)

O segundo ocorre também quando o algoritmo está percorrendo a imagem. Ele deveria fazer a “escada”, porém acaba tomando um “atalho” como mostra a Figura 37.

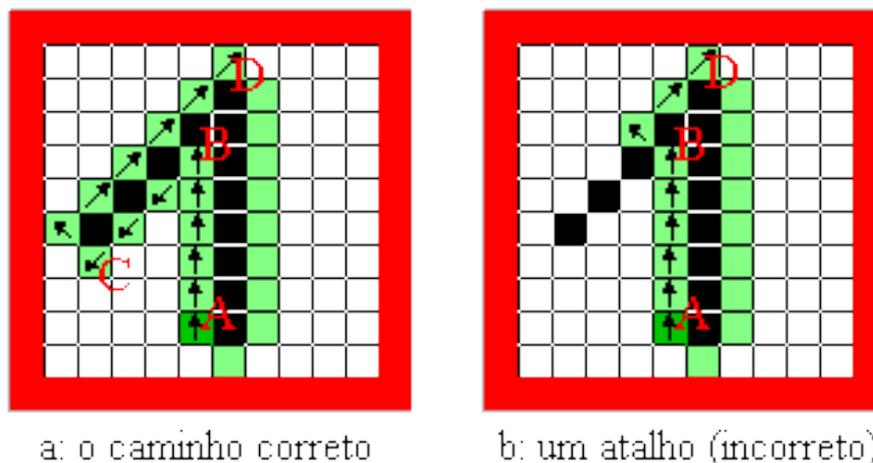


Figura 37 (a) caminho (“A B C D”) que deveria ser feito (b) caminhamento quando há erro  
 Fonte: (Pinho, 1999)

A solução proposta para este problema é simples. Neste novo algoritmo cada *pixel* é dividido em uma 4x4 de pedaços menores. Fazendo isso, o primeiro problema não ocorre mais, já que não haverá mais buracos estreitos. Para o segundo, basta que linhas retas tenham prioridades sobre linhas diagonais.

O problema desse novo algoritmo é que uma entrada de tamanho 1x foi transformada em uma de tamanho 4x. Para retornar a uma imagem de 1x é necessário analisar a codificação gerada e realizar algumas transformações. Considerando C como sendo a codificação de um ponto resultante da vetorização:

- CCCC reduz para uma cópia C;
- CCC elimina os C's;
- CC ignora;
- C retorna o próprio C.

Com isso, o novo código não será muito extenso, além de evitar que figuras com linhas de 45 graus se tornem uma longa “escada”, ao invés de uma linha oblíqua.

Um método utilizado para melhorar o desempenho desse algoritmo é guardar a última direção do movimento e usá-la como primeiro caso de teste na próxima iteração, visto que a cada uma delas, o algoritmo deve comparar a direção do próximo ponto. Isso melhora desempenho já que muitas vezes o algoritmo se move no mesmo sentido.

Com este algoritmo implementado seria possível vetorizar as imagens de Moiré e possibilitaria a obtenção das características descritas no capítulo 3.4.

## 5 SISTEMA DESENVOLVIDO

O software desenvolvido neste projeto auxilia na identificação das estruturas de Moiré, utilizando os conhecimentos da área de processamento de imagens e do padrão de Moiré.

O software foi desenvolvido com a ferramenta Visual Studio 2008, na linguagem C++ e com o auxílio da biblioteca de OpenGL\*. Ele está dividido em três camadas. A primeira é responsável por fazer a interação com o usuário. Esta camada se comunica com o arquivo *OpenGL.h* (segunda camada), o qual faz a carga e o armazenamento das imagens, além da parte comportamental do sistema. Para isso, a segunda camada utiliza a classe *ImageClass* que é usada para carregar e armazenar as imagens e comunica com a última camada, onde os algoritmos que processam as imagens são implementados (arquivo *AlgoritmosProImagens.cpp*).

As estruturas destacadas no *software* podem ser utilizadas em diversas áreas, tais como a geração de modelos digitais de objetos tridimensionais, a análise de simetria de objetos e o auxílio ao diagnóstico de escolioses (Figura 38) e deformidades na sola do pé.



Figura 38 Aplicação da Técnica de Moiré de Sombra nas costas do paciente

Fonte: (Hertz H, 2005)

---

\* <http://www.opengl.org/> - The Industry's Foundation for High Performance Graphics

Neste *software*, foram aplicados sobre imagens com o padrão de Moiré os algoritmos descritos no capítulo 4 e, a partir da imagem afinada, foram implementados algoritmos para destacar as características descritas na seção 3.4. Detalhes sobre como foram implementados os recursos disponíveis no sistema são apresentados nos capítulos 6.1 e 6.2.

No sistema, o usuário pode realizar várias tarefas, entre elas destacam-se a possibilidade **selecionar e editar as franjas**.

A seleção das franjas é a primeira tarefa a ser realizada pelo usuário, pois permite que este informe sobre quais estruturas deseje realizar as próximas operações.

Para facilitar a seleção das franjas, foi implementada uma funcionalidade chamada de *Destacar Franjas*, na qual todas as franjas da imagem original são separadas em cores diferentes e copiadas para uma segunda imagem, para possibilitar uma melhor manipulação (Figura 45).

Como a separação das franjas pode gerar incorreções devido a baixa qualidade das imagens ou a erro de digitalização, após a separação das franjas, o usuário pode selecionar um segmento da franja, removendo-o ou unindo com outro. Para a seleção de uma franja o usuário clica em um ponto próximo a ela e o *software* copia a estrutura para uma terceira imagem (Figura 46).

Conforme foi mencionado antes, é possível selecionar mais de uma franja, montando uma imagem que contenha somente as estruturas que interessem ao usuário. Isso facilita a utilização do programa, mostrando para o usuário, de forma mais clara, as franjas selecionadas, para que este tenha certeza que as características estarão sendo extraídas do local desejado.

A partir das estruturas selecionadas, é possível realizar operações de medida sobre elas. Estas operações são:

- **Determinação dos pontos de máximo e mínimo:** estes representam os pontos onde a altura (eixo  $y$ ) de uma franja é maior (ponto de máximo) ou menor (ponto de mínimo) e a partir destes pontos é calculada a inclinação. O programa permite ao usuário selecionar a franja que os pontos de máximo e mínimo devem ser encontrados para o calculado da inclinação. Um exemplo está na Figura 47.

- **Medição da inclinação de uma franja:** nesta operação, a partir dos pontos de máximo e mínimo definidos anteriormente, é possível calcular a inclinação entre eles. Para a visualização desta funcionalidade, uma reta é desenhada ligando estes pontos e o valor da inclinação é mostrado na tela do programa Figura 62.

- **Cálculo do período:** para calcular o período, o usuário seleciona dois pontos que desejar, o programa então calcula a distância e desenha uma linha entre eles (Figura 60).

- **Exibição dos pontos de inflexão:** A partir das franjas selecionadas, o programa encontra os pontos de inflexão destas franjas, ou seja, pontos em que ocorram mudanças no sentido do eixo y.

- **Medição da variação angular entre dois pontos de inflexão:** É possível selecionar dois pontos de inflexão destacados em uma imagem, e então exibir a informação referente à variação angular entre eles (Figura 39).

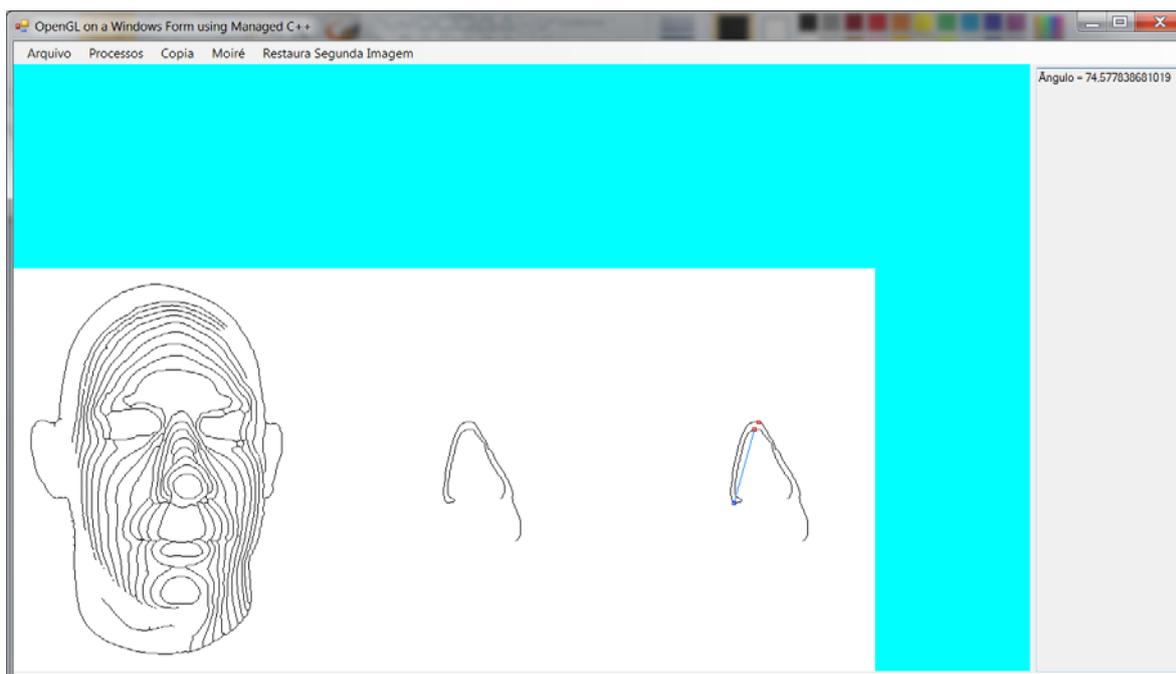


Figura 39 Variação angular entre pontos de inflexão

Fonte: O Autor (2010)

No Trabalho de Conclusão I, foi dito que seria possível realizar o cálculo da profundidade do objeto sobre o qual a sombra de Moiré está projetada, porém como foram utilizadas imagens de trabalhos de outros autores, não é possível realizar esta operação, pois para isto seriam necessárias algumas informações a respeito da

configuração do equipamento utilizado para a captura da imagem. Mesmo não sendo possível realizar este cálculo, é feito o cálculo do período, que pode ser usado futuramente para descobrir a profundidade.

## 6 DETALHES DE IMPLEMENTAÇÃO DO PROCESSAMENTO DAS FRANJAS

Nesta seção serão explicados os algoritmos utilizados para o pré-processamento da imagem e para implementação das funcionalidades descritas no capítulo 5. Junto à explicação sobre a implementação, serão feitos comentários sobre os resultados e as dificuldades encontradas em certas etapas.

Embora todos os algoritmos apresentados neste capítulo tenham sido implementados ao longo do trabalho, alguns não foram utilizados na versão final do programa desenvolvido, pois não alcançaram o resultado previsto em termos de qualidade. Nestes casos os algoritmos foram substituídos por outras ferramentas de processamento de imagem que também implementam estes algoritmos. Mesmo assim, para apresentar o resultado dos algoritmos desenvolvidos, foi feito um comparativo entre estes resultados e os resultados obtidos pela utilização das referidas ferramentas.

É importante ressaltar que a qualidade do resultado do pré-processamento da imagem, está diretamente ligada à qualidade da imagem de entrada, ou seja, os métodos desenvolvidos ao longo do trabalho não funcionam para qualquer imagem. Serão demonstrados no capítulo referente aos resultados, casos em que a imagem possibilita a aplicação destes métodos, e casos em que o resultado obtido não foi satisfatório.

O software de apoio utilizado foi o **ImageJ**\*. Este é uma ferramenta de utilização gratuita para fins não comerciais, multiplataforma e desenvolvida na linguagem de programação Java. O programa é capaz de mostrar, analisar, editar, salvar e imprimir imagens em 8-bit, 16-bit e 32-bit.

---

\* <http://rsbweb.nih.gov/ij/> - Image Processing and Analysis in Java

## 6.1 Pré-processamento da imagem

Para a etapa do pré-processamento, o algoritmo inicial utilizado no tratamento da imagem foi a segmentação das franjas de Moiré, através do método implementado neste software. Detalhes sobre o algoritmo podem ser vistos no capítulo 4.1. Este método, porém não deu bons resultados, pois as imagens utilizadas apresentavam muitos *pixels* de “sujeira”. Verificou-se então que para melhorar o resultado, poderia ser utilizado, previamente à segmentação, um algoritmo de suavização Figura 40.

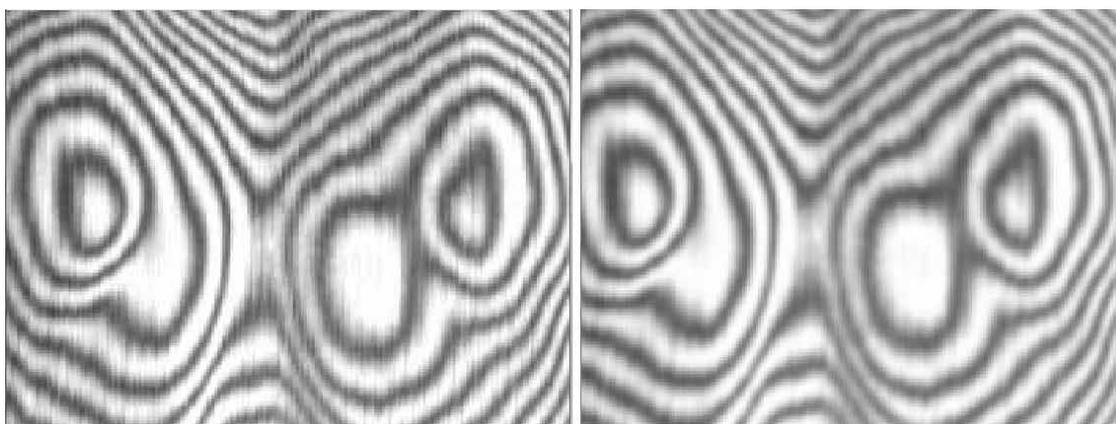


Figura 40 Imagem normal (esq.) e Imagem com Smooth (dir.)

Fonte: O Autor (2010)

O algoritmo utilizado para desenvolver o filtro de suavização (Smooth) é apresentado no Algoritmo 3.

### Algoritmo 3 - Suavização de Imagens

```
suaviza(Imagem Origem, Imagem Destino){
  para todos pontos do eixo x{
    para todos pontos do eixo y{
      calcula valor médio da intensidade da cor neste ponto
      para o ponto(x,y) da imagem destino seta RGB com a média
    }
  }
}
```

Após a aplicação da suavização, aplicou-se então a segmentação das franjas. Este passo garantiu um resultado mais preciso, com franjas mais definidas Figura 41.

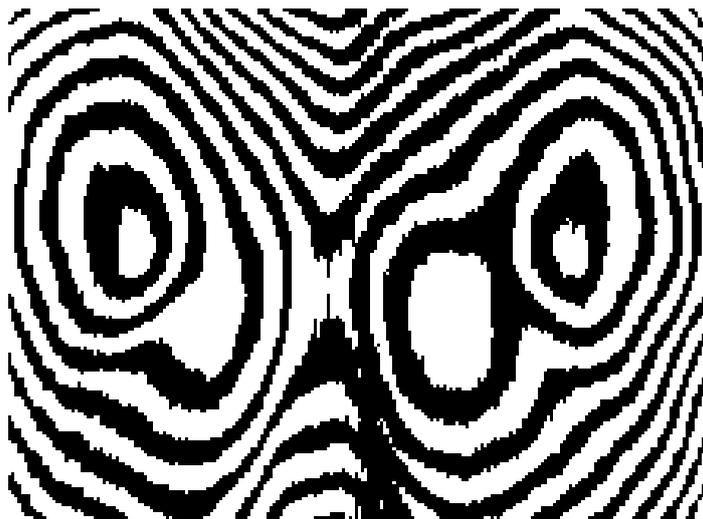


Figura 41 Resultado da segmentação

Fonte: O Autor (2010)

Com isto pode-se então aplicar o algoritmo de afinamento. A aplicação correta deste algoritmo encerra a parte de pré-processamento da imagem, gerando como resultado uma imagem que contenha franjas de apenas um pixel de largura.

A implementação do afinamento, descrito na seção 4.2, não gerou resultados satisfatórios, pois o algoritmo não eliminava os *pixels* de sujeira que restaram na imagem, e criava linhas ligando as franjas a estes pontos de sujeira. Um exemplo deste caso pode ser visto no lado esquerdo da Figura 42.

Para resolver este problema utilizou-se o algoritmo *Skeletonization* do *software ImageJ*. Este conseguiu resolver, em grande parte, o problema dos pontos soltos, gerando um resultado mais adequado aos propósitos do sistema, como visto à direita na Figura 42.

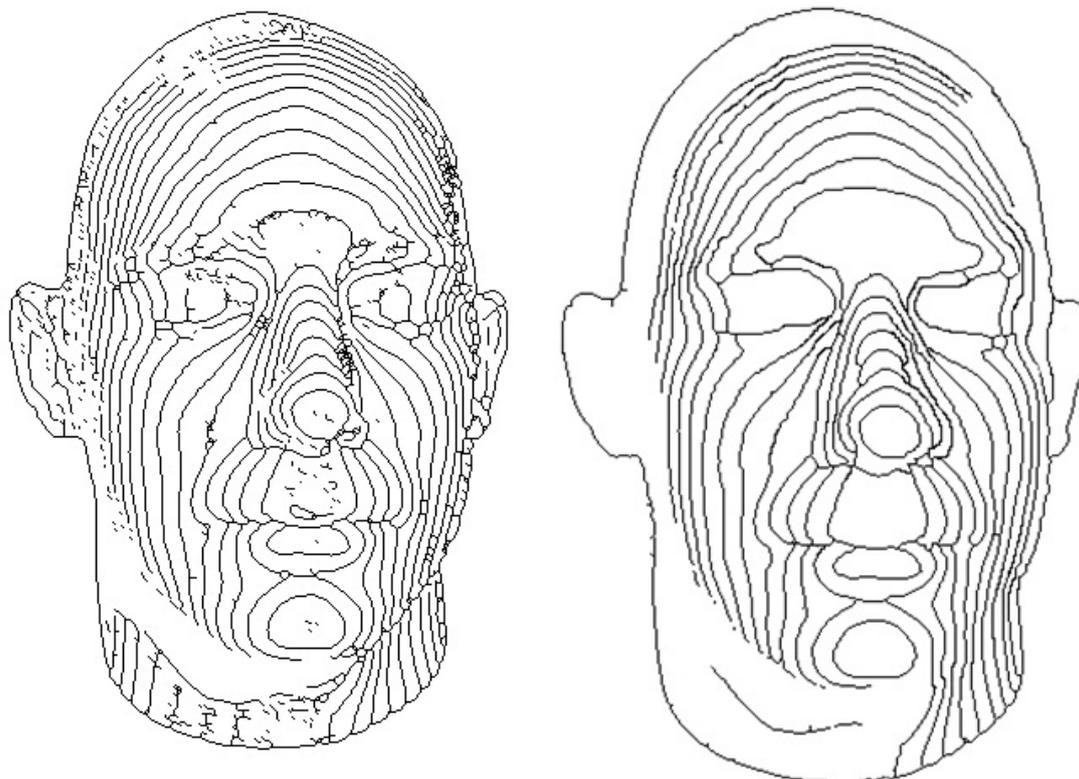


Figura 42 Imagem afinada pelo algoritmo implementado no trabalho (e) e pelo algoritmo de Skeletonization do software ImageJ (d)

Fonte: O Autor (2010)

Segundo a pesquisa realizada no *Trabalho de Conclusão I*, para facilitar o processo de extração de informações, poderia ser necessário a aplicação do algoritmo de vetorização sobre a imagem afinada. Este foi desenvolvido conforme o capítulo 4.3, porém verificou-se que o resultado gerado não ajudaria no resultado final do processo, pois as mesmas operações poderiam ser realizadas de maneira mais simples e eficaz sobre a imagem afinada. A vetorização desenvolvida neste trabalho contornou toda a imagem afinada, não oferecendo nenhuma vantagem, já que não acrescentou nenhuma facilidade na implementação das características descritas na seção 6.2. Além disso, trata-se de um processo lento, visto que a imagem original necessitava ser aumentada para que o algoritmo funcionasse e após isso redimensionada ao seu tamanho original.

Um exemplo de vetorização é exibido na Figura 43, onde os pontos em roxo apresentam todos pontos que o algoritmo gerou.

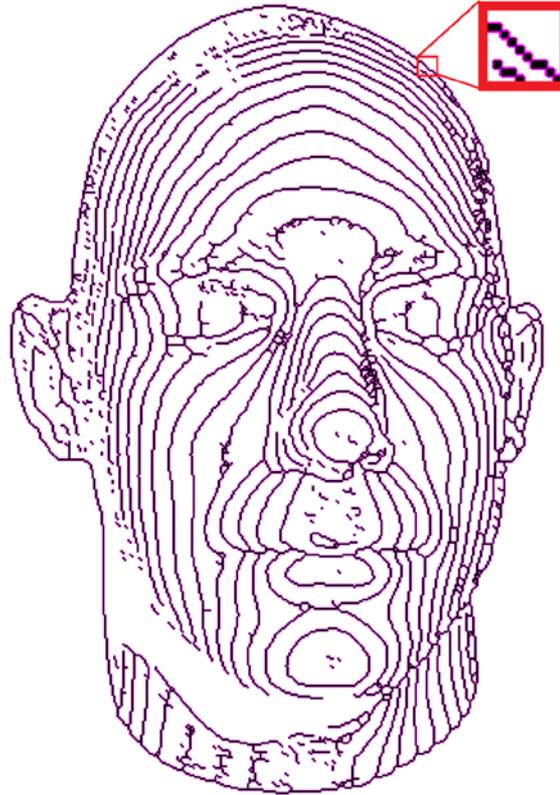


Figura 43 Imagem vetorizada - pixels roxos pontos gerados - pixels pretos são a imagem original aumentada

Fonte: O Autor (2010)

## 6.2 Extração das Características das Curvas de Moiré

Este capítulo apresenta como os métodos de extração de características foram implementados e como utilizá-los. Para a aplicação destes métodos assume-se que a imagem já foi pré-processada com os algoritmos descritos na seção 6.1.

Foram desenvolvidos os seguintes algoritmos para auxiliar na obtenção das características das franjas de Moiré descritas na seção 3.4:

- Cálculo do Período entre duas Franjas
- Pontos de máximo e de mínimo de uma Franja
- Pontos de inflexão
- Destacar franjas
- Selecionar franjas
- Métodos auxiliares

### 6.2.1 Destacar franjas

A funcionalidade de destacar franja é bastante útil neste software, pois só após sua aplicação o usuário poderá montar uma imagem a ser usada para obter as informações desejadas de maneira precisa.

Este método é aplicado sobre uma imagem afinada e separa as franjas em partes quando há pontos com intersecção, isto é, locais onde um ponto é ligado à outros 3 ou mais *pixels* pretos como mostra na Figura 44, pintando cada parte com uma cor diferente (Figura 45).

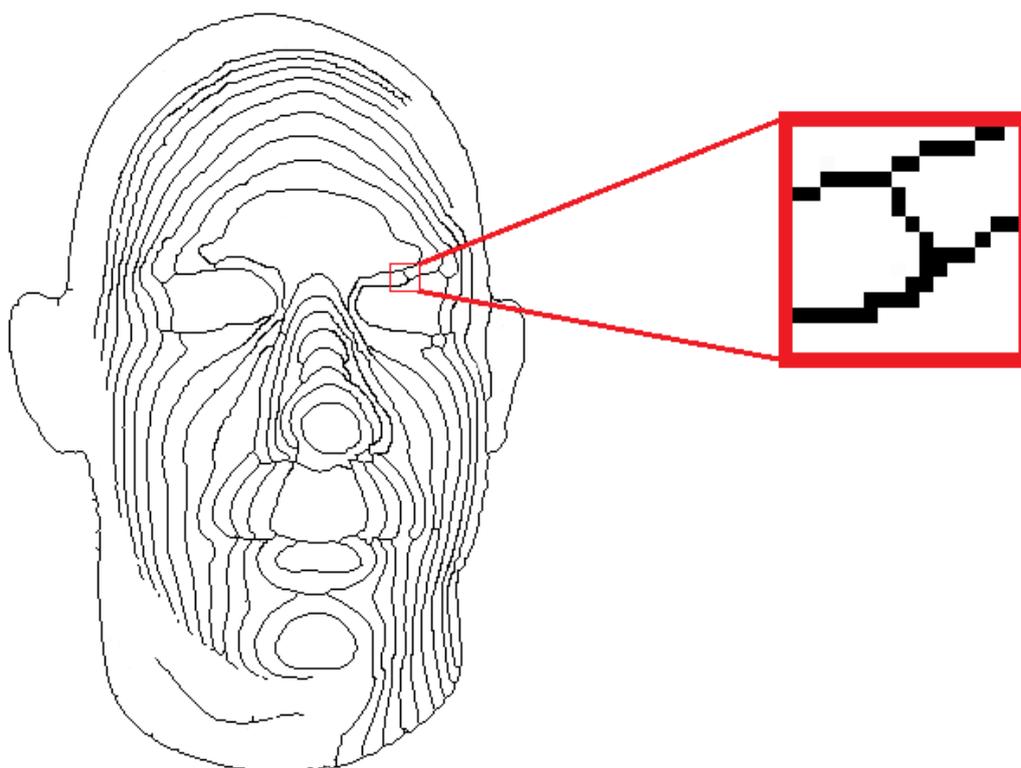


Figura 44 Franjas com intersecção.

Fonte: O Autor (2010)

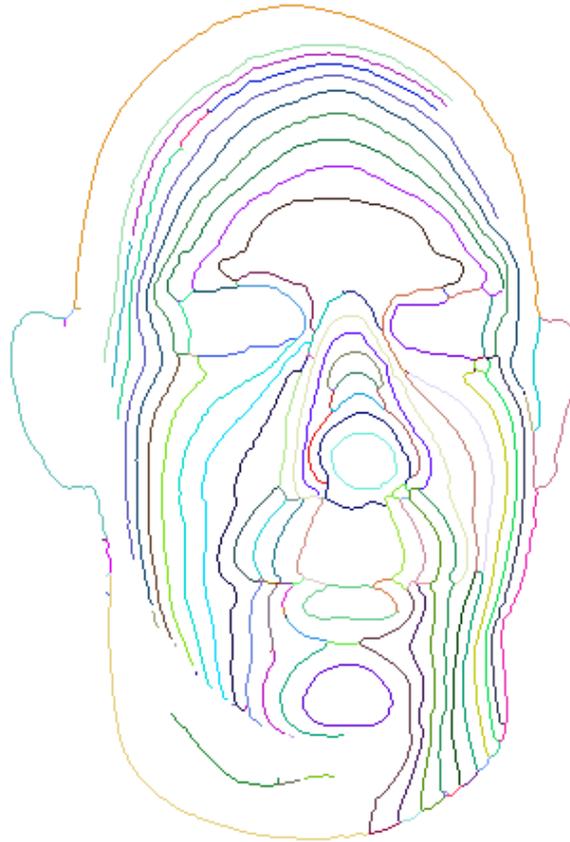


Figura 45 Franjas destacadas com várias cores.

Fonte: O Autor (2010)

Este algoritmo consiste em procurar pontos pretos na imagem. Quando um ponto deles é encontrado, se ele é o início de uma franja ou o ponto é ligado a outros 3 pontos pretos, o algoritmo percorre a franja até chegar ao fim dela ou chegar a outro ponto de intersecção, pintando-a com alguma cor aleatória.

Um pseudo-algoritmo é apresentado a seguir:

#### Algoritmo 4 - Destaca Franjas

```

DestacaFranjas (imagem afinada)
  cria imagem em branco saida
  Enquanto há pontos p que são pretos na imagem afinada
    se p é início ou ponto de intersecção de uma franja
      gera cor c
      Enquanto houver pontos f na franja e f não for
intersecção
        pinta f em saida com a cor c
  
```

O resultado deste algoritmo é mostrado na Figura 45.

### 6.2.2 Selecionar franjas

A função de selecionar franjas permite ao usuário montar uma imagem somente com as franjas que lhe interessam, assim permitindo que os resultados sejam mais precisos.

Este método é aplicado sobre a imagem com as franjas destacadas, resultantes da aplicação da função descrita acima. Esta permite ao usuário montar imagens com segmentos de franjas que desejam, somente clicando perto delas para adicionar ou clicando novamente perto para retirá-las.

O algoritmo foi implementado de forma que, dado um ponto  $(x, y)$ , é buscado o ponto preto mais próximo de  $(x, y)$  com o método descrito na seção 6.2.6. Tendo o um ponto da franja, o algoritmo busca na imagem de saída se o ponto correspondente já está pintado. Caso ele seja branco, o algoritmo percorre a franja, na imagem destacada pela sua cor e pinta este ponto na imagem de saída com a cor preta. Se o ponto já estiver pintado na imagem de saída, o algoritmo percorre na imagem com as franjas separadas e pinta a saída de branco.

A seguir é apresentado o algoritmo:

#### Algoritmo 5 - Seleciona Franjas

```

SelecionaFranjas(imagem, x, y)
    imagem em branco saida
    ponto p = pega ponto proximo
    se saida em p é branco
        Enquanto há pontos f da franja na imagem
destacada
            pinta f de preto na saida
        senao
            Enquanto há pontos f da franja na imagem
destacada
                pinta f de branco na saida

```

O resultado deste algoritmo pode ser visto na Figura 46.

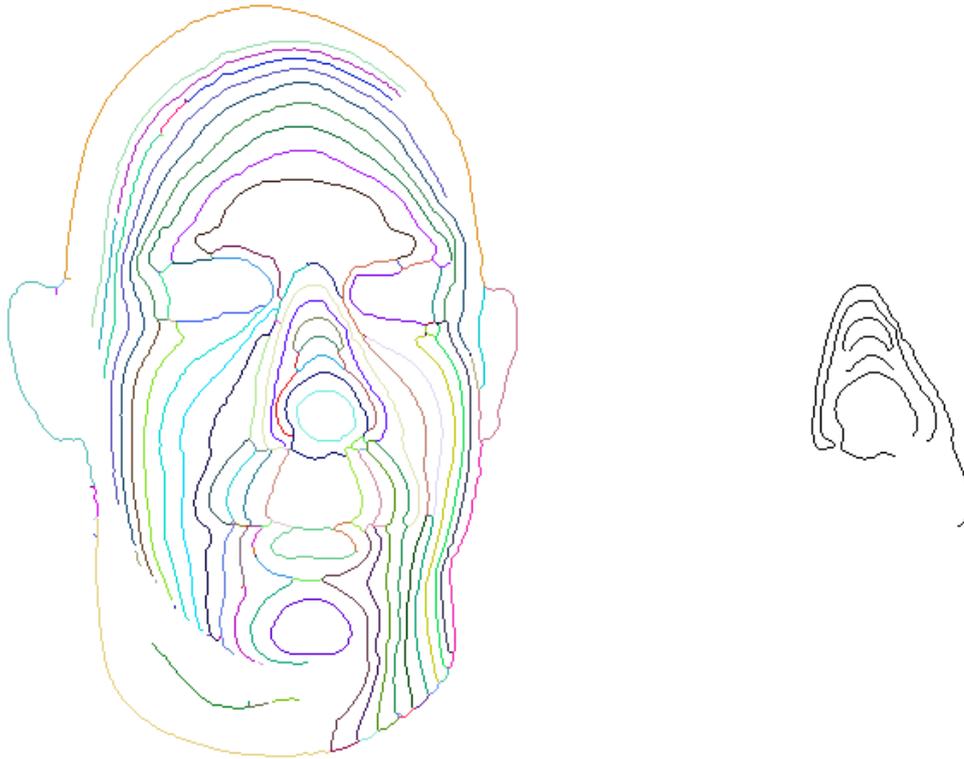


Figura 46 Imagem com franjas destacadas e com algumas partes de franjas selecionadas  
Fonte: O Autor (2010)

Na Figura 46 pode-se perceber que foram selecionadas as franjas que podem conter informações úteis ao usuário, além de deixar a imagem clara.

### 6.2.3 Cálculo do Período entre duas Franjas

Conforme foi definido na seção 3.4 o período é a distância em *pixels* entre duas franjas. Esta informação pode ser usada para determinar a profundidade de uma imagem. Para isso, utilizou-se o algoritmo, que a partir de dois pontos ( $x$ ,  $y$ ) na imagem, calcula-se a distância entre eles conforme o Algoritmo 6. Uma reta é traçada entre os dois pontos para melhor visualização.

#### Algoritmo 6 - Distância entre Duas Franjas

```

calculaPeriodo(xPonto1, yPonto1, xPonto2, yPonto2){
    calcula distancia entre Ponto1 e Ponto2 com
    raiz(potência de 2(xPonto1 - xPonto2) somado a potência de
2(yPonto1 - yPonto2))
    traça uma reta entre os dois pontos
    retorna o valor da distancia
}

```

### 6.2.4 Pontos de máximo e mínimo

O método que calcula o ângulo e exibe os pontos máximo e mínimo de uma franja é aplicado sobre uma imagem afinada e em preto e branco. Ele consiste em, a partir de um ponto  $(x, y)$  de uma franja que o usuário seleciona na figura, percorre todos os pontos da franja, comparando-os para ver se eles são pontos de máximo ou de mínimo.

Para passar por todos os pontos a partir de  $(x, y)$ , adiciona este ponto numa pilha *pontos*. Para iniciar o resto da franja, este ponto é marcado como ponto máximo e mínimo. Após isso é feito um *loop* onde é executado enquanto a pilha não está vazia. Dentro do laço, é desempilhado o primeiro ponto marcando-o com a cor branca, comparando a altura (eixo  $y$ ) dos pontos de máximo e mínimo, caso seja, o ponto desempilhado é atribuído ao novo ponto de máximo, caso o valor de seu  $y$  seja maior que o antigo, ou atribuído ao novo ponto de mínimo caso  $y$  seja menor que o ponto mínimo. Após, é verificado se tem algum vizinho preto. Caso tenha, este é adicionado na pilha e o loop reinicia enquanto houver pontos na pilha.

Posteriormente os pontos de máximo e de mínimo já são conhecidos, porém eles devem ser aproximados na horizontal para que o ângulo seja calculado de maneira mais precisa. Para resolver este problema, foi necessário 2 passos. No primeiro, o ponto máximo tenta andar o mais que conseguir para o lado do ponto máximo enquanto houver vizinhos e o valor de  $y$  for o mesmo. O segundo consiste em fazer o mesmo para o ponto mínimo.

Após isso, basta calcular o ângulo, o qual está descrito na seção 6.2.6.

Um pseudocódigo deste algoritmo é descrito a seguir.

#### Algoritmo 7 - Acha Ponto máximo e mínimo

```

pMax = ponto(x, y)
pMin = ponto(x, y)
pilha pontos.empilha(ponto(x, y))

enquanto pilha não está vazia
    ponto p = pilha.desempilha
    pinta p de branco
    se p.y > pMax.y
        pMax = p
    se p.y < pMin.y
        pMin = p
    para cada vizinho V de p
        se V é preto
            pontos.empilha(v)
    se há algum ponto a na franja com a.y igual ao pMin.y e
    a.x é mais próximo de pMax.x

```

```

    pMin = a
    se há algum ponto a na franja com a.y igual ao pMax.y e
    a.x é mais próximo de pMin.x
    pMax = a
    pinta pMax de verde e pMin de vermelho
    faz reta entre pMax e pMin
    retorna calcula angulo(pMax, pMin)

```

O resultado deste algoritmo é exibido na Figura 47.

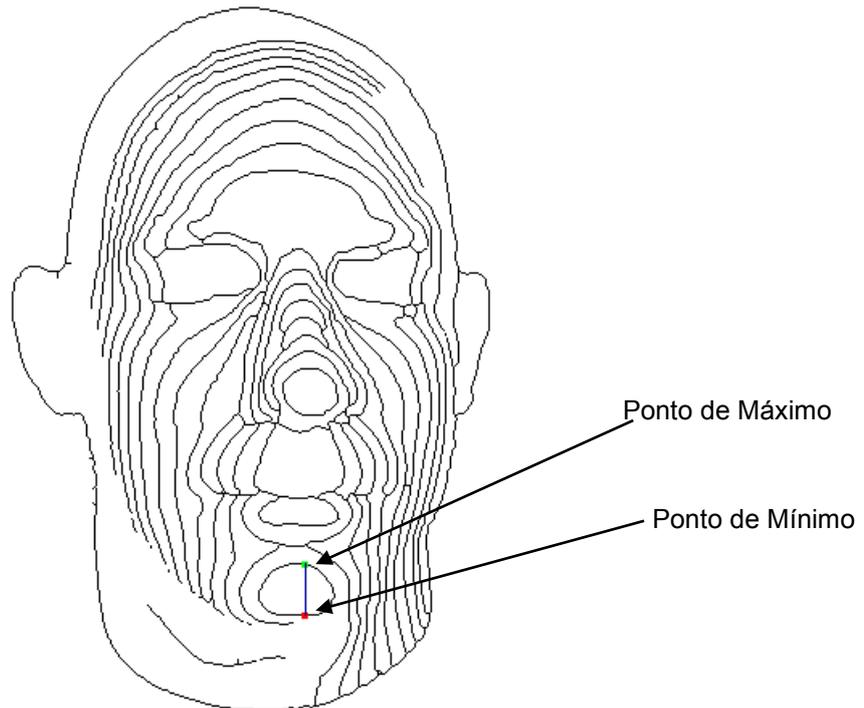


Figura 47 Pontos máximo (verde) e mínimo (vermelho).

Fonte: O Autor (2010)

### 6.2.5 Pontos de inflexão

Os pontos de inflexão permitem que ângulos entre dois pontos selecionados sejam calculados. Primeiramente o algoritmo passa por toda a imagem procurando pelos pontos e, após, permite que o usuário selecione 2 para que seja exibido o ângulo.

O método que procura pelas inflexões na imagem afinada, consiste em achar o início de uma franja, percorrendo-a até achar 15 pixels que são colocados num vetor circular. Quando eles são encontrados, o ponto central deste vetor é comparado com os demais pontos do *array*, caso a soma da altura (eixo *y*) dos pontos anteriores seja menor que zero e dos seguintes também seja menor que

zero, então ele é um ponto de inflexão. Ou então, caso a soma da altura dos pontos anteriores seja maior que zero e dos seguintes também seja maior, é outro ponto de inflexão. Quando um ponto de inflexão é encontrado (Figura 48), então o vetor é reinicializado. Se o ponto não for uma inflexão, a função segue buscando os pontos na franja e inserindo no início do vetor circular, e sempre retirando o ponto mais antigo (último).



Figura 48 Pontos de Inflexão: Vermelhos - soma dos pontos vizinhos são menores que 0.  
Azul - soma dos pontos vizinhos são maiores que 0.

Fonte: O Autor (2010)

Um pseudo-algoritmo é apresentado a seguir:

#### Algoritmo 8 - Acha ponto de inflexão

```
AchaInflexao(imagem)
  Para cada franja f da imagem
    vetor[15]
    contador = 0;
    para cada ponto p de f
      vetor.adiciona(p)
      vetor.removeMaisAntigo
      contador++
      marca p
      se contador >= 15
        acumDireita = soma y's a partir do meio do
vetor
        acumEsquerda = soma y's até o meio do vetor
        se(acumDireita < 0 && acumEsquerda < 0){
          é ponto de inflexão
          contador = 0
        se(acumDireita > 0 && acumEsquerda > 0){
          é ponto de inflexão
          contador = 0
```

### 6.2.6 Métodos auxiliares

Esta seção destina-se a mostrar outros algoritmos que foram desenvolvidos e são usados em mais de um método descrito nas seções anteriores.

O primeiro deles serve para medir o ângulo entre 2 pontos. Ele é usado nos algoritmos descritos nas seções 6.2.4 e 6.2.5. Esta função faz um triângulo retângulo com os dois pontos passados por parâmetro e mede ângulo do primeiro ponto em relação a uma linha reta horizontal. O resultado é retornado em graus radianos.

O pseudocódigo do algoritmo implementado é:

#### Algoritmo 9 - Ângulo

```
angulo(ponto p1, ponto p2){
    catetoLargura = modulo (p2.x - p1.x
    catetoAltura = modulo(p2.y - p1.y)
    hipotenusa = raiz ((catetoLargura^2)+(catetoAltura^2))
    senoAnguloBaixo = catetoAltura / hipotenusa
    retorna arcseno (senoAnguloBaixo) * 180 / PI
}
```

Outro método que auxiliou no desenvolvimento, foi o de pegar a franja mais próxima de um ponto  $(x, y)$ . Este, normalmente é usado quando o usuário interage com a imagem através do mouse, pegando a franja mais próxima do clique. O algoritmo recebe o ponto do clique por parâmetro e busca nas 4 direções (esquerda, direita, acima e abaixo). Quando encontra um ponto diferente de branco, este é retornado.

## 7 RESULTADOS DE TESTES

Este capítulo apresenta os resultados obtidos pela aplicação dos métodos explicados no trabalho. Serão mostrados resultados de cada etapa do processo.

### 7.1 Imagem original

Para demonstração dos resultados serão utilizadas quatro imagens (Figura 49). A primeira delas é a imagem das franjas de Moiré aplicadas em um rosto de um manequim. A segunda mostra o uso da técnica sobre uma calota. A terceira e a quarta imagens mostram a aplicação do efeito nas costas de uma pessoa, porém a qualidade da terceira é melhor do que a quarta.

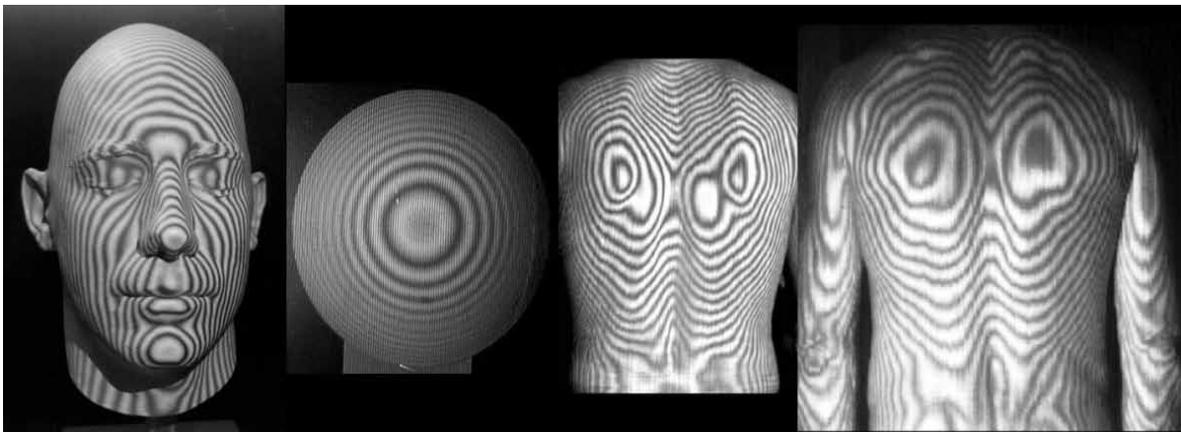


Figura 49 Imagens originais.

Fonte: O Autor (2010)

Nota-se que a qualidade da quarta imagem está comprometida devido a menor nitidez.

### 7.2 Aplicação da *Suavização (Smooth)*

Como explicado anteriormente, este método prepara a imagem para o uso posterior da segmentação. O filtro deixa a imagem com menos contraste nos pixels, diminuindo a sujeira. É possível perceber isso nas Figura 50, Figura 51, Figura 52 e na Figura 53.

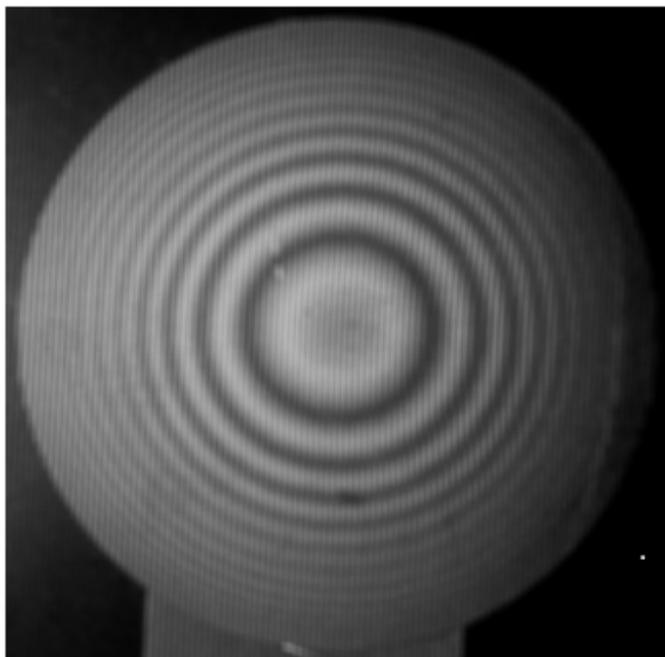


Figura 50 Calota com aplicação da suavização  
Fonte: O Autor (2010)



Figura 51 Suavização aplicada as Costas de uma pessoa  
Fonte: O Autor (2010)



Figura 52 Exemplo de suavização aplicada as costas.

Fonte: O Autor (2010)



Figura 53 Suavização na face de um manequim.

Fonte: O Autor (2010)

### 7.3 Aplicação da Segmentação

Nesta etapa já é possível perceber que a qualidade da imagem original interfere no resultado. Os resultados da Figura 54 e Figura 55, mostra que o processo não funciona para qualquer tipo de entrada. Como o resultado da segmentação para estas duas imagens não foram bons, estas não serão utilizadas nos passos seguintes.

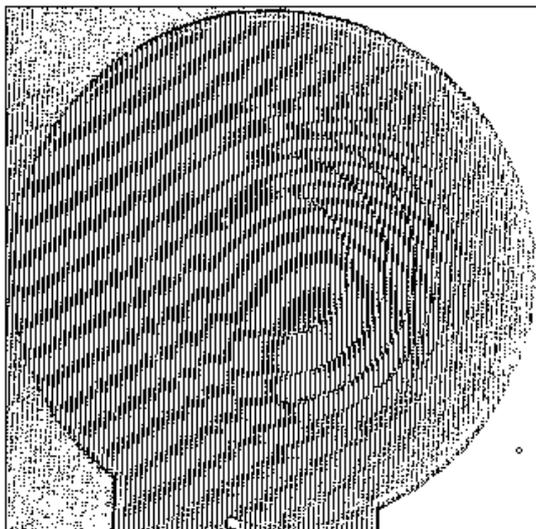


Figura 54 Segmentação aplicada a Figura 50.

Fonte: O Autor (2010)

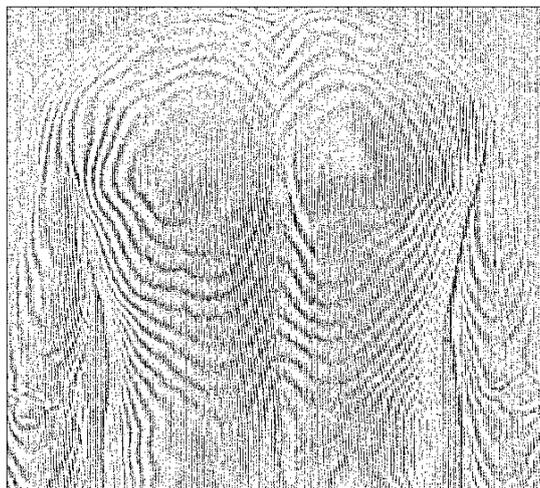


Figura 55 Segmentação aplicada a Figura 51.

Fonte: O Autor (2010)

Já os resultados obtidos nas imagens abaixo (Figura 56 e Figura 57), são adequados para aplicação do próximo passo do pré-processamento da imagem.

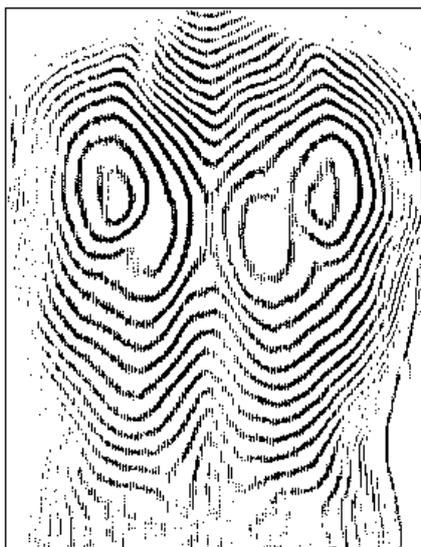


Figura 56 Segmentação aplicada a Figura 52.

Fonte: O Autor (2010)

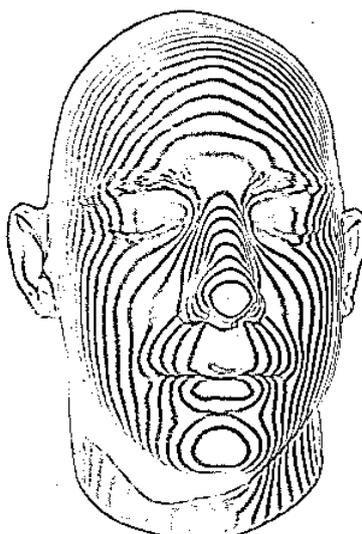


Figura 57 Segmentação aplicada a Figura 53.

Fonte: O Autor (2010)

#### 7.4 Aplicação do *Afinamento*

Aplicando o afinamento do *ImageJ*, obteve-se o resultado visto na Figura 58 e Figura 59. Esta etapa conclui a fase de pré-processamento, gerando como resultado imagens aptas a receber a aplicação dos algoritmos para extração das estruturas de Moiré.

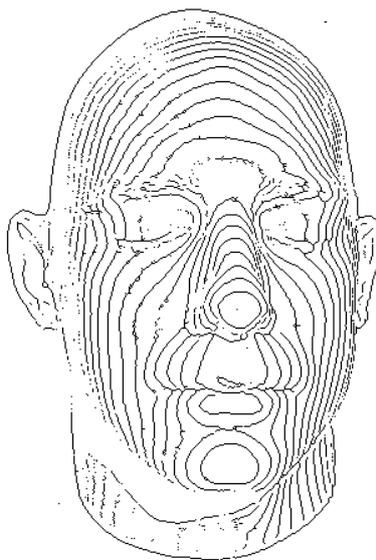


Figura 58 Imagem de manequim afinada.

Fonte: O Autor (2010)

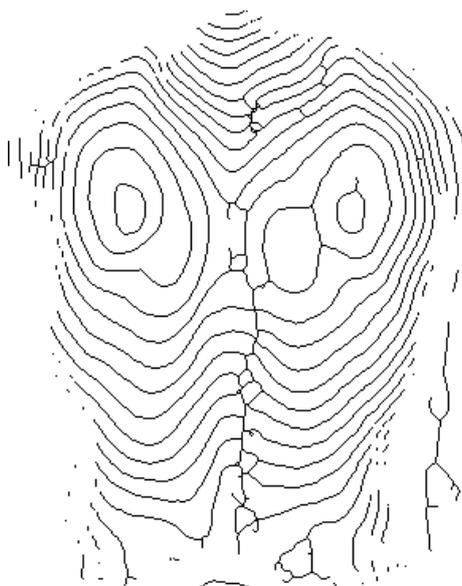


Figura 59 Imagem das costas afinada.

Fonte: O Autor (2010)

### **7.5 Aplicação do *Cálculo do Período***

Para testar o cálculo do período na imagem das costas, pegaram-se dois pontos próximos a região escapular esquerda, como mostra a Figura 60. Estes pontos foram ligados por uma linha vermelha para melhor representação. O valor da distância entre eles aparece no canto direito da tela.

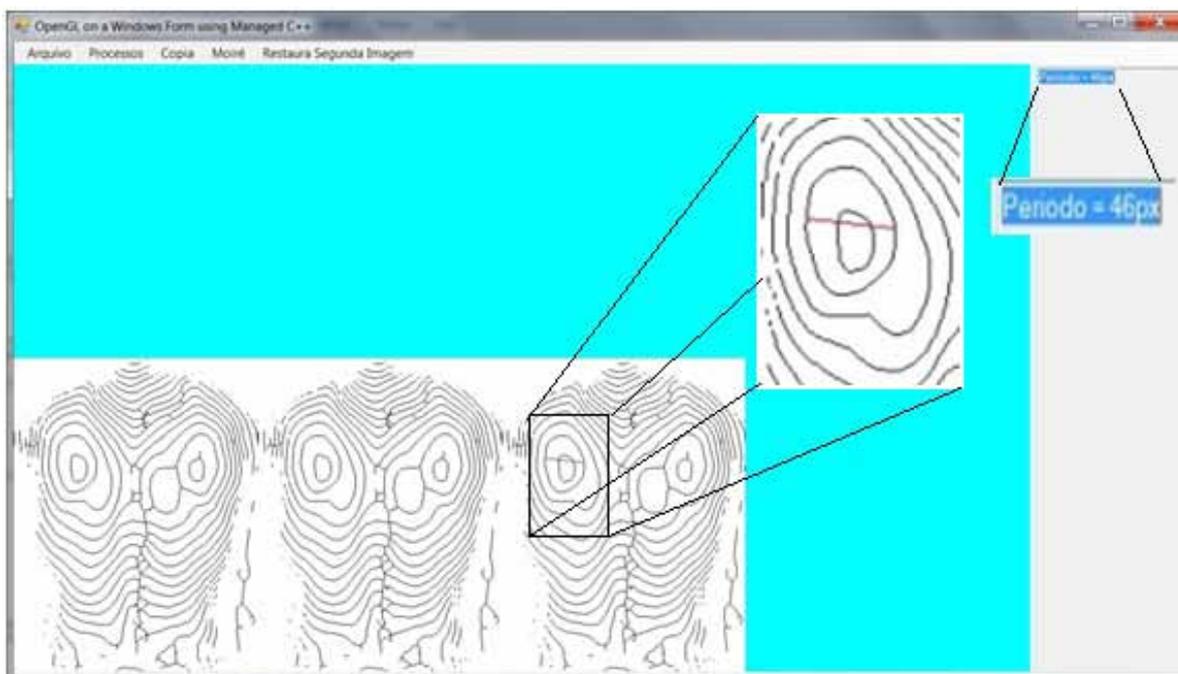


Figura 60 Cálculo do Período

Fonte: O Autor (2010)

O mesmo foi feito para a imagem do rosto, pegou-se dois pontos próximos a região central do nariz, traçou-se uma linha entre eles e o valor da distância foi exibido Figura 61.

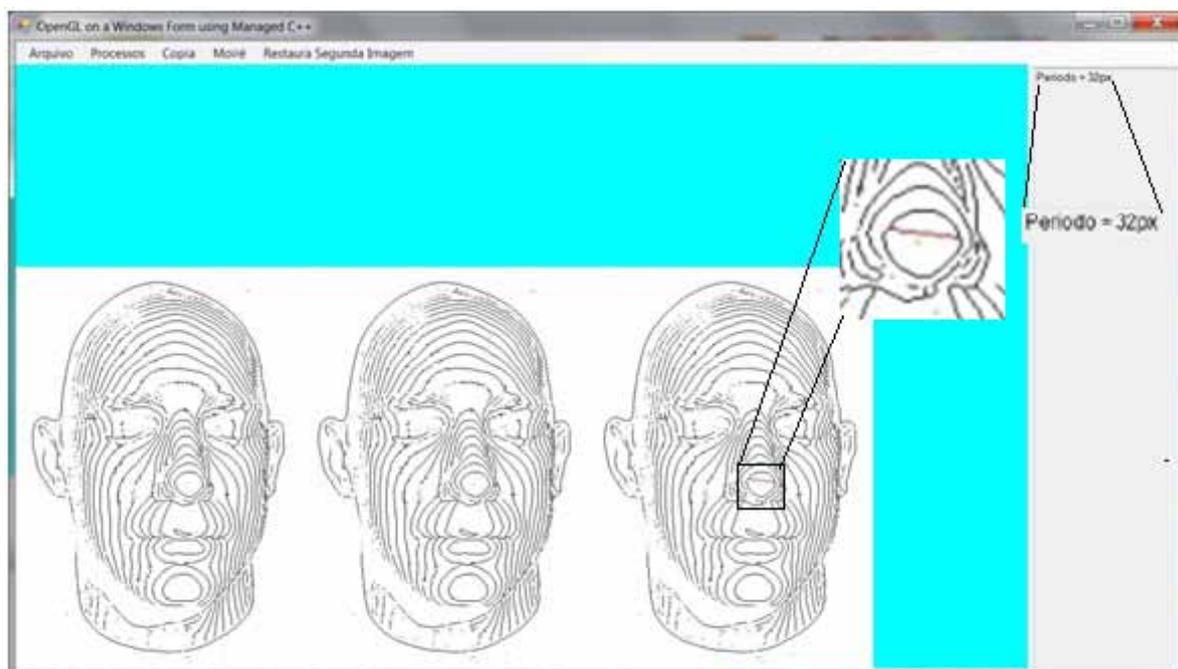


Figura 61 Cálculo do Período

Fonte: O Autor (2010)

O resultado para o cálculo desta estrutura foi obtido com sucesso para as duas imagens.

## 7.6 Aplicação do *Ponto de Máximo e Mínimo*

Existem duas maneiras de mostrar os pontos de máximo e mínimo de uma franja. A primeira forma é separando a franja que se deseja calcular, como na Figura 62. O ponto máximo é representado pelo ponto verde, o mínimo pelo vermelho e o ângulo entre estes pontos é exibido no canto superior direito.

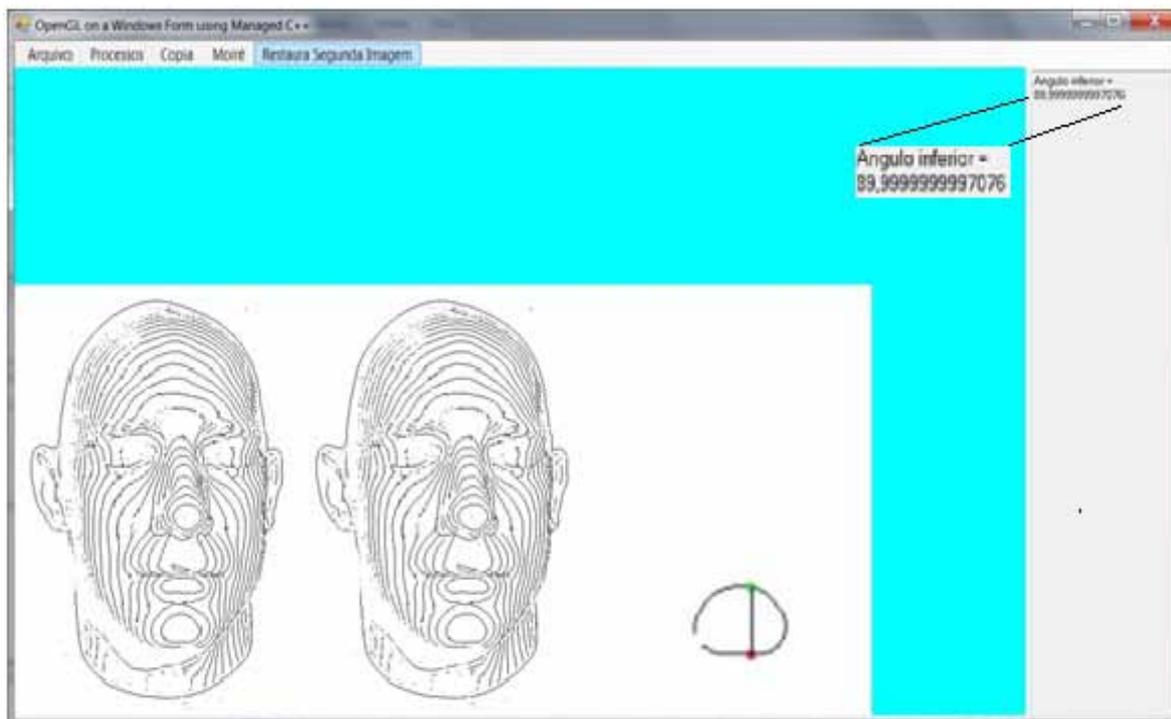


Figura 62 Pontos de máximo e mínimo.

Fonte: O Autor (2010)

A outra forma é mostrar os pontos dentro da figura completa. Da mesma maneira, o ponto máximo é representado pelo ponto verde, o mínimo pelo vermelho e o ângulo entre estes pontos é exibido no canto superior direito, como na Figura 63.

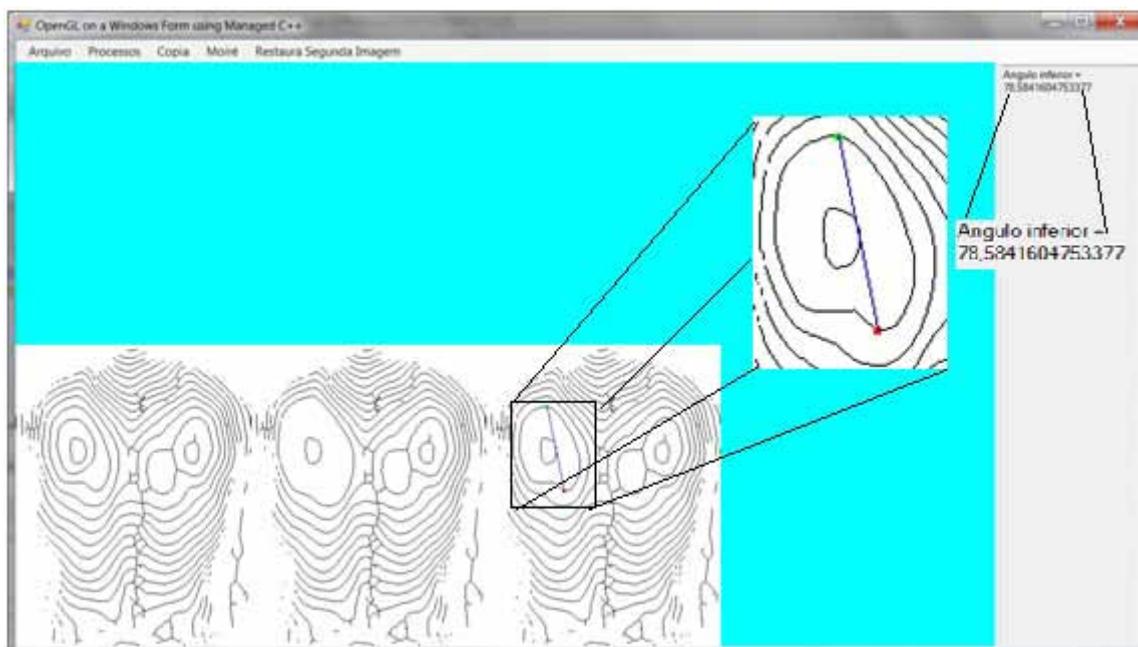


Figura 63 Ponto de Máximo e Mínimo

Fonte: O Autor (2010)

### 7.7 Aplicação do *Pontos de Inflexão*

Para calcular os pontos de inflexão, primeiro foi preciso selecionar as franjas desejadas. Como visto na Figura 64 que mostra franjas selecionadas da imagem das costas e na Figura 65 com franjas da imagem do rosto, foram encontrados pontos de inflexão nos locais corretos.

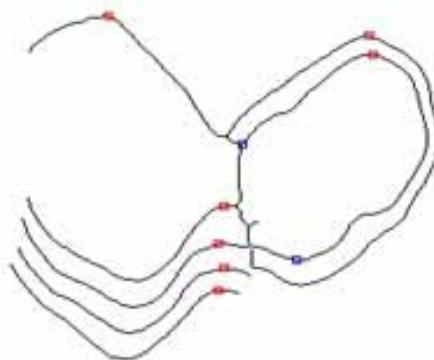


Figura 64 Pontos de Inflexão

Fonte: O Autor (2010)

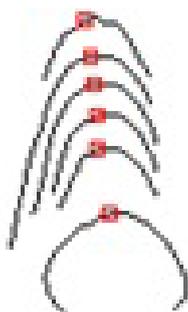


Figura 65 Pontos de Inflexão.

Fonte: O Autor (2010)

### 7.8 Aplicação do *Destacar Franjas*

Esta funcionalidade agregou facilidade para trabalhar os outros algoritmos. Facilitou a visualização, dividiu a parte que importa da imagem em diversas estruturas menores que podem ser analisadas individualmente. Os resultados vistos na Figura 66 e na Figura 67, mostram que as franjas são divididas em pontos de intersecção com outras franjas. Logo aquelas com um resultado melhor na etapa de pré-processamento, como a Figura 66, apresentaram franjas mais bem definidas.

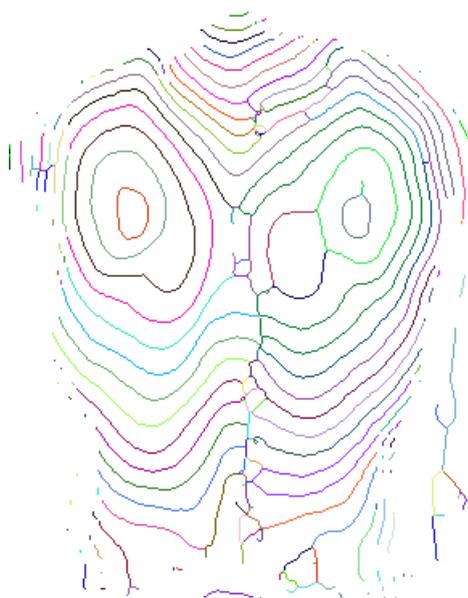


Figura 66 Franjas das costas destacadas.

Fonte: O Autor (2010)

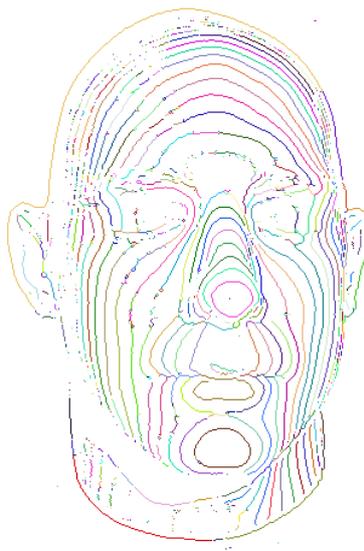


Figura 67 Franjas do rosto destacadas.

Fonte: O Autor (2010)

### 7.9 Aplicação do *Selecionar Franjas*

A partir de uma imagem com franjas destacadas, é possível selecionar àquelas que interessam, e a partir da imagem resultante das franjas selecionadas (Figura 68 e Figura 69), aplicar os outros algoritmos para extração de informações.

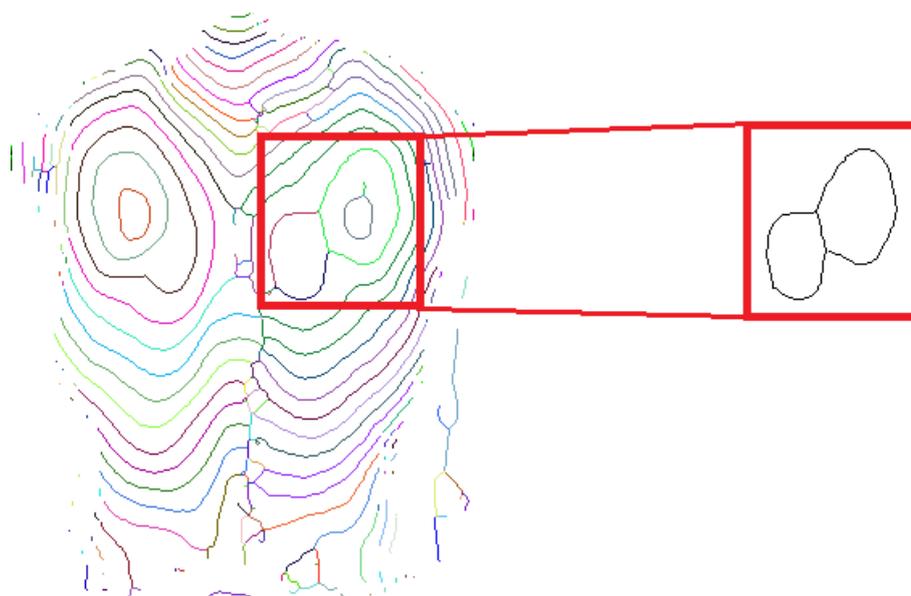


Figura 68 Franjas selecionadas a partir de uma imagem com as franjas das costas destacadas.

Fonte: O Autor (2010)

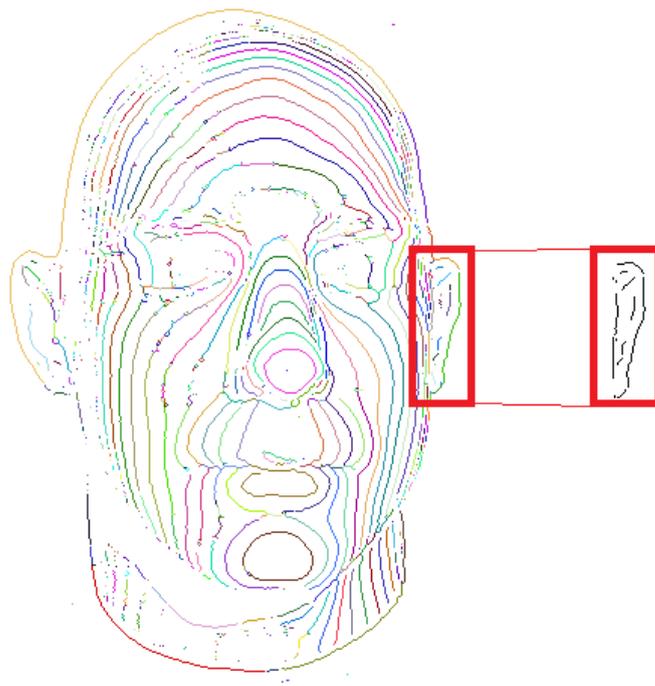


Figura 69 Franjas selecionadas a partir de uma imagem com as franjas do rosto destacadas.

Fonte: O Autor (2010)

## 8 CONCLUSÃO

Pode-se concluir que os algoritmos desenvolvidos ao longo do trabalho obtiveram um resultado satisfatório para todos os casos que tiveram como entrada imagens com boa qualidade. Como possíveis extensões para o trabalho poderão ser implementados filtros de pré-processamento que aperfeiçoem as etapas descritas no capítulo 4.

Foi possível demonstrar e calcular as principais estruturas, porém os passos seguidos não foram exatamente os mesmos previstos no Trabalho de Conclusão I. Alguns algoritmos não precisaram ser utilizados, como a vetorização, enquanto outros não alcançaram o resultado esperado, e foram substituídos por outra ferramenta de processamento de imagens.

Em suma, foi um trabalho que desenvolveu uma ferramenta que poderá servir como base para a criação de trabalhos, em áreas mais específicas, que façam o uso do fenômeno de Moiré.

## 9 BIBLIOGRAFIA

**A., Turner-Smith. 1997.** *Structured light surface measurement techniques*. 1997.

**Bastos, Vanessa Parodes. 2008.** Técnicas de Segmentação de Imagens para Recuperação de Informações Visuais . [Online] 2008. <http://paginas.ucpel.tche.br/~vbastos/segmenta.htm>.

**Batouche, M., Benlamri, R. e Kholadi, M. K. 1996.** A computer vision system for diagnosing scoliosis using Moiré images. *Elsevier Science*. 4, 1996, Vol. 26.

**CA, Sciammarella. 1982.** *The moiré method: a review*. 1982.

**Cavaco, Marco Antonio Martins. 2006.** EMC 5236 – Medição de Grandezas Mecânicas. *LabMetro*. [Online] 2006. [http://www.labmetro.ufsc.br/Disciplinas/EMC5236/OTICA\\_1.pdf](http://www.labmetro.ufsc.br/Disciplinas/EMC5236/OTICA_1.pdf).

**Cavalcante, Sedecias Lopes. 1999.** Esqueletonização, A linha Essencial. [Online] setembro de 1999. <http://www.inf.ufsc.br/~visao/1999/sedecias/index.html>.

**Centeno, Tania Mezzadri. 2009.** CPGei. [Online] 2009. <http://www.cpgei.cefetpr.br/~mezzadri/PDI/Parte07.htm>.

**Cloud, Gary. 1998.** *Optical methods of engineering analysis*. Cambridge: Cambridge University Press, 1998. p. 405.

**CreativePro. 2008.** CreativePro Graphic Design Software, Reviews, Tutorials, News and Resources. *www.creativepro.com*. [Online] 2008.

**Doué, Jean-François e Rubio, Ruben Gonzales. 1995.** Efficient and Robust 2D Shape Vectorization. [A. do livro] Alan W. Paeth. *Graphics Gems V*. s.l. : Academic Press, 1995.

**Durelli, A.J. e Riley, W.F. 1962.** Application of moiré methods to the determination of transient stress and strain distributions. *Journal of Applied Mechanics*. 4, 1962, Vol. 29.

**Ferreira, Flávia. 2008.** *Avaliação postural dos idosos de Porto Alegre-RS com uso da Técnica de Moiré de Sombra*. 2008.

**Foxx, John. 2006.** FotoSearch.  
<http://comps.fotosearch.com/bigcomps/JFI/JFI173/SK1811.jpg>. [Online] 2006.

**Gåsvik, K. J. 2002.** *Optical Metrology*. s.l. : John Wiley & Sons Ltd, 2002.

**Gonzales, Rafael C. e Woods, Richard E. 2000.** *Processamento de Imagens Digitais*. Knoxville : Edgard Blücher LTDA, 2000.

**HAMRA, ALBERTO e VOLPON, JOSÉ B. 1995.** . *Revista Brasileira de Ortopedia*. [Online] agosto de 1995. <http://www.rbo.org.br/materia.asp?mt=1277&ididioma=1>.

**Hertz, H., et al. 2005.** *Desenvolvimento da técnica de Moiré de Sombra como alternativa de baixo custo para análise postural*. Porto Alegre : s.n., 2005.

**Holt, C. M., et al. 1987.** An improved parallel thinning. *Communications of the ACM*. 2. 1987, Vol. v. 30, pp. 156-160.

**Jung, Cláudio Rosito. 2000.** Inf Unisinos.  
[http://www.inf.unisinos.br/~crjung/pesquisa\\_arquivos/image028.jpg](http://www.inf.unisinos.br/~crjung/pesquisa_arquivos/image028.jpg). [Online] 2000.

**Lino, A. C. L. 2002.** Técnica óptica de moiré visando a aplicação no estudo de. *Dissertação*. Campinas : Universidade Estadual de Campinas, 2002.

**Lino, Antônio Carlos Loureiro. 2008.** Aplicação de Moiré de Projeção com deslocamento de fase na geração de modelos digitais topográficos de órgãos vegetais. Campinas : s.n., 2008.

**Llibre, Llar del. 2009.** encarte.org. *Llar del Llibre*. [Online] 2009.  
<http://www.encarte.org/catalog/images/moire.gif>.

**Morton, Michael. 2009.** *Ziggyware*. [Online] 2009. [Citado em: 11 de novembro de 2009.] <http://www.ziggyware.com/ZiggywareImages/Articles/virtualearth/quadtrees.jpg>.

**Oliveira, Bruno Silva de. 2006.** Imagens Tridimensionais por meio da técnica de Moiré: Uma nova proposta. *Monografia*. Lavras : s.n., 2006.

**Oral, Diagnósticos Radiologia. 2009.** Diagnósticos Radiologia Oral.  
[www.diagnosticos.odo.br/img/exames/03/02.jpg](http://www.diagnosticos.odo.br/img/exames/03/02.jpg). [Online] 2009.

**Oster, G. e Nishijima, Y. 1964.** Moiré patterns. *Scientific American Resource Library: Readings in the Physics Sciences and Technology*. 1964, Vol. 3, pp. 291-326.  
**Parker, J. R.** *Extracting Vector from Raster Images*.

**Parks, V.J. e Durelli, A.J. 1970.** *Moiré analysis of strain*. s.l. : Prentice Hall, 1970. p. 399. 978-0135996058.

**Pinho, Márcio Sarroglia. 1999.** *Faculdade de Informática - PUCRS*. [Online] 1999. [www.inf.pucrs.br/~pinho/CG/Aulas/Vetorizacao/vec.htm](http://www.inf.pucrs.br/~pinho/CG/Aulas/Vetorizacao/vec.htm).

**R. Fisher, S. Perkins, A. Walker and E. Wolfart. 2003.** Morphology Thining. <http://homepages.inf.ed.ac.uk/rbf/HIPR2/thin.htm>. [Online] 2003.

**Rayleigh, Lord. 1874.** *On the manufacture and theory of diffraction gratings*. 1874.

**Rubio, Jean-François Doué e Ruben Gonzalez. 1995.** Efficient and Robust 2D Shape Vectorization. [A. do livro] Alan W. Paeth. *Graphics Gems V*. s.l. : Academic Press, 1995.

**Sergento, Cláudia. 2004.** Semana da Informática. <http://www.semanainformatica.xl.pt/680/est/i/p13.jpg>. [Online] 2004.

**Schouten, Theo. 2004.** Radboud University Nijmegen - Computing Science Department. [Online] 2004. <http://www.cs.ru.nl/~ths/rt2/col/h9/9gebiedENG.html>.

**Sciammarella, C.A. 1982.** *The moiré method: a review*. 1982.

**Seara, Daniela Marta. 1998.** Algoritmos para detecção de bordas. *Publicação em WebSite*. Florianópolis : s.n., 1998.

**Stentiford e R.G., Mortimer. 1997.** Some New Heuristics for Thinning Binary Handprinted Characters for OCR. *Algorithms for Image Processing and Computer Vision*. s.l. : JR Parker- John Wiley & Sons, 1997.

**Suen, Zhang e. 1984.** *A fast parallel algorithm for thinning digital patterns*. 3ª Edição. s.l. : Comm. ACM, 1984. pp. 236-239. Vol. vol. 27.

**Takasaki, H. 1970.** *Moiré Topography*. Washington : s.n., 1970. Vol. 9.

**Theocaris, P.S. 1969.** *Moiré fringes in strain analysis*. New York : Pergamon Press, 1969.

**Túlio Gomes, Roberto Júnior, Antônio Lino. 2009.** *Calibration method applied to a moiré technique in experiments of mechanical archetypes*. 2009.

**Turner-Smith, A. 1997.** *Structured light surface measurement techniques*. 1997.

**Wangenheim, Aldo von. 2008.** Seminário Introdução à Visão Computacional . *Visão Computacional*. [Online] 2008. <http://www.inf.ufsc.br/~visao/>.

**Y. Ohtsuka, A. Shinoto, and S. Inoue. 1979.** *Mass school screening for early detection of scoliosis by use of moiré topography camera and low dose X-ray imaging*. 1979.

## 10 APÊNDICE

O software para estudo das características de Moiré é formado por três principais partes, a *Barra de Ferramentas*, a *Tela Principal* na cor azul e a *Tela de Informações* em cinza no canto direito da tela Figura 70.

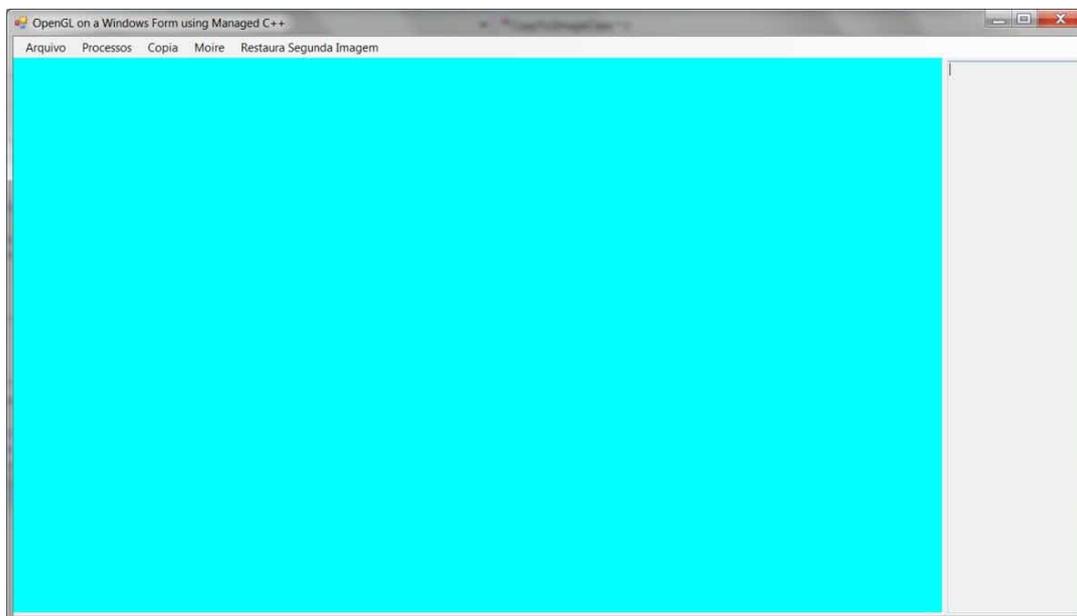


Figura 70 Tela principal do software

Fonte: O Autor (2010)

Na barra de Ferramentas tem o item *Arquivo*. Ao clicar nesse o usuário terá acesso a duas operações – *Abrir Imagem* e *Salvar* Figura 71. Ao selecionar a primeira, o usuário terá acesso a uma janela de *Explorer*, onde poderá procurar pela imagem que deseja usar. A imagem utilizada precisa ser um *bitmap de 24 bits*.

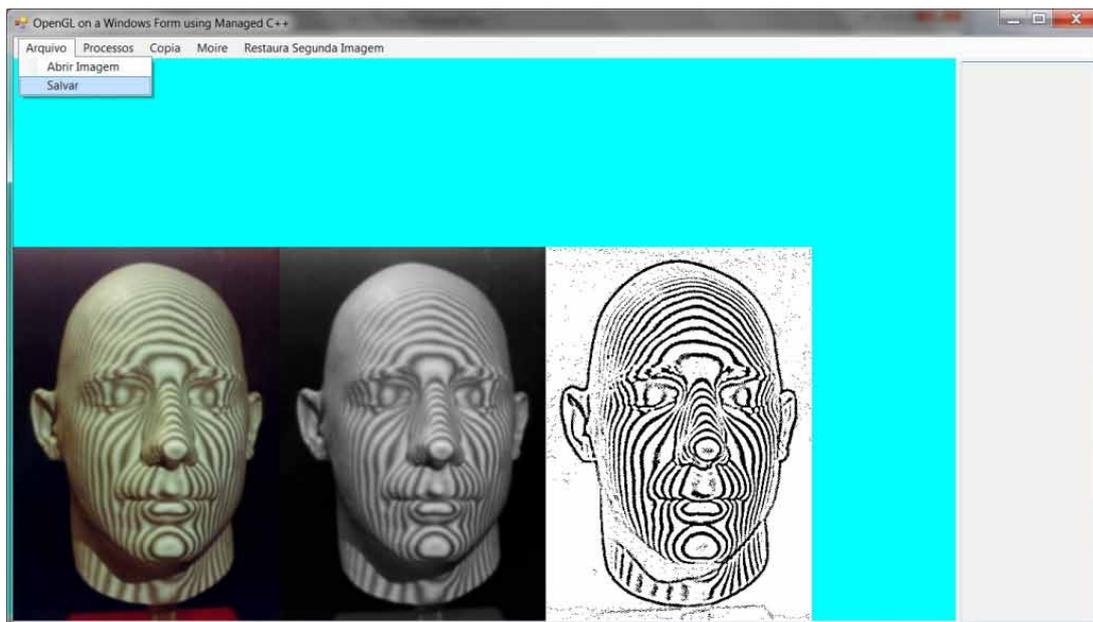


Figura 71 Opção *Arquivo* da barra

Fonte: O Autor (2010)

Ao clicar em *Salvar*, o usuário estará salvando as modificações que operou sobre a imagem original, contidas na segunda e terceira imagem na *Tela Principal*.

Outra opção da *Barra de Ferramentas* é a aba *Processos*, neste tem as operações de pré-processamento da imagem. Neste o usuário pode selecionar o filtro que deseja usar e o resultado aparecerá na terceira imagem da tela, Figura 72.

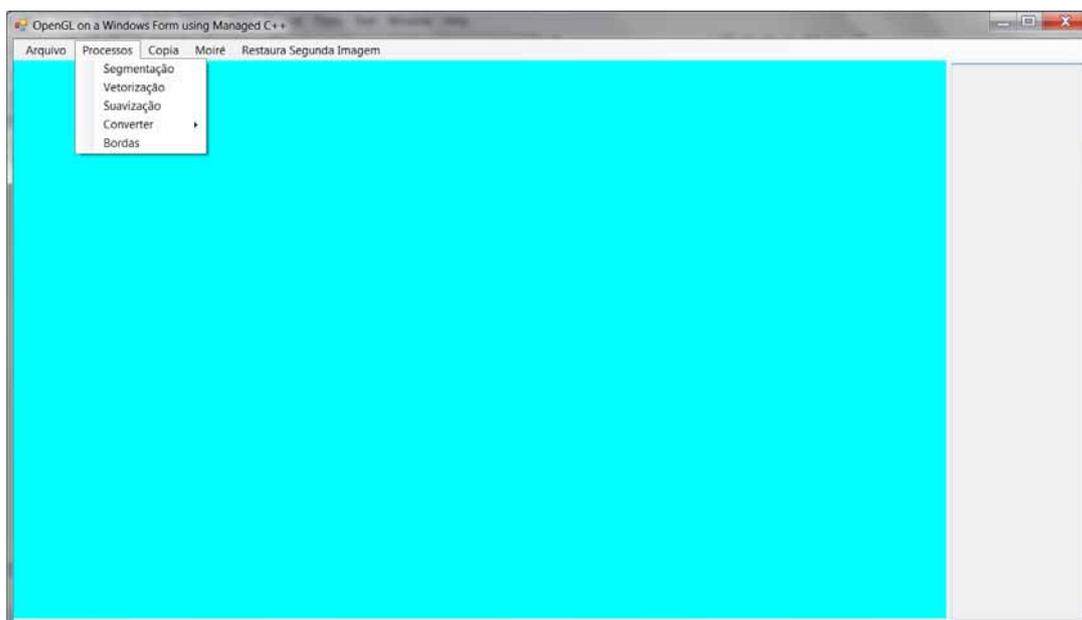


Figura 72 Opção *Processos* da barra

Fonte: O Autor (2010)

A função do item *Cópia* é passar o conteúdo da terceira imagem para a segunda imagem da tela, Figura 73.

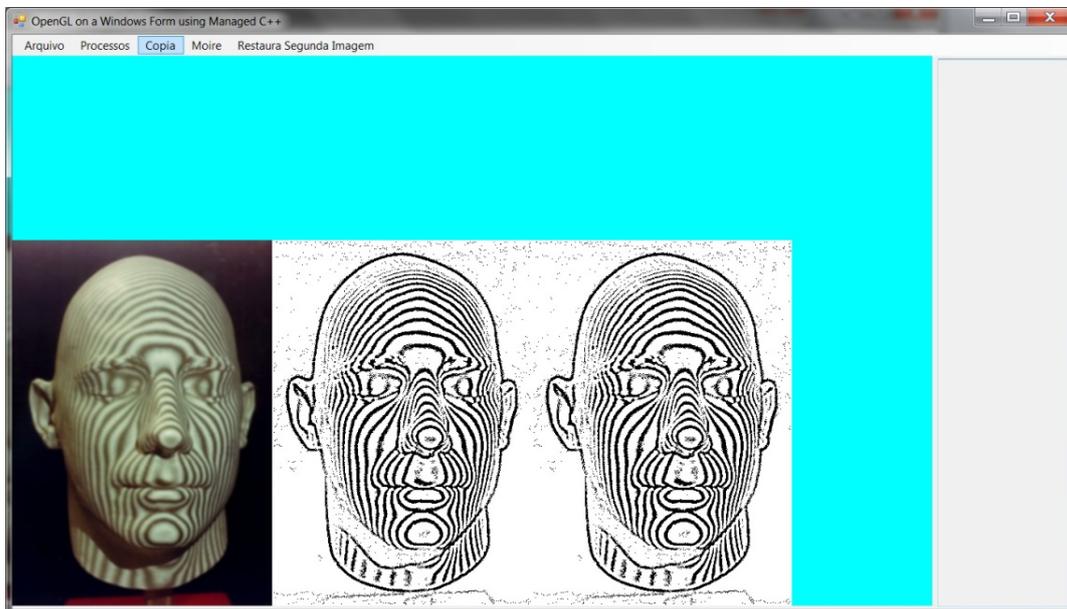


Figura 73 Opção *Cópia* da barra.

Fonte: O Autor (2010)

A próxima opção é *Moiré*. Neste estão contidos os algoritmos responsáveis pelos cálculo das estruturas de Moiré, 74.

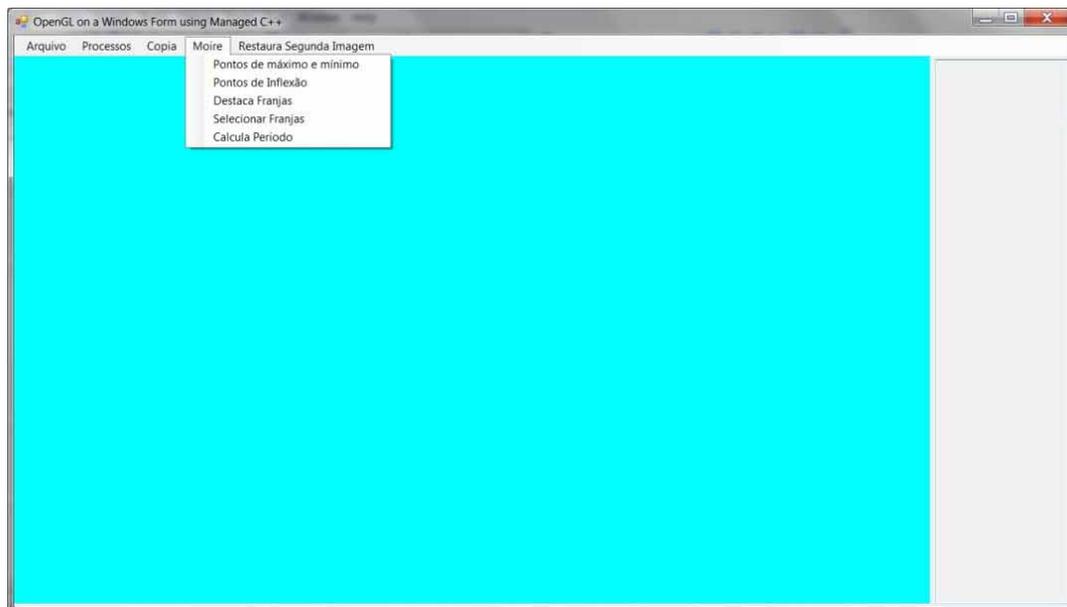


Figura 74 Opção *Moiré* da barra

Fonte: O Autor (2010)