

PONTIFÍCIA UNIVERSIDADE CATÓLICA
DO RIO DE JANEIRO



Diogo Menezes Duarte

**RECONSTRUÇÃO DO ESPAÇO TRIDIMENSIONAL A PARTIR DA
DEFORMAÇÃO DE TEXTURA DE IMAGENS**

Dissertação de Mestrado

Dissertação apresentada ao Programa de Pós-graduação em Engenharia Elétrica da PUC-Rio como requisito parcial para a obtenção do título de Mestre em Ciências da Engenharia Elétrica

Orientador: Raul Queiroz Feitosa
Co-orientador: Jorge Luís Nunes e Silva Brito

Rio de Janeiro
Abril de 2006

Livros Grátis

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.



Diogo Menezes Duarte

Reconstrução do Espaço Tridimensional a partir da Deformação de Textura de Imagens

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo Programa de Pós-Graduação em Engenharia Elétrica do Departamento de Engenharia Elétrica do Centro Técnico Científico da PUC-Rio. Aprovada pela Comissão Examinadora abaixo assinada.

Prof. Raul Queiroz Feitosa

Orientador

Departamento de Engenharia Elétrica - PUC-Rio

Prof. Jorge Luís Nunes e Silva Brito

Co-orientador

UERJ

Prof. Guilherme Lucio Abelha Mota

UERJ

Prof. Sidnei Paciornik

Departamento de Ciências dos Materiais e Metalurgia – PUC-Rio

Prof. José Eugenio Leal

Coordenador Setorial do Centro

Técnico Científico – PUC-Rio

Rio de Janeiro, 06 de Abril de 2006

Todos os direitos reservados. É proibida a reprodução total ou parcial do trabalho sem autorização da universidade, do autor e do orientador.

Diogo Menezes Duarte

Graduou-se em Engenharia Elétrica (Sistemas e Computação) na UERJ (Universidade do Estado do Rio de Janeiro) em 2003.

Ficha Catalográfica

Duarte, Diogo Menezes

Reconstrução do Espaço Tridimensional a partir da Deformação de Textura de Imagens / Diogo Menezes Duarte; orientador: Raul Queiroz Feitosa; co-orientador: Jorge Luís Nunes e Silva Brito – Rio de Janeiro: PUC, Departamento de Engenharia Elétrica, 2006.

121 f. : il. ; 30 cm.

Dissertação (mestrado) – Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Engenharia Elétrica.

Inclui referências bibliográficas

1. Engenharia elétrica – Teses.
2. Processamento digital de imagens. 3. Visão Computacional. 4. Textura. 5. Deformação de Textura. 6. Forma a partir da textura. 7. Estimativa da transformação afim. 8. Reconstrução tridimensional. 9. Reconstrução de cena. 10. Filtros de Gabor. I. Feitosa, Raul Queiroz. II. Brito, Jorge Luís Nunes e Silva. III. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Engenharia elétrica. IV. Título

CDD: 621.3

Agradecimentos

Aos meus pais, Tonho e Lu, pelo carinho, amor e apoio incondicional em tudo.

Às minhas queridas irmãs Dani, Taty e Thaissa, pelo amor e por aturarem meu mau humor nas fases mais “divertidas” deste trabalho.

Aos meus amigos e amigas, irmãos de coração, que agüentaram minha ausência em muitos momentos (talvez eles tenham gostado pois só sabia falar de texturas). Tentar citar nomes aqui seria injusto pois com certeza acabaria esquecendo de alguém, além do que boa parte de minha memória foi tomada por nomes de autores de artigos. Mas saibam que não esqueço do sorriso de nenhum de vocês.

Ao meu orientador Raul, pela preocupação quase que paterna, amizade, força e ajuda em todas as fases deste trabalho assim como ao meu co-orientador Prof. Nunes.

Ao CNPq e a PUC-Rio, pelos auxílios concedidos.

Aos grandes amigos, irmãos do LVC: Rodrigo, Vá (Vanessa), Japa-boy (André), Thiago e Clara. Pelas horas de trabalho, companheirismo, piadas, em longos cafés, em bares, alegrias, tristezas etc etc etc.

Ao Fabian Graefe, da universidade de Braunschweig – Alemanha, pela ajuda no início deste trabalho e atual amizade, além de piadas internacionais.

Ao pessoal da 442 da PUC, Manuel, Isnarde e cia, sempre dispostos a nos ajudar.

Aos autores Ruth Rosenholtz e Boaz Super, pela ajuda em dúvidas sobre seus trabalhos. Em especial também a autora Angeline Loh, pela amigável troca de e-mails com dúvidas e idéias.

Resumo

Duarte, Diogo Menezes; Feitosa, Raul Queiroz. **Reconstrução do Espaço Tridimensional a partir da Deformação de Textura de Imagens**. Rio de Janeiro, 2006, 121p. Dissertação de Mestrado – Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro.

O presente trabalho apresenta um estudo sobre técnicas de construção de um modelo tridimensional de objetos a partir unicamente da informação de textura. Estas técnicas são baseadas na medida da deformação da textura ao longo de uma superfície, obtendo assim a orientação do vetor normal à superfície em cada ponto. De posse da orientação é possível construir um modelo tridimensional do objeto. São avaliados três métodos. O primeiro emprega Filtros de Gabor e momentos de segunda ordem como medida de textura e os outros dois estimam a transformação afim entre recortes de igual tamanho na imagem. A estimativa da transformação afim tem ênfase especial neste trabalho por ser um passo fundamental no algoritmo que mede a deformação da textura. Os métodos foram validados em diferentes etapas, de forma a avaliar: estimativa da transformação afim; decomposição em ângulos; e reconstrução do modelo 3D a partir do mapa de orientação, também conhecido como mapa de agulhas. A avaliação experimental foi realizada com imagens sintéticas e fotos de objetos reais. Os resultados mostram a aplicabilidade, dificuldades e restrições dos métodos analisados.

Palavras-chave

Processamento digital de imagens; Visão Computacional; textura; deformação de textura; forma a partir da textura; estimativa da transformação afim; reconstrução tridimensional; reconstrução de cena; Filtros de Gabor.

Abstract

Duarte, Diogo Menezes; Feitosa, Raul Queiroz. **Scene Reconstruction using Shape from Texture**. Rio de Janeiro, 2006, 121p. Msc. Dissertation – Department of Electrical Engineering, Pontifical Catholic University of Rio de Janeiro.

The current work presents a study about methods for 3D object shape reconstruction based on their texture information. These methods, called “Shape from Texture”, measure texture deformation along object surface, obtaining the orientation in each point of the image. Having the orientation in each point (a needle map) it is possible to construct the object 3D model. Three methods are studied in this dissertation. One of these methods uses Gabor Filters and second order moments, and other two that estimate the affine transform between images patches. The affine estimation problem gets emphasis in the present work since it is an essential step in most Shape from Texture algorithms. The methods were tested in separate steps: evaluate the affine transform estimation; the decomposition of the affine matrix in slant and tilt angles; and the 3D model reconstruction using the needle map. Both synthetic and real images were used on the experiments. The results clearly show the applicability, difficulties and restrictions of the investigated methods.

Keywords

Digital Image Processing; Computer Vision; texture; texture deformation; Shape from X; Shape from Texture; affine transform estimate; 3D reconstruction; scene reconstruction; Gabor filters.

Sumário

1 Introdução	15
1.1 Reconstrução do espaço tridimensional	16
1.2 Objetivo	18
1.3 Organização da dissertação	18
2 Trabalhos Relacionados	19
2.1 Visão Estéreo (<i>Shape from Stereopsis</i>)	19
2.2 Forma a partir da intensidade (<i>Shape from Intensity</i>)	21
2.3 Forma a partir do Movimento (<i>Shape from Motion/Vídeo</i>)	23
2.4 Forma a partir do Foco (<i>Shape from Focus/Defocus</i>)	23
2.5 Forma a partir de sensores de distância (<i>Shape from Range Sensor</i>)	24
2.6 Forma a partir da textura (<i>Shape from Texture</i>)	24
2.6.1 Estimativa da orientação de planos inclinados	25
2.6.2 Estimativa da forma de superfícies curvas	27
3 Conceitos Básicos	29
3.1 Afinal, o que é textura?	29
3.1.1 Homogeneidade e isotropia	30
3.2 Modelo de Projeção Ortográfica	31
3.3 Representação do espaço 3D através da orientação	32
3.4 Modelo de transformação geométrica	33
3.4.1 Decomposição do modelo de transformação por SVD	34
4 Metodologia	36
4.1 Método de Super e Bovik	37
4.1.1 Filtros de Gabor	38
4.1.2 Convolução com os filtros e momentos de segunda ordem	42
4.1.3 Momentos canônicos	45
4.1.4 Projeção da textura	46
4.1.5 Cálculo da orientação	46

4.2. Método de Rosenholtz e Malik	48
4.2.1 Construção dos <i>espectrogramas</i>	49
4.2.2 Análise diferencial	50
4.3. Método de Loh e Zisserman	54
4.3.1 Transformação para textura isotrópica	54
4.3.2 Cálculo da rotação	56
4.3.3 Transformação final	58
4.4 Algoritmo completo de reconstrução de superfícies	58
4.4.1 Extração de recortes	59
4.4.2 Segmentação dos pontos de interesse	61
4.4.3 Seleção de modelo frontal	62
4.4.4 Estimativa da transformação afim	62
4.4.5 Reconstrução do modelo 3D	63
5 Descrição dos experimentos	64
5.1. Decomposição da transformação afim na presença de ruído	66
5.2 Estimativa da transformação afim	71
5.3 Reconstrução da forma de superfícies	79
5.3.1 Método de Super e Bovik	80
5.3.2 Reconstrução da forma da superfície usando a estimativa da transformação afim de Loh	88
5.3.3 Estudo de caso	97
5.4 Avaliação Final dos experimentos	106
6 Conclusão	108
6.1 Trabalhos futuros	109
7 Referências	111
Apêndice I Tabelas extras dos experimentos da estimação da transformação afim	116
Anexo I Derivação das equações de Super e Bovik	120

Lista de figuras

Figura 1 – Exemplo de cena complexa.	15
Figura 2 – Matriz $M \times N \times 3$ que representa uma imagem digital colorida.	16
Figura 3 - Exemplo de objeto texturizado.	17
Figura 4 - O problema da fusão binocular: no diagrama da esquerda temos apenas um ponto e seu homólogo facilmente encontrado. No da direita temos quatro pontos e seus homólogos podem ser qualquer um dos quatro.	20
Figura 5 - Exemplos de pares estereoscópios.	21
Figura 6 - Exemplos de imagens onde o sombreamento evidencia a noção de profundidade.	22
Figura 7 – Exemplos de texturas. Texturas naturais: mar (a) e grama cortada (b). Texturas sintéticas (c) e (d), que possuem distribuição idêntica de intensidade, porém diferem-se no arranjo espacial.	29
Figura 8 – Exemplos de textura. Textura homogênea (a) e textura isotrópica (b).	31
Figura 9 – Ilustração de projeções. (a) Perspectiva. (b) Ortográfica.	32
Figura 10 - Representação da orientação por slant e tilt.	32
Figura 11 – Exemplo de Transformação T_M passo a passo. (a) imagem original. (b) após rotação por θ . (c) compressão de (b) por slant . (d) rotação de (c) por tilt .	33
Figura 12 - Cilindro sintético com textura senoidal aplicada.	37
Figura 13 - Exemplos de Filtros de Gabor no domínio espacial. (a) e (b), parte real e imaginária, respectivamente, de um filtro tendo $F=12$ ciclos/imagem, $\kappa=1$ e $\Phi=0^\circ$. (c) e (d), parte real e imaginária, respectivamente, de um filtro tendo $F=32$ ciclos/imagem, $\kappa=1/2$ e $\Phi=45^\circ$.	41
Figura 14 - Transformadas de Fourier dos Filtros de Gabor. (a) $F=12$ ciclos/imagem, $\kappa=1$ e $\Phi=0^\circ$. (b) $F=32$ ciclos/imagem, $\kappa=1/2$ e $\Phi=45^\circ$.	41

Figura 15 - Diagrama do método de Rosenholtz e Malik.	53
Figura 16 - Transformação de uma textura em sua versão isotrópica.	54
Figura 17 - Transformada de Fourier da versão isotrópica da primeira imagem (a) e da segunda (b).	56
Figura 18 - Transformada de Fourier em coordenadas polares da versão isotrópica da primeira (a) e da segunda (b) imagem.	57
Figura 19 - Sinais obtidos a partir da média das transformadas de Fourier nas coordenadas polares. (a) primeira imagem e (b) segunda imagem.	57
Figura 20 – Exemplo de extração de recortes de uma imagem.	59
Figura 21 – Exemplo de detecção de pontos de interesse. As marcações “+” representam os pontos de interesse detectados pelo método de Lowe.	60
Figura 22 – Grupos de pontos de interesse encontrados pelo método de Lowe e segmentados por <i>K-means</i> . (a) grupo de pontos que representam bem os elementos de textura. (b) grupo que representa mal os elementos de textura (concentrados nas bordas da bola de golfe).	61
Figura 23 – Exemplo de modelo frontal extraído da imagem.	62
Figura 24 - (a) Mapa de agulhas. (b) Modelo 3D reconstruído usando método de Kovesi.	63
Figura 25 - Exemplo de superfície de erro para o <i>tilt</i> . <i>Theta</i> é fixo para todas as combinações.	65
Figura 26 - Superfície de erro do <i>slant</i> para decomposição na presença de ruído. Variância = 0.0001, <i>theta</i> = 0°.	68
Figura 27 - Superfícies de erro do <i>slant</i> para decomposição na presença de ruído. Variância = 0.01, <i>theta</i> = 0°.	69
Figura 28 - Superfícies de erro do <i>tilt</i> para decomposição na presença de ruído. Variância = 0.0001, <i>theta</i> = 0°.	70
Figura 29 - Superfícies de erro do <i>tilt</i> para decomposição na presença de ruído. Variância = 0.01, <i>theta</i> = 0°.	70
Figura 30 - Diagrama da avaliação visual do erro do método.	72
Figura 31 - Texturas de Brodatz utilizadas nos experimentos. (a) D18, (b) D20, (c) D95, (d) D101 e (e) D103.	72

Figura 32 - Diferença absoluta entre transformações afim geradas (T_M) e a matriz identidade.	74
Figura 33 - Diferença entre matriz original T_M e estimada T_R' para a textura D95 usando o método de Rosenholtz.	75
Figura 34 - Diferença entre matriz original T_M e estimada T' para a textura D95 usando o método de Loh.	75
Figura 35 - Superfície de erro para o <i>slant</i> para a textura D95, usando o método de Ruth Rosenholtz. $\Theta = 0^\circ$.	76
Figura 36 - Superfície de erro para o <i>tilt</i> para a textura D95, usando o método de Ruth Rosenholtz. $\Theta = 0^\circ$.	77
Figura 37 - Superfície de erro para o <i>slant</i> para a textura D95, usando o método de Angeline Loh. $\Theta = 30^\circ$.	77
Figura 38 - Superfície de erro para o <i>tilt</i> para a textura D95, usando o método de Angeline Loh. $\Theta = 30^\circ$.	78
Figura 39 - Exemplos de estimativas da transformação afim. (a) Recorte original. (b) Recorte com uma transformação afim A aplicada. (c) Aplicação da transformada inversa \hat{A} (estimada pelo método de Loh) sobre o recorte (b).	79
Figura 40 – (a) imagem sintética de um cilindro texturizado com o ponto frontal encontrado demarcado.(b) Mapa de agulhas obtido pelo método de Super.	81
Figura 41 - (a) Modelo 3D a partir do mapa de agulhas da figura anterior. (b) Modelo 3D com a textura sobreposta.	82
Figura 42 - (a) imagem real de uma bola de golfe com ponto frontal indicado. (b) mapa de agulhas obtido pelo método de Super.	83
Figura 43 - Modelo 3D a partir do mapa de agulhas (estimado pelo método de Super) da bola de golfe.	84
Figura 44 - Modelo 3D da figura anterior com a textura sobreposta.	85
Figura 45 - Imagem sintética de uma superfície irregular com ponto frontal indicado.	85
Figura 46 - Mapa de agulhas obtido pelo método de Super.	86
Figura 47 - (a) Modelo 3D a partir do mapa de agulhas (estimado pelo método de Super) da superfície irregular. (b) Modelo 3D com a textura sobreposta.	87

Figura 48 – Exemplo de grupos de pontos de interesse cujo ponto frontal é distinto para cada. (a) Grupo 1. (b) Grupo 2.	88
Figura 49 - Resultados obtidos para a imagem de um cilindro sintético. (a) Imagem original com pontos frontais selecionados indicados. (b) Mapa de agulhas.	89
Figura 50 – Cilindro sintético com as normais sobrepostas.	90
Figura 51 - Modelo 3D reconstruído a partir do mapa de agulhas.	90
Figura 52 – Modelo 3D com a textura sobreposta.	91
Figura 53 - Imagem de uma bola de golfe com ponto frontal selecionado indicado.	91
Figura 54 - Mapa de agulhas encontrado	92
Figura 55 – Bola de golfe com as normais sobrepostas.	92
Figura 56 - Modelo 3D reconstruído a partir do mapa de agulhas da bola de golfe.	93
Figura 57 – Modelo 3D da figura anterior com a textura sobreposta.	94
Figura 58 - Superfície sintética irregular com pontos frontais selecionados indicados.	94
Figura 59 - Mapa de agulhas da figura anterior	95
Figura 60 – Superfície irregular com as normais sobrepostas.	95
Figura 61 - (a) Modelo 3D reconstruído a partir do mapa de agulhas. (b) Modelo 3D com a textura sobreposta.	96
Figura 62 – Cena real de uma colheita.	97
Figura 63 – Exemplo de modelagem do conteúdo da caçamba.	98
Figura 64 – (a) Primeira imagem da câmara da colheitadeira com ponto frontal detectado pelo método de Super. (b) Primeira imagem com o ponto frontal determinado manualmente para o algoritmo <i>SFT</i> com estimativa afim de Loh.	99
Figura 65 – (a) Modelo tridimensional da primeira imagem da câmara da colheitadeira obtido a partir do mapa de agulhas oriundo do método de Super. (b) O mesmo modelo com a textura sobreposta.	100
Figura 66 - (a) Modelo tridimensional da primeira imagem da câmara da colheitadeira obtido a partir do mapa de agulhas oriundo do algoritmo <i>SFT</i> com estimativa afim de Loh. (b) O mesmo modelo com a textura sobreposta.	101

Figura 67 - (a) Segunda imagem da câmara da colheitadeira com ponto frontal detectado pelo método de Super. (b) Segunda imagem com o ponto frontal determinado manualmente para o algoritmo *SFT* com estimativa afim de Loh. 102

Figura 68 – (a) Modelo tridimensional da segunda imagem da câmara da colheitadeira obtido a partir do mapa de agulhas oriundo do método de Super. (b) O mesmo modelo com a textura sobreposta. 103

Figura 69 – (a) Modelo tridimensional da segunda imagem da câmara da colheitadeira obtido a partir do mapa de agulhas oriundo do algoritmo *SFT* com estimativa afim de Loh. (b) O mesmo modelo com a textura sobreposta. 104

Figura 70 – As duas imagens segmentadas da câmara da colheitadeira. 105

Lista de tabelas

Tabela 1 - Algoritmo <i>SFT</i> de Super e Bovik.	48
Tabela 2 - Erros (em graus) da decomposição da transformação afim por <i>SVD</i> na presença de ruído.	67
Tabela 3 - Erros calculados (em graus) para o método de Ruth Rosenholtz.	73
Tabela 4 - Erros calculados (em graus) para o método de Angeline Loh.	73
Tabela 5 - Erros da decomposição da transformação afim por <i>SVD</i> na presença de ruído. Versão completa da Tabela 2.	116
Tabela 6 – Continuação da Tabela 5.	117
Tabela 7 - Erros calculados (em graus) para o método de Ruth Rosenholtz.	118
Tabela 8 - Erros calculados (em graus) para o método de Angeline Loh.	119

1

Introdução

Dentre as inúmeras habilidades humanas como andar, pensar ou segurar um objeto, a capacidade de enxergar é uma das mais fascinantes e intrigantes. O ato de observar uma cena e identificar objetos e suas formas como carros, copos ou flores é tão comum no dia a dia que, por vezes, aparenta ser uma tarefa simples. Ledo engano; a atividade mental por trás disso é tão engenhosa que pesquisadores de diversas áreas ainda não desvendaram todos os seus mistérios, muito menos conseguem reproduzi-los em sua totalidade.

A Visão Computacional busca reproduzir através de métodos computacionais inúmeras atividades corriqueiras da visão humana, de forma a extrair de imagens informações úteis para processamento. Um exemplo de problema pode ser visto na Figura 1 onde facilmente identificamos um carro, uma casa, e há também a noção de distância relativa entre os elementos da cena.



Figura 1 – Exemplo de cena complexa.

Entretanto, ao transpor esse problema para um método computacional, temos na realidade, uma matriz com diversos valores numéricos, correspondentes às intensidades luminosas que, à primeira vista, estão longe de se parecerem com a figura anterior:

	82	73	63	46	77	81	80	57	27	38	45	58	56	62	57	
45	44	49	51	42	28	26	17	4	55	75	109	80	65	126	55	
42	43	44	44	44	42	42	41	44	42	39	37	37	38	40	224	88
48	47	46	46	48	45	45	42	43	44	42	40	44	48	48	204	43
45	44	43	43	46	44	45	42	40	43	42	43	46	52	49	62	27
42	42	42	42	41	38	41	44	46	41	41	44	44	31	44	41	29
43	41	41	42	42	42	43	46	44	45	46	43	46	43	56	70	36
43	41	39	41	43	43	44	46	39	34	40	52	55	41	45	120	32
43	41	40	41	42	42	43	45	43	47	47	39	37	46	51	134	35
43	43	44	44	43	41	43	45	44	49	51	42	28	26	17	136	38
42	43	45	46	44	42	43	46	42	37	55	86	107	134	165	138	34
42	42	44	45	43	41	40	42	49	87	143	181	198	223	220	148	23
43	42	41	43	42	39	36	35	52	68	85	76	54	59	48	145	08
41	43	43	41	42	44	45	43	40	51	50	59	54	52	45	156	07
41	42	42	40	41	43	43	42	49	58	56	85	80	73	63	150	25
41	42	42	41	41	42	42	41	46	53	56	124	131	131	127	150	23
42	42	42	41	41	41	42	41	51	51	47	135	137	136	135	153	31
44	43	43	43	43	42	42	43	47	43	37	136	138	137	141	152	34
43	42	41	42	42	41	41	42	45	40	36	131	135	132	134	153	35
43	42	41	43	43	41	42	43	42	39	40	130	140	139	139	155	35
45	43	43	45	45	43	43	45	45	40	43	127	138	139	138	154	28
41	43	45	44	42	41	42	44	44	45	43	135	135	135	142	154	27
43	43	43	43	42	42	43	43	45	45	44	135	135	135	141	154	26
44	43	42	41	41	42	42	43	45	44	47	134	135	135	140	154	23
43	42	41	39	38	39	41	43	44	43	50	135	135	135	140	154	
40	41	41	38	35	35	39	43	43	43	55	134	136	135	140		

Figura 2 – Matriz MxNx3 que representa uma imagem digital colorida.

Os valores apresentados na Figura 2 são na verdade o que método computacional tem acesso. Para um sistema computacional encontrar nessas matrizes os mesmos objetos que a visão humana identifica numa cena é um processo difícil.

1.1

Reconstrução do espaço tridimensional

A capacidade da visão humana de perceber formas de objetos, assim como distâncias entre eles, está baseada em diferentes informações presentes na imagem. Por exemplo a estereoscopia, onde temos diferentes imagens de uma mesma cena obtidas por cada um de nossos olhos. Ou ainda a variação de iluminação no objeto também permite perceber sua forma.

Na literatura da Visão Computacional existem técnicas que se baseiam nessas peculiaridades da visão humana para reconstruir o espaço tridimensional de um objeto ou de uma cena completa. Ao conjunto dessas técnicas dá-se o nome *Shape from X*.

Essa reconstrução torna-se útil em campos como navegação de robôs e arquitetura, dentre outros. No primeiro caso seu emprego ajuda a calcular distâncias até objetos, por exemplo. No segundo caso se encontra a área de fotogrametria arquitetônica, usada, por exemplo, para gerar modelos precisos de antigas construções e monumentos, de forma que, caso aconteça alguma avaria, tenha-se uma referência para a restauração.

O presente trabalho foca em uma técnica específica do conjunto *Shape from X*, baseada somente na informação de textura de um objeto para perceber sua forma e/ou distâncias relativas, chamada *Shape from Texture (SFT)*. Um exemplo pode ser visto na Figura 3, onde a única informação presente é a textura; não há outra como sombreamento, movimento ou estereoscopia, mesmo assim temos a clara sensação que se trata de um cilindro.

De fato, a visão humana é capaz de perceber a forma do objeto baseando-se apenas na deformação presente na textura ao longo da superfície. Iremos explorar neste trabalho algumas técnicas baseadas neste artifício da visão e observar seu comportamento.

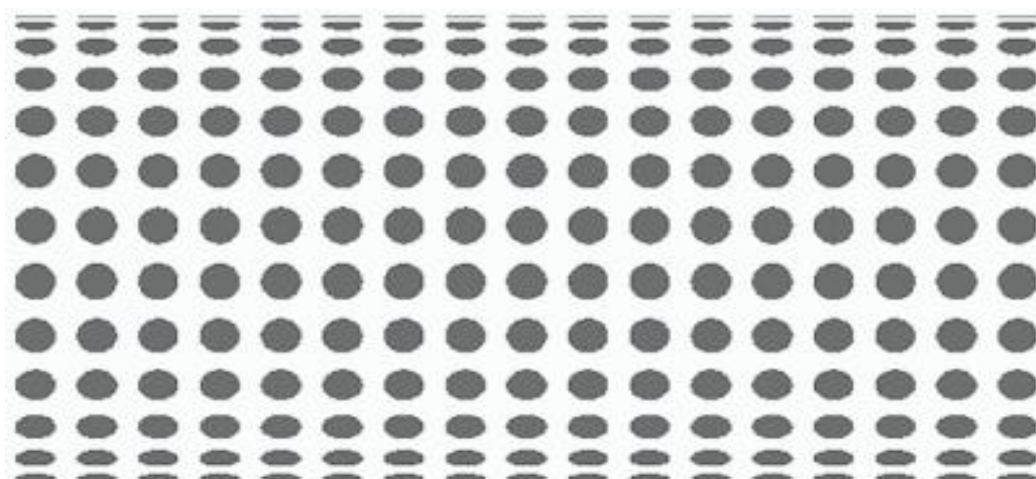


Figura 3 - Exemplo de objeto texturizado.

1.2 Objetivo

A área de *SFT* historicamente divide-se em duas: a que estima a orientação de planos texturizados e a que estima a forma de superfícies curvas texturizadas. O enfoque neste trabalho é o último, tendo como objetivos:

- Investigar as técnicas de reconstrução de superfícies curvas baseadas somente na informação de textura.
- Estimar o desempenho e estabilidade de algumas técnicas selecionadas.
- Avaliar sua aplicabilidade a problemas práticos de modelagem 3D.
- Apontar pontos de possível melhora nas técnicas.

1.3 Organização da dissertação

O restante da dissertação está organizado, conforme descrito nos parágrafos subsequentes.

O capítulo 2 apresenta um breve resumo sobre as técnicas de reconstrução *Shape from X*, ao final dando ênfase maior à *Shape from Texture*, apresentando um histórico das técnicas e o estado da arte.

O capítulo 3 define alguns conceitos básicos amplamente repetidos nos artigos sobre *SFT*, que convencionou-se centralizar em um único capítulo, para facilitar a leitura desta dissertação. Nele são apresentadas definições de modelos de projeção e de orientação, os ângulos *tilt* e *slant*, termos repetidos com frequência na área de *SFT*.

O capítulo 4 descreve detalhadamente os algoritmos das técnicas empregadas nos experimentos e o capítulo 5 mostra os resultados numéricos destes.

O capítulo 6 apresenta então as conclusões da pesquisa. Este capítulo é seguido das referências bibliográficas.

2 Trabalhos Relacionados

A visão humana e sua capacidade fantástica de extrair informação do espaço tridimensional à sua volta é uma área de pesquisa que envolve campos como Biologia, Física, Psicologia, Robótica e Computação.

Dentre as possíveis informações que a visão humana se utiliza para estimar a forma de um objeto tridimensional a partir de imagens bidimensionais temos: textura, movimento, sombreamento, estereoscopia, foco, sobreposição etc. No campo da Visão Computacional foram criados métodos computacionais baseados nestas informações e ao conjunto deles damos o nome de *Shape-from-X*.

Neste capítulo são apresentadas breves notas sobre algumas técnicas de *Shape from X*, que podem ser encontradas de maneira mais abrangente na tese de Jong [1]. Ao final é dada ênfase na área de *Shape from Texture*, tema desta pesquisa, mostrando assim um histórico das principais pesquisas desenvolvidas sobre o tema.

2.1 Visão Estéreo (*Shape from Stereopsis*)

A estereoscopia é baseada na diferença relativa entre posições de objetos em duas ou mais imagens, a qual chamamos disparidade. Diz respeito à percepção de formas 3D e distâncias que resultam de diferentes imagens de uma mesma cena produzidas por pontos de vista distintos. A fusão das imagens capturadas por nossos olhos permite uma forte percepção da profundidade da cena.

A visão estereo envolve dois processos: a fusão das características observadas pelos olhos e a reconstrução de seu modelo tridimensional. O primeiro consiste em encontrar características ou pontos de interesse que sejam facilmente localizáveis em ambas as imagens. Em uma imagem temos milhares de *pixels*; porém, nem todos podem ser usados como pontos de interesse, uma vez que seu par correspondente na segunda imagem (par homólogo) não seria adequadamente

localizado. Este é chamado de “problema da fusão binocular”, ilustrado na Figura 4. Exemplos de bons pontos de interesse incluem bordas e cantos.

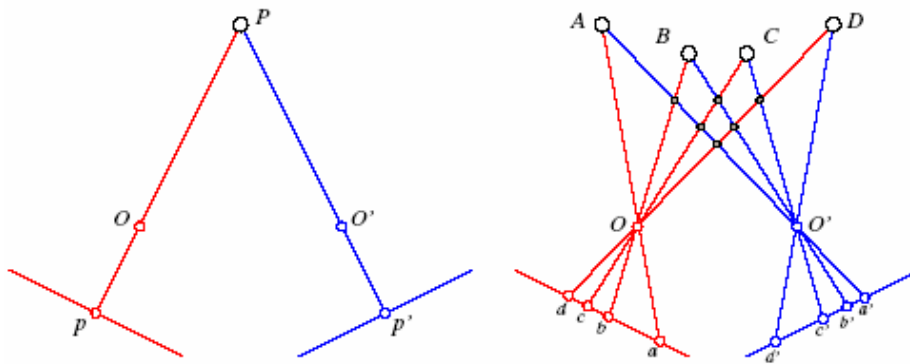


Figura 4 - O problema da fusão binocular: no diagrama da esquerda temos apenas um ponto e seu homólogo facilmente encontrado. No da direita temos quatro pontos e seus homólogos podem ser qualquer um dos quatro.

A segunda fase de um sistema estéreo envolve a reconstrução do modelo tridimensional dados os pontos de interesse e seus homólogos na outra imagem. Na Figura 5 temos exemplos de pares estereoscópios de objetos. Note que até aqui foi considerado um sistema estereoscópio baseado na visão humana, ou seja, com apenas duas câmeras, porém podem ser empregadas mais de duas, como um sistema trinocular [2].



Figura 5 - Exemplos de pares estereoscópios.

2.2

Forma a partir da intensidade (*Shape from Intensity*)

A visão humana consegue perceber diferenças de profundidade a partir da intensidade, em função da iluminação que um objeto recebe. Exemplos de imagens bidimensionais podem ser vistos na Figura 6, onde temos uma clara percepção de forma apesar da única informação presente ser a de iluminação na superfície. Este recurso de sombreado já é empregado há muito tempo por pintores e artistas da área de computação gráfica, conseguindo passar com maestria a sensação de profundidade.

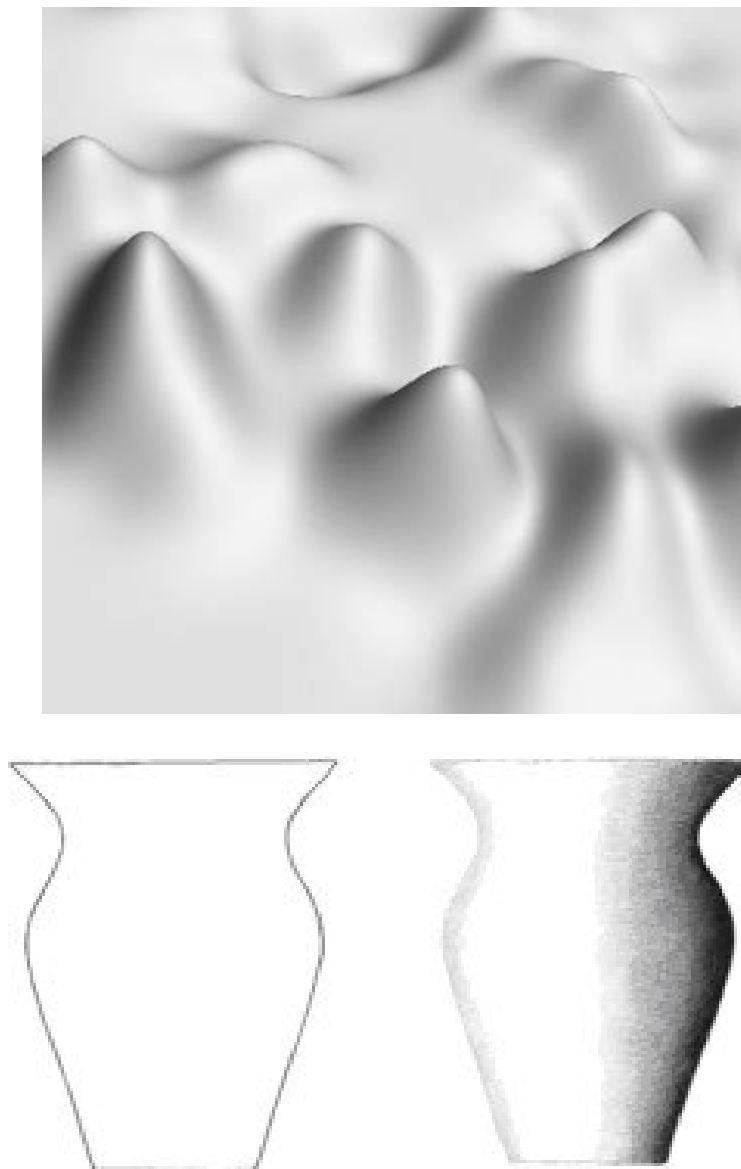


Figura 6 - Exemplos de imagens onde o sombreado evidencia a noção de profundidade.

Partindo desta característica da visão humana surgiram na Visão Computacional técnicas que exploram este efeito. Esta abordagem é dividida em dois grupos: Reconstrução a partir do sombreado, *Shape from Shading (SFS)* e Reconstrução a partir de Estereoscopia Fotométrica, *Shape from Photometric Stereo (SFPS)*.

O primeiro, *SFS*, teve seu trabalho inicial com Horn [3] em 1970. Consegue medir profundidade através do sombreado partindo de uma única imagem. Normalmente admite uma superfície Lambertiana (reflete a luz incidente com a mesma luminância em todas as direções) e que existe uma única fonte de luz

incidente, cuja direção é conhecida. O problema desta abordagem é que, na prática, poucas são as superfícies Lambertianas e as imagens teriam de ser adquiridas em ambiente controlado, pois senão mais de uma fonte de iluminação teria de ser considerada, com parâmetros distintos, além da reflexão das superfícies. Com isso, *SFS* não é um método muito acurado para toda e qualquer situação real.

O segundo, *SPFS*, utiliza mais de uma imagem, ainda em ambiente controlado, mas sendo que agora cada imagem é adquirida usando uma fonte de iluminação com direção diferente, mantendo a posição da câmera e do objeto constantes. Algumas técnicas são descritas em [4, 5].

2.3

Forma a partir do Movimento (*Shape from Motion/Vídeo*)

A reconstrução a partir do movimento [6] é similar a estereoscopia, pois parte de mais de uma imagem de um objeto tomada de diferentes pontos de vista. É possível estimar a forma de um objeto móvel partindo de uma sequência de imagens de uma câmera fixa, assim como o contrário também é verdadeiro, partindo de um objeto fixo e uma câmera móvel.

2.4

Forma a partir do Foco (*Shape from Focus/Defocus*)

A análise do Foco apresenta uma poderosa ferramenta para recuperar a forma tridimensional de superfícies. Esta técnica possui emprego em superfícies ásperas microscópicas, onde outras técnicas de Visão Computacional não possuem bons resultados. Um exemplo pode ser visto no trabalho de Nayar e Nakagawa [7].

2.5

Forma a partir de sensores de distância (*Shape from Range Sensor*)

As outras técnicas em *Shape from X* são ditas passivas, porque utilizam sensores como câmeras para captura de imagem e então aplicam algoritmos de processamento de imagem. Porém, existe uma área baseada em sensores ativos [8, 9], que emprega sensores lasers ou ultra-sônicos. Por exemplo, usando esses sensores como radares *time-of-flight* (tempo de voo), pode-se medir com precisão a distância entre o sensor e o ponto no objeto, mensurando a diferença entre um pulso eletromagnético recebido e um transmitido.

2.6

Forma a partir da textura (*Shape from Texture*)

A idéia de que a visão humana usava como uma de suas medidas de profundidade a deformação de textura foi apontada inicialmente por Gibson [10], em 1950. Foi ele que cunhou o termo gradiente de textura, que descreve o fenômeno pelo qual pequenos recortes (*patches*) de uma superfície que possui um padrão repetido, ou seja, uma textura, projetam no plano da imagem os mesmos recortes, porém com aparência diferente devido à orientação, distância e forma da superfície. Gibson sugere que se pudermos medir este gradiente, essa diferença entre os recortes, podemos encontrar a orientação local da superfície na cena. Foi a partir desse trabalho que começaram as pesquisas sobre *Shape from Texture* (*SFT*) em Visão Computacional.

Os métodos de *SFT* se dividem inicialmente em dois grupos de abordagens: medir a orientação de uma superfície plana inclinada; e, medir a orientação local de pequenos recortes de uma superfície curva, podendo assim integrar essa informação e obter a forma da superfície. Os métodos são apresentados nas subseções a seguir, divididos pelas duas abordagens.

2.6.1

Estimativa da orientação de planos inclinados

O primeiro método computacional na área foi o de Bajscy e Lieberman [11] em 1976, apesar deste não calcular a orientação de um plano. Na verdade, eles desenvolveram uma forma de medir distâncias relativas entre objetos em cenas reais, estes sobre um plano texturizado, a partir da diferença de tamanho entre elementos de textura (*texels*) ao longo do plano. Para medir essa diferença é empregada análise do espectro de frequência, usando a transformada de Fourier em janelas [12]. O método é limitado a planos inclinados naturais como pisos, solos e mar.

Para medir o gradiente de textura nos *texels* utilizam-se informações como mudança na densidade, área, escala, compressão (*foreshortening*), e ainda uma abordagem estatística usando desvio da isotropia - uma característica física, que significa que um objeto possui a mesmas propriedades em todas as direções -, quando uma textura isotrópica é projetada perde esta propriedade. Na literatura o primeiro trabalho que apontou e classificou essas informações distintas foi o de Stevens [13], em 1981.

Muitos trabalhos, no início, basearam-se em extração de *texels*, usando detecção de bordas, por exemplo. Exemplos incluem os trabalhos de Witkin [14] em 1981, que utiliza um algoritmo de detecção de bordas para encontrar os *texels* e busca medir dois efeitos: que eles aparecem menores de acordo com a distância; e que são comprimidos na direção da inclinação da superfície. Usando um modelo de projeção ortográfica, que equivale a uma transformação afim na imagem, são aplicadas uma série de possíveis transformações ao plano. Com os possíveis planos reconstruídos, a solução se resume em determinar qual destes é o mais similar à imagem original. O método é estendido para superfícies curvas, porém não apresenta bons resultados.

No trabalho de Aloimonos [15] a orientação do plano é calculada assumindo que a densidade dos *texels* é uniforme. Aloimonos emprega um modelo de projeção distinto, o paraperspectivo, que é mais simples que o perspectivo porém mais acurado e complexo do que o ortográfico. Apresenta bons resultados, mas apenas voltado para planos inclinados.

Blostein e Ahuja [16], em 1989, propõem um método voltado apenas para superfícies inclinadas, contendo uma única textura natural. O método extrai os *texels* usando um algoritmo de detecção de região multiescala. Uma relação entre o tamanho da textura e um plano texturizado hipotético é derivada do modelo de projeção, o problema recai em uma procura pela orientação que melhor se ajusta aos *texels* encontrados.

Dentre outros exemplos temos o de Kender [17], Ohta et al. [18], Ikeuchi [19], Kanatani e Chou [20]. A grande desvantagem destes métodos é a abordagem por extração de *texels*, uma vez que esse é um passo difícil, que requer normalmente intervenção do usuário, além do que, uma textura pode conter sub-texturas, dependendo da escala em que se trabalha. Esses métodos também acabam por não utilizar toda a informação contida na imagem, somente as existentes nos *texels*.

Uma outra abordagem na literatura de *SFT* é baseada em um modelo probabilístico de textura. O problema de determinar a forma e orientação de uma superfície recai então em um problema de inferência estatística (estimativa dos parâmetros do modelo dadas amostras na imagem). Normalmente nesses casos, admite-se que a textura é isotrópica, isto é, possui a mesma aparência independente de rotação e quando é projetada perde esta propriedade. Então pode-se usar o desvio da isotropia como meio de inferir a orientação. Exemplos dessa abordagem são os trabalhos de Witkin [14], Davis et al. [21], Blake e Marinos [22], Garding [23] e Brown e Shvaytser [24], sendo que os últimos usam análise de Fourier e a função de autocorrelação para encontrar a orientação de planos.

O problema da hipótese de isotropia é a dificuldade de encontrar texturas com essas características no mundo real, pois normalmente as texturas são direcionais.

Krumm e Shafer [25] estimam a transformação afim presente na projeção no domínio da frequência, porém usando o conceito de *espectrograma* (melhor descrito no capítulo 4 desta dissertação). Esses autores procuraram por todas as combinações de orientação em um conjunto discreto para estimar a correta. Entretanto seu método é empregado somente para planos.

2.6.2

Estimativa da forma de superfícies curvas

Garding [26], em 1992, desenvolveu toda uma base matemática para *SFT* em superfícies curvas. O trabalho de Garding foi de extrema importância para a área, pois, além de desenvolver uma *metodologia* para análise dos gradientes baseado em geometria diferencial [27], também mostrou que estas simples informações de distorção não são suficientes para uma medida completa da curvatura da superfície (ex.: sinal da curvatura gaussiana).

Garding [28] e Super e Bovik [29] usam o domínio da frequência e momentos de segunda ordem para estimar a forma de superfícies curvas, sendo que o primeiro trabalho é focado na estrutura matemática da estimativa. Super e Bovik empregaram filtros de Gabor para a análise de textura e projeção ortográfica, que demonstraram resultados encorajadores. Também apontam uma maneira de encontrar um ponto visto frontalmente na imagem, possuindo assim um referencial para seu algoritmo, que estima a deformação comparando pontos na imagem.

Rosenholtz e Malik [30] desenvolveram um método de *SFT* para superfícies curvas usando a base matemática desenvolvida por Garding. O método utiliza transformada de Fourier e análise diferencial para estimar a transformação afim entre dois recortes pequenos da superfície curva. Com a análise diferencial e algumas aproximações usando álgebra linear e mínimos-quadrados, consegue chegar a uma solução para a orientação, sem a necessidade de determinar um ponto visto frontalmente, como no trabalho de Super e Bovik. Tanto o trabalho de Super e Bovik quanto parte do de Malik e Rosenholtz são empregados nesta dissertação e melhor descritos no Capítulo 4.

Dando continuidade ao trabalho de Malik e Rosenholtz, Ribeiro et al. [31] apresenta uma solução fechada para o método de Rosenholtz, sem empregar otimização numérica iterativa, através dos autovetores das matrizes de transformação afim.

Clerc e Mallat [32] apresentaram um método que usa *wavelets* como medida de textura espectral e modelam o gradiente de textura como uma deformação de um processo estocástico estacionário [33].

Loh e Kovesi [34] apresentaram um método para superfícies curvas que utiliza o conceito de inércia local [35] sobre os momentos de segunda ordem da transformada de Fourier dos recortes de textura na imagem. Os resultados desse método são promissores porém ainda limitados, uma vez que apenas calcula um dos ângulos da orientação (*tilt*) e precisa que um ponto frontal na imagem seja indicado manualmente.

Forsyth [36] apresenta uma técnica bem mais elaborada, que extrai recortes pequenos na superfície curva e calcula a orientação relativa em cada ponto sem a necessidade de determinar um ponto frontal na imagem, visto que este não é tido como existente na imagem e sim, estimado a partir dos recortes extraídos. Forsyth vai além e sugere uma outra aplicação prática de *SFT*, em conjunto com a computação gráfica, na renderização de tecidos, que pode ser empregado em lojas virtuais de roupas.

Loh e Zisserman [37] apresentam um trabalho semelhante ao de Forsyth. Assim como ele, não encontram um ponto frontal na imagem e sim o estimam. Entretanto, empregam como cálculo da transformação afim entre o ponto frontal e os recortes na imagem um método próprio, apresentado em outro trabalho [38] (usado nesta dissertação e descrito em detalhes no Capítulo 4).

Até aqui foi apresentado um resumo sobre os artigos pesquisados neste trabalho, com ênfase em *SFT* voltado para superfícies curvas. O capítulo seguinte apresenta alguns conceitos amplamente usados em *SFT*. O capítulo 4 descreve em detalhes três artigos selecionados deste resumo, que foram analisados nos experimentos, apresentados no capítulo 5.

3

Conceitos Básicos

Este capítulo tem por função apresentar alguns conceitos que serão repetidos com frequência ao longo deste trabalho. Idéias como textura homogênea e isotrópica, projeções ortográficas, transformações afim etc se encontram resumidas nas seções abaixo, com o intuito de facilitar a leitura.

3.1

Afinal, o que é textura?

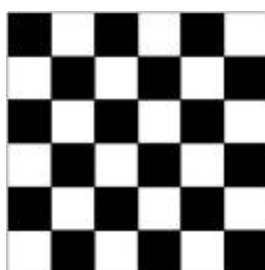
O termo textura é facilmente associado quando lido, ouvido ou visto, entretanto uma definição é difícil. A visão humana reconhece texturas de maneira tão corriqueira que usualmente passa despercebido.



(a)



(b)



(c)



(d)

Figura 7 – Exemplos de texturas. Texturas naturais: mar (a) e grama cortada (b). Texturas sintéticas (c) e (d), que possuem distribuição idêntica de intensidade, porém diferem-se no arranjo espacial.

Em uma imagem a textura nos dá a informação sobre o arranjo espacial das cores e intensidades. Um exemplo pode ser visto na Figura 7, em (c) e (d), onde as imagens possuem a mesma distribuição de intensidade (50% branco – 50% preto), porém suas texturas são diferentes, têm uma disposição espacial na imagem distinta. A textura é uma informação adicional na segmentação, muito útil em alguns casos.

Na área de processamento de imagens texturas podem ser definidas como padrões de repetição de determinados elementos formadores, os chamados *texels*. Essa é a definição necessária para *SFT*, como veremos mais a frente, uma vez que esperamos que padrões apareçam de maneira homogênea na imagem.

Em *SFT* são empregados normalmente a detecção de *texels* ou uma análise de frequência (Fourier) como medidas de textura. No presente trabalho os métodos apresentados baseiam-se em Fourier e Filtros de Gabor, que serão melhor descritos no capítulo seguinte. Descrições mais abrangentes de outras medidas de texturas podem ser encontradas na literatura como em [39, 40, 41].

3.1.1 Homogeneidade e isotropia

Na área de *SFT* as texturas são divididas em dois tipos: homogêneas e isotrópicas.

Uma textura **homogênea** é aquela onde recortes locais pequenos da textura parecem iguais, independente de onde tenham sido retirados. Ou seja, há um padrão uniforme de repetição ao longo da superfície. Em um algoritmo *SFT*, ao tomarmos uma textura como homogênea assumimos que algumas estatísticas como área e densidade são constantes na superfície; desta forma, estas irão variar devido a distorções causadas pela mudança de orientação.

Uma textura **isotrópica** é aquela onde a probabilidade de se encontrar um elemento formador da textura (*texel*) independe de sua orientação. Não há orientação tendenciosa. Ou seja, ao rotacionar a textura ainda assim teremos a impressão de que esta não sofreu nenhuma alteração. O termo isotrópico, por definição, significa que um objeto apresenta as mesmas propriedades físicas em todas as direções.

Na Figura 8 há um exemplo de textura homogênea e isotrópica.



Figura 8 – Exemplos de textura. Textura homogênea (a) e textura isotrópica (b).

3.2

Modelo de Projeção Ortográfica

Ao transportamos um objeto tridimensional no mundo real para uma imagem, um plano bidimensional, usamos um modelo de projeção dos pontos. Em um caso real, uma câmera comum, o modelo é perspectivo (Figura 9 (a)), ou seja, os pontos na imagem convergem para o ponto focal da lente e então divergem para o plano da imagem.

Nos métodos apresentados neste trabalho foi empregado um modelo de projeção ortográfico, cuja modelagem matemática (uma transformação afim) é mais simples que o perspectivo. Ainda assim o ortográfico é uma boa aproximação para imagens de objetos cuja superfície está a uma curta distância da câmera.

Em um problema de *SFT* usando projeção ortográfica, o ângulo *tilt* (descrito na seção seguinte) possui uma ambigüidade de 180° (π). Isso significa que neste modelo, utilizando somente informação de textura, não há como saber se uma superfície é côncava ou convexa.

Na projeção ortográfica os pontos são projetados paralelamente ao plano imagem, conforme ilustrado na Figura 9 (b).

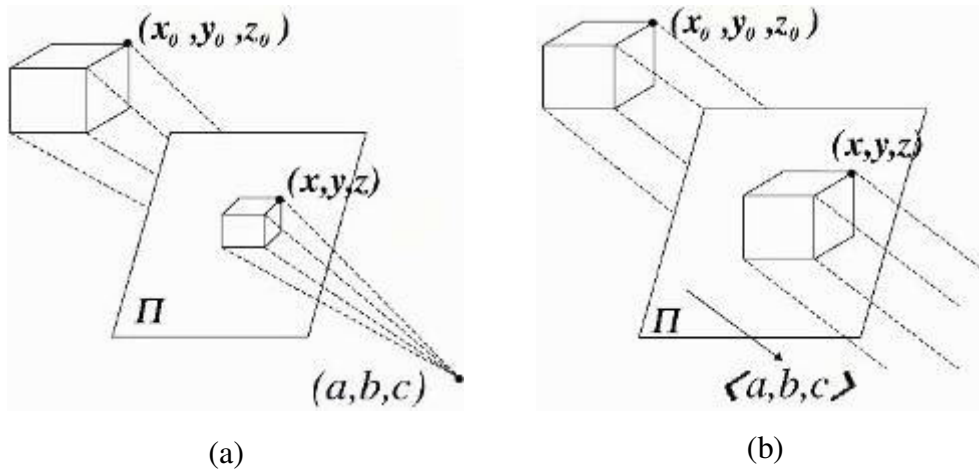


Figura 9 – Ilustração de projeções. (a) Perspectiva. (b) Ortográfica.

3.3

Representação do espaço 3D através da orientação

A representação do espaço tridimensional na área de *SFT* se dá através da orientação, sendo esta representada pelos ângulos *slant* e *tilt*.

Dado um sistema de coordenadas (x, y, z) tal que o plano x - y coincide com o plano da imagem e o eixo óptico coincide com a coordenada z . O *slant*, σ , é o ângulo entre a normal da superfície e a coordenada z . O *tilt*, τ , é o ângulo entre a normal projetada no plano x - y e a coordenada x , como pode ser visto na Figura 10. Esta representação foi proposta por Stevens [42].

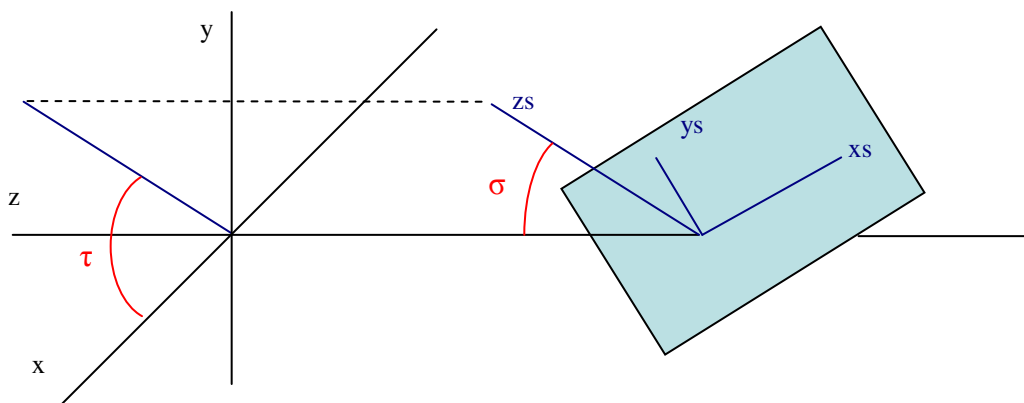


Figura 10 - Representação da orientação por slant e tilt.

3.4 Modelo de transformação geométrica

Conforme visto na seção anterior a orientação de um ponto na imagem é representada por dois ângulos, *slant* e *tilt*. O *slant* representa uma compressão na direção do *tilt*.

Este modelo de orientação, um uma projeção ortográfica, pode ser representado através de uma transformação afim T_M , segundo [36]:

$$T_M = \begin{bmatrix} \cos \tau & -\sin \tau \\ \sin \tau & \cos \tau \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & \cos \sigma \end{bmatrix} \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}, \quad (1)$$

onde τ é o *tilt*, σ é o *slant* e θ é o ângulo *theta*. Esta transformação representa a seguinte operação em uma imagem: primeiro é rotacionada de θ , depois comprimida em um dos eixos por um fator equivalente ao co-seno do *slant* (σ) e finalmente rotacionada pelo ângulo *tilt* (τ). Na Figura 11 há um exemplo passo-a-passo da transformação afim T_M .

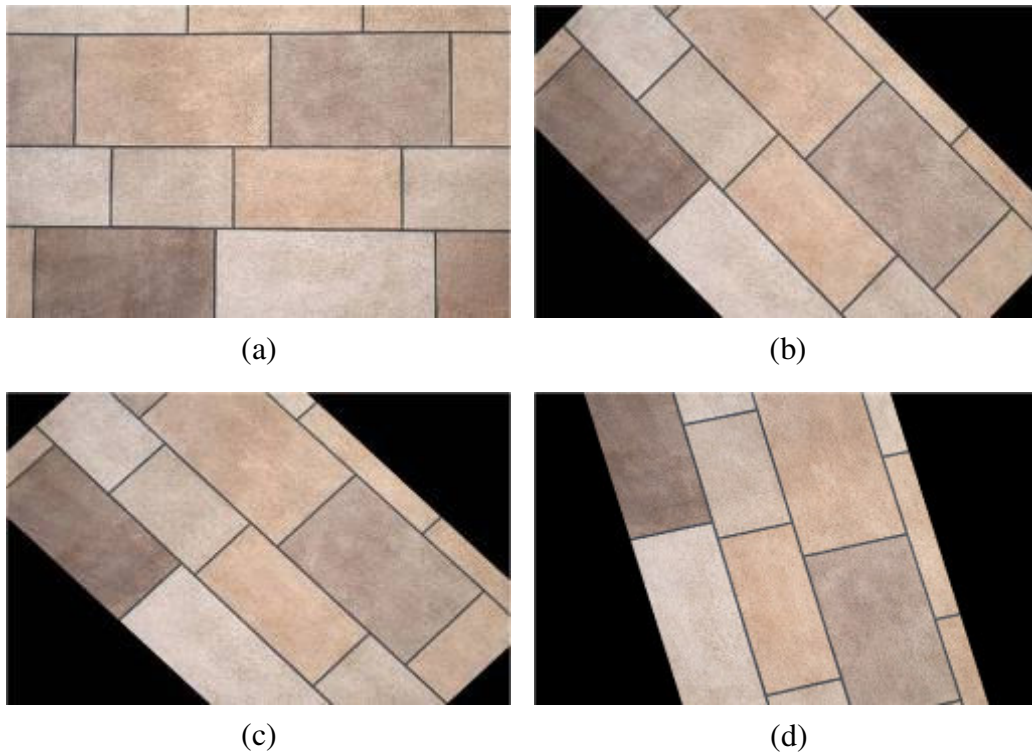


Figura 11 – Exemplo de Transformação T_M passo a passo. (a) imagem original. (b) após rotação por *theta*. (c) compressão de (b) por *slant*. (d) rotação de (c) por *tilt*.

Esse modelo de transformação de textura é o adotado em todos os experimentos. Representa a deformação esperada da textura em um local da imagem, causada pela projeção.

Uma convenção feita ao longo do texto é que qualquer transformação afim é representada por T . Entretanto, uma transformação que siga a forma da eq. (1) é representada por T_M . As transformações afins estimadas pelos métodos apresentados no capítulo a seguir são representadas por T_M' . Onde o apóstrofo significa que é uma estimativa e não a transformação original.

3.4.1

Decomposição do modelo de transformação por SVD

O modelo visto na eq. (1) é o esperado numa transformação afim, estimada pelos métodos descritos no capítulo adiante. Entretanto, para obter a orientação nos pontos da imagem é necessário decompor a equação. Para tal é empregado o método da Álgebra Linear conhecido como *Singular Value Decomposition* (SVD) [43], decomposição por valores singulares.

A decomposição de uma matriz P por SVD dá-se como:

$$P = R\Sigma Z^t, \quad (2)$$

onde R e Z são matrizes ortogonais e Σ :

$$\Sigma = \begin{bmatrix} \varsigma_{11} & 0 \\ 0 & \varsigma_{22} \end{bmatrix}$$

Admitimos então na decomposição que R corresponde à matriz de rotação definida pelo *tilt* e Z a matriz de rotação pelo *theta* transposta. *Tilt* e *theta* são calculados então pelas equações:

$$\tau = \arccos(R_{11}) \quad (3)$$

$$\theta = \arccos(Z_{11}) \quad (4)$$

A matriz Σ representa a compressão pelo *slant*. Entretanto, usualmente o valor de ς_{11} é diferente de 1. Para que esta siga os moldes de uma transformação T_M , Eq. (1), o valor de um dos elementos da diagonal deve ser igual a 1. Para garantir isso assumimos então que o *slant* é o arco cujo co-seno é igual a $\frac{\varsigma_{22}}{\varsigma_{11}}$, conforme a equação:

$$\sigma = \arccos\left(\frac{\varsigma_{22}}{\varsigma_{11}}\right) \quad (5)$$

O capítulo seguinte apresenta, em detalhes, as técnicas utilizadas neste trabalho.

4 Metodologia

Este capítulo descreve os métodos selecionados para este estudo, nomeadamente, os trabalhos, Super e Bovik [29], Rosenholtz e Malik [30], e Loh e Zisserman [44].

O primeiro método foi selecionado porque, ao contrário da maioria dos demais, não se baseia em definir recortes na imagem, mas emprega um banco de filtros de Gabor, em diferentes configurações, não sendo assim necessário definir explicitamente um tamanho de recorte.

Do segundo temos aqui explicitado somente o algoritmo que estima a transformação afim entre duas imagens, uma vez que o método completo de *SFT* apresentado no artigo utiliza projeção em perspectiva e, neste trabalho, optamos por projeções ortográficas. A estimativa da transformação afim do método de Rosenholtz e Malik também se torna atrativa para uma comparação com o terceiro método.

O terceiro, Loh e Zisserman, é o trabalho mais recente em estimativa de transformações afins, apresentando resultados promissores. Loh possui também um trabalho na área de *SFT* [45] que se baseia nesta estimativa, entretanto, também busca estimar o modelo frontal de textura. Nesta dissertação partiu-se da premissa que o modelo frontal é conhecido e dado como entrada para o algoritmo.

A estimativa da transformação afim é relacionada em *SFT* da seguinte maneira: dados dois recortes extraídos de pontos distintos na imagem, o primeiro com uma vista frontal de uma textura e o segundo contendo a projeção ortográfica do anterior, determinar a transformação afim que os relaciona. De posse dessa transformação tem-se os ângulos *tilt* e *slant*, como visto no capítulo anterior.

As subseções a seguir descrevem em detalhes os métodos supracitados. Ao final deste capítulo também é apresentado o algoritmo *SFT* usado nos experimentos.

4.1

Método de Super e Bovik

A grande dificuldade em muitos trabalhos de *SFT* está em encontrar os elementos de textura, os *texels*. Normalmente se utiliza detecção de bordas, através de um processamento local que prescinde da informação presente na imagem como um todo, e ainda é capaz de encontrar sub-texturas, algo não desejado dependendo da escala em que se trabalha.

Super e Bovik [29] procuram contornar estas dificuldades e utilizam a frequência espacial local como medida de textura, uma vez que essa pode representar qualquer textura e nenhuma informação é desprezada.

A técnica clássica de análise de frequência em imagens é a transformada de Fourier [40], que mostra a amplitude e fase da variação espacial em cada frequência espacial 2D. Porém, esta representação é global e mostra o comportamento da frequência espacial na imagem como um todo, sendo, portanto, imprópria para *SFT* sobre superfícies curvas, uma vez que se pretende analisar localmente a frequência em determinados pontos na imagem.

Com essa análise local é possível estimar a variação da frequência entre dois pontos e, usando um modelo de projeção adequado, pode-se estimar a orientação relativa entre eles. Na Figura 12 há um exemplo de um cilindro texturizado. A frequência (número de barras numa mesma unidade de medida) aumenta a medida que caminhamos do centro para as bordas. Se conseguirmos medir essa variação de frequência podemos encontrar a orientação relativa.

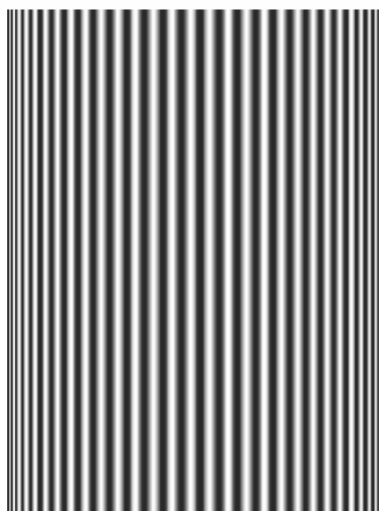


Figura 12 - Cilindro sintético com textura senoidal aplicada.

Entretanto, o exemplo da figura é relativamente simples. Em um caso mais genérico, como determinar automaticamente a frequência espacial a ser analisada na imagem, e ainda, a direção desta? Para resolver este problema são empregados Filtros de Gabor, que podem ser sintonizados em determinada frequência e orientação. Utilizam-se então inúmeras configurações de filtros na imagem, sintonizados em frequências e orientações distintas, cujos resultados são agregados usando momentos de segunda ordem.

Filtros de Gabor, momentos de segunda ordem e outras etapas do algoritmo são descritas nas subseções a seguir.

4.1.1 Filtros de Gabor

Gabor [46], em 1946, viu que a teoria da comunicação baseava-se em dois métodos de processamento de sinais: um descrevia o sinal como função do tempo, outro em termos de sua frequência (Fourier). A representação no domínio do tempo define a amplitude do sinal em cada instante, enquanto que a do domínio da frequência utiliza senóides infinitamente longas (no domínio do tempo), definidas pela frequência, amplitude e fase.

Como então poderíamos decompor um sinal de forma a obter informação de frequência sem perder a localização? Poderíamos usar a transformada de Fourier em janelas. Como definir neste caso o tamanho ideal da janela? Gabor introduziu então filtros que possuem o melhor compromisso de localização simultânea no domínio da frequência e espacial permitidos pelo princípio da incerteza:

$$\Delta t \Delta f \geq \frac{1}{2},$$

onde Δt é a incerteza da localização do sinal no domínio do tempo e Δf é a incerteza da frequência do sinal.

Para satisfazer essa condição Gabor utiliza uma senóide complexa modulada por uma Gaussiana, dada pela equação (6):

$$h(t) = e^{-\gamma^2(t-t_0)^2} \cdot (\cos[2\pi f(t-t_0) + \delta] + j \sin[2\pi f(t-t_0) + \delta]), \quad (6)$$

onde γ é a constante da Gaussiana, inversamente proporcional à largura da função, t_0 define o centro da Gaussiana, f e δ são, respectivamente, a frequência e a fase da senóide complexa.

Daugman [47, 48], estendeu os filtros de Gabor para 2D e percebeu que estes possuíam resposta similar aos de algumas células do córtex visual de mamíferos. Na verdade, foi a partir do trabalho de Daugman, em processamento de imagem, que as funções de Gabor passaram a se chamar filtros de Gabor.

A equação de um filtro de Gabor h em 2D fica da forma:

$$h(x, y) = g(x', y') \cdot \exp[2\pi j(Ux + Vy)], \quad (7)$$

onde U é a frequência horizontal do filtro, V é a frequência vertical do filtro e x' e y' são rotações de x e y :

$$\begin{aligned} x' &= x \cos \phi + y \sin \phi \\ y' &= -x \sin \phi + y \cos \phi, \end{aligned} \quad (8)$$

onde ϕ é o ângulo de rotação.

Também na eq. (7) g é uma gaussiana do tipo

$$g(x, y) = \left(\frac{1}{2\pi\kappa\gamma^2} \right) \cdot \exp \left[-\frac{(x/\kappa)^2 + y^2}{2\gamma^2} \right] \quad (9)$$

onde γ é o desvio-padrão da gaussiana e κ representa a razão entre os dois semi-eixos da elipse, definida pela interseção da gaussiana com um plano paralelo a xy .

A eq. (10) mostra a transformada de Fourier do filtro:

$$H(u, v) = \exp \left\{ -2\pi^2 \gamma^2 \left[(u' - U')^2 \kappa^2 + (v' - V')^2 \right] \right\}, \quad (10)$$

onde u' e v' são rotações dos valores de frequência u e v , semelhantes à eq.(8):

$$\begin{aligned} u' &= u \cos \phi + v \sin \phi \\ v' &= -u \sin \phi + v \cos \phi \end{aligned}$$

Também é possível representar as frequências U e V do filtro através da frequência radial F e orientação Φ :

$$\begin{aligned} F &= \sqrt{U^2 + V^2} \\ \Phi &= \tan^{-1} \left(\frac{V}{U} \right) \end{aligned} \quad (11)$$

Normalmente se utilizam filtros de Gabor em que a orientação da senóide complexa e da gaussiana coincidem ($\phi = \Phi$). Resulta daí uma nova fórmula para o filtro de Gabor, em função da frequência radial:

$$h(x, y) = g(x', y') \cdot \exp(2\pi j F x') \quad (12)$$

e sua transformada de Fourier;

$$H(u, v) = \exp \left\{ -2\pi^2 \gamma^2 \left[(u' - F)^2 \kappa^2 + (v')^2 \right] \right\} \quad (13)$$

Na Figura 13 temos um exemplo de filtro de Gabor, seguido por sua transformada de Fourier na Figura 14.

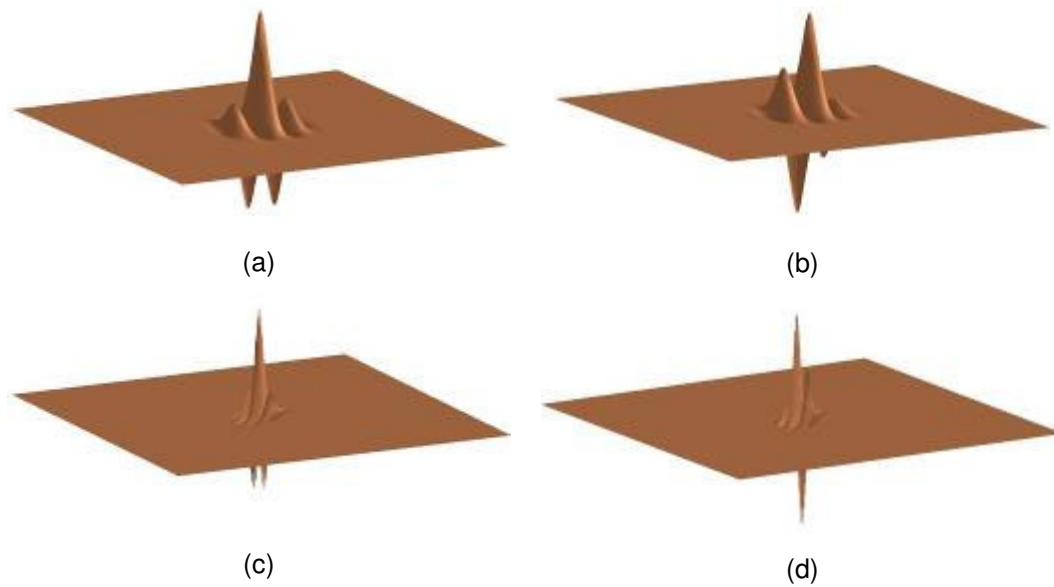


Figura 13 - Exemplos de Filtros de Gabor no domínio espacial. (a) e (b), parte real e imaginária, respectivamente, de um filtro tendo $F=12$ ciclos/imagem, $\kappa=1$ e $\Phi=0^\circ$. (c) e (d), parte real e imaginária, respectivamente, de um filtro tendo $F=32$ ciclos/imagem, $\kappa=1/2$ e $\Phi=45^\circ$.

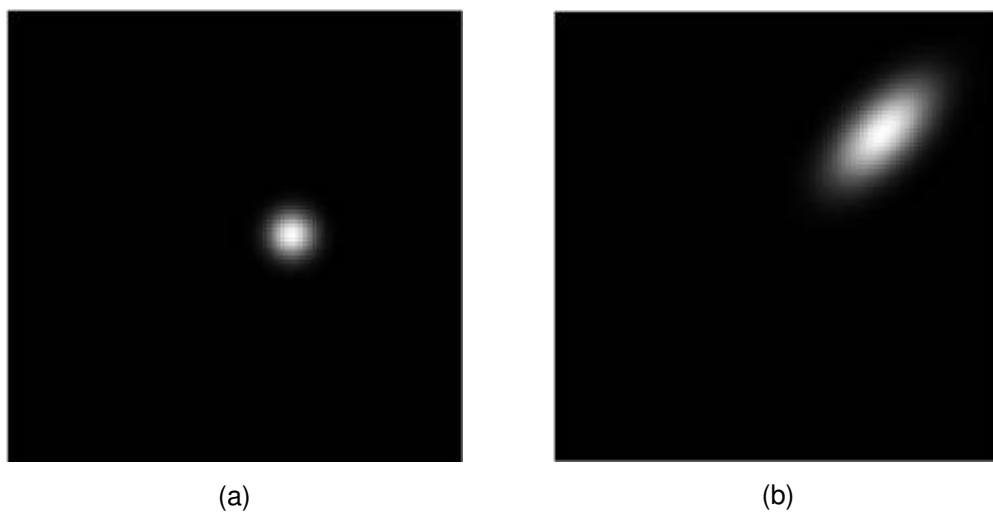


Figura 14 - Transformadas de Fourier dos Filtros de Gabor. (a) $F=12$ ciclos/imagem, $\kappa=1$ e $\Phi=0^\circ$. (b) $F=32$ ciclos/imagem, $\kappa=1/2$ e $\Phi=45^\circ$.

Para construir os filtros pode-se especificá-los em termos de sua largura de banda β (medida em oitavas) e orientação Ω (em radianos) [49, 50], conforme:

$$\begin{aligned}\beta &= \log_2 \left[\frac{\pi F \kappa \gamma + \alpha}{\pi F \kappa \gamma - \alpha} \right] \\ \Omega &= 2 \tan^{-1} \left[\frac{\alpha}{\pi F \gamma} \right]\end{aligned}\tag{14}$$

onde α é uma constante, $\alpha = \sqrt{\frac{\ln 2}{2}}$

Finalizando, no presente trabalho os filtros de Gabor foram usados no domínio da frequência, segundo a eq. (13), onde são necessários os parâmetros: F – frequência radial, Φ – orientação (mesmo ângulo usado para a gaussiana – φ), κ e γ – parâmetros da gaussiana. Para a construção dos filtros foram determinados alguns valores para F e Φ , mantendo um valor fixo de $\kappa=1$ e também um valor fixo para a largura de banda β . Com esses valores pode-se calcular γ para cada filtro utilizado, com a eq. (15), obtida através da manipulação algébrica da eq.(14).

$$\gamma = \frac{\alpha(2^\beta + 1)}{\pi F \kappa(2^\beta - 1)}\tag{15}$$

4.1.2

Convolução com os filtros e momentos de segunda ordem

Usando os filtros de Gabor podemos então estimar a frequência local de uma imagem fazendo a convolução com o filtro:

$$A_i(x, y) = |f(x, y) * h_i(x, y)|,\tag{16}$$

onde $A_i(x,y)$ é a amplitude em cada ponto, $f(x,y)$ é uma imagem 2D e $h_i(x,y)$ é um filtro de Gabor, eq. (12). A energia local é definida como $(A_i(x,y))^2$. Esta medida

expressa a quantidade de energia próxima à frequência e orientação do filtro h_i no ponto (x,y) . A saída do filtro de Gabor revela se na vizinhança do ponto (x,y) existe um padrão de repetição, com frequência e orientação específicas, definidas pelos parâmetros.

Segundo o teorema da convolução a operação representada na eq. (16) pode ser realizada no domínio da frequência através da multiplicação entre a transformada de Fourier da imagem e a função de transferência do filtro dada pela eq.(13).

Como não se sabe *a priori* a frequência e orientação de uma textura qualquer em uma imagem, utilizam-se vários filtros de Gabor, sintonizados em diferentes frequências e orientações. Para denotar os diferentes filtros utilizamos aqui o índice i . E para agregar as diferentes saídas dos filtros os autores empregam os momentos de segunda ordem:

$$\begin{aligned} a(x, y) &= \sum_i U_i^2 A_i^2(x, y) \\ b(x, y) &= 2 \sum_i U_i V_i A_i^2(x, y) \\ c(x, y) &= \sum_i V_i^2 A_i^2(x, y) \end{aligned} \tag{17}$$

onde U_i e V_i são estimativas da frequência local do componente de textura. No primeiro trabalho dos autores [51] são utilizados os próprios valores de frequência, U_i e V_i , em que o filtro $h_i(x,y)$ de Gabor está sintonizado. O problema neste caso é que a frequência real do componente de textura pode não ser a mesma em que o filtro está sintonizado, uma vez que a banda de passagem do filtro admite certa gama de frequências, próximas à frequência central que foi passada como parâmetro. Com isso pode-se ter uma estimativa ruim da frequência. Na literatura esta peculiaridade é chamada de “problema da estimativa de frequência”.

Uma possível solução seria aumentar o número de filtros de Gabor utilizados, ou seja, realizar uma amostragem mais densa do espaço de frequência. Entretanto, isso elevaria o número de convoluções e o conseqüente custo computacional.

Para contornar o problema apresentado, os autores empregam uma técnica desenvolvida por Havlicek, Bovik e Maragos [52] que gera melhores estimativas de U_i e V_i através da eq. (18):

$$\begin{aligned} |U_i(x, y)| &\approx \frac{B_i(x, y)}{2\pi A_i(x, y)} \\ |V_i(x, y)| &\approx \frac{C_i(x, y)}{2\pi A_i(x, y)}, \end{aligned} \quad (18)$$

onde B_i e C_i são calculados como A_i , por uma convolução com a imagem porém, agora usando as derivadas parciais do filtro de Gabor.

$$\begin{aligned} B_i(x, y) &= \left| f(x, y) * \frac{\partial h_i(x, y)}{\partial x} \right| \\ C_i(x, y) &= \left| f(x, y) * \frac{\partial h_i(x, y)}{\partial y} \right| \end{aligned} \quad (19)$$

Note que, U_i e V_i agora possuem ao longo de (x, y) valores distintos, diferentes dos valores fixos da configuração do filtro.

Os sinais de $U_i(x, y)$ e $V_i(x, y)$ na eq. (18) são resolvidos minimizando $|U_i(x, y) - U_i|$ e $|V_i(x, y) - V_i|$, respectivamente, onde U_i e V_i são os valores de frequência no qual está sintonizado um filtro h_i .

Substituindo a eq. (18) na eq. (17) encontramos:

$$\begin{aligned} a(x, y) &= \sum_i \frac{B_i^2(x, y)}{4\pi^2} \\ b(x, y) &= 2 \sum_i \frac{B_i(x, y)C_i(x, y)}{4\pi^2} \\ c(x, y) &= \sum_i \frac{C_i^2(x, y)}{4\pi^2} \end{aligned} \quad (20)$$

Calcula-se, desta forma, apenas a convolução da imagem com as derivadas parciais do filtro. Após a convolução realiza-se uma suavização na saída do filtro de Gabor com um filtro gaussiano [40].

A_i pode ser utilizada ainda para normalizar os momentos com relação a variações de iluminação e contraste na imagem, dividindo cada momento da eq. (20) pelo termo $\sum_i A_i^2(x, y)$.

4.1.3 Momentos canônicos

Os momentos da eq. (20) dependem do sistema de coordenadas escolhido. Para tornar a informação independente de orientação estes são transformados para sua forma canônica, onde os momentos máximos e mínimos de segunda ordem, M e m , respectivamente, são invariantes à orientação do sistema de coordenadas e o ângulo ψ define o eixo do menor segundo momento. Estes momentos podem ser interpretados como elipses, onde M é o eixo maior e m o eixo menor, e ψ o ângulo de rotação. M , m e ψ são calculados da seguinte forma [53]:

$$\begin{aligned} M &= \frac{1}{2} \left(a + c + \sqrt{b^2 + (a - c)^2} \right) \\ m &= \frac{1}{2} \left(a + c - \sqrt{b^2 + (a - c)^2} \right) \\ \psi &= \frac{1}{2} \arctan \left(\frac{b}{a - c} \right) \end{aligned} \tag{21}$$

4.1.4 Projeção da textura

De posse dos momentos devidamente calculados os autores derivam, através de um modelo de projeção ortográfico, a projeção dos momentos na superfície (espaço 3D) para o plano da imagem (ver derivações no Anexo I):

$$\begin{bmatrix} a_s \\ b_s \\ c_s \end{bmatrix} = \begin{bmatrix} \cos^2 \sigma \cos^2 \tau & \cos^2 \sigma \cos \tau \sin \tau & \cos^2 \sigma \sin^2 \tau \\ -2 \cos \sigma \cos \tau \sin \tau & \cos \sigma (\cos^2 \tau - \sin^2 \tau) & 2 \cos \sigma \cos \tau \sin \tau \\ \sin^2 \tau & -\cos \tau \sin \tau & \cos^2 \tau \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix}, \quad (22)$$

onde σ é o *slant* e τ o *tilt*. O índice s significa que são os momentos do objeto na superfície, calculados caso não houvesse deformação da projeção, ou seja, o ponto visto de frente. Sem o índice s são os momentos calculados na imagem. A eq. (22) provê um método para calcular os valores de *slant* (σ) e *tilt* (τ), conhecendo os momentos da vista frontal de uma textura (a_s, b_s e c_s) e os momentos da projeção da mesma textura sobre o plano da imagem (a, b e c).

4.1.5 Cálculo da orientação

Como foi dito na seção 4.1.3 os momentos podem ser representados por elipses, tendo sua área definida como $4ac - b^2$. O *slant* (σ) então pode ser definido pela razão entre as áreas da elipse na imagem e a elipse na superfície, tendo o momento canônico na superfície (M_s e m_s) e na imagem (M e m), conforme a eq. (23) abaixo:

$$\cos \sigma = \sqrt{\frac{4a_s c_s - b_s^2}{4ac - b^2}} = \sqrt{\frac{M_s m_s}{Mm}} \quad (23)$$

Expandindo a igualdade $M_s + m_s = a_s + c_s$ usando a eq. (22), os autores encontram também a equação para o *tilt* (τ).

$$\tau = \left\{ \psi \pm \frac{1}{2} \arccos \lambda, \psi \pm \frac{1}{2} \arccos \lambda + \pi \right\} \quad (24)$$

onde ψ é a orientação dos momentos canônicos e λ é dado pela eq. (25):

$$\lambda = \frac{(\cos^2 \sigma + 1)(M + m) - 2(M_s + m_s)}{\sin^2 \sigma (M - m)} \quad (25)$$

Nota-se que na eq. (24) existem 4 soluções diferentes para o *tilt*. Duas correspondem à ambigüidade do sinal do arco co-seno de λ , pois desconhecemos em qual quadrante se encontra o ângulo.

As outras duas soluções vêm da ambigüidade de 180° (π). Essa não pode ser resolvida somente a partir da informação de textura em projeção ortográfica.

É importante ressaltar que tanto para a solução do *slant*, eq. (23), quanto para a do *tilt*, eq. (24), é necessário conhecer os momentos canônicos tanto na superfície quanto na imagem. Neste trecho reside o problema, pois usualmente se dispõe apenas da medida na imagem.

Uma solução para este problema seria conhecer a orientação em algum ponto da imagem e usar este como referência. Ao compararmos este ponto com algum outro teríamos a orientação relativa entre ambos, e, conhecendo a orientação de um deles com relação à origem do sistema de coordenadas, facilmente encontramos a outra.

A solução acima é facilitada se tomarmos como referência um ponto na imagem onde os ângulos sejam zero, ou seja, um ponto visto de frente na imagem. Nesse caso a orientação relativa corresponde diretamente a orientação do ponto que está sendo comparado.

Os autores supõem que sempre existe na superfície um ponto frontal. Usando os momentos canônicos deste ponto como referências (M_s e m_s) podemos encontrar a orientação em outros pontos da imagem! Para encontrar este ponto procuramos pelo ponto de mínimo *slant*, ou seja, pontos de mínimo \sqrt{Mm} .

Na tabela abaixo se encontra o resumo do algoritmo de Super e Bovik.

1.	Realize a convolução da imagem com os filtros de Gabor e suas derivadas parciais. Eqs.(16) e (19);
2.	Suavize as saídas dos filtros realizando uma convolução com uma gaussiana;
3.	Calcule os momentos de segunda ordem (a , b e c) com as saídas dos filtros. Eq. (20);
4.	Normalize os momentos dividindo-os pela energia total $\sum_i A_i^2(x, y)$;
5.	[passo opcional – para texturas não totalmente homogêneas] Suavize os momentos realizando uma convolução com um filtro gaussiano;
6.	Calcule os momentos canônicos (M , m e θ) usando a eq.(21);
7.	Encontre a ponto frontal (ponto de menor valor \sqrt{Mm}) e assumo M_s e m_s como sendo os valores de M e m deste ponto;
8.	Calcule <i>slant</i> e <i>tilt</i> em cada ponto usando as eqs.(23) e (24).

Tabela 1 - Algoritmo *SFT* de Super e Bovik.

A seguir encontra-se descrito o segundo método do presente estudo.

4.2.

Método de Rosenholtz e Malik

O método de Rosenholtz e Malik [30] para a estimativa de uma transformação afim entre duas imagens é parte do artigo dos autores na área de *SFT* aqui referenciado. Como foi dito no início deste capítulo estudaremos nesta seção somente o procedimento que estima a transformação afim.

O método é baseado numa análise diferencial da transformação que relaciona as duas imagens. Porém, ao invés de estimar esta transformação no domínio espacial, Rosenholtz o faz no domínio da frequência. Isso é possível devido à propriedade da transformada de Fourier, expressa pela equação abaixo [54]:

$$F[g(Tx)] = \frac{1}{|T|} G(T^{-t} w), \quad (26)$$

onde $G = F(g)$ representa a transformada de Fourier da imagem g , T representa uma transformação afim. A eq. (26) mostra que aplicar uma transformação afim em uma imagem equivale a aplicar a inversa transposta desta mesma transformação afim à transformada de Fourier da imagem, dividida pelo

determinante da transformação afim. Encontrando então a transformação afim no domínio da frequência podemos encontrar a mesma no domínio espacial.

A vantagem de trabalhar no domínio da frequência é que, se utilizarmos somente a magnitude da transformada de Fourier, o algoritmo será invariante à pequenas mudanças de posição, uma vez que estas variações causam pequenas diferenças apenas na fase. Se trabalhássemos no domínio espacial teríamos de assegurar que os centros dos recortes da imagem estivessem corretamente alinhados.

4.2.1 Construção dos *espectrogramas*

O primeiro passo para encontrar a transformação afim consiste em obter o chamado *espectrograma* das imagens. Isso envolve a seguinte sequência de ações:

1. **Normalização das imagens:** Normaliza-se cada imagem para que tenham a mesma escala de tons de cinza, e subtrai-se da média. Esse passo torna as imagens menos sensíveis a variações de contraste e sombreamento;

Em seguida filtra-se a imagem com uma janela de Welch [55], para amenizar o efeito das bordas da textura. A equação de uma janela de Welch é dada por:

$$W(x, y) = \left[1 - \left(\frac{(x-1) - \frac{1}{2}(L-1)}{\frac{1}{2}(L-1)} \right)^2 \right] * \left[1 - \left(\frac{(y-1) - \frac{1}{2}(C-1)}{\frac{1}{2}(C-1)} \right)^2 \right], \quad (27)$$

onde $W(x, y)$ é o valor da função no ponto (x, y) , L e C são o tamanho da imagem, altura e largura, respectivamente. No caso de uma matriz, L e C são, respectivamente, o número de linhas e colunas.

Após a filtragem realiza-se o *detrend* [56], isto é, subtrai-se das imagens o plano que melhor se ajusta às intensidades da mesma. Este procedimento reduz os efeitos da iluminação não-uniforme.

2. **Transformada de Fourier:** realiza-se a transformada de Fourier das imagens, adicionando zeros antes de realizar a transformada [57]; usualmente dobra-se a imagem de tamanho nessa concatenação com zeros. Toma-se a Magnitude da Transformada de Fourier:

$$|F|$$

3. **Extração dos componentes de baixa frequência:** A maior parte da informação costuma ficar nas baixas frequências da MTF. Para extrair essa zona de baixa frequência utiliza-se o seguinte critério: recorta-se uma janela de tamanho tal que cada ponto fora da janela possua magnitude inferior a 50% do máximo. A esta região dá-se o nome de *espectrograma*.
4. **Normalização:** é necessário normalizar os *espectrogramas* das duas imagens, de forma que possuam picos semelhantes. Para tal aplica-se a seguinte normalização: subtrai-se o mínimo de todos os valores e divide-se o resultado pelo máximo. Isso faz com que tenhamos valores entre 0 e 1.

4.2.2 Análise diferencial

De posse dos *espectrogramas* de ambas as imagens pode-se determinar a transformação afim que as relaciona. Para tal é usando um método diferencial. Para expor o método, primeiro o descreveremos em 1D.

Suponha que temos duas curvas, f e g , tal que sejam relacionadas por uma transformação afim. Em 1D, uma curva é uma versão escalada da outra:

$$g(w) = f(aw), \quad (28)$$

onde a é o fator de escala, que podemos reescrever como $1 + \Delta a$, onde assumimos que Δa é pequeno. Sabemos, da expansão em série de Taylor [58], que:

$$f(w + \Delta w) \approx f(w) + \frac{df}{dw}(w) \cdot \Delta w \quad (29)$$

Substituindo Δw por $\Delta a \cdot w$ temos:

$$f(aw) \approx f(w) + \frac{df}{dw}(w) \cdot \Delta aw \quad (30)$$

Usando a eq. (28):

$$g(w) - f(w) \approx \frac{df}{dw}(w) \cdot \Delta aw \quad (31)$$

A eq. (31) é para o caso 1D, estendendo para 2D, usando uma transformação afim T_r , igual a $I + \Delta T_r$, onde I é a matriz identidade, temos:

$$F_2(\vec{w}) - F_1(\vec{w}) \approx \vec{\nabla} F_1 \circ \Delta T_r \vec{w}, \quad (32)$$

onde F_1 e F_2 são os *espectrogramas* das imagens, w é a frequência e ∇F_1 é o gradiente do *espectrograma* da primeira imagem. Reescrevendo a eq. (32), desmembrando o ΔT_r em seus elementos, de uma matriz 2x2, t_{r11} , t_{r12} , t_{r21} e t_{r22} , temos:

$$\left[\frac{\partial F_1}{\partial w_x} w_x \frac{\partial F_1}{\partial w_x} w_y \frac{\partial F_1}{\partial w_y} w_x \frac{\partial F_1}{\partial w_y} w_y \right] \begin{bmatrix} t_{r11} \\ t_{r12} \\ t_{r21} \\ t_{r22} \end{bmatrix} = F_2(\vec{w}) - F_1(\vec{w}) \quad (33)$$

Cada ponto no *espectrograma* fornece uma nova equação. Isto remete a um problema clássico de sistema de equações lineares, $Dz = l$, onde l é o termo do lado direito da eq. (33), D o produto do gradiente com a frequência w e z os elementos da transformação ΔT_r , que compõem a solução.

Este sistema pode ser resolvido pelo método de mínimos-quadrados [55], caso tenhamos mais do que quatro pontos, algo fácil em se tratando de imagens. Entretanto, para reduzir o efeito de ruído e rejeitar *outliers* no algoritmo, os

autores utilizam duas técnicas na escolha dos pontos que serão empregados. São elas:

- 1) Rejeitar pontos no *espectrograma* com magnitude menor que 10% do máximo;
- 2) Rejeitar pontos, usando uma técnica de mínimos-quadrados ponderada, da estatística robusta [59], conforme o procedimento abaixo:
 - Obter a estimativa inicial do ΔT_r , que chamamos de ΔT_{r0} , e calcular o resíduo r , $r=l-l'$. O resíduo representa a diferença entre os valores reais de l e os estimados, l' , com ΔT_{r0} aplicado na eq. (32).
 - Dividir os resíduos pelo valor $\frac{1}{F_1}$, agora chamados de r' .
 - Rejeitar os pontos que possuam o resíduo ponderado r' maior do que o valor $thr = 2.5 * 1.4826 \sqrt{\text{mediana}(r'^2)}$.

Tendo rejeitado alguns pontos nos dois passos anteriores, calcula-se com os pontos restantes a estimativa do ΔT_r , chamada $\Delta T_r'$. Encontramos a estimativa da transformação afim T_r' fazendo $T_r' = I + \Delta T_r'$.

Essa estimativa pode ainda ser melhorada usando um método iterativo:

- 1) Aplica-se a estimativa inicial T_{r1}' no *espectrograma* da primeira imagem, gerando assim um *espectrograma* transformado;
- 2) Calcula-se uma nova estimativa T_{r2}' , entre o *espectrograma* transformado e o segundo *espectrograma*. Se a primeira estimativa estiver correta, esta nova estimativa será bem pequena;
- 3) Repetem-se os passos 1 e 2 iterativamente, até que a diferença entre uma estimativa e a seguinte seja pequena ou que seja atingido um número limítrofe de iterações;
- 4) A estimativa final é o conjunto de todas as estimativas:

$$T_r' = T_{r1}' * T_{r2}' * \dots * T_{rm}'.$$

Vale lembrar que esta estimativa final T_r' se encontra no domínio da frequência, sendo necessário trazê-la para o domínio espacial. Para tal basta calcular a matriz inversa da transposta, como indicado na eq. (26).

Na Figura 15 é mostrado o diagrama do método.

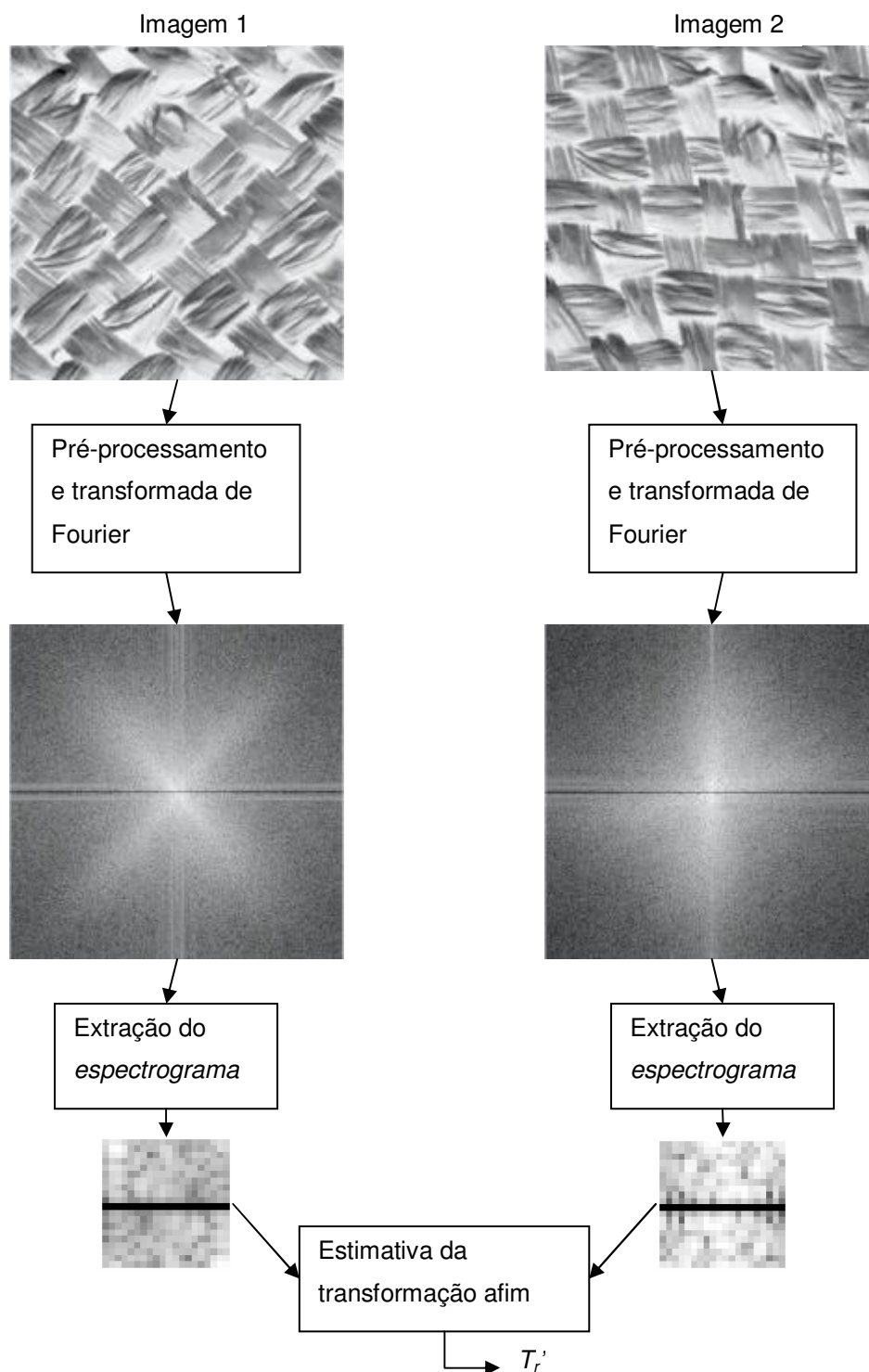


Figura 15 - Diagrama do método de Rosenholtz e Malik.

Na seção a seguir é descrito o terceiro método utilizado neste trabalho.

4.3. Método de Loh e Zisserman

O método de Loh e Zisserman [44] é um dos mais recentes na área de estimativa da transformação afim. O algoritmo transforma cada uma das imagens em sua versão isotrópica e depois encontra a rotação que relaciona as duas. De posse dessas três transformações, duas isotrópicas e a rotação, calcula-se a estimativa final da transformação afim como sendo uma combinação delas.

As subseções a seguir descrevem o algoritmo em detalhes:

4.3.1 Transformação para textura isotrópica

O primeiro passo é realizar uma transformação afim nas imagens que produza uma textura isotrópica (ver definição no Capítulo 3). Um exemplo pode ser visto na Figura 16:

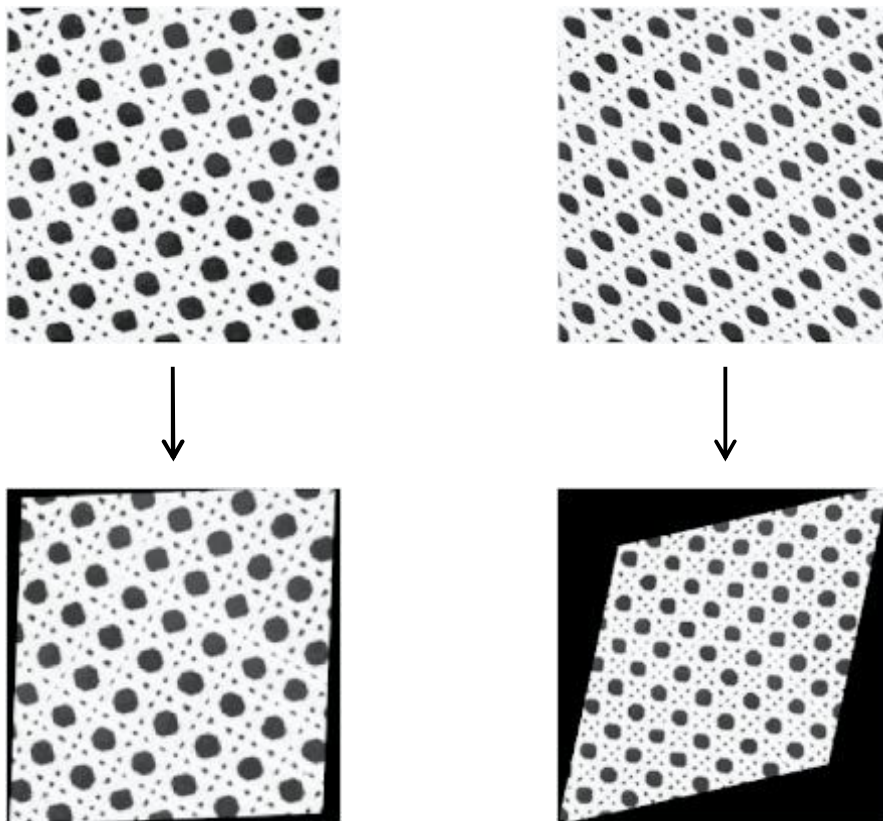


Figura 16 - Transformação de uma textura em sua versão isotrópica.

Essa idéia tem origem no trabalho de Schaffalitzky e Zisserman [60], que a utilizam para gerar um descritor de texturas invariante a transformações afins. Para tal, inicialmente é calculada a matriz do momento de segunda ordem da imagem, conforme dado pela equação abaixo:

$$\mathbf{M} = \int_{\Omega} \nabla I \otimes \nabla I \frac{dxdy}{|\Omega|} = \int_{\Omega} \begin{pmatrix} I_x I_x & I_x I_y \\ I_x I_y & I_y I_y \end{pmatrix} \frac{dxdy}{|\Omega|}, \quad (34)$$

onde I é uma imagem, I_x e I_y são os gradientes na direção horizontal e vertical, respectivamente, e Ω é a área da imagem, no caso, o número de linhas multiplicado pelo número de colunas. Em termos de implementação a eq. (34) se resume a:

$$\mathbf{M} = \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix} \frac{1}{|\Omega|}, \quad (35)$$

onde:

$$\begin{aligned} m_{11} &= \sum I_x^2 \\ m_{12} &= m_{21} = \sum I_x I_y \\ m_{22} &= \sum I_y^2 \end{aligned}$$

A transformação afim T_{iso} que torna então a imagem I isotrópica é a que satisfaz $T_{iso}^t T_{iso} = M$. Uma solução é possível decompondo a matriz M em seus autovalores (Λ) e autovetores (Vet), substituindo os últimos por suas raízes quadradas:

$$\begin{aligned} M &= Vet \Lambda Vet^{-1} \\ T_{iso} &= Vet \sqrt{\Lambda} Vet^{-1} \end{aligned}$$

Aplica-se então a transformação T_{iso} à imagem. Note que são calculadas duas transformações, T_{iso1} para a primeira imagem e T_{iso2} para a segunda, cada uma oriunda do momento de segunda ordem de cada imagem.

4.3.2 Cálculo da rotação

É necessário agora encontrar a rotação existente entre as duas imagens transformadas em isotrópicas no passo anterior.

Primeiro, calcula-se a transformada de Fourier das imagens para reduzir efeitos de translação. Porém, antes de calcular a transformada, a imagem é submetida a um filtro passa-baixa gaussiano, para reduzir o efeito das bordas sobre a textura.

As duas imagens transformadas diferem por uma rotação e uma escala. Um exemplo pode ser visto na Figura 17.

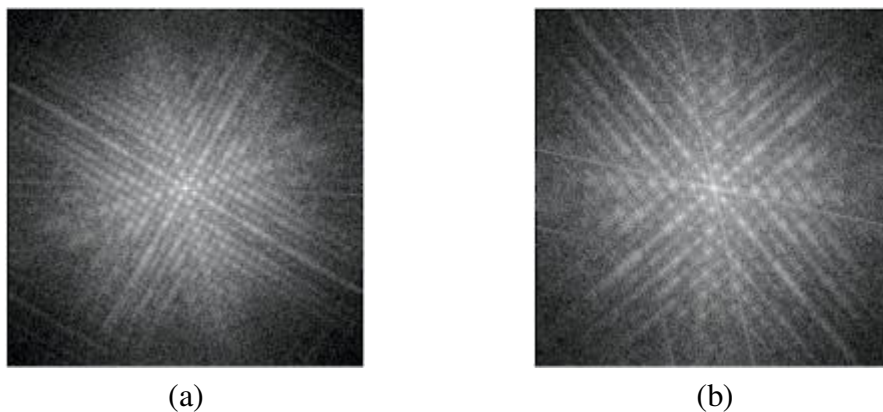
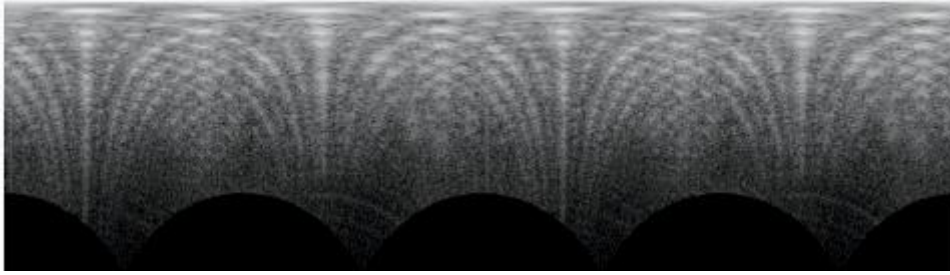
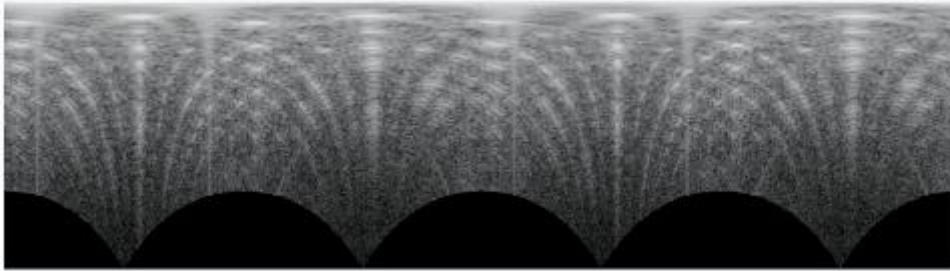


Figura 17 - Transformada de Fourier da versão isotrópica da primeira imagem (a) e da segunda (b).

Uma rotação no domínio espacial resulta numa rotação equivalente no domínio da frequência, enquanto que uma expansão equivale a uma compressão no domínio da frequência, sendo que somente a rotação nos interessa. Com o intuito de facilitar o cálculo da rotação, os autores realizam duas manobras: primeiro convertem a transformada de Fourier de cada imagem para coordenadas polares (Figura 18); segundo, calculam a média de cada coluna na representação polar. Antes de calcular a média, as primeiras linhas são excluídas. Isto se dá porque correspondem a valores de baixa frequência, que acabam dominando a média. Nos experimentos descritos no próximo capítulo foram excluídas 20% das linhas, valor definido empiricamente.



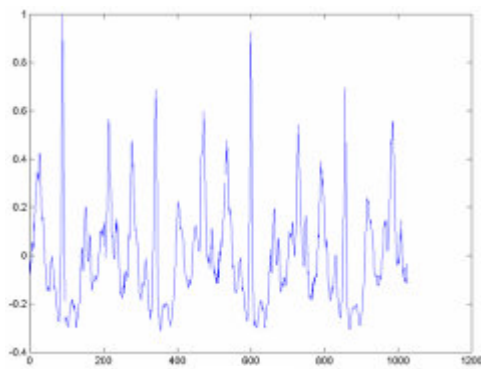
(a)



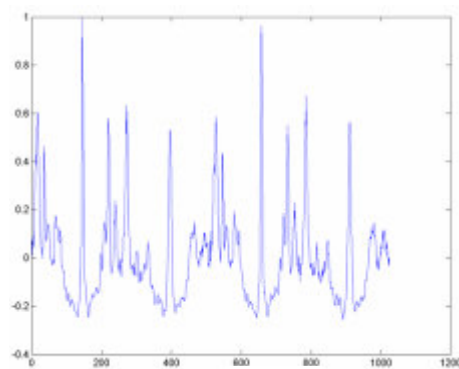
(b)

Figura 18 - Transformada de Fourier em coordenadas polares da versão isotrópica da primeira (a) e da segunda (b) imagem.

Após o cálculo da média, temos agora dois sinais, correspondentes a cada imagem, um exemplo é visto na Figura 19. Estes são centrados em zero (diminui-se da média) e normalizados com valores entre 0 e 1.



(a)



(b)

Figura 19 - Sinais obtidos a partir da média das transformadas de Fourier nas coordenadas polares. (a) primeira imagem e (b) segunda imagem.

Em coordenadas polares a rotação corresponde a uma alteração de fase. O valor da rotação é determinado deslocando-se um dos sinais de modo circular e calculando a correlação entre ambos. O deslocamento que possuir maior correlação corresponde à diferença de fase procurada.

4.3.3

Transformação final

De posse das três transformações (T_{iso1} , T_{iso2} e a rotação) encontramos a transformação afim final entre as duas imagens originais como sendo:

$$T_L' = T_{iso1} Rot(\psi) T_{iso2}^{-1}, \quad (36)$$

onde $Rot(\psi)$ é uma matriz de rotação, equivalente à matriz que representa a rotação do *tilt* na eq. (1) do Capítulo 3.

A eq. (36) representa a seguinte transformação: transforma-se a primeira imagem em sua versão isotrópica, aplica-se uma rotação de ψ , e depois a transformação inversa que torna a segunda imagem isotrópica.

Encerra-se aqui a descrição dos três métodos citados na introdução deste capítulo. A seção a seguir apresenta um algoritmo de *SFT*.

4.4

Algoritmo completo de reconstrução de superfícies

Dos três métodos apresentados, somente no primeiro é apresentado um algoritmo completo para a obtenção da orientação em todos os pontos. Os outros dois métodos são apresentados apenas para estimar a transformação afim entre duas imagens. Nesta seção é apresentado um algoritmo *SFT*, que extrai recortes na imagem de um objeto texturizado e os fornece como entrada para um algoritmo de estimativa da transformação afim, como os citados anteriormente. Este procedimento é baseado nos passos iniciais do método de Forsyth [36]. Entretanto, o método busca também estimar o modelo frontal da textura do objeto.

Este passo está fora do escopo do presente trabalho, e considerou-se que o modelo frontal é uma entrada manual do algoritmo.

4.4.1

Extração de recortes

Nesta etapa são extraídos da imagem recortes de igual tamanho. Aqui dois pontos são importantes: o tamanho do recorte e sua localização. O primeiro é empírico, depende de fatores como tamanho da imagem e tamanho dos elementos formadores da textura, de forma que, em cada recorte, tenhamos informação suficiente sobre a textura. De fato, este é um problema em visão computacional difícil de ser automatizado e neste trabalho o recorte foi definido em função do tamanho da imagem, de modo empírico. Por exemplo, em imagens de tamanho 256x256 *pixels* foram extraídos recortes de 40x40 *pixels*. O segundo ponto, localização do recorte, não pode ser aleatório, pois um recorte, centralizado em pontos diferentes pode ter conteúdos muito distintos, como na Figura 20.

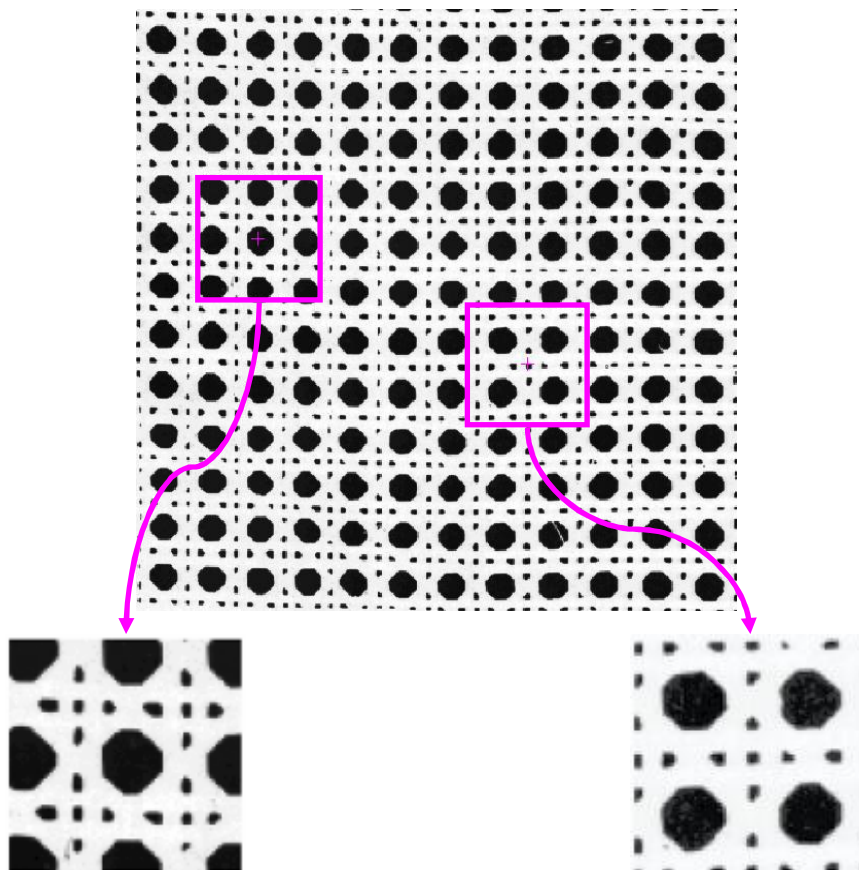


Figura 20 – Exemplo de extração de recortes de uma imagem.

Para solucionar o problema da localização dos recortes de maneira automática empregamos um método de detecção de pontos de interesse, uma área da Visão Computacional bem desenvolvida nos campos de visão estéreo e reconhecimento de objetos. Foi utilizada a técnica de Lowe [61], disponível para MATLAB em [62], que encontra pontos de interesse invariantes à escala e à rotação e, é robusta contra distorções como transformações afins, ruído e mudança de iluminação.

Um exemplo de pontos de interesse detectados se encontra na Figura 21.

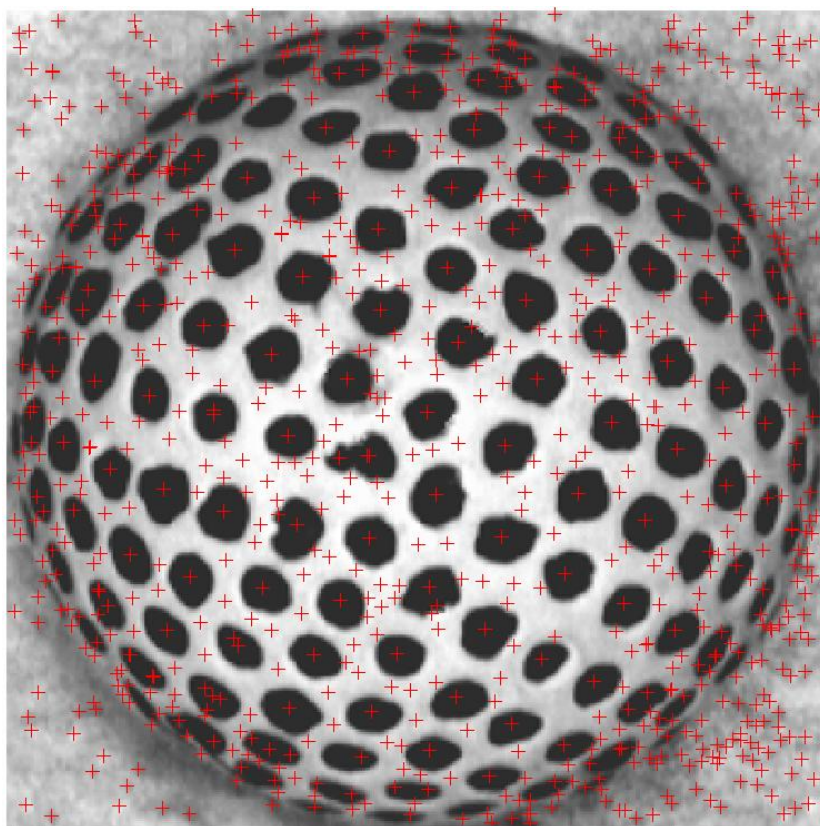


Figura 21 – Exemplo de detecção de pontos de interesse. As marcações “+” representam os pontos de interesse detectados pelo método de Lowe.

Nota-se na Figura 21 que alguns pontos de interesse são localizados no fundo da imagem, portanto, fora do objeto; há ainda outros pontos cuja centralização pode trazer dificuldades na estimativa da transformação afim. Isto justifica o segundo passo do algoritmo de *SFT*, a escolha de que pontos utilizar.

4.4.2 Segmentação dos pontos de interesse

Conforme visto na etapa anterior é necessário selecionar quais os pontos de interesse a utilizar para a extração de recortes. A partir do vetor de medidas retornado pelo método de Lowe realizamos uma clusterização com o algoritmo *K-means* [63]. O número de grupos deve ser alto, algo entre 8 e 15 grupos, uma vez que a quantidade de pontos de interesse detectada em uma imagem de tamanho 400x400 *pixels* beira facilmente a casa dos milhares.

Após a segmentação são selecionados os grupos cujos pontos de interesse serão utilizados. Este processo é manual. O critério é descartar grupos de pontos fora do objeto, grupos muito próximos à borda e aqueles cujo ponto central não aparenta ser o mesmo nos outros pontos de interesse, o que pode causar resultados ruins na estimativa da transformação afim. Na Figura 22 existem dois exemplos de grupos de pontos de interesse, um que representa bem os elementos formadores da textura e outro que não representa.

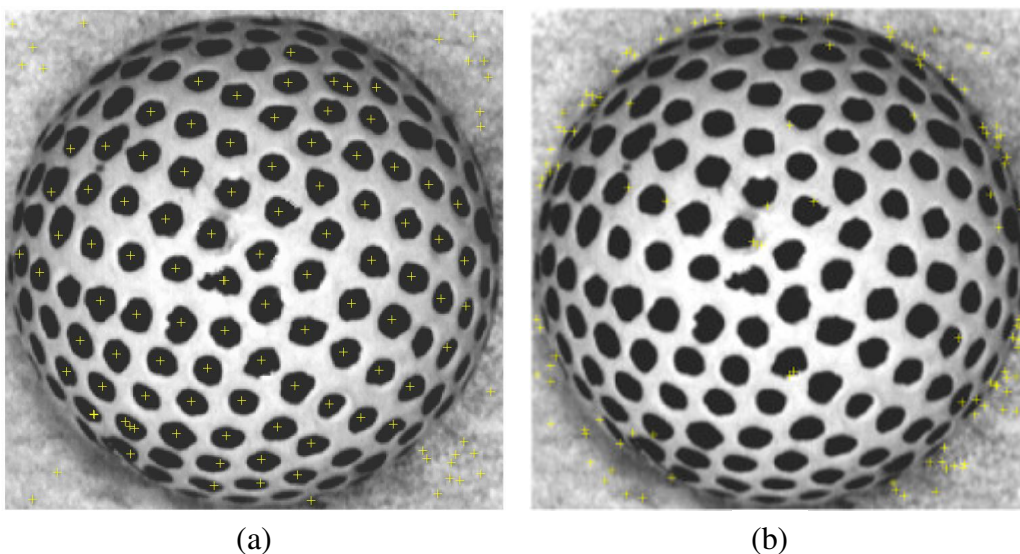


Figura 22 – Grupos de pontos de interesse encontrados pelo método de Lowe e segmentados por *K-means*. (a) grupo de pontos que representam bem os elementos de textura. (b) grupo que representa mal os elementos de textura (concentrados nas bordas da bola de golfe).

4.4.3 Seleção de modelo frontal

Com os pontos de interesse definidos e os recortes extraídos a partir deles, torna-se necessário encontrar um modelo frontal da textura do objeto, para servir de referência na estimativa da transformação afim. Este processo foi realizado de forma manual, selecionando da imagem um recorte que não sofra deformação da projeção, ou seja, possua ângulos *tilt* e *slant* iguais a zero. Nem sempre isso será possível, porém parte-se da premissa que haverá um ponto frontal na imagem. Um exemplo é visto na Figura 23.

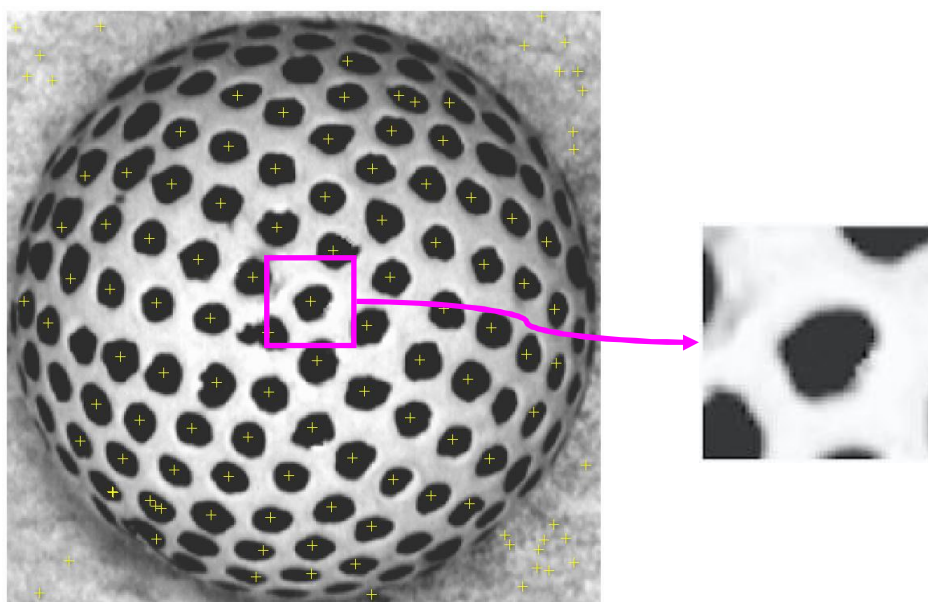


Figura 23 – Exemplo de modelo frontal extraído da imagem.

4.4.4 Estimativa da transformação afim

Com o modelo frontal definido e os demais recortes extraídos da imagem podemos estimar a transformação afim entre ambos. Cada recorte é comparado ao modelo frontal para estimar a transformação afim que relaciona um ao outro. De posse da transformação é possível extrair a orientação de cada recorte com a decomposição por *SVD* mostrada na Seção 3.4.1 do Capítulo 3, gerando assim um mapa de agulhas.

O mapa de agulhas, nesse algoritmo, está incompleto, uma vez que não temos pontos de interesse para cada *pixel* da imagem. É realizada então uma interpolação para preencher a imagem toda.

4.4.5 Reconstrução do modelo 3D

O mapa de agulhas representa a orientação da imagem. Porém, através dele nem sempre é possível ter uma correta visualização da forma do objeto. Torna-se necessário então um modelo de profundidade do objeto, que é o objetivo primário de outras técnicas de *Shape from X*.

É importante notar que no caso de *SFT* o modelo tridimensional não é acurado, uma vez que este possui tamanho relativo ao original, proporcional. E também como não há nenhuma calibração de câmera e a única informação utilizada é a de textura, *SFT* permite apenas uma reconstrução mais grosseira da superfície.

Para reconstruir o modelo tridimensional de profundidade a técnica usada aqui foi a de Kovesi [64], cujo código para MATLAB se encontra em [65]. O método consiste em ajustar funções *Shapelets* de diversas escalas ao mapa de agulhas, levando em consideração inclusive a ambigüidade do *tilt* na construção. Um exemplo do método pode ser visto na Figura 24.

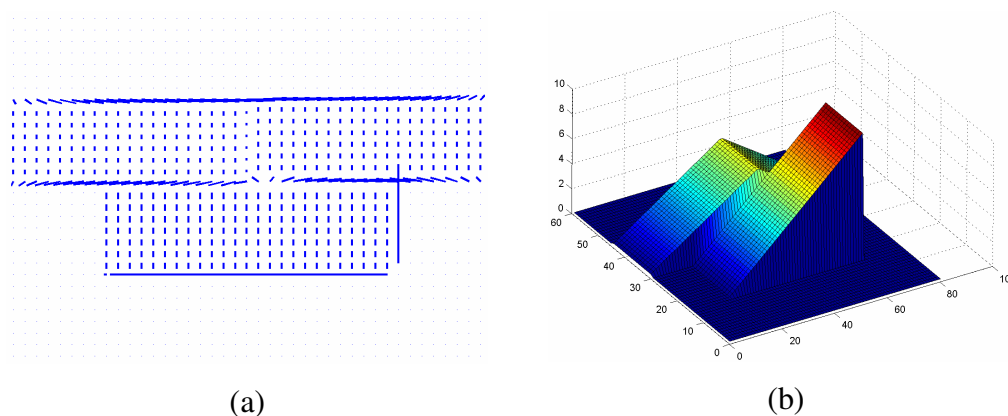


Figura 24 - (a) Mapa de agulhas. (b) Modelo 3D reconstruído usando método de Kovesi.

5 Descrição dos experimentos

Neste capítulo são mostrados os resultados alcançados, com cada uma das técnicas apresentadas anteriormente. O objetivo é avaliar seus pontos fracos e fortes. Para uma melhor compreensão, a descrição dos experimentos foi dividida em etapas, da seguinte forma:

- Decomposição da transformação afim na presença de ruído;
- Estimativa de uma transformação afim entre duas imagens;
- Testes com os algoritmos de *SFT* propriamente ditos.

A primeira etapa avalia como se comporta o método de decomposição de uma matriz de transformação afim na presença de ruído; a segunda avalia os métodos de estimativa da transformação afim que relaciona dois recortes de imagem; e a terceira apresenta resultados dos algoritmos de *SFT*, descritos no capítulo 4.

Todos os experimentos apresentados aqui foram implementados no MATLAB 7 (*Release* 14). Para uma melhor visualização do erro, em alguns casos, foram empregadas superfícies tridimensionais. Isto porque, nas duas primeiras seções de experimentos, são utilizadas diversas combinações dos ângulos *tilt*, *slant* e *theta* para compor uma possível transformação afim. A visualização tridimensional pode ajudar a encontrar combinações críticas de ângulos, onde o erro é maior.

As superfícies foram montadas da seguinte maneira: o eixo *x* representa os valores do ângulo *tilt*, que variou entre 0° e 180°, não sendo necessário ir até 360° uma vez que, como visto no Capítulo 3, a própria limitação da projeção ortográfica faz com que seja esperada uma ambigüidade de 180° no valor do *tilt*; O eixo *y* representa o *slant*, variando de 0° a 80° (como o *slant* equivale a uma compressão num dos eixos da imagem utilizar valores muito próximos ou iguais a 90° não trará bons resultados, uma vez que o co-seno de 90° é igual a 0, representando uma total perda de informação naquele eixo); Tanto o eixo *x* quanto

o y representam os valores reais dos ângulos empregados para construir a transformação afim.

O eixo z representa o erro absoluto dos ângulos. Sempre são geradas duas superfícies para cada experimento, uma com o erro absoluto do *tilt* e outra com o erro do *slant*.

Na Figura 25 temos um exemplo. Nela cada ponto no eixo x - y representa uma configuração de ângulos *tilt* e *slant*, que geram uma transformação afim T_M , conforme a eq. (1). Esta transformação é utilizada no experimento, que retorna uma outra transformação afim T_M' (ruidosa ou estimada, dependendo do experimento). O erro calculado (eixo z), nesse exemplo, é o valor da diferença absoluta entre o ângulo original (na figura, *tilt*) e o obtido da matriz T_M' .

Note que, como foi visto na eq. (1) do Capítulo 3, para compor uma transformação afim ainda é necessário o ângulo *theta*. Como não era possível manter a visualização com mais um eixo, optou-se por fixar o valor de *theta* para cada superfície. A fim de avaliar a contribuição de *theta* no erro foram construídas superfícies para diferentes valores do mesmo, como 0° , 30° , 45° , 60° e 90° .

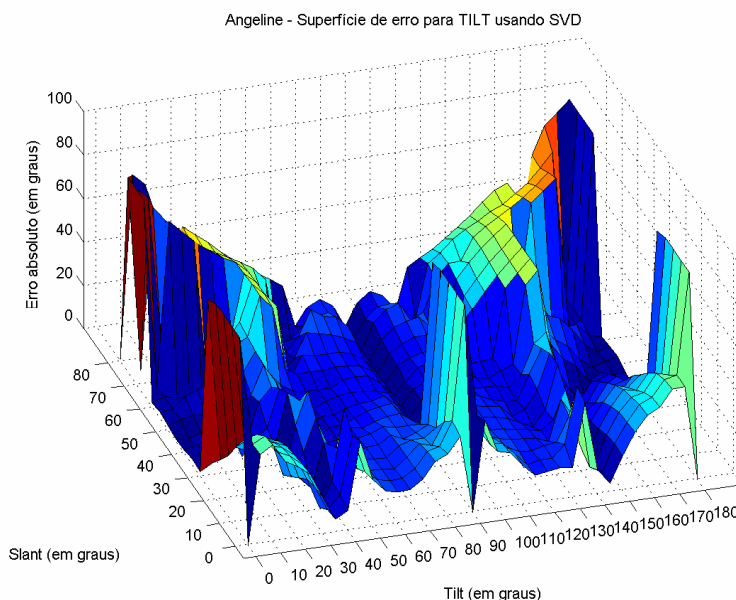


Figura 25 - Exemplo de superfície de erro para o *tilt*. *Theta* é fixo para todas as combinações.

5.1.

Decomposição da transformação afim na presença de ruído

O presente experimento tem como objetivo avaliar a decomposição de uma transformação afim pelo método *Singular Value Decomposition* (SVD), na presença de ruído. Através da decomposição obtemos os ângulos *tilt* e *slant*, usando as eqs. (3) e (5) do Capítulo 3. Entretanto, os métodos aqui apresentados sobre estimativa da transformação afim entre duas imagens não são perfeitos e o resultado pode conter algum ruído. A intenção é avaliar o erro nas estimativas dos ângulos produzidos pelo método SVD quando a estimativa da matriz de transformação afim está afetada por ruído, uma vez que na ausência de ruído a decomposição encontra valores exatos. O procedimento é descrito a seguir.

Partindo-se dos valores de *tilt*, *slant* e *theta* produz-se a correspondente matriz de transformação afim T_M , eq. (1). Acrescenta-se em seguida um ruído aleatório Gaussiano η , gerando assim uma nova matriz T , como no exemplo abaixo:

$$T = T_M + \eta$$

$$T = \begin{bmatrix} 0.8660 & 0.5000 \\ -0.4924 & 0.8529 \end{bmatrix} + \begin{bmatrix} 0.0337 & -0.0030 \\ 0.0019 & -0.0263 \end{bmatrix}$$

$$T = \begin{bmatrix} 0.8998 & 0.4970 \\ -0.4905 & 0.8265 \end{bmatrix}$$

Aplica-se então SVD nessa nova matriz e os valores de ângulo encontrados são comparados aos originais. Este procedimento foi repetido para uma gama de valores de *tilt*, *slant* e *theta*, gerando as superfícies de erro.

O ruído η acrescentado provém de uma distribuição normal, com média 0, e os seguintes valores de variâncias: 0.00001, 0.0001, 0.001, 0.01, 0.1 e 1. Em todos os casos foi feita uma média de 20 experimentos.

Inicialmente, é apresentado o resultado na Tabela 2. A tabela é formada pela média de todos os erros calculados, onde constam:

Erro médio: média do módulo do erro (diferença) dos ângulos encontrados (em graus).

Erro máximo: maior diferença absoluta encontrada (em graus).

Erro mínimo: menor diferença absoluta encontrada (em graus).

Desvio-padrão do erro: desvio-padrão da diferença absoluta (em graus).

ΔT médio: média do somatório das diferenças absolutas entre os elementos da matriz T_M original e a matriz ruidosa T .

O somatório das diferenças absolutas entre duas matrizes foi calculado segundo a equação:

$$\begin{aligned}\Delta T &= \sum |T_M - T| \\ \Delta T &= \sum \begin{bmatrix} t_{M11} - t_{11} & t_{M12} - t_{12} \\ t_{M21} - t_{21} & t_{M22} - t_{22} \end{bmatrix} \\ \Delta T &= |t_{M11} - t_{11}| + |t_{M12} - t_{12}| + |t_{M21} - t_{21}| + |t_{M22} - t_{22}|\end{aligned}\quad (37)$$

Conforme mencionado anteriormente, o valor de θ é fixo. Para verificar como o valor de θ influencia o erro, foram geradas superfícies para cinco valores diferentes deste ângulo (0°, 30°, 45°, 60° e 90°). Observou-se que os erros sofreram poucas mudanças em função da variação de θ . Por isso, optou-se por apresentar na Tabela 2 apenas os valores médios obtidos nos experimentos para os valores de θ considerados. No Apêndice I encontra-se a tabela completa, discriminando os resultados para cada valor de θ individualmente.

variância		erro médio	erro máx.	erro mín.	desvio-padrão do erro	ΔT médio
0.00001	<i>slant</i>	0,6	4,6	0,1	1,0	0,00966
	<i>tilt</i>	4,0	52,3	0,1	8,8	
0.0001	<i>slant</i>	1,4	7,9	0,3	1,8	0,02993
	<i>tilt</i>	6,2	51,3	0,4	9,9	
0.001	<i>slant</i>	3,8	14,2	1,3	3,1	0,10375
	<i>tilt</i>	9,5	49,6	1,2	10,6	
0.01	<i>slant</i>	10,0	26,5	3,8	6,1	0,33076
	<i>tilt</i>	15,7	50,2	3,3	10,2	
0.1	<i>slant</i>	22,8	45,8	7,6	10,2	1,00303
	<i>tilt</i>	23,2	50,2	9,0	8,2	
1	<i>slant</i>	37,7	60,5	23,0	9,4	3,27798
	<i>tilt</i>	30,3	52,8	18,9	7,4	

Tabela 2 - Erros (em graus) da decomposição da transformação afim por SVD na presença de ruído.

Podemos destacar da Tabela 2 duas conclusões:

Para o *slant*: o erro aumenta gradativamente de acordo com o ruído, porém de maneira uniforme, pois tanto o erro máximo quanto o desvio-padrão acompanham o crescimento. Para ruídos altos (variância 0.1 e 1) o *slant* passa a ter erros mais grosseiros que o *tilt*.

Observando a superfície de erro outros detalhes vêm à tona: na Figura 26 e na Figura 27 pode-se observar que o erro é mais grosseiro para valores muito baixos (até 10°, pior em 0°) do *Slant* (eixo *y*). Isso se deve ao fato da matriz de transformação afim nesses casos se aproximar de uma matriz de rotação, onde a mais branda perturbação faz com que o *SVD* retorne valores diferentes de zero para o ângulo.

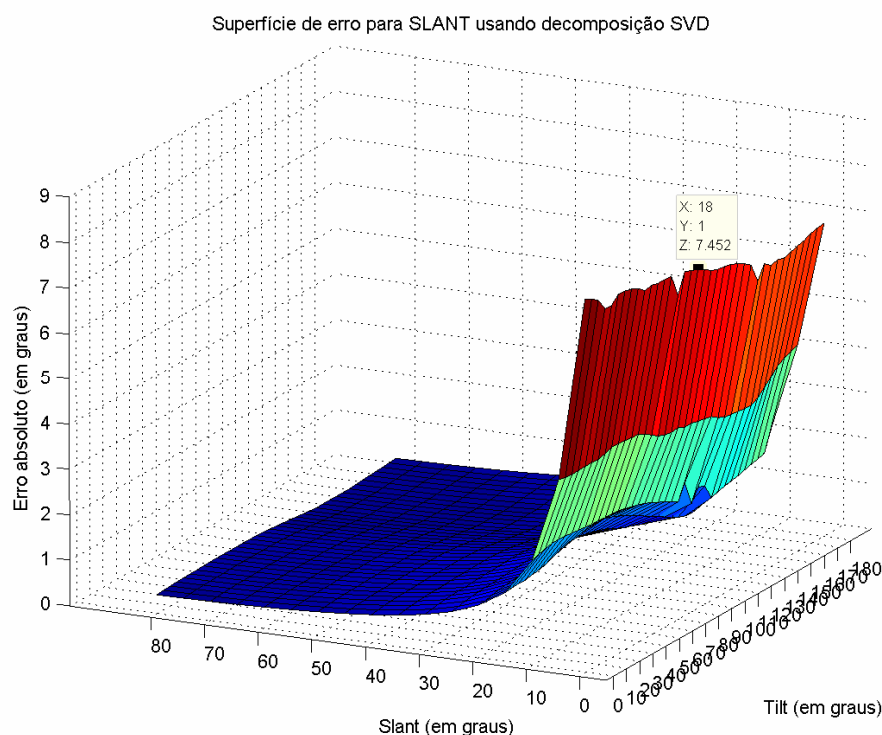


Figura 26 - Superfície de erro do *slant* para decomposição na presença de ruído. Variância = 0.0001, $\theta = 0^\circ$.

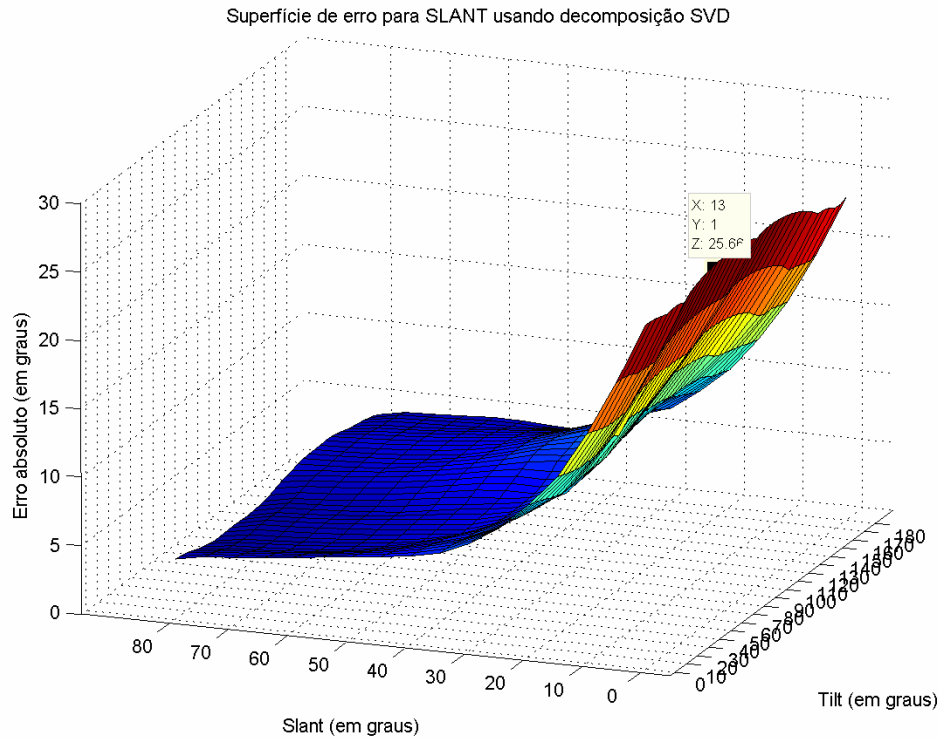


Figura 27 - Superfícies de erro do *slant* para decomposição na presença de ruído. Variância = 0.01, $\theta = 0^\circ$.

Para o *tilt*: o erro se comporta da seguinte maneira: enquanto o erro médio cresce gradualmente, com valores não muito distantes da média do *slant*, o pico apresenta erros grosseiros mesmo com ruído ínfimo, conforme se pode verificar o ΔT na Tabela 2, para a menor variância, 0.00001. Isto também pode ser verificado pelo desvio-padrão, mais elevado que no *slant*. Observando as duas superfícies de erro para o *tilt*, na Figura 28 e na Figura 29, vê-se que o erro grosseiro acontece para valores de Tilt (eixo x) iguais a 0° , 90° e 180° , quando o Slant (eixo y) é baixo. Isso ocorre pois, como foi mencionado anteriormente, a estimativa para o *slant* é muito sensível a ruído para valores de *slant* baixo.

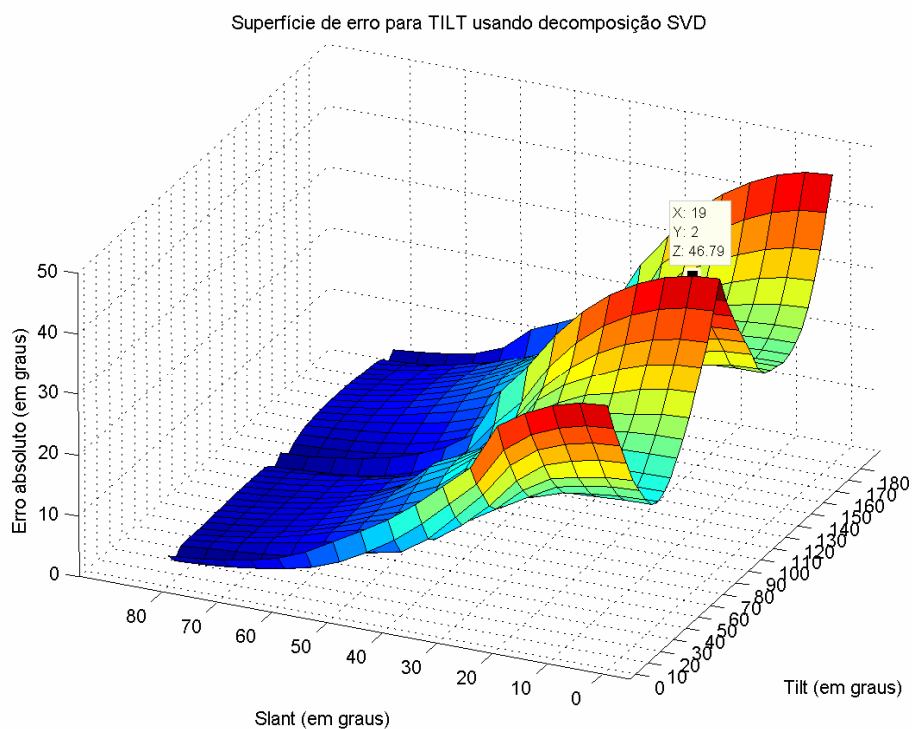


Figura 28 - Superfícies de erro do *tilt* para decomposição na presença de ruído. Variância = 0.0001, $\theta = 0^\circ$.

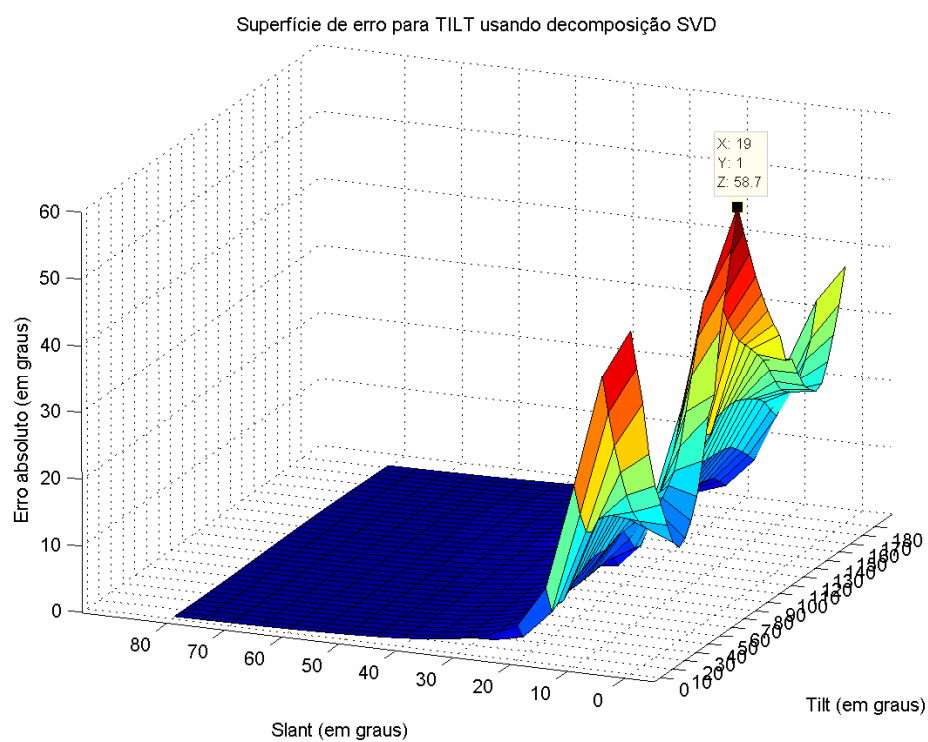


Figura 29 - Superfícies de erro do *tilt* para decomposição na presença de ruído. Variância = 0.01, $\theta = 0^\circ$.

5.2

Estimativa da transformação afim

Nesta etapa são apresentados os experimentos com os dois métodos de estimativa da transformação afim apresentados no Capítulo 4, o de Loh e Zisserman e o de Rosenholtz e Malik.

Somente esses métodos são avaliados, pois possuem algoritmos específicos para a estimar a transformação entre dois recortes extraídos da imagem. Já o método de Super e Bovik não é voltado para recortes mas sim para a imagem como um todo.

O procedimento adotado nos experimentos é descrito abaixo:

1) Combinando *tilt*, *slant* e *theta* geramos a correspondente matriz de transformação afim T_M com a eq. (1). O conjunto de valores possíveis para os ângulos é o mesmo usado no experimento anterior para gerar a superfície de erros.

2) Para cada imagem I do banco de imagens é aplicada a transformação afim T_M , gerando assim uma nova imagem transformada IT .

3) Extraí-se de cada imagem (I e IT) um recorte. Os dois recortes são submetidos ao algoritmo, que estima a transformação afim que explica a relação geométrica entre os recortes, que chamamos de T' .

4) De posse da estimativa T' são calculados os erros: diferença absoluta entre as matrizes segundo a eq. (37); diferença absoluta entre os ângulos *tilt* e *slant* originais (obtidos de T_M); e os estimados (obtidos de T' com as eqs. (3) e (5)).

5) Aplica-se à imagem IT a transformação inversa da estimada, gerando um recorte I' que é então comparado visualmente com I .

Na Figura 30 há um diagrama que ilustra o procedimento. Nota-se que a parte escura nos quatro cantos de I' é resultante da perda de informação ao realizar uma rotação na imagem. Cabe ressaltar que este efeito não ocorre na transformação de I para IT pois parte-se de uma imagem maior, sendo I um recorte.

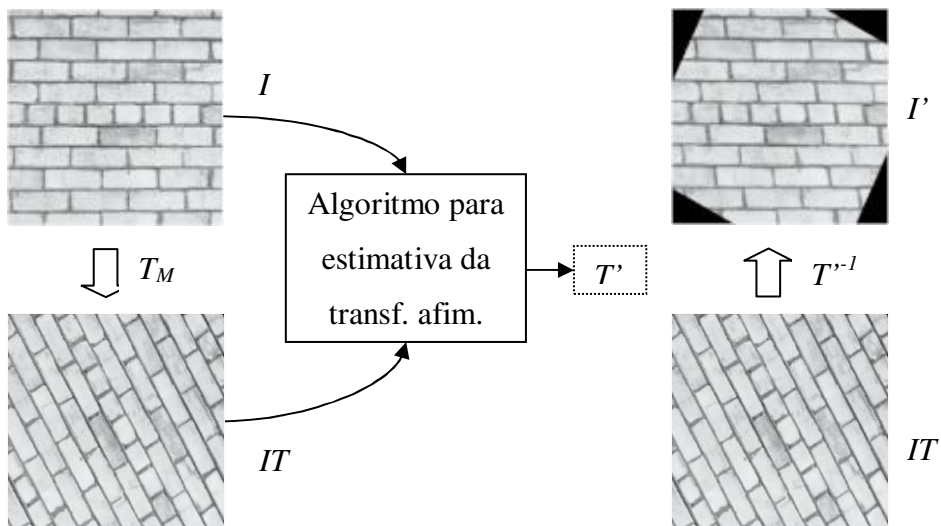


Figura 30 - Diagrama da avaliação visual do erro do método.

Banco de dados

As imagens empregadas nesse teste foram selecionadas do banco de texturas de Brodatz [66] e podem ser vistas na Figura 31.

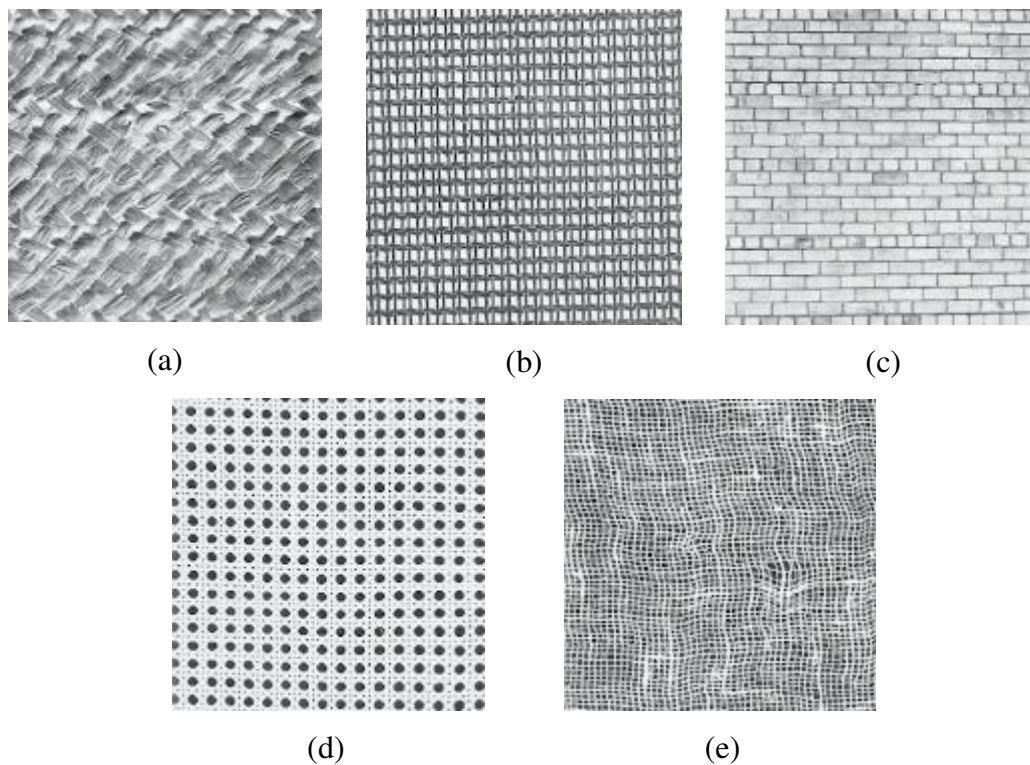


Figura 31 - Texturas de Brodatz utilizadas nos experimentos. (a) D18, (b) D20, (c) D95, (d) D101 e (e) D103.

Todas as imagens do banco possuem o tamanho original de 643x643 *pixels* e foram reduzidas para 400x400 *pixels*. O tamanho do recorte em todos os casos foi de 128x128 *pixels*, extraídos a partir do centro da imagem.

Os resultados consolidados para o método de Rosenholtz e Loh são mostrados na Tabela 3 e Tabela 4, respectivamente, onde a média dos erros dos ângulos, para todo o conjunto de transformações afins aplicadas, é apresentada. Também são mostrados os erros mínimos e máximos e seus desvios-padrão.

imagem		erro médio	erro máx.	erro mín.	desvio-padrão do erro
D20	<i>slant</i>	30,1	71,1	0,0	20,8
	<i>tilt</i>	34,9	89,0	0,0	23,7
D18	<i>slant</i>	22,6	72,3	0,0	16,4
	<i>tilt</i>	26,8	85,0	0,0	19,5
D95	<i>slant</i>	23,4	66,4	0,0	17,6
	<i>tilt</i>	29,8	88,9	0,0	21,0
D101	<i>slant</i>	26,8	72,7	0,0	20,2
	<i>tilt</i>	34,9	89,6	0,1	23,2
D103	<i>slant</i>	23,2	68,8	0,0	15,6
	<i>tilt</i>	31,1	89,5	0,0	22,9

Tabela 3 - Erros calculados (em graus) para o método de Ruth Rosenholtz.

imagem		erro médio	erro máx.	erro mín.	desvio-padrão do erro
D20	<i>slant</i>	11,2	54,4	0,0	9,2
	<i>tilt</i>	15,8	74,9	0,0	15,3
D18	<i>slant</i>	17,2	66,3	0,0	15,2
	<i>tilt</i>	18,9	79,1	0,0	19,1
D95	<i>slant</i>	9,8	35,7	0,0	7,2
	<i>tilt</i>	15,6	76,7	0,1	16,5
D101	<i>slant</i>	9,0	29,7	0,0	6,9
	<i>tilt</i>	11,5	68,5	0,0	14,5
D103	<i>slant</i>	11,9	57,3	0,0	11,4
	<i>tilt</i>	16,5	72,7	0,0	15,6

Tabela 4 - Erros calculados (em graus) para o método de Angeline Loh.

Observa-se nas tabelas o desempenho superior do método de Loh, que apresenta média e desvio-padrão menores que o de Rosenholtz, para todas as imagens. Isso era previsto pois o método de Rosenholtz admite um pequeno desvio (ΔT_r) na matriz de transformação afim T em relação a matriz identidade, uma condição que freqüentemente não se verifica na prática, como pode ser visto

na Figura 32, onde é mostrada a diferença absoluta (ΔT) entre todas as matrizes T_M geradas e a matriz identidade.

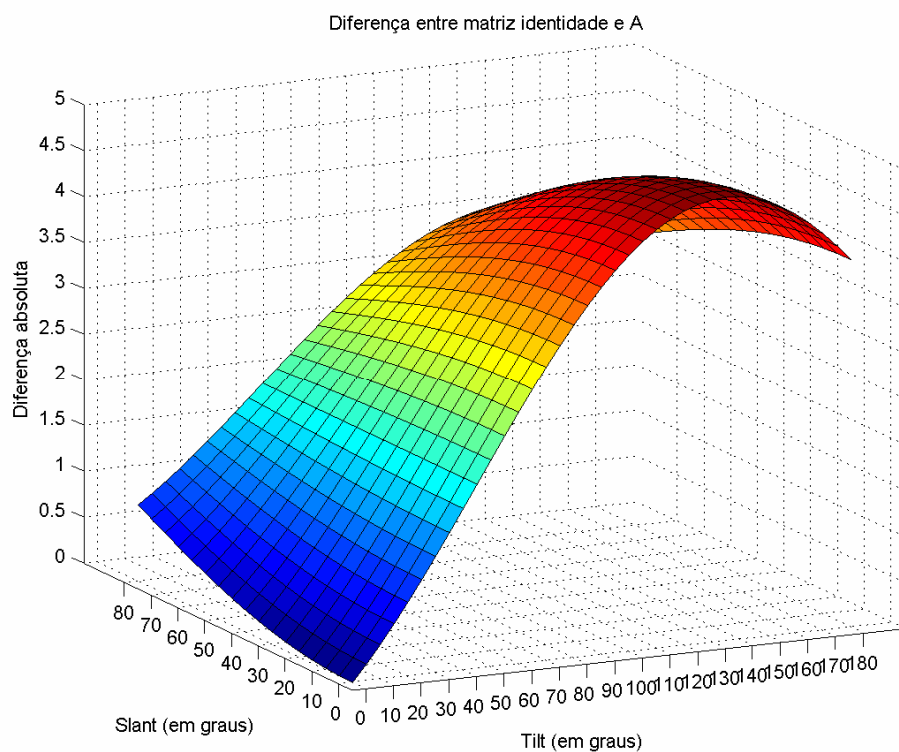


Figura 32 - Diferença absoluta entre transformações afim geradas (T_M) e a matriz identidade.

Na figura anterior é mostrado que para a maioria das transformações aqui testadas a diferença absoluta para a matriz identidade é elevada.

Podemos observar também duas superfícies geradas do ΔT para cada método, na Figura 33 e Figura 34.

Na Figura 33 observa-se claramente a limitação do método de Rosenholtz. Os valores de ΔT são elevados em quase toda a faixa de valores de *tilt* e *slant*.

A Figura 34 mostra o resultado da avaliação para método de Loh. Fica clara a superioridade deste método sobre o método de Rosenholtz.

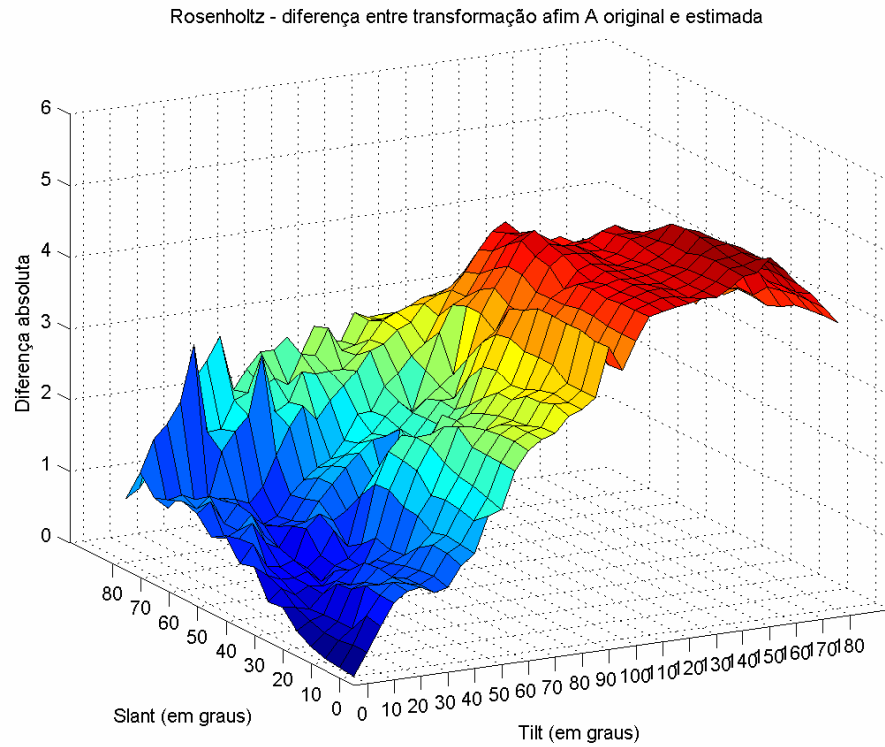


Figura 33 - Diferença entre matriz original T_M e estimada T_R' para a textura D95 usando o método de Rosenholtz.

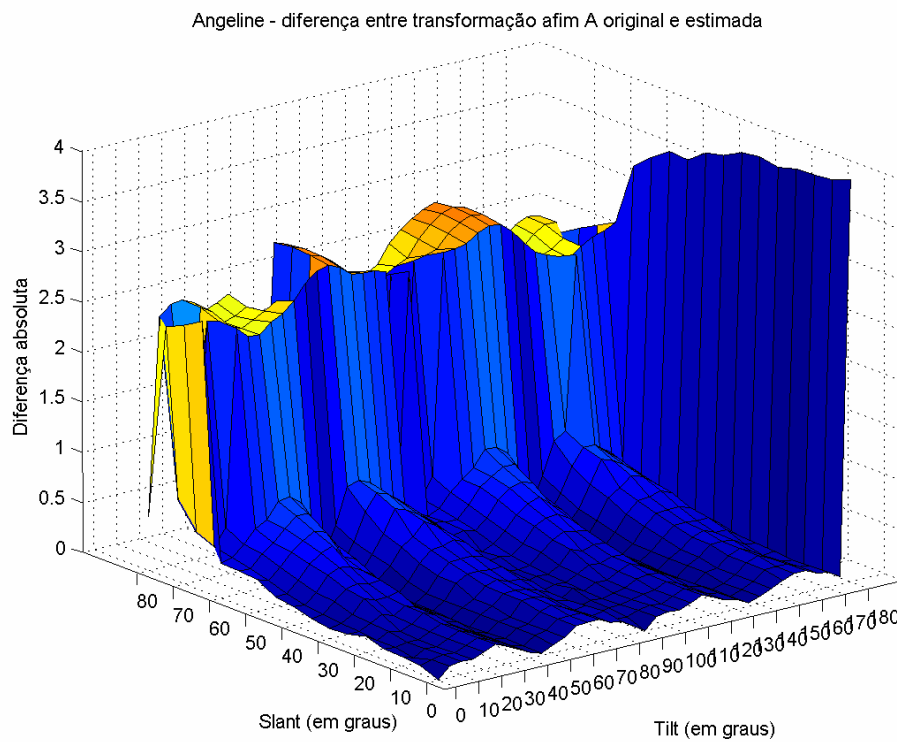


Figura 34 - Diferença entre matriz original T_M e estimada T' para a textura D95 usando o método de Loh.

As superfícies de erro (ΔT) mostradas anteriormente também contribuem para uma melhor avaliação da performance dos métodos discutidos. Isto porque, como visto anteriormente, a decomposição de uma transformação afim em valores de ângulo *tilt* e *slant* está sujeita a erros grosseiros em algumas situações, dado que a decomposição *SVD* é sensível a ruído. Portanto, ao apresentar somente o erro dos ângulos acaba-se por fazer uma avaliação tendenciosa, que desconsideraria o erro oriundo da decomposição.

Nas figuras a seguir são mostradas algumas superfícies de erro para o *tilt* e *slant* de ambos os métodos, com base numa das texturas de Brodatz.

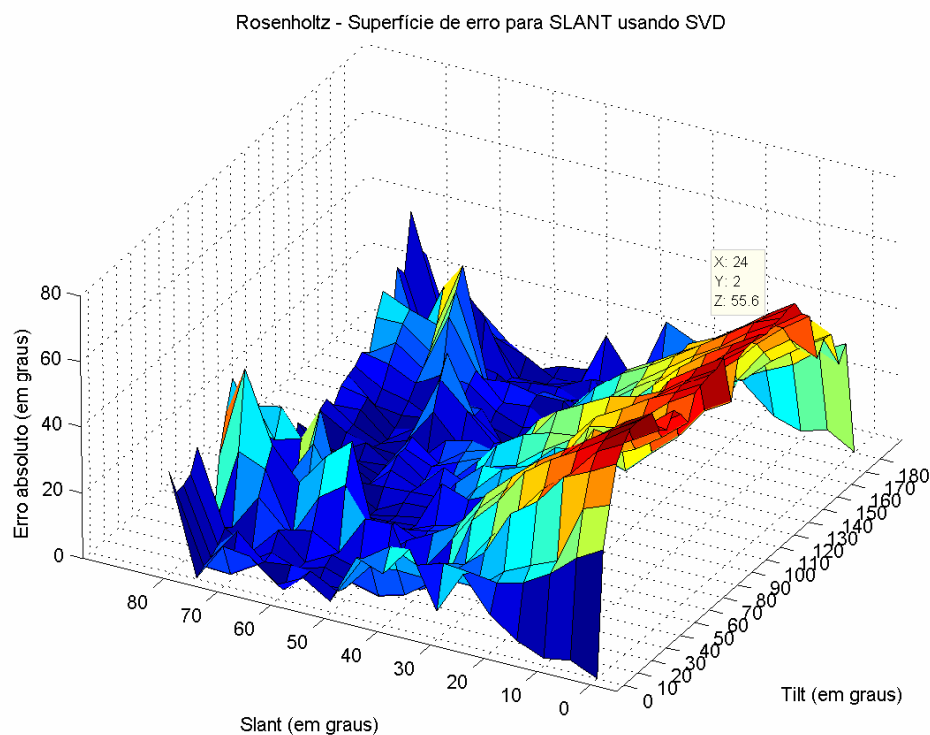


Figura 35 - Superfície de erro para o *slant* para a textura D95, usando o método de Ruth Rosenholtz. $\theta = 0^\circ$.

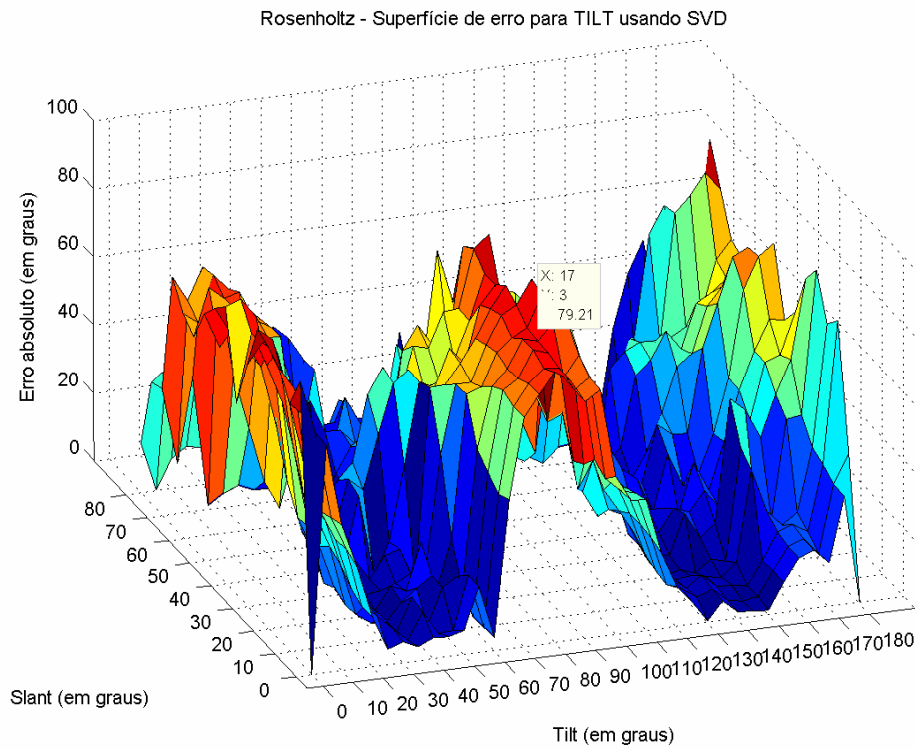


Figura 36 - Superfície de erro para o *tilt* para a textura D95, usando o método de Ruth Rosenholtz. $\Theta = 0^\circ$.

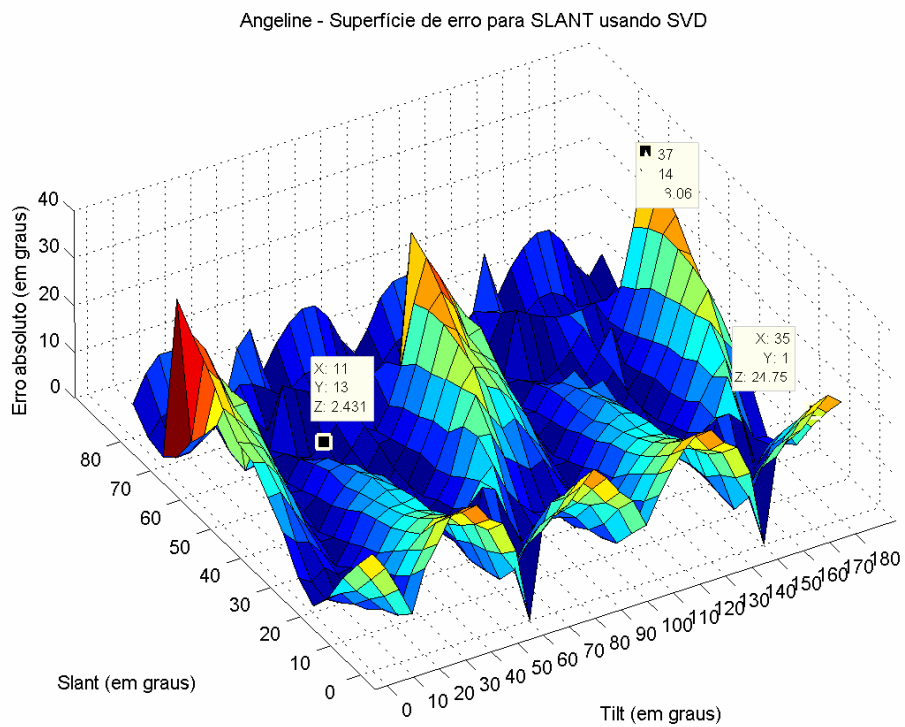


Figura 37 - Superfície de erro para o *slant* para a textura D95, usando o método de Angeline Loh. $\Theta = 30^\circ$.

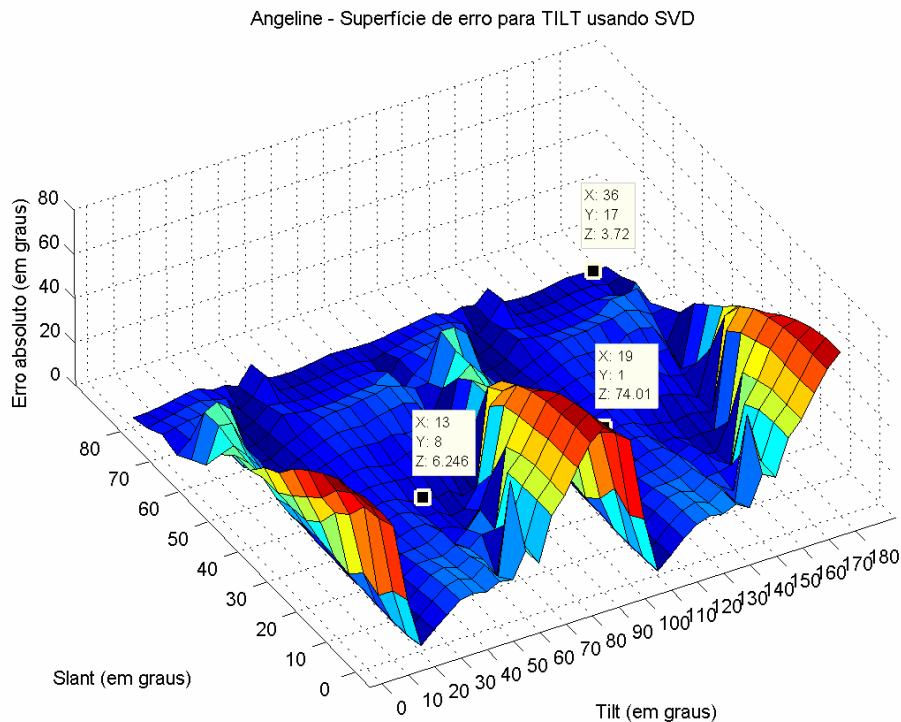


Figura 38 - Superfície de erro para o *tilt* para a textura D95, usando o método de Angeline Loh. $\theta = 30^\circ$.

Como um complemento, na Figura 39 são mostrados alguns exemplos de imagens, suas transformadas e conseqüentes transformadas inversas (com a estimativa T' da transformação afim). Em (a) temos as imagens originais, em (b) as imagens após a aplicação de uma transformação afim qualquer. Aplica-se o método de Loh para estimar a transformação afim entre (a) e (b). O resultado final é mostrado em (c), onde foi aplicada a inversa da transformação afim estimada. Mostra-se aqui que o método de Loh apresenta bons resultados, com as imagens em (c) assemelhando-se muito às imagens originais (a), a não ser pelos quatro cantos, onde a informação foi perdida devido à rotação.

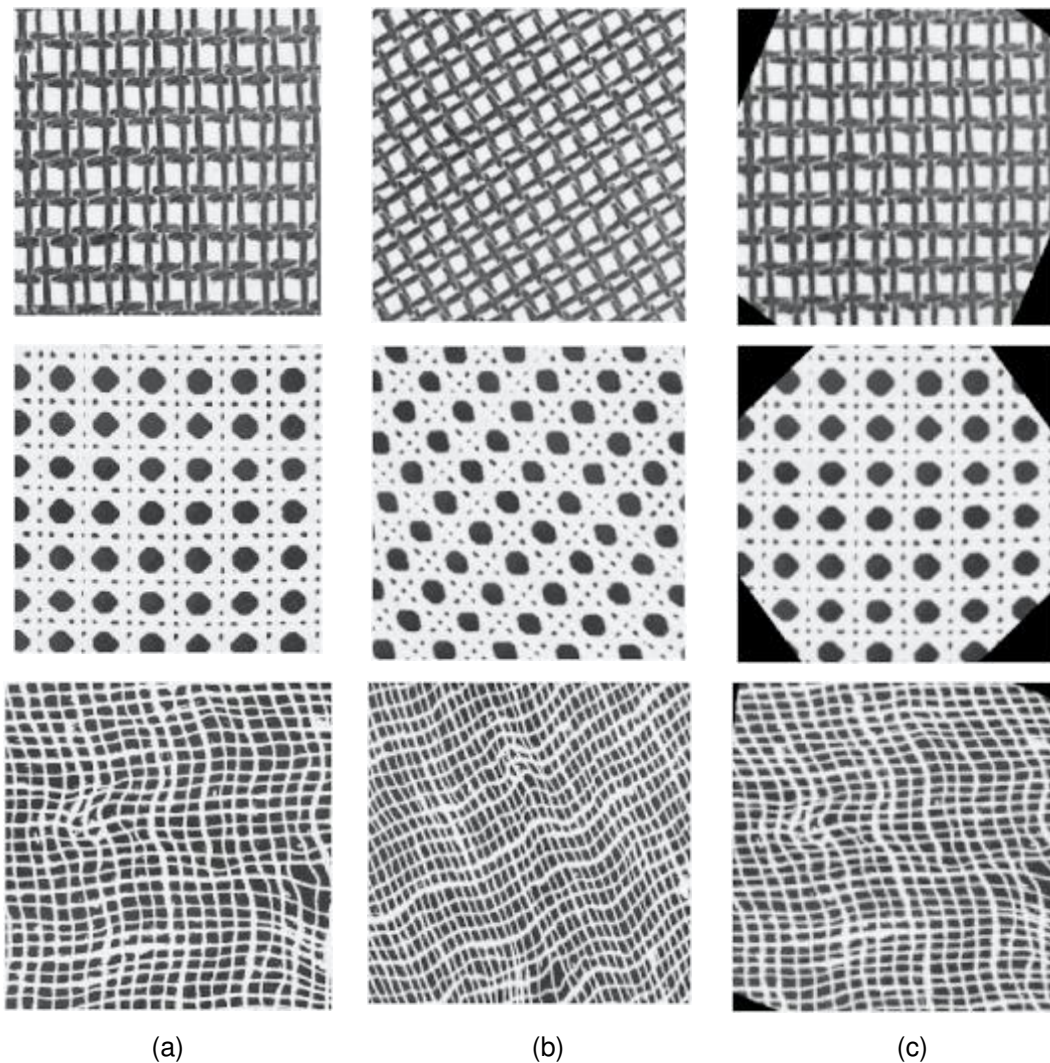


Figura 39 - Exemplos de estimativas da transformação afim. (a) Recorte original. (b) Recorte com uma transformação afim A aplicada. (c) Aplicação da transformada inversa \hat{A} (estimada pelo método de Loh) sobre o recorte (b).

5.3 Reconstrução da forma de superfícies

Nesta seção serão apresentados os resultados de dois algoritmos completos de *SFT*, cuja teoria foi apresentada no Capítulo 4.

O primeiro é o de Super enquanto que o segundo corresponde ao algoritmo apresentado ao final do Capítulo 4, onde aqui optou-se por usar a estimativa da transformação afim de Loh, uma vez que o experimento anterior apontou clara superioridade com relação à estimativa de Rosenholtz.

Os resultados foram divididos em três subseções:

- Resultados do algoritmo de Super;

- Resultados do algoritmo de *SFT* com estimativa da transformação afim de Loh.
- Estudo de caso: reconhecimento do estado de carregamento de uma colheitadeira.

Nos dois primeiros foram empregadas três imagens de objetos distintos: um cilindro (sintético), uma bola de golfe (real) e uma superfície irregular (sintética). Todas foram segmentadas manualmente para tratar somente o objeto de interesse.

Para cada imagem são ilustrados: a imagem propriamente dita com a indicação do ponto frontal (detectado automaticamente no primeiro algoritmo e determinado visualmente no segundo); o mapa de agulhas encontrado pelo método; o modelo 3D encontrado usando a técnica citada na Seção 4.4.5; e o mesmo modelo 3D encontrado porém com a imagem original sobreposta.

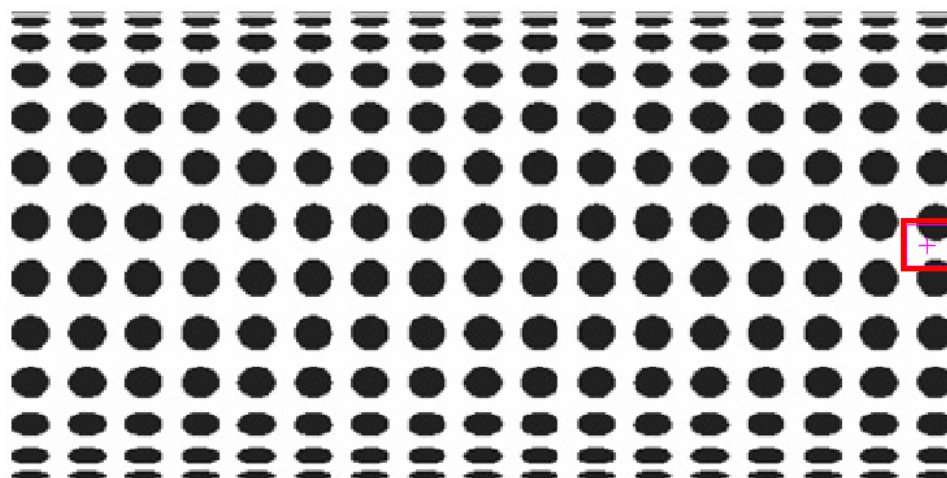
Cabe lembrar que os modelos 3D encontrados possuem tamanho proporcional ao modelo original, não sendo possível em um algoritmo *SFT* obter distâncias precisas, somente relativas. Uma vez que, em nenhuma imagem aqui apresentada possui-se um modelo numérico preciso de suas medidas no mundo real e, aliado ao fato que *SFT* produz aproximações imprecisas de superfícies, tem-se uma análise subjetiva, baseada na interpretação visual dos modelos.

5.3.1

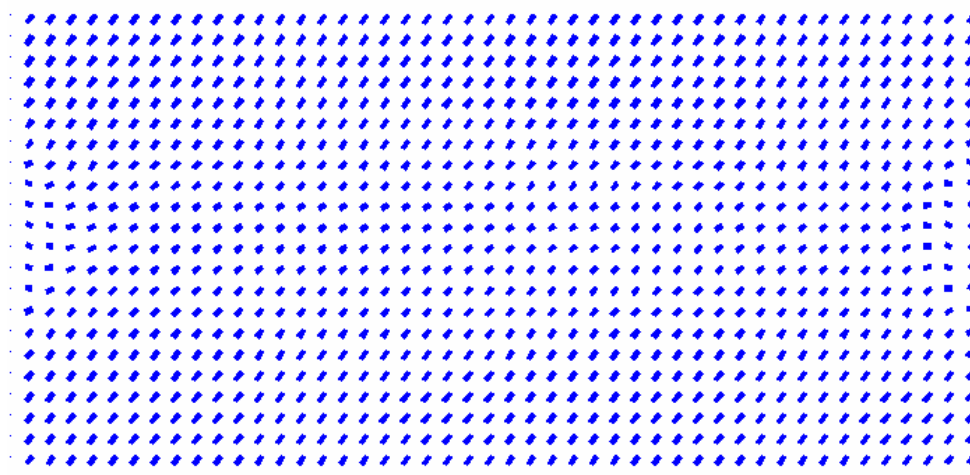
Método de Super e Bovik

Foram reproduzidos aqui parte dos experimentos indicados no artigo original. Foram empregadas 63 configurações de filtros de Gabor, usando os seguintes valores de frequência radial, F : 12, 16.97, 24, 33.94, 48, 67.88 e 96 ciclos/imagem. Valores de orientação, Φ : -70° , -50° , -30° , -10° , $+10^\circ$, $+30^\circ$, $+50^\circ$, $+70^\circ$, $+90^\circ$. A largura de banda B foi de uma oitava. A gaussiana, isotrópica (simétrica), ou seja, o aspecto κ igual a unidade na eq. (9). Com os valores de F , Φ , B e κ podemos calcular o valor de γ na eq. (14). Todas as imagens usadas aqui foram redimensionadas para o tamanho 256×256 pixels, para otimizar a transformada de Fourier, uma vez que os filtros de Gabor foram utilizados em sua versão no domínio da frequência, eq. (13).

Na Figura 40 e Figura 41 encontram-se os resultados para o cilindro sintético apresentado no primeiro capítulo deste trabalho.



(a)

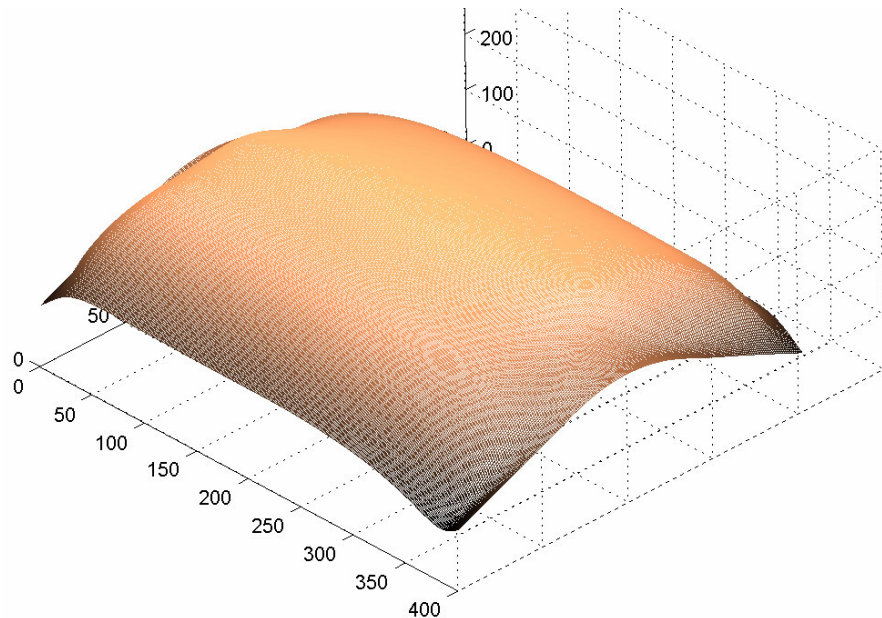


(b)

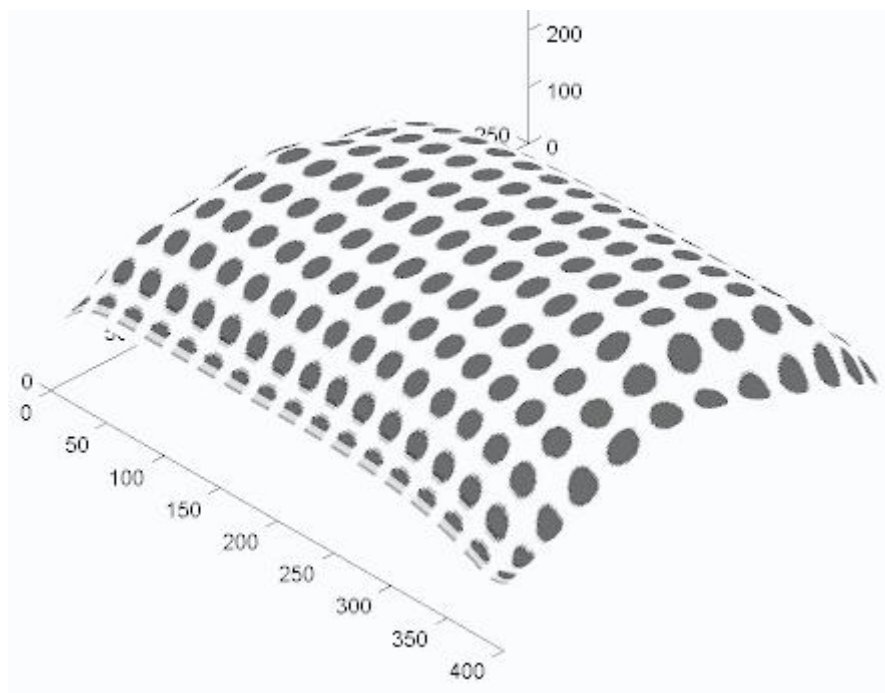
Figura 40 – (a) imagem sintética de um cilindro texturizado com o ponto frontal encontrado demarcado.(b) Mapa de agulhas obtido pelo método de Super.

A Figura 40(b) apresenta o mapa de agulhas encontrado pelo método, nota-se que o *slant* (representado pelo comprimento das agulhas) é bem próximo ao real, aumentando a medida que se caminha para as bordas. O *tilt* não possui resultado tão acurado. Tal fato se deve ao ponto frontal encontrado pelo método se encontrar muito próximo à borda. Outro ponto que influencia o resultado é a largura da gaussiana empregada para suavizar os momentos; nos experimentos foi usado um critério tal que a largura fosse equivalente a um décimo do tamanho da imagem.

Na Figura 41 encontram-se as superfícies tridimensionais reconstruídas. Pode-se observar aqui uma aproximação da superfície coerente, apenas com pequenas deformações nas bordas.



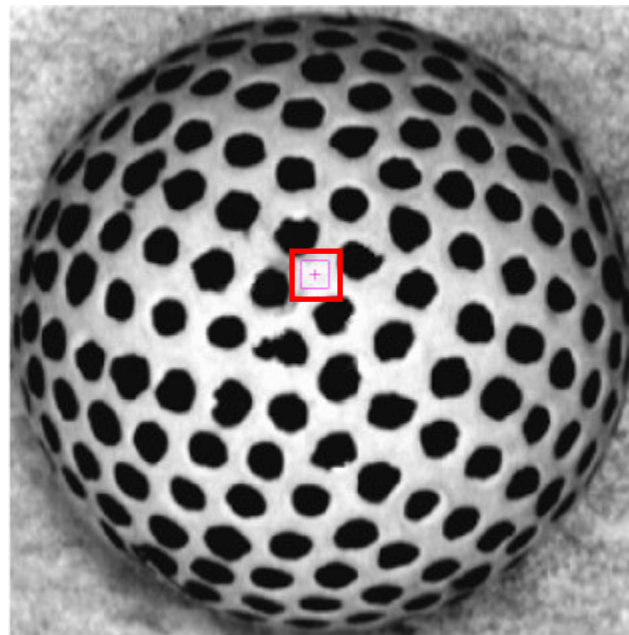
(a)



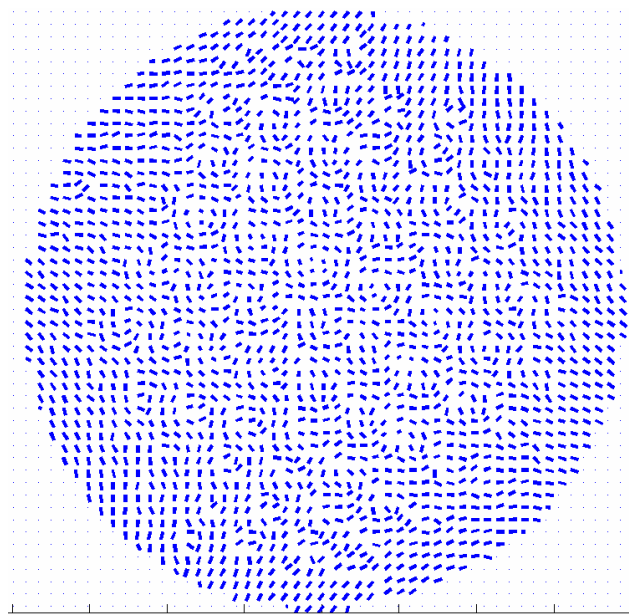
(b)

Figura 41 - (a) Modelo 3D a partir do mapa de agulhas da figura anterior. (b) Modelo 3D com a textura sobreposta.

Na Figura 42 encontra-se o resultado para a bola de golfe:



(a)



(b)

Figura 42 - (a) imagem real de uma bola de golfe com ponto frontal indicado. (b) mapa de agulhas obtido pelo método de Super.

A Figura 42(b) mostrou o mapa de agulhas encontrado. O mapa em questão apresenta inúmeras inconsistências ao centro da imagem, apesar de encontrar orientações corretas em determinados pontos. Essas inconsistências aparecem principalmente por dois fatores:

O primeiro é o fato de que esta é uma imagem real, sujeita a variações de iluminação e os elementos de textura (as cavidades da esfera) foram pintados de forma a melhorar o resultado da filtragem com os filtros de Gabor. Entretanto, como se pode notar na imagem, esta pintura contém falhas, tornando os elementos não-uniformes.

O segundo é o mesmo que do experimento anterior: a largura da gaussiana empregada. Em testes realizados, foi visto que quanto maior a largura da gaussiana menores as inconsistências. Porém, a medida que os momentos são demasiadamente suavizados, as estimativas da orientação tornam-se ruins, perdendo detalhes do objeto.

Na Figura 43 e Figura 44 são vistos os modelos tridimensionais obtidos. Devido às inconsistências na região central da esfera o modelo se apresenta ruidoso. Ainda assim é possível uma aproximação da superfície, apesar de grosseira.

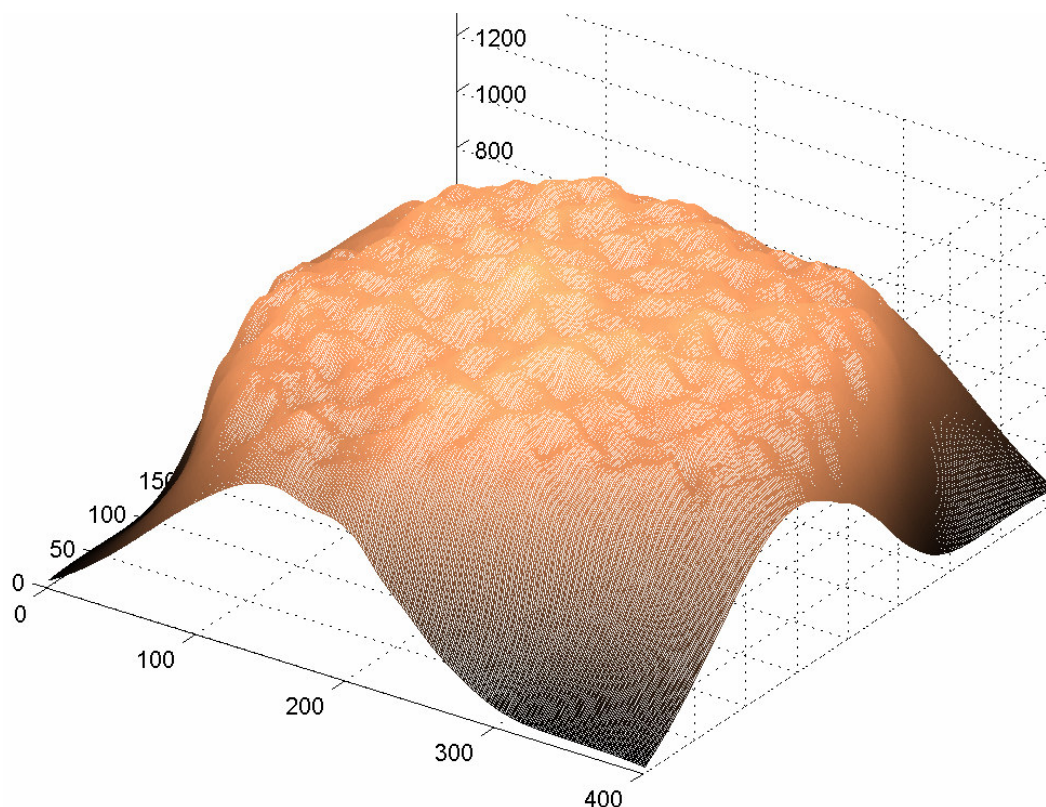


Figura 43 - Modelo 3D a partir do mapa de agulhas (estimado pelo método de Super) da bola de golfe.

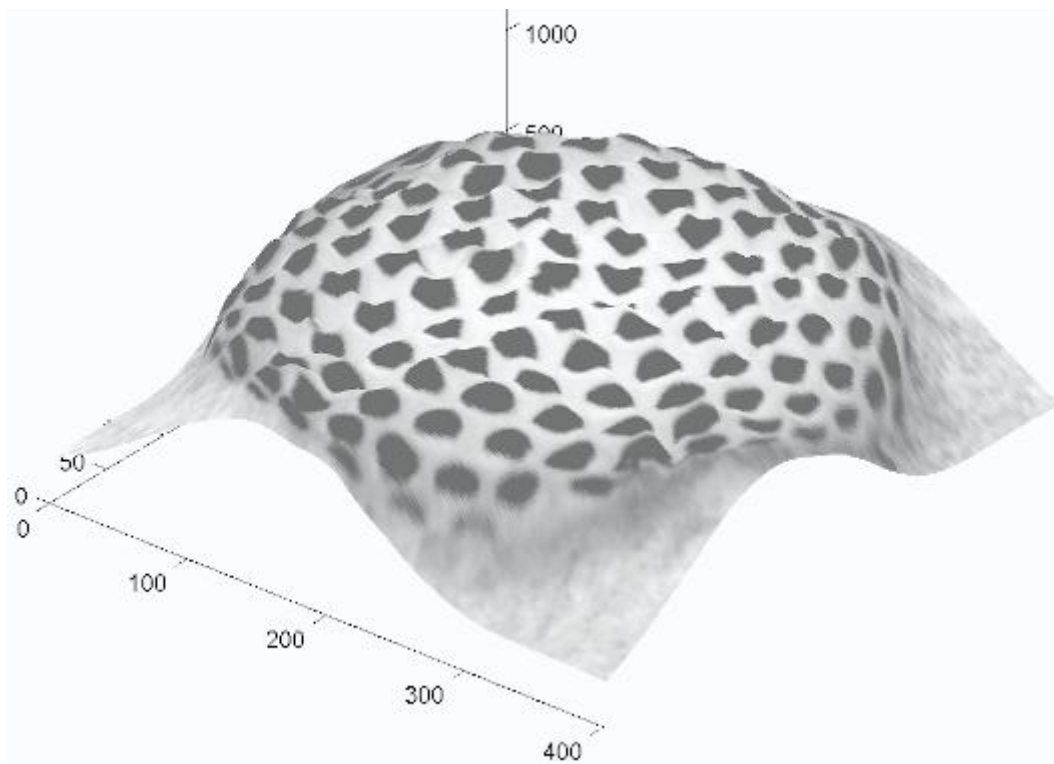


Figura 44 - Modelo 3D da figura anterior com a textura sobreposta.

A Figura 45 e Figura 46 a seguir apresentam um teste com uma superfície irregular.

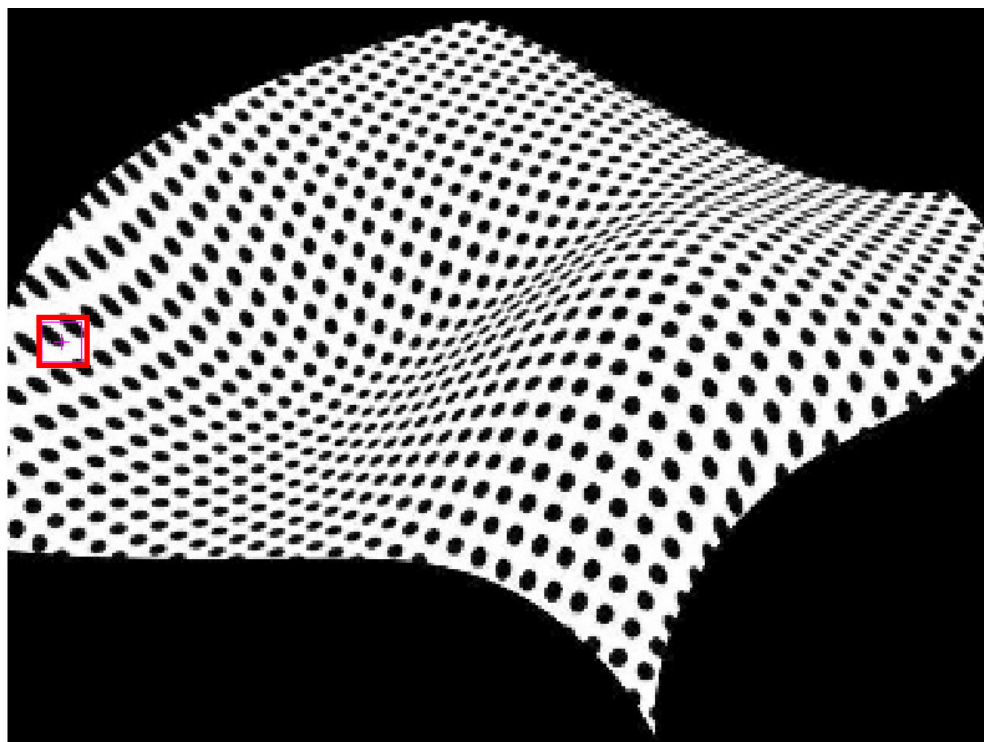


Figura 45 - Imagem sintética de uma superfície irregular com ponto frontal indicado.

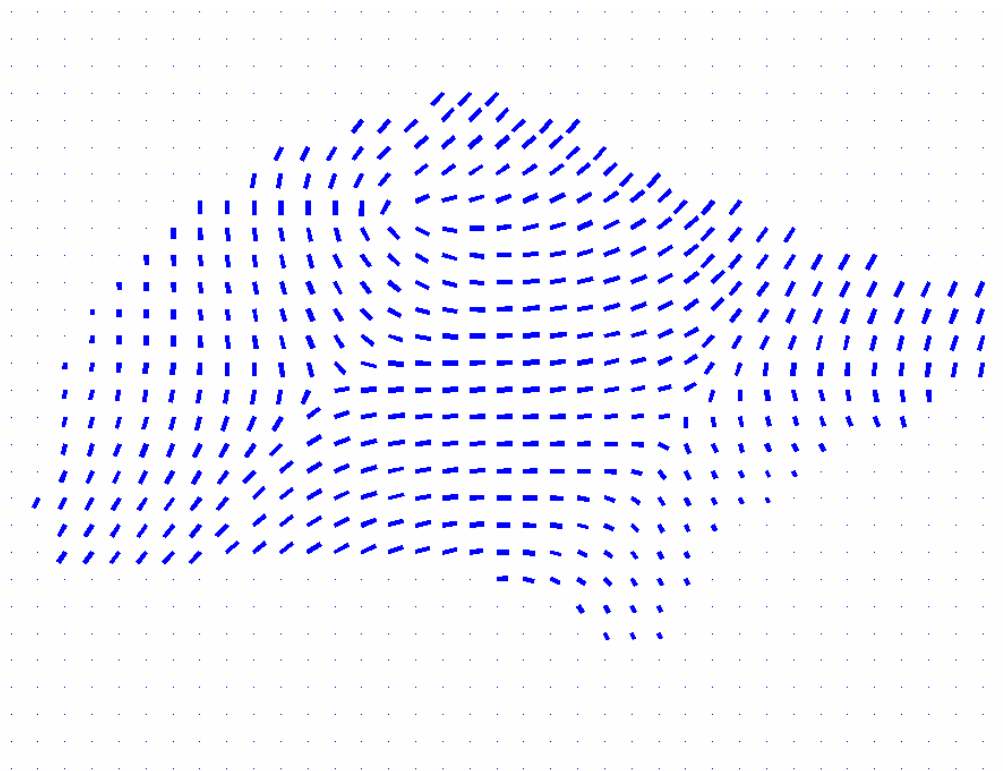
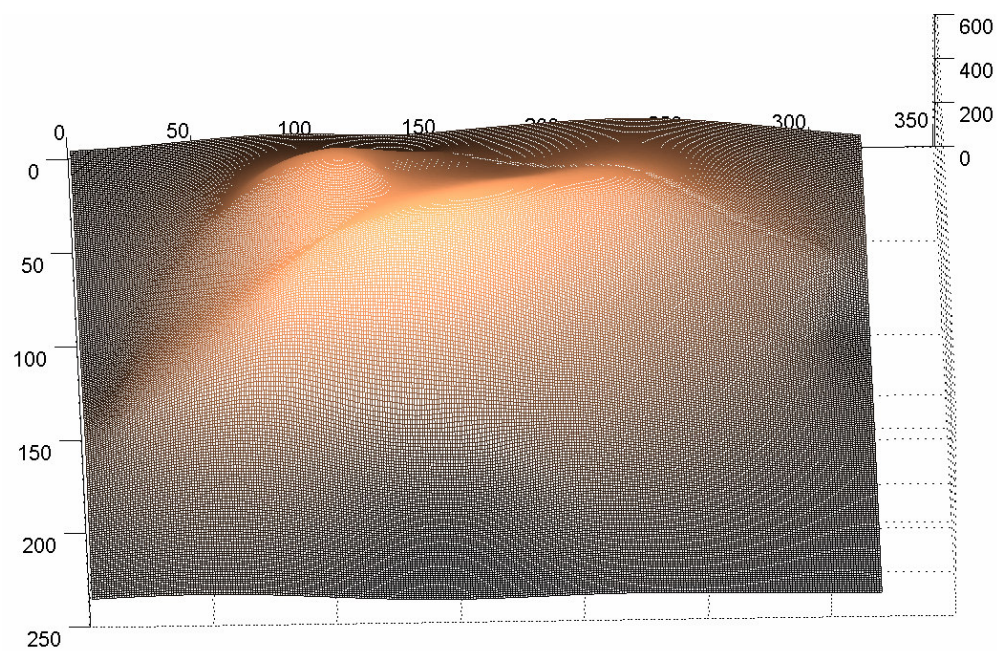


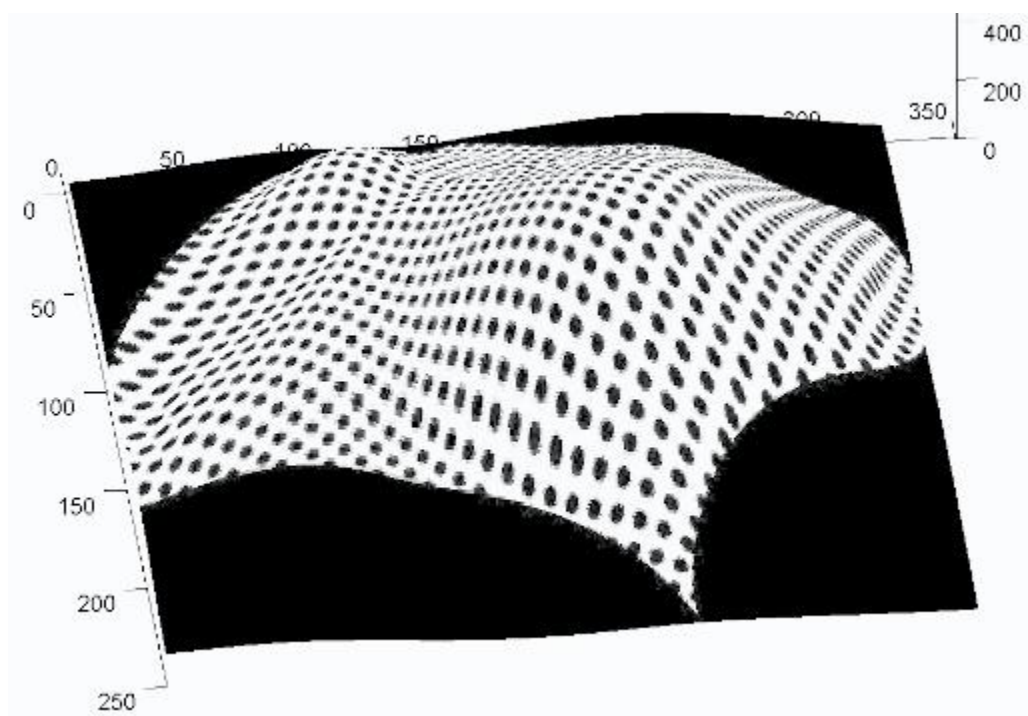
Figura 46 - Mapa de agulhas obtido pelo método de Super.

A Figura 45 mostra uma superfície irregular gerada artificialmente. Analisando o mapa de agulhas na Figura 46 observa-se que a região central da imagem possui uma estimativa ruim, enquanto que a região superior e à direita (próximas à borda) possuem uma boa aproximação. A região central onde o erro é maior deve-se ao fato de que, ali, a compressão (*slant*) é alta, ou seja, a perda de informação também o é, o que torna difícil a precisão do método. Nota-se também que o ponto frontal detectado pelo algoritmo não representa adequadamente o elemento de textura no mundo real, outro fator que desencadeia erros.

A Figura 47 apresenta os modelos reconstruídos. Como era esperado, a parte central da superfície possui uma reconstrução não condizente com a realidade. Entretanto, a borda superior e a direita possuem, ambas, boas aproximações da real forma da imagem, fato que pode ser melhor visualizado na Figura 47(b) com a textura sobreposta.



(a)



(b)

Figura 47 - (a) Modelo 3D a partir do mapa de agulhas (estimado pelo método de Super) da superfície irregular. (b) Modelo 3D com a textura sobreposta.

5.3.2

Reconstrução da forma da superfície usando a estimativa da transformação afim de Loh

Neste experimento foi empregado o algoritmo descrito na Seção 4.4, utilizando o método de Loh para a estimativa da transformação afim.

O tamanho das imagens usadas aqui é de 400x400 *pixels*. Os pontos de interesse detectados pelo método de Lowe foram agrupados em 10 classes com o algoritmo *K-means*, para todas as imagens. O tamanho dos recortes extraídos nas imagens, centralizados nos pontos de interesse, manteve-se fixado em 40x40 *pixels* para todos os casos. Para a suavização foi empregado um filtro gaussiano de tamanho fixo, 5x5 *pixels*.

Os pontos frontais indicados nas imagens foram determinados visualmente, e em alguns casos podem ser empregados diferentes pontos frontais para a mesma imagem. Isso ocorre quando pontos de interesse de um grupo estão centralizados na textura de forma diferente que os de outro grupo. Um exemplo pode ser visto na Figura 48 onde o grupo em (a) é centralizado nos pontos negros, enquanto que (b) tem centro nos grupos de quatro pontos.

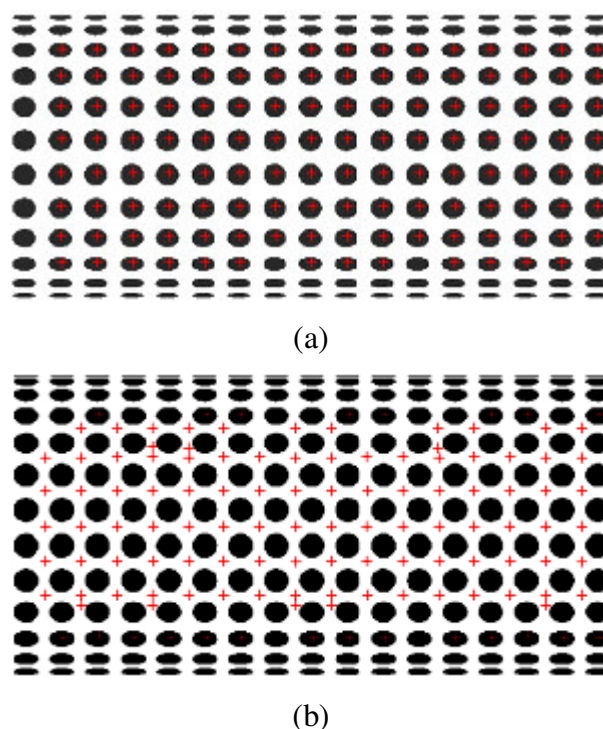
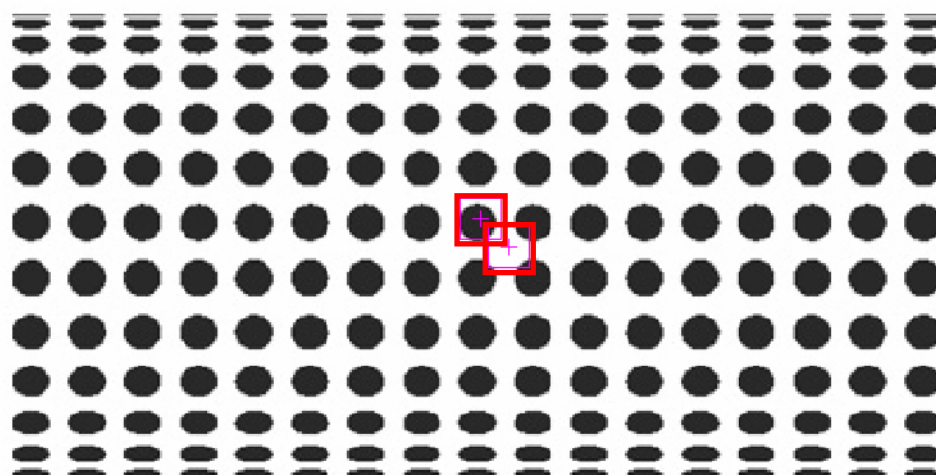
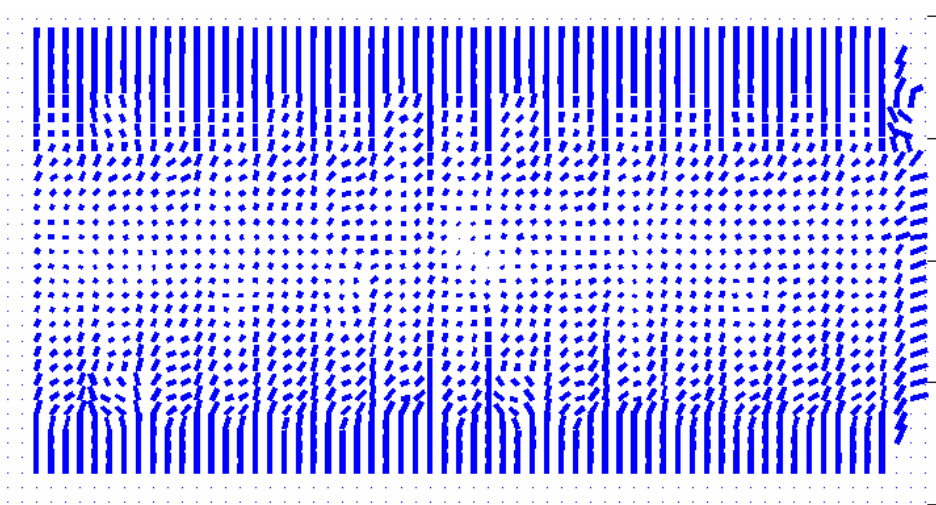


Figura 48 – Exemplo de grupos de pontos de interesse cujo ponto frontal é distinto para cada. (a) Grupo 1. (b) Grupo 2.

Foram repetidos aqui os mesmos experimentos realizados com o método de Super. Entretanto, acrescentou-se uma figura aos resultados, a das agulhas, representando a orientação sobreposta à textura. Isto porque temos a orientação em cada ponto de interesse e assim visualizada permite uma melhor análise dos resultados.



(a)



(b)

Figura 49 - Resultados obtidos para a imagem de um cilindro sintético. (a) Imagem original com pontos frontais selecionados indicados. (b) Mapa de agulhas.

Na Figura 49(b) se encontra o mapa de agulhas calculado pelo algoritmo. Nele pode-se observar que o resultado do *slant* é coerente, elevando-se à medida que se caminha do centro para as bordas superiores. Porém, o *tilt* apresenta erro de orientação, nos casos em que o ponto de interesse está no centro dos círculos

negros. Na Figura 50 o mapa de agulhas é mostrado sobreposto à imagem. Dele nota-se que o erro do *tilt* não é oriundo de uma má localização dos pontos de interesse e sim da própria estimativa da transformação afim para essa textura.

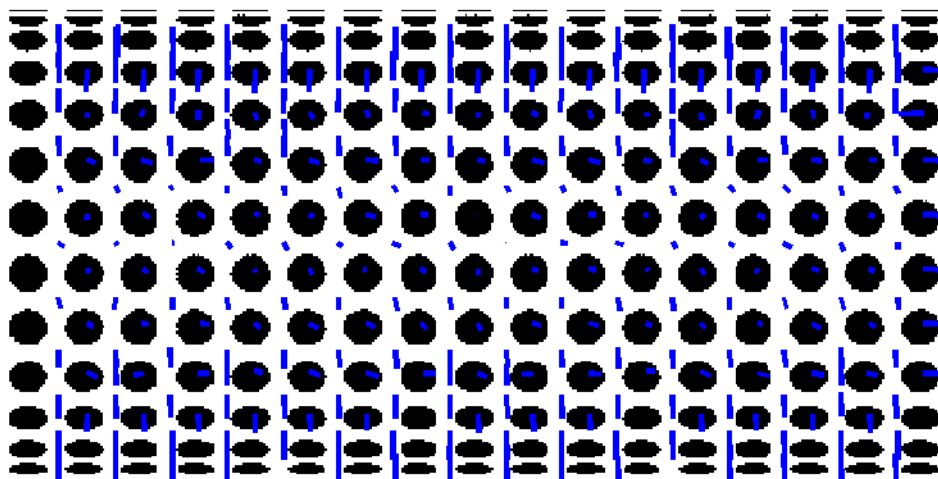


Figura 50 – Cilindro sintético com as normais sobrepostas.

Na Figura 51 e Figura 52 são apresentados os modelos tridimensionais do cilindro sintético. Devido ao ruído mostrado no mapa de agulhas a reconstrução também é ruidosa, apresentando saliências.

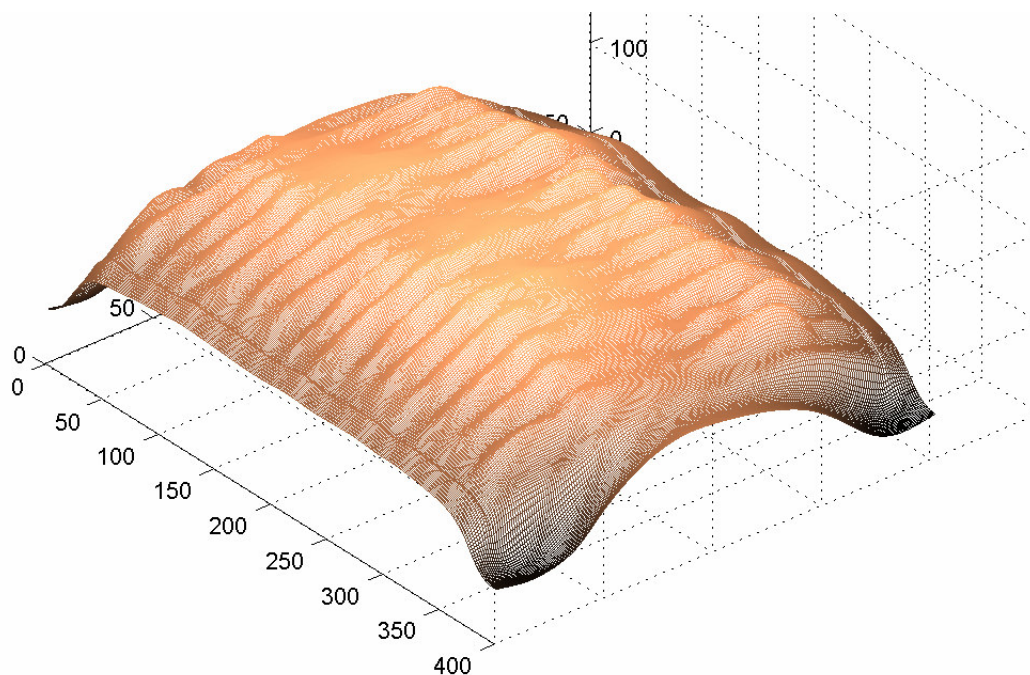


Figura 51 - Modelo 3D reconstruído a partir do mapa de agulhas.

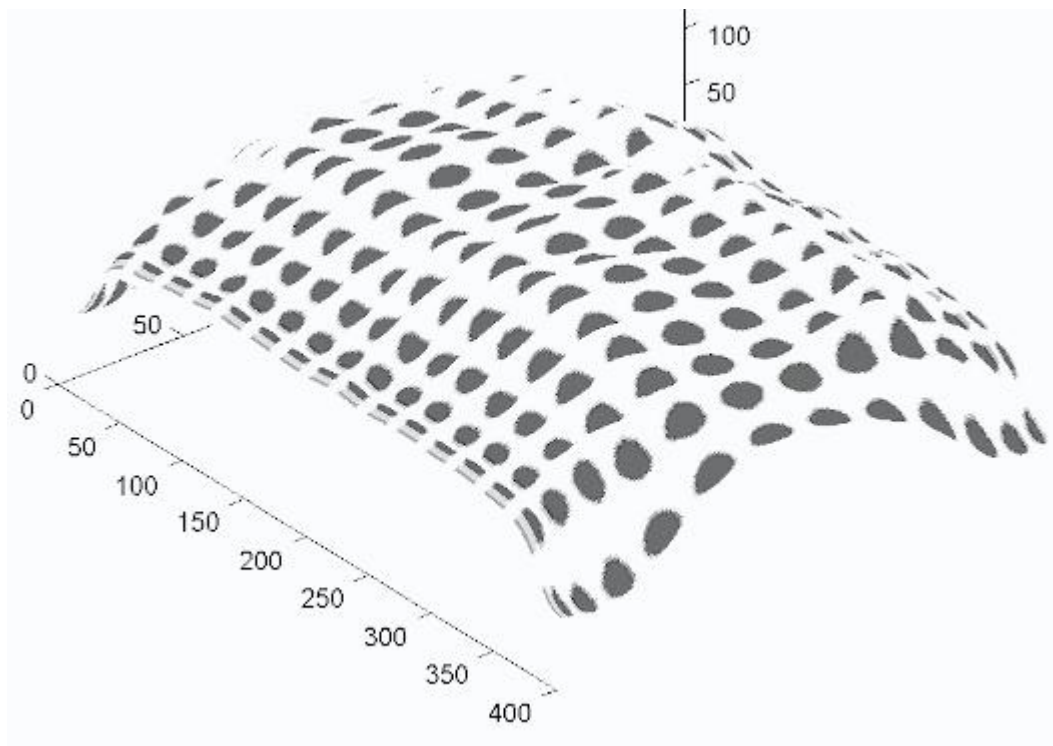


Figura 52 – Modelo 3D com a textura sobreposta.

A Figura 53 mostra uma bola de golfe e a Figura 54 seu mapa de agulhas:

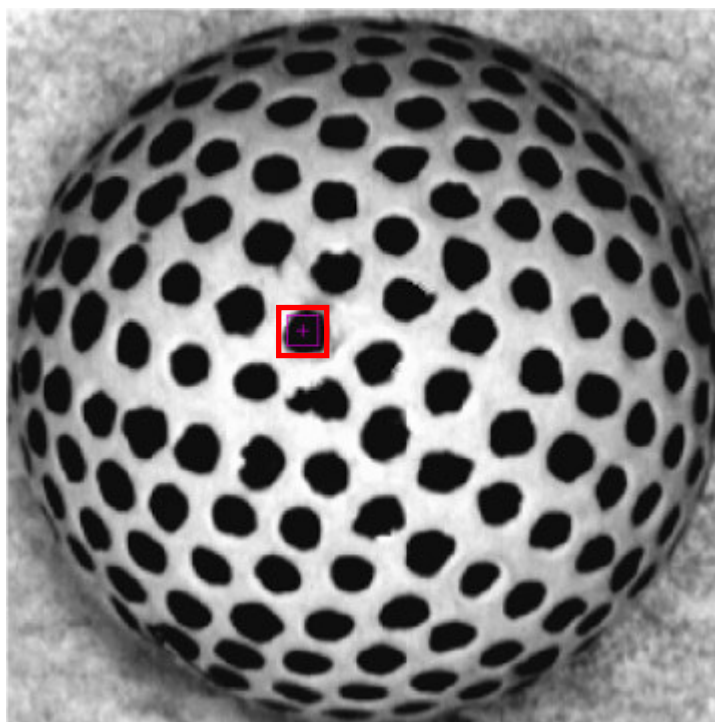


Figura 53 - Imagem de uma bola de golfe com ponto frontal selecionado indicado.

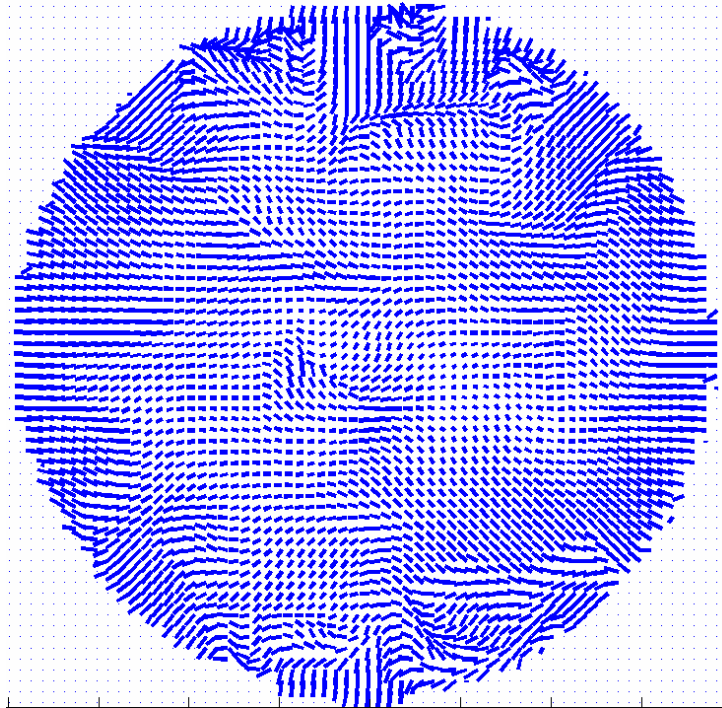


Figura 54 - Mapa de agulhas encontrado

Na Figura 54 o mapa de agulhas visto é coerente nas bordas do objeto. Entretanto, o centro é mais ruidoso e a estimativa da transformação afim apresenta erro em alguns pontos, como visto na Figura 55. Estes erros se dão, além da limitação do método, pela irregularidade na textura da bola de golfe.

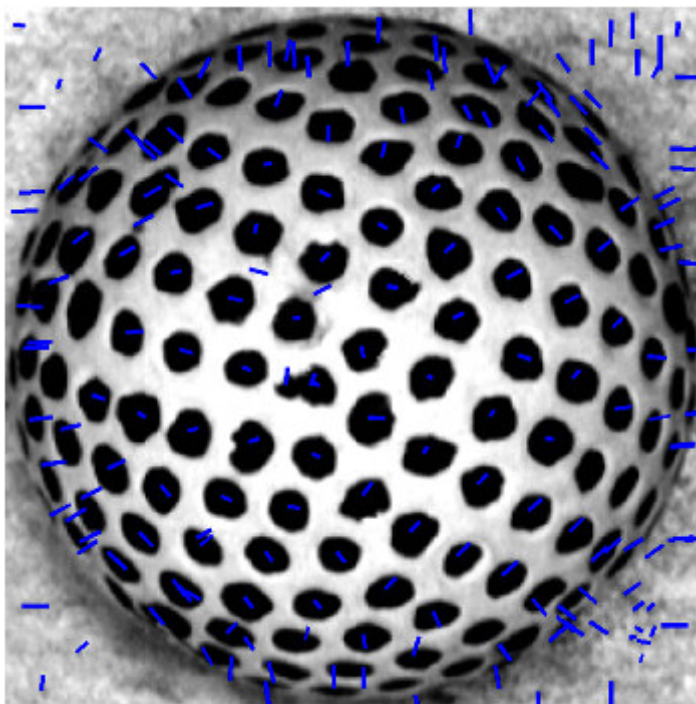


Figura 55 – Bola de golfe com as normais sobrepostas.

O modelo tridimensional apresentado na Figura 57 é ruidoso e termina por mostrar uma aproximação bastante grosseira da superfície da bola de golfe. O método mostra aqui sua sensibilidade à textura do objeto, uma vez que essa deveria ser o mais homogênea possível.

Como já foi dito no experimento com o método de Super a pintura não está uniforme, o que acaba por adicionar ruído à textura. Soma-se a este ruído o próprio erro do método de estimativa e mais a sensibilidade da decomposição SVD para encontrar os ângulos.

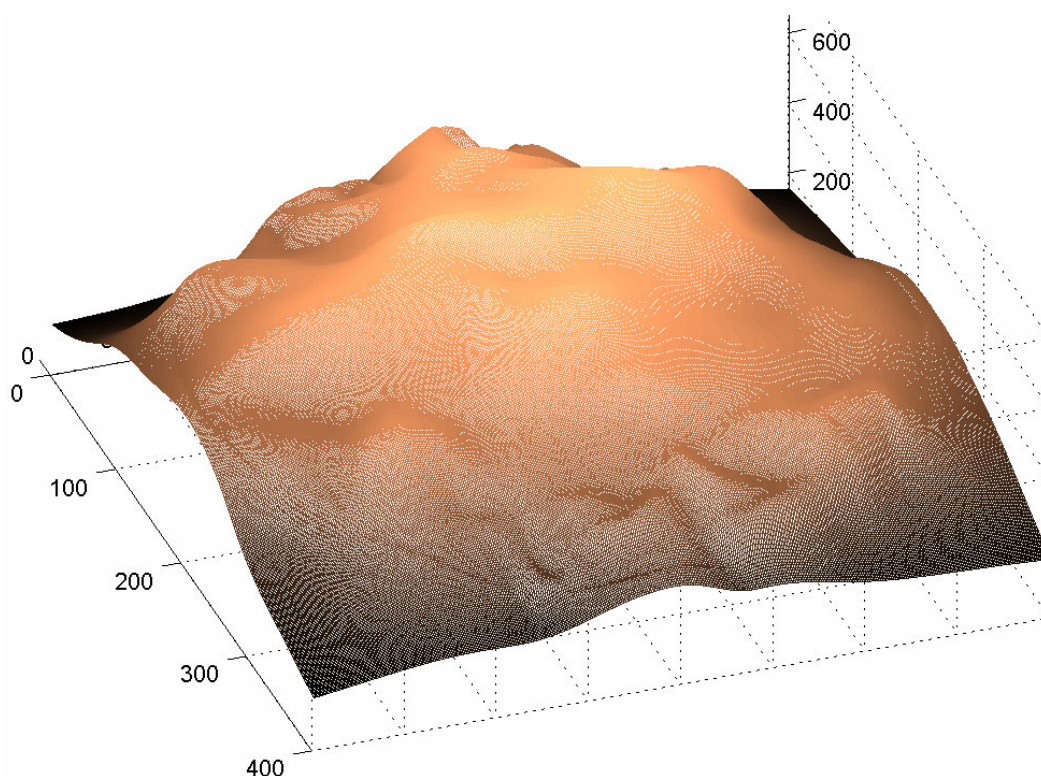


Figura 56 - Modelo 3D reconstruído a partir do mapa de agulhas da bola de golfe.

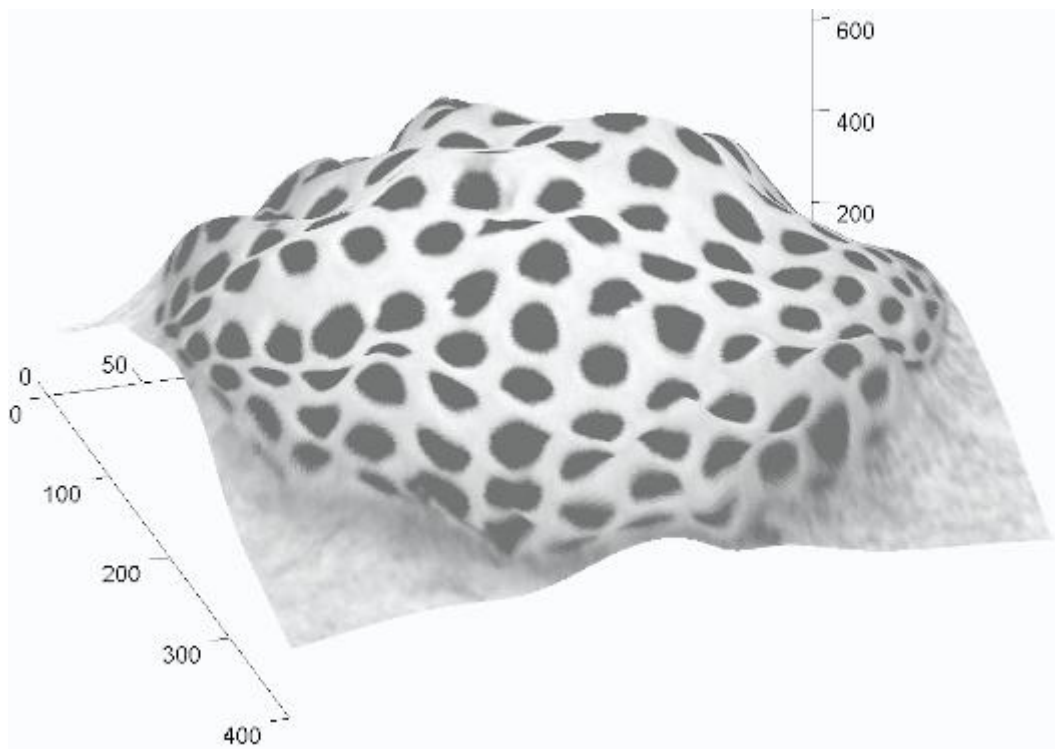


Figura 57 – Modelo 3D da figura anterior com a textura sobreposta.

O resultado para uma superfície irregular encontra-se na Figura 59:

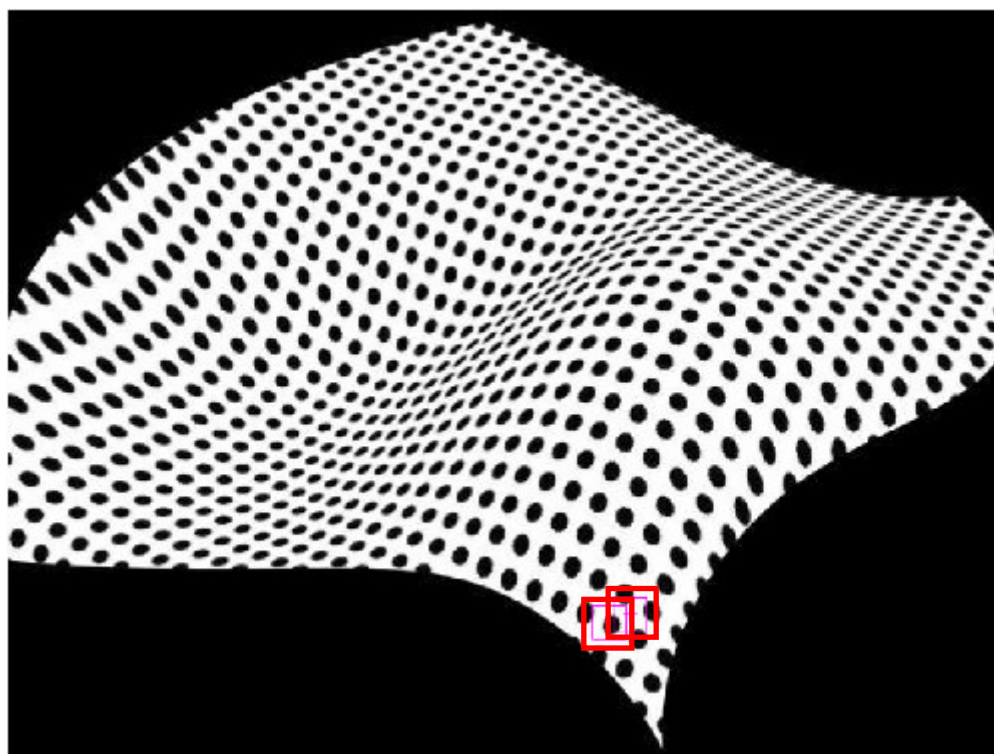


Figura 58 - Superfície sintética irregular com pontos frontais selecionados indicados.

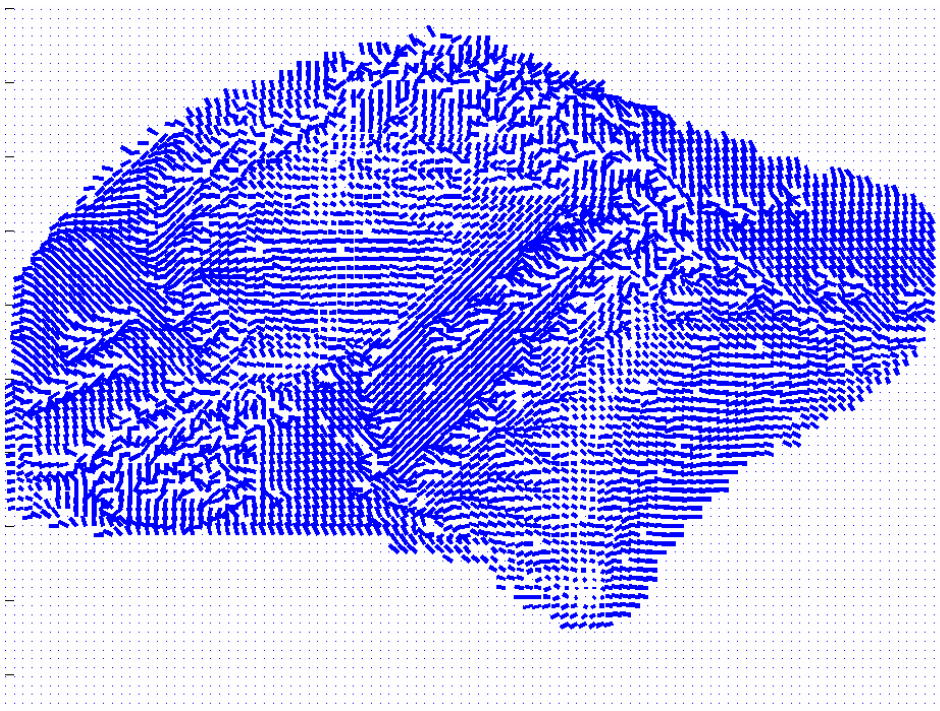


Figura 59 - Mapa de agulhas da figura anterior

O mapa de agulhas na Figura 59 é de difícil interpretação, mas ao olharmos para as normais sobrepostas nos pontos de interesse na Figura 60 podemos perceber que o método obtém ótimos resultados, tendo uma boa aproximação da normal na maior parte dos pontos.

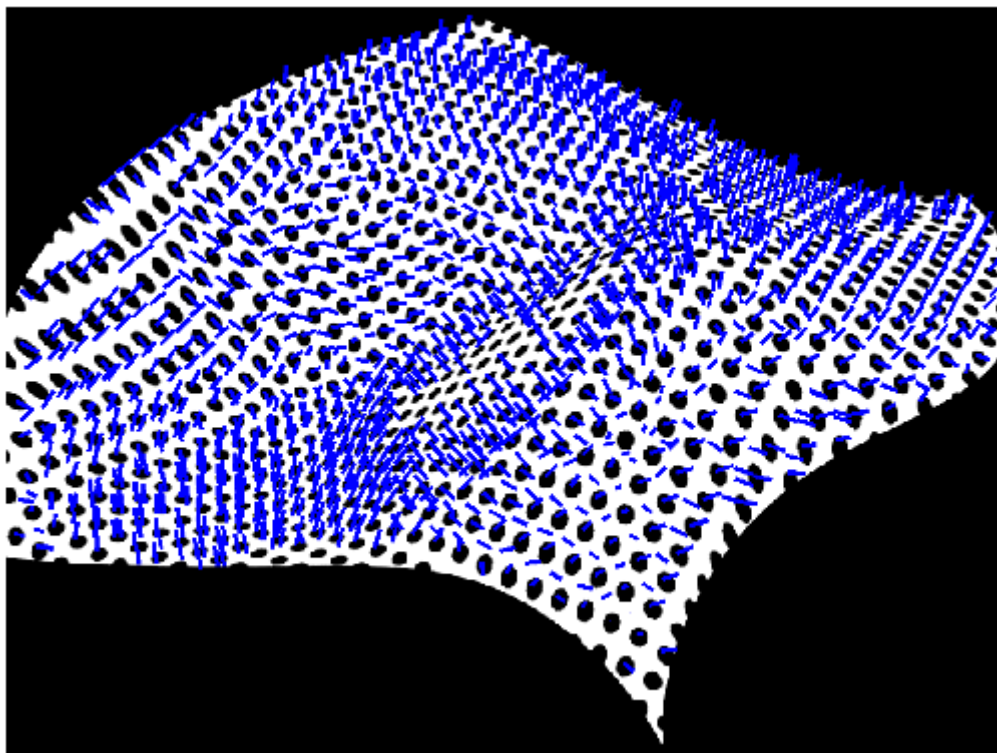
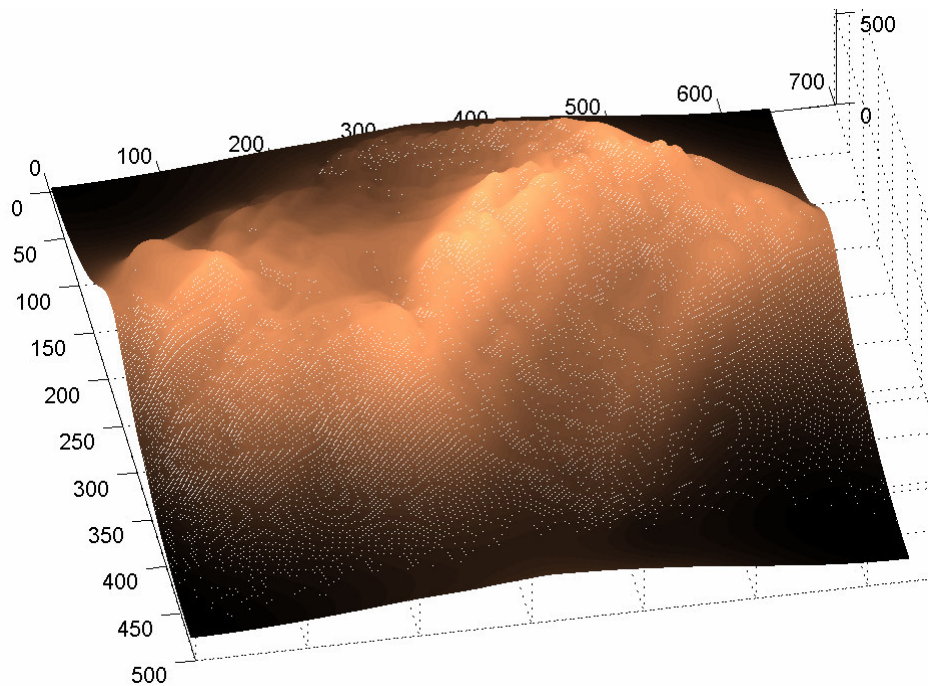
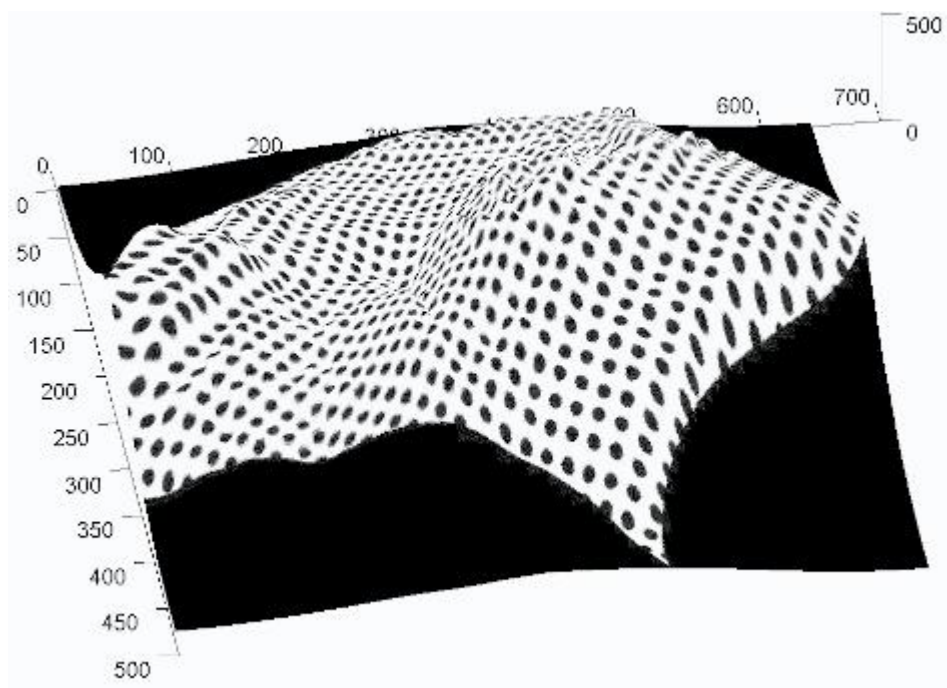


Figura 60 – Superfície irregular com as normais sobrepostas.

A reconstrução do modelo a partir do mapa de agulhas segue na Figura 61, onde temos uma aproximação da superfície, com esta possuindo erro grosseiro somente na depressão ao centro-esquerdo da imagem porém, ainda assim, a reconstrução consegue mostrar que naquela região há um vale.



(a)



(b)

Figura 61 - (a) Modelo 3D reconstruído a partir do mapa de agulhas. (b) Modelo 3D com a textura sobreposta.

5.3.3 Estudo de caso

No presente experimento testou-se os algoritmos de *SFT* em uma situação real, na agricultura, para reconhecer o estado de carregamento para uma colheitadeira (Figura 62).



Figura 62 – Cena real de uma colheita.

O objetivo é obter o modelo tridimensional do conteúdo no interior da caçamba do caminhão que acompanha a colheitadeira em uma colheita semi-automatizada.

O projeto original é oriundo do trabalho de Graefe [67], da Universidade de Braunschweig, na Alemanha, onde a empresa fabricante da colheitadeira desejava um sistema de controle móvel do braço que deposita o material colhido em um caminhão. O objetivo do sistema de controle é manter o braço sempre sobre a caçamba. Numa situação real de colheita manter ambos os veículos bem alinhados é uma tarefa difícil, principalmente em terrenos irregulares. Para solucionar o problema foi colocada uma câmara presa no braço da colheitadeira, e usando processamento de imagem detectam-se as bordas da caçamba do caminhão. Uma segunda fase do projeto é detectar o estado de carregamento de um caminhão, além de encontrar um modelo tridimensional do conteúdo, permitindo assim

otimizar a distribuição do material colhido na caçamba, coordenando o despejo nas áreas menos preenchidas. Um exemplo pode ser visto na Figura 63.



Figura 63 – Exemplo de modelagem do conteúdo da caçamba.

Neste experimento foram utilizadas duas imagens reais, de tamanho 640×480 pixels, obtidas a partir da câmara presa ao braço da colheitadeira. Foram aplicadas nelas os dois algoritmos de *SFT* apresentados anteriormente. São mostrados os pontos frontais escolhidos e as superfícies tridimensionais reconstruídas. Omitiram-se os mapas de agulhas uma vez que o objetivo neste passo é avaliar o modelo 3D alcançado.

Na Figura 64 temos ilustrados os pontos frontais usados nos métodos para a primeira imagem. Na Figura 64(a) encontra-se demarcado o ponto frontal determinado automaticamente pelo algoritmo de Super e em (b) o ponto frontal escolhido manualmente para o segundo método.

Na Figura 65(a) é mostrado o modelo tridimensional reconstruído a partir do mapa de agulhas obtido com o método de Super. A Figura 66(a) mostra a reconstrução usando o segundo método, o algoritmo *SFT* com a estimativa da transformação afim de Loh.

A mesma seqüência é usada para a segunda imagem da câmara da colheitadeira, conforme pode ser visto em Figura 67, Figura 68 e Figura 69.

Ao final das figuras são apresentadas as conclusões sobre estes resultados.

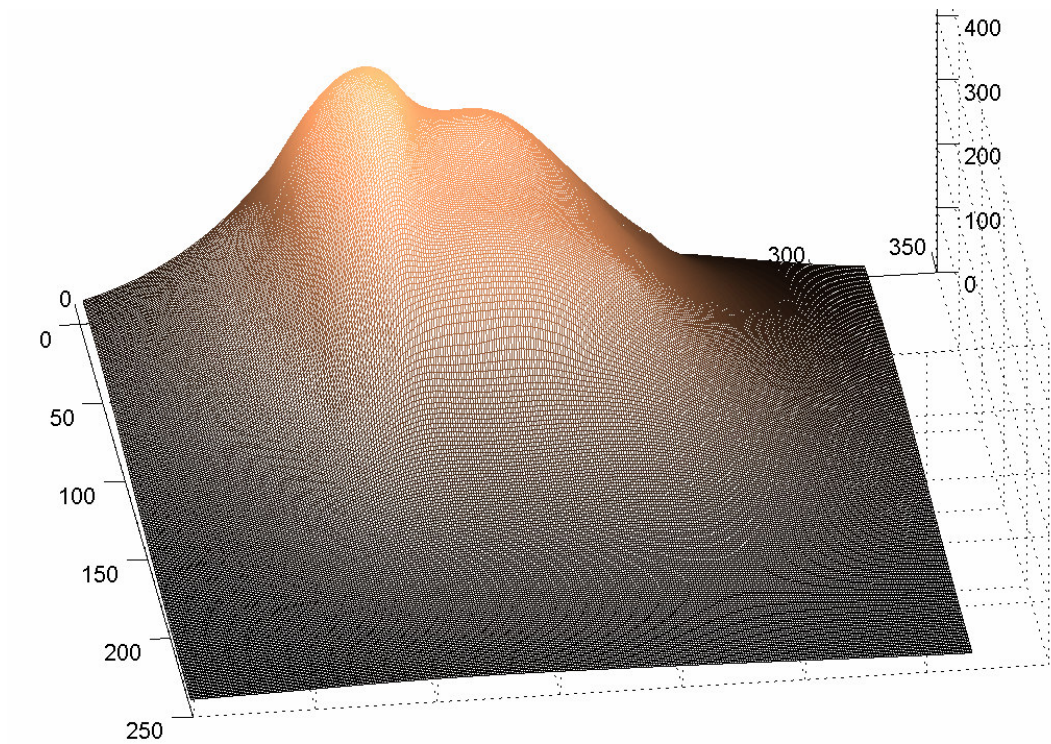


(a)

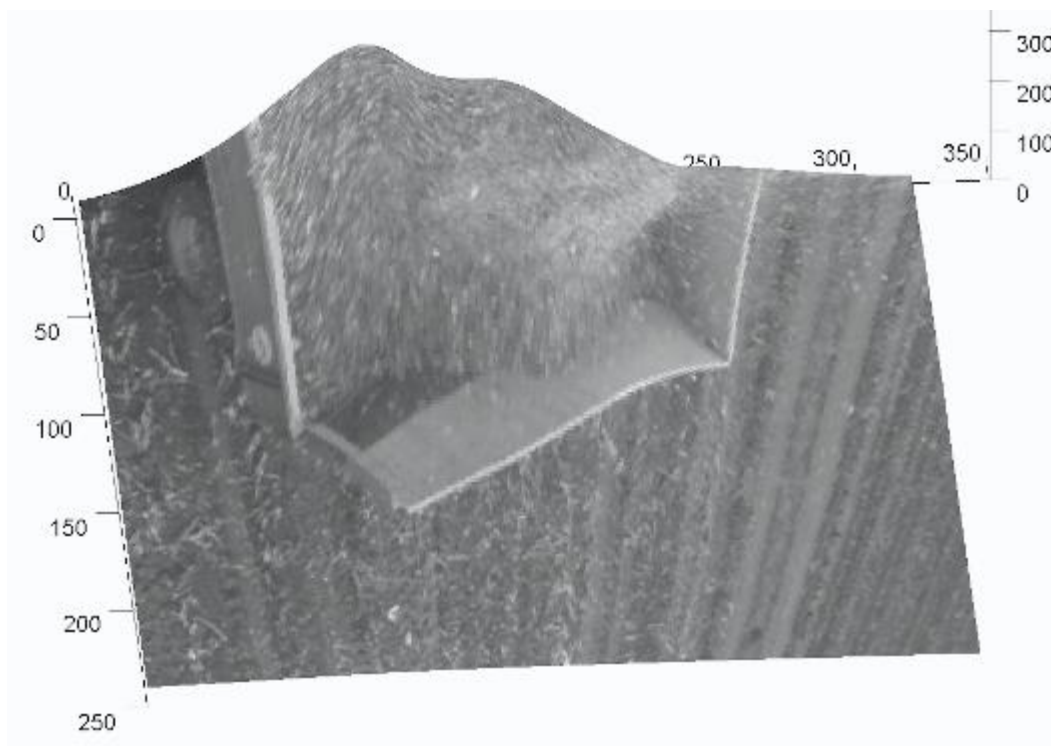


(b)

Figura 64 – (a) Primeira imagem da câmara da colheitadeira com ponto frontal detectado pelo método de Super. (b) Primeira imagem com o ponto frontal determinado manualmente para o algoritmo *SFT* com estimativa afim de Loh.

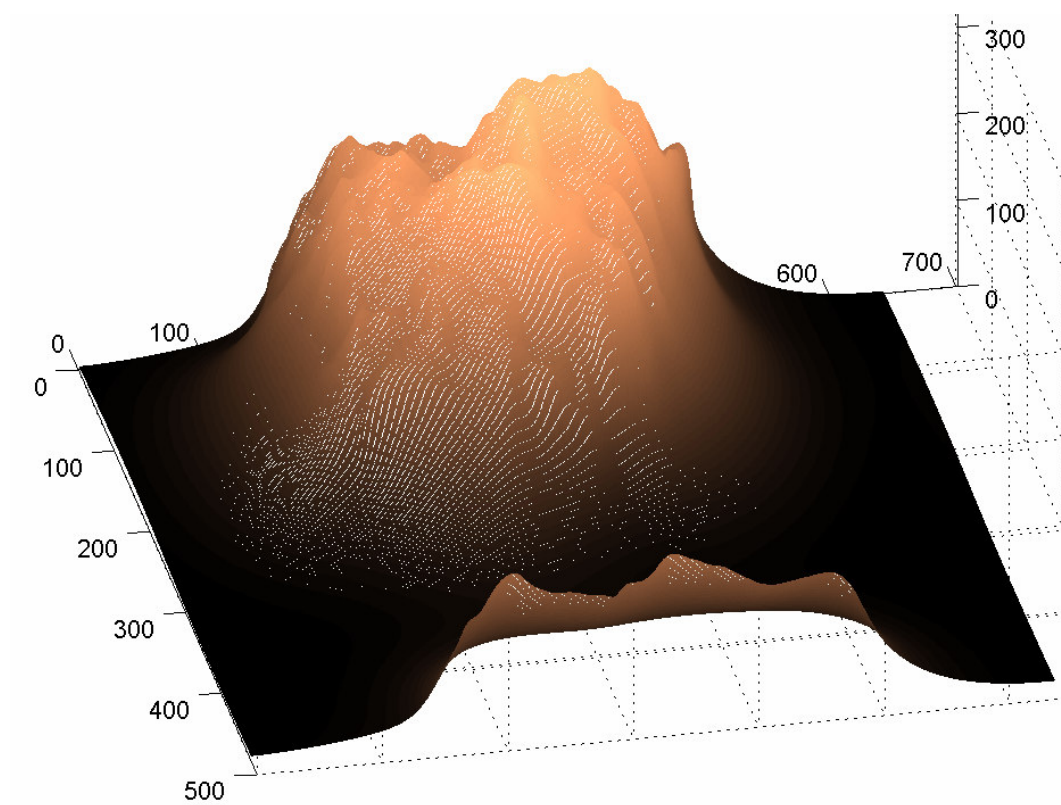


(a)

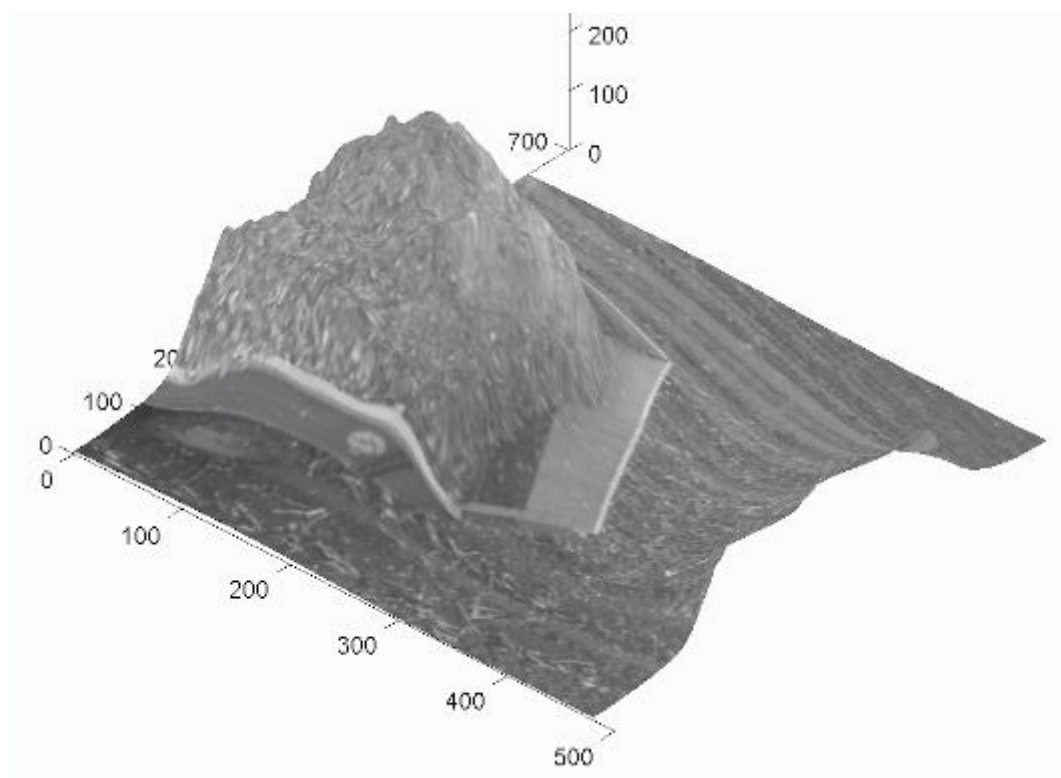


(b)

Figura 65 – (a) Modelo tridimensional da primeira imagem da câmara da colheitadeira obtido a partir do mapa de agulhas oriundo do método de Super. (b) O mesmo modelo com a textura sobreposta.

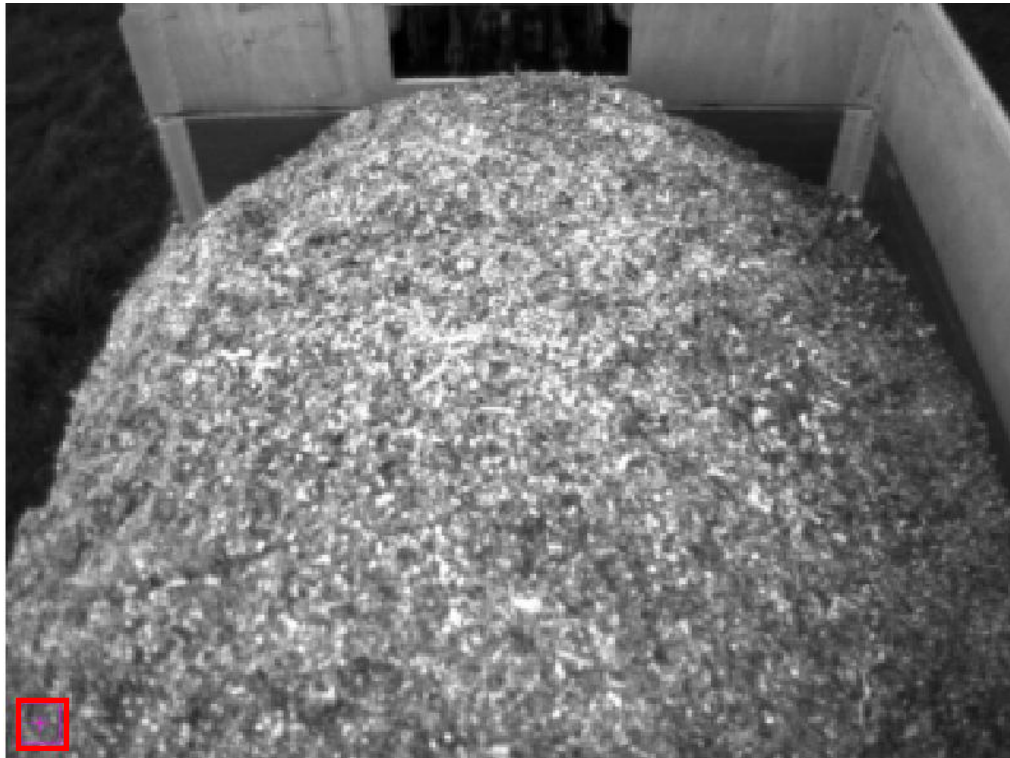


(a)

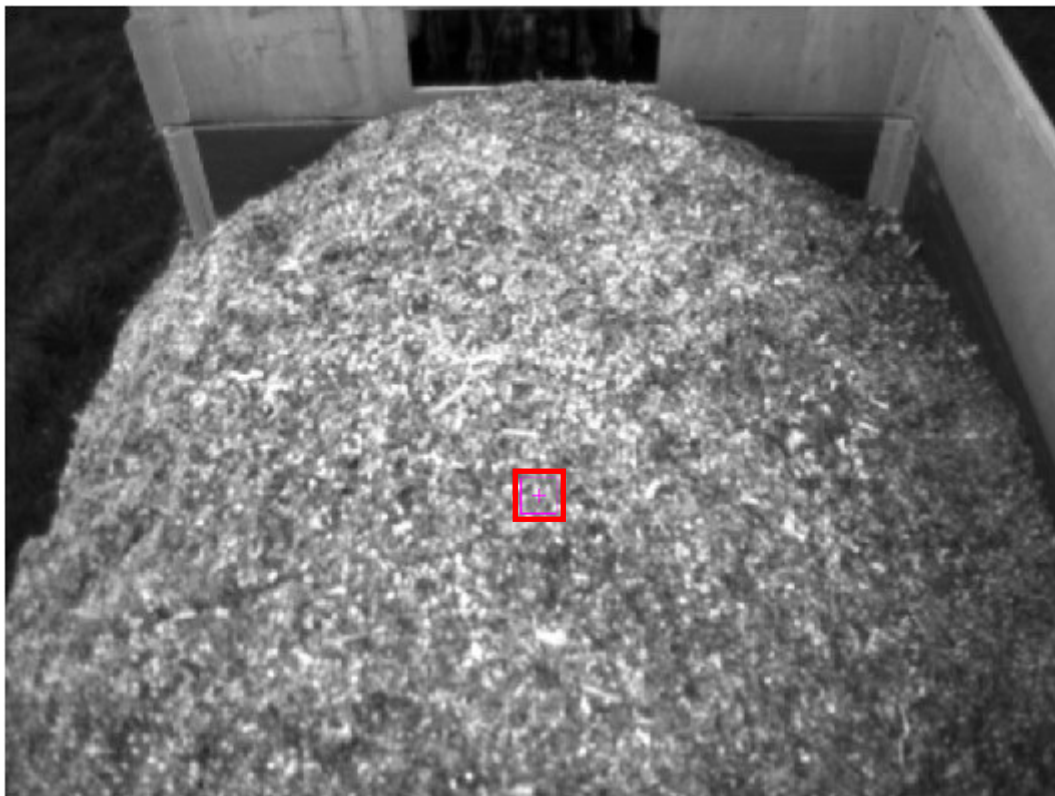


(b)

Figura 66 - (a) Modelo tridimensional da primeira imagem da câmara da colheitadeira obtido a partir do mapa de agulhas oriundo do algoritmo *SFT* com estimativa afim de Loh. (b) O mesmo modelo com a textura sobreposta.

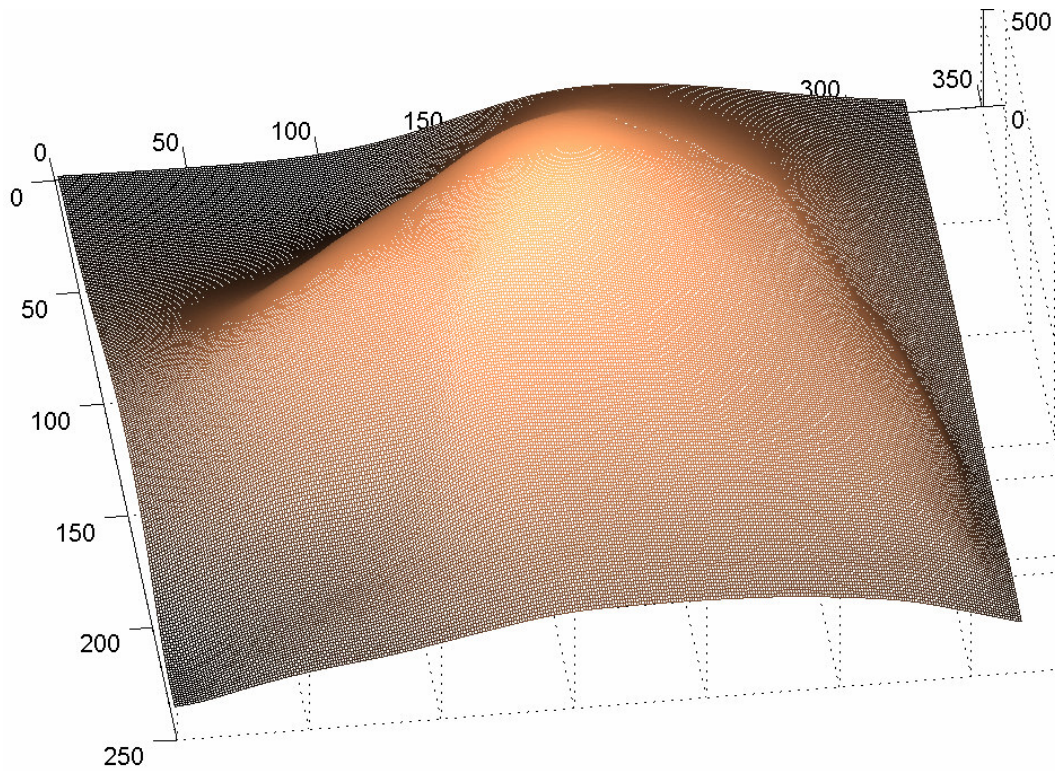


(a)

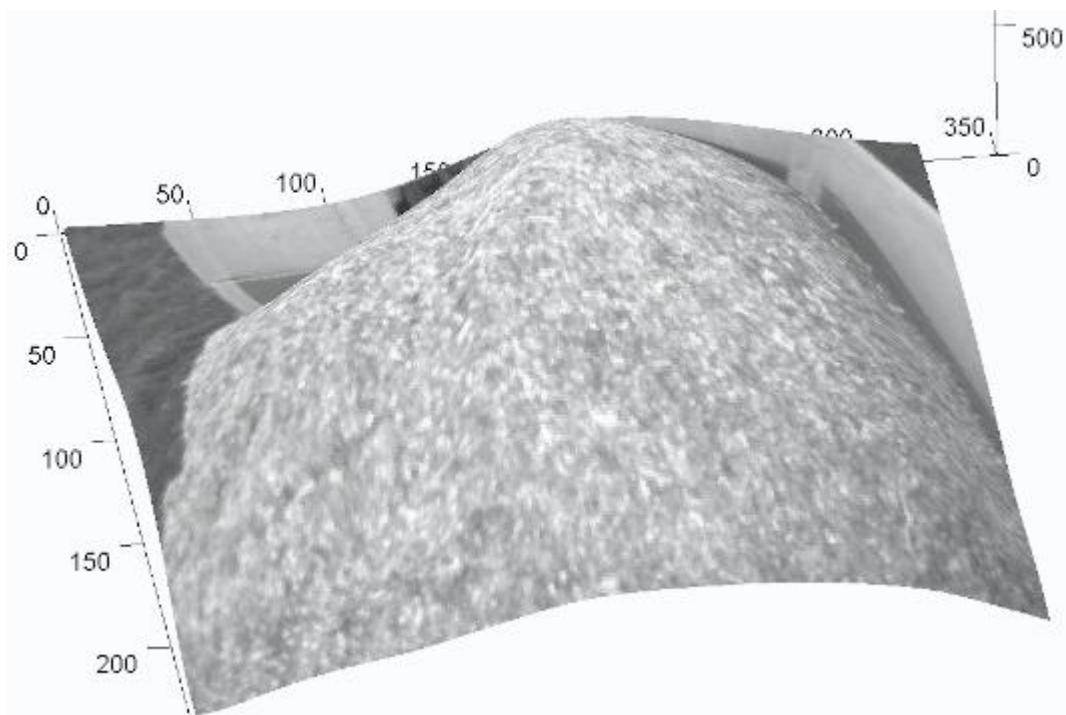


(b)

Figura 67 - (a) Segunda imagem da câmara da colheitadeira com ponto frontal detectado pelo método de Super. (b) Segunda imagem com o ponto frontal determinado manualmente para o algoritmo *SFT* com estimativa afim de Loh.

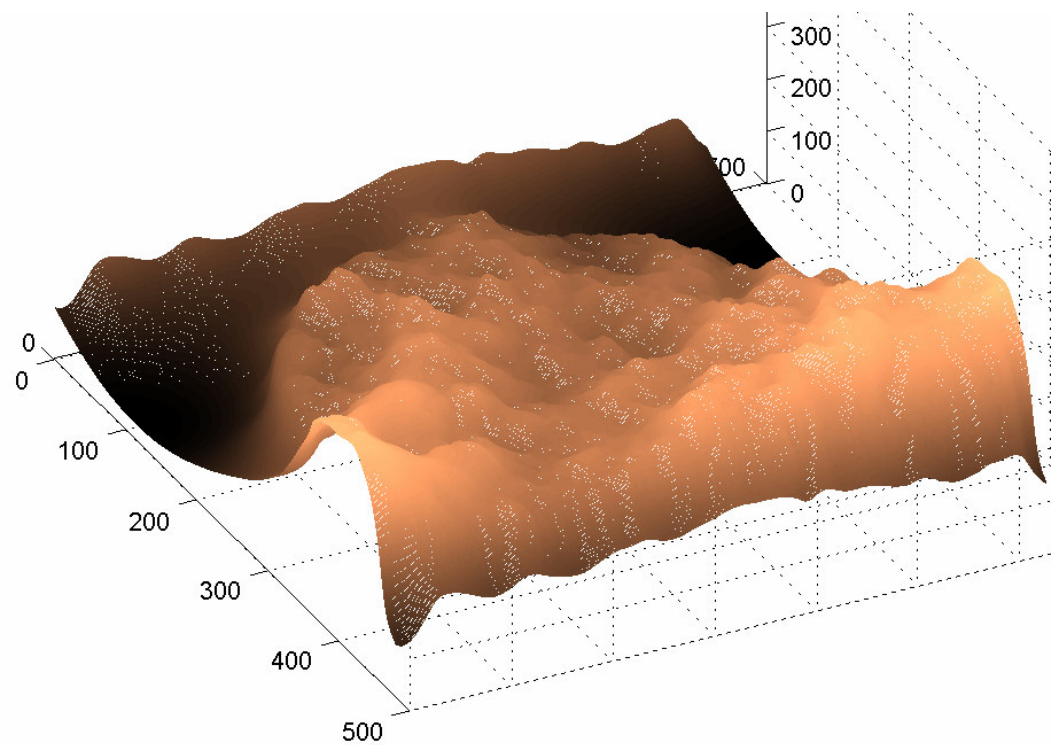


(a)

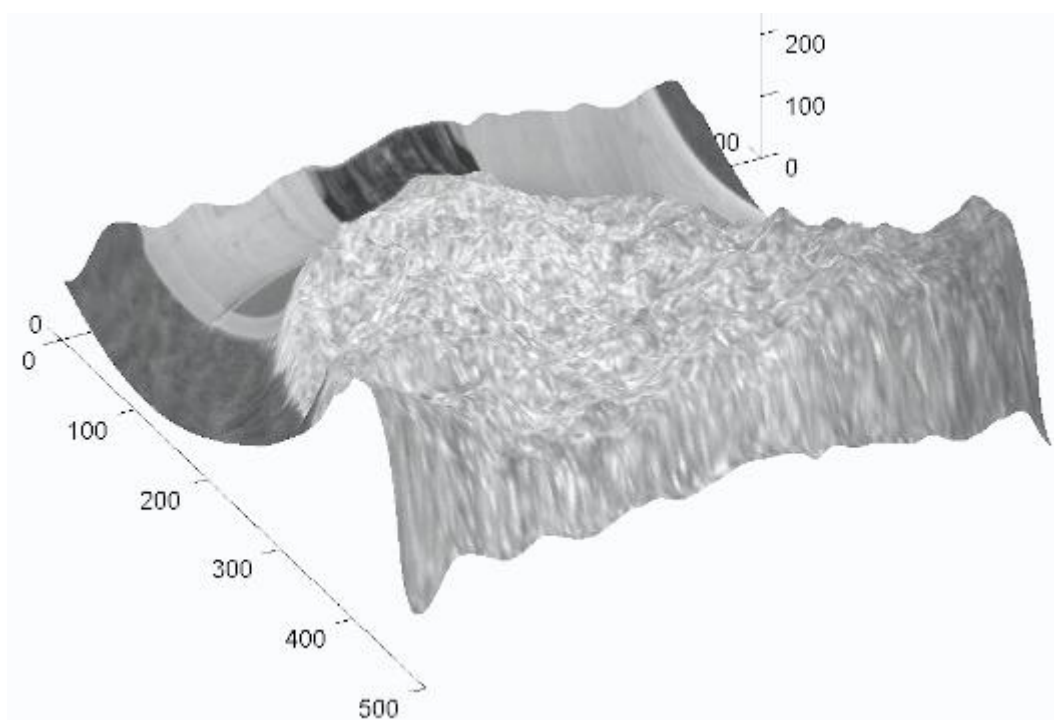


(b)

Figura 68 – (a) Modelo tridimensional da segunda imagem da câmara da colheitadeira obtido a partir do mapa de agulhas oriundo do método de Super. (b) O mesmo modelo com a textura sobreposta.



(a)



(b)

Figura 69 – (a) Modelo tridimensional da segunda imagem da câmara da colheitadeira obtido a partir do mapa de agulhas oriundo do algoritmo *SFT* com estimativa afim de Loh. (b) O mesmo modelo com a textura sobreposta.

Os resultados apresentados, em ambos os métodos, realizam aproximações bastante grosseiras da superfície. Em alguns casos (Figura 69) a reconstrução em nada lembra a imagem original.

Conforme exemplificado nos resultados anteriores o problema da modelagem automática do conteúdo da caçamba está longe de ser trivial. É algo complexo até mesmo para a visão humana. Isto pode ser confirmado na Figura 70, ao retirarmos da imagem outras informações, como o contorno do caminhão. Passamos agora também a observar somente a textura e assim é possível perceber as limitações dos métodos de *SFT*.



Figura 70 – As duas imagens segmentadas da câmara da colheitadeira.

5.4

Avaliação Final dos experimentos

Conforme mencionado no início deste capítulo, a divisão em etapas dos experimentos apontou detalhes cruciais no método, melhor explicitadas abaixo:

Decomposição de uma transformação afim: foi visto que a própria decomposição por *SVD* é sensível a ruído, apresentando erros grosseiros em alguns casos isolados. Este erro não foi divulgado em nenhum trabalho sobre *SFT* publicado até aqui. O trabalho de Forsyth [36] realiza a decomposição por autovetores e autovalores da matriz TT^t , cujo resultado também foi testado e é idêntico ao do *SVD*, fato pelo qual não foi exposto no presente texto.

Estimativa da transformação afim: o experimento realizado mostrou o desempenho dos métodos em diferentes texturas e para toda uma gama de possíveis transformações afim. Como foi visto no experimento anterior, pequenos ruídos na estimativa da transformação afim podem implicar em erros grosseiros na estimativa dos ângulos por *SVD*. Com o intuito de analisar o desempenho somente a estimativa da transformação afim, avaliou-se a diferença absoluta entre as matrizes originais e as estimadas. Os experimentos indicaram que, neste aspecto, o método de Loh é superior ao método de Rosenholtz.

Mapa de agulhas e reconstrução 3D da superfície: os resultados encontrados aqui foram coerentes com as superfícies, apesar de alguns trechos com erros grosseiros. Entretanto, cabe lembrar que *SFT* não é um método exato; comporta-se adequadamente em superfícies suaves, com texturas bem definidas. Desta forma, os resultados obtidos são aproximações.

A reconstrução tridimensional de objetos a partir do mapa de agulhas leva em conta as distâncias entre pontos. Assim, um erro grosseiro na estimativa de *tilt* ou *slant* num ponto resulta numa deformação importante na forma da superfície calculada.

Comparar os resultados dos algoritmos de *SFT* é uma tarefa complexa. Os primeiros trabalhos apresentam como resultado somente o mapa de agulhas; reconstruções de superfícies existem apenas nos trabalhos mais recentes. Também

não existe um banco de imagens amplamente adotado para *benchmark* e cada pesquisador emprega suas próprias imagens. Devido a esses fatores a comparação entre resultados de diferentes trabalhos é difícil.

Em última instância, a análise dos resultados finais é subjetiva, uma vez que tanto o mapa de agulhas quanto a reconstrução tridimensional são avaliadas visualmente, observando sua semelhança frente à imagem original. Faltam nos experimentos, por exemplo, os valores reais de *tilt* e *slant* em cada ponto do objeto, bem como os valores de profundidade reais de cada ponto do objeto. Ambos são dados difíceis de serem adquiridos, e neste trabalho não foi possível obtê-los.

6 Conclusão

Esta dissertação investigou a literatura sobre *Shape from Texture* e implementou alguns métodos a fim de avaliar seu desempenho. Os objetivos são:

- Investigar as técnicas de reconstrução de superfícies curvas baseadas somente na informação de textura.
- Estimar o desempenho e estabilidade de algumas técnicas selecionadas.
- Avaliar sua aplicabilidade a problemas práticos de modelagem 3D.
- Apontar pontos de possível melhora nas técnicas.

O estudo realizado trouxe as seguintes conclusões:

- Nas etapas dos experimentos foi observado um ponto importante: a extração dos valores de *tilt* e *slant* a partir da decomposição *SVD* é sensível ao ruído. Mesmo pequenos erros na estimativa da transformação afim podem gerar erros grosseiros na estimativa dos ângulos *tilt* e *slant*.
- A reconstrução tridimensional baseada no mapa de agulhas apresenta resultados satisfatórios em exemplos simples. Em casos reais, os métodos de reconstrução por *SFT* analisados podem incorrer em erros significativos. Isso vale também para texturas complexas e objetos em ambientes não controlados ou com iluminação não-uniforme.

É válido lembrar que, em Visão Computacional, é raro encontrar um método robusto que aceite qualquer imagem como entrada e apresente bons resultados. Em *SFT* não é diferente; os métodos funcionam adequadamente quando as texturas são bem definidas e homogêneas, e ainda assim podem ocorrer erros em trechos do algoritmo, como na determinação do ponto frontal, decomposição da matriz em ângulos e reconstrução a partir do mapa de agulhas.

Os métodos estudados aqui são complexos e possuem inúmeros parâmetros distintos, que precisam ser ajustados adequadamente para cada situação, o que prejudica a sua utilização em aplicações reais. Além disso, a literatura disponível é fragmentada e incompleta. Devido a esses fatores *SFT* é ainda um tema de pesquisa em aberto.

Como foi mencionado na introdução deste trabalho, reproduzir aspectos da visão humana é demasiadamente complexo; uma tarefa árdua ainda que se tratando somente da informação de textura! Os resultados aqui apresentados indicam, contudo, que a abordagem por *SFT* para reconstrução 3D é promissora, e apontam caminhos possíveis para investigação futura.

6.1 Trabalhos futuros

A área de *SFT* pode ser muito útil quando empregada em conjunto com outras informações para a reconstrução tridimensional. Por exemplo, em um sistema híbrido empregando visão estéreo e deformação da textura. Uma abordagem combinando diferentes técnicas de *Shape from X* é um tema promissor para estudos futuros.

Além deste, pode-se destacar outros pontos a serem pesquisados:

- Decomposição da transformação afim em *tilt* e *slant* usando um método diferente do *SVD*. Com isso espera-se reduzir o erro nesta etapa;
- Elaborar uma maneira de rejeitar automaticamente grupos de pontos de interesse que não propiciem uma boa estimativa da transformação afim. Forsyth [36] realiza algo nessa linha em seu trabalho;
- Desenvolver uma estimativa automática do ponto frontal, como no trabalho de Forsyth [36] ou de Loh [37];
- Investigar outras técnicas de estimativa da transformação afim [68];
- Desenvolver métodos baseados em projeção perspectiva, como por exemplo, o trabalho de Clerc e Mallat [32];

- Elaborar testes com os algoritmos cujos resultados possam ser avaliados numericamente, ao invés de visualmente. Um exemplo seria obter um objeto texturizado cujo mapa de profundidade é conhecido;
- Elaborar uma base de dados e um procedimento experimental para *benchmark* na área de *SFT*, o que facilita a comparação de trabalhos, auxiliando na evolução das técnicas.

7 Referências

-
- 1 JONG, S. A Study of Shape from X by Imaging System Parameters. **Tese de Doutorado**, Chung Yuan Christian University, Departamento de Engenharia Mecânica, 2000.
 - 2 HANSEN, C.; AYACHE, N.; LUSTMAN, F. Towards Real-Time Trinocular Stereo. **Proceedings of 2nd International Conference on Computer Vision**, p. 129-133, 1988.
 - 3 HORN, B. K. P. Shape from Shading: A Method for Obtaining the Shape of a Smooth Opaque Object from One View. **Technical Report**, AI-TR-232, MIT, 1970.
 - 4 WOODHAM, R. J. Photometric Method for Determining Surface Orientation from Multiple Images. **Optical Engineering**, Vol. 19, p. 139-144, 1980.
 - 5 CLARK, J. J. Active Photometric Stereo. **IEEE Proceedings of Computer Vision and Pattern Recognition (CVPR)**, p. 29-34, 1992.
 - 6 TOMASI, C.; KANADE, T. Shape and Motion from Image Streams: a Factorization Method (Full Report on the Orthographic Case). **Technical Report**, CMU-CS-91-132, Carnegie Mellon University, 1991.
 - 7 NAYAR, S.K.; NAKAGAWA, Y. Shape from Focus, **IEEE Transactions on Pattern Analysis and Machine Intelligence**, vol. 16, no. 8, p. 824-831, Aug., 1994.
 - 8 JARVIS, R. A. A perspective on Range Finding Techniques for Computer Vision. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, Vol. 5, p. 122-139, 1983.
 - 9 BANNO, A.; IKEUCHI, K. Shape Recovery of 3D Data Obtained from a Moving Range Sensor by Using Image Sequences. **IEEE International Conference on Computer Vision**, Vol. 1, p. 792-799, 2005.
 - 10 GIBSON, J. **The Perception of the Visual World**. Boston: Houghton Mifflin, 1950.
 - 11 BAJCSY, R.; LIEBERMAN, L. Texture gradient as a depth cue. **Computer Graphics and Image Processing**, Vol. 5, p. 52-67, 1976.
 - 12 VAN VLIET, L. J. **Windowed Fourier Transform**. Disponível em <<http://www.ph.tn.tudelft.nl/~lucas/education/tn254/index.html>>. Acesso em 10 Nov. 2005.
 - 13 STEVENS, K. A. The information content of texture gradients. **Biological Cybernetics**, Vol. 42, p. 95-105, 1981.

-
- 14 WITKIN, A. P. Recovering surface shape and orientation from texture. **Artificial Intelligence**. Vol. 17, p. 17-45, 1981.
 - 15 ALOIMONOS, J. Shape from texture. **Biological Cybernetics**. Vol. 58, p. 345-360, 1988.
 - 16 BLOSTEIN, D.; AHUJA, N. Shape from texture: integrating texture-element extraction and surface estimation. **IEEE Transactions on Pattern Analysis and Machine Intelligence**. Vol. 11, no. 12, p. 1233-1251, December 1989.
 - 17 KENDER, J. R. Shape from texture: An aggregation transform that maps a class of textures into surface orientation. **In: IJCAI79**, p. 475-480, 1979.
 - 18 OHTA, Y.; MAENOBU, K.; SAKAI, T. Obtaining surface orientation from texels under perspective projection. **In: IJCAI81**, p. 746-751, 1981.
 - 19 IKEUCHI, K. Shape from regular patterns. **Journal of Artificial Intelligence**, Vol. 22, p. 49-75, 1984.
 - 20 KANATANI, K. I.; CHOU, T. C. Shape from texture: General principle. **Journal of Artificial Intelligence**, Vol. 38, p. 1-48, February 1989.
 - 21 DAVIS, L. S.; JANOS, L.; DUNN, S. M. Efficient recovery of shape from texture. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, Vol. 5, 1983.
 - 22 BLAKE, A.; MARINOS, C. Shape from texture estimation, isotropy and moments. **Journal of Artificial Intelligence**. Vol. 45, p. 323-380, 1990.
 - 23 GARDING, J. Shape from texture and contour by weak isotropy. **Journal of Artificial Intelligence**, Vol. 64, p. 243-297, 1993.
 - 24 BROWN, L. G.; SHVAYTSER, H. Surface orientation from projective foreshortening of isotropic texture autocorrelation. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, Vol. 12, p. 584-588, 1990.
 - 25 KRUMM, J.; SHAFER, S. A. A characterizable shape-from-texture algorithm using the spectrogram. **In: Proceedings of the IEEE-SP International Symposium on Time-Frequency and Time-Scale Analysis**, p. 322-325, 1994.
 - 26 GARDING, J. Shape from texture for smooth curved surfaces in perspective projection. **Journal of Mathematical Imaging and Vision**, Vol. 2, p. 327-350, 1992.
 - 27 O'NEILL, B. **Elementary Differential Geometry**. Academic Press, Orlando, Florida, 1966.
 - 28 GARDING, J. Shape from texture for smooth curved surfaces. **In: European Conference on Computer Vision**, p. 630-638, 1992.
 - 29 SUPER, B. J.; BOVIK, A. C. Shape from texture using local spectral moments. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, Vol. 17, p. 333-343, April 1995
 - 30 ROSENHOLTZ, R.; MALIK, J. Computing Local Surface Orientation and Shape from Texture for Curved Surfaces. **International Journal of Computer Vision**, Vol. 23, p. 149-168, 1997.

-
- 31 RIBEIRO, E.; HANCOCK, E. R. Shape from Periodic Texture Using the Eigenvectors of Local Affine Distortion. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, Vol. 23, no. 12, December 2001.
 - 32 CLERC, M.; MALLAT, S. The texture gradient equation for recovering shape from texture. **IEEE Transactions on Pattern Analysis and Machine Intelligence**. Vol. 24, no. 4, p. 536-549, April 2002.
 - 33 CLERC, M.; MALLAT, S. Estimating Deformations of Stationary Processes. **Technical Report** 2000-192, CERMICS, L' Ecole Nationale des Ponts et Chaussées, 2000.
 - 34 LOH, A.M.; KOVESI, P. Estimation of surface normal of a curved surface using texture. **Proceedings of the Digital Image Computing - Techniques and Applications**, 2003.
 - 35 HUYNH, D. **Binary Images**. Disponível em <<http://undergraduate.csse.uwa.edu.au/units/233.412/>>. Acesso em 20 Jan 2005.
 - 36 FORSYTH, D. A. Shape from Texture without Boundaries. **Proceedings of European Conference on Computer Vision**, 2002.
 - 37 LOH, A. M.; HARTLEY, R. Shape from non-homogeneous, non-stationary, anisotropic, perspective texture. **Proceedings of the British Machine Vision Conference**, 2005.
 - 38 LOH, A. M.; ZISSERMAN, A. Estimating the affine transformation between textures. **In Proceedings of the Digital Image Computing: Techniques and Applications Conference**, 2005.
 - 39 FORSYTH, D. A.; PONCE, J. **Computer Vision - A Modern Approach**, Prentice Hall, 2003.
 - 40 GONZALEZ, R.G.; WOODS, R. E. **Digital Image Processing**. Prentice Hall, 2002.
 - 41 SHAPIRO, L. G.; STOCKMAN, G. C. **Computer Vision**, Prentice Hall, 2001.
 - 42 STEVENS, K. A. Slant-Tilt: The Visual Encoding of Surface Orientation. **Biological Cybernetics**, Vol. 46, no. 3, p. 183-195, 1983.
 - 43 GOLUB, G.; VAN LOAN, C. **Matrix Computations**, 3rd ed., Johns Hopkins Press, Baltimore, 1996.
 - 44 LOH, A. M.; ZISSERMAN, A. Estimating the affine transformation between textures. **In Proceedings of the Digital Image Computing: Techniques and Applications Conference**, 2005.
 - 45 LOH, A. M.; HARTLEY, R. Shape from non-homogeneous, non-stationary, anisotropic, perspective texture. **Proceedings of the British Machine Vision Conference**, 2005.
 - 46 GABOR, D. Theory of Communication. **Journal of the Institute of Electrical Engineering.**, Vol. 93, p. 429-457, 1946.

-
- 47 DAUGMAN, J. G.. Two dimensional spectral analysis of cortical receptive field profiles. **Vision Research**, Vol. 20, p. 847-856, 1980.
 - 48 DAUGMAN, J. G. Uncertainty relation for resolution in space, spatial frequency, and orientation optimised by two-dimensional cortical filters. **Journal of the Optical Society of America**, Vol. 2, p. 1160-1169, 1985.
 - 49 BOVIK, A. C.; CLARK, M.; GEISLER, W. S. Multichannel texture analysis using localized spatial filters. **IEEE Transactions Pattern Analysis and Machine Intelligence**, Vol. 12, p. 55-73, 1990.
 - 50 MOVELLAN, J. R. **Tutorial on Gabor Filters**, 2002. Disponível em: <<http://mplab.ucsd.edu/tutorials/tutorials.html>>. Acesso em: 10 Jan. 2005.
 - 51 SUPER, B. J.; BOVIK, A. C. Shape-from-texture by wavelet-based measurement of local spectral moments. **IEEE Proceedings of the Conference on Computer Vision and Pattern Recognition**, p. 296-301, 1992.
 - 52 HAVLICEK, J. P.; BOVIK, A. C.; MARAGOS, P. Modulation models for image processing and wavelet-based image demodulation. **Proceedings of 26th IEEE Asilomar Conference on Signals, Systems and Computers**, p. 805-810, 1992.
 - 53 HORN, B. K. P. **Robot Vision**. MIT Press, Cambridge, Massachusetts, 1986.
 - 54 MARKS, R. J. **Introduction to Shannon Sampling and Interpolation Theory**. Springer-Verlag, 1991.
 - 55 PRESS, W. H.; FLANNERY, B. P.; TEUKOLSKI, S. A; VETTERLING, W. T. **Numerical Recipes in C**, Cambridge University Press, 1988.
 - 56 RUSS, J. C. **The Image Processing Handbook**, 3 ed. CRC Press, 1998.
 - 57 FEITOSA, R. Q. **Notas de aula em Visão Computacional – Realce no Domínio da Frequência**, 2005. Disponível em <<http://www.ele.puc-rio.br/~visao/>>. Acesso em 20 de Agosto de 2005.
 - 58 DAUGMAN, J. **Continuous Mathematics**, 1999. Disponível em <<http://www.cl.cam.ac.uk/Teaching/1999/ContMaths/Notes.pdf>>. Acesso em 5 Jun 2005.
 - 59 ROUSSEUW, P. J.; LEROY, A. M. **Robust Regression and Outlier Detection**, Wiley, 1987.
 - 60 SCHAFFALITZKY, F.; ZISSERMAN, A. Viewpoint invariant texture matching and wide baseline stereo. In: **International Conference on Computer Vision**, Vol. 01, p. 636-643, 2001.
 - 61 LOWE, D. G. Distinctive image features from scale-invariant keypoints, **International Journal of Computer Vision**, Vol. 60, p. 91-110, 2004.
 - 62 LOWE, D. G. **Demo Software: SIFT Keypoint Detector**. Disponível em <<http://www.cs.ubc.ca/~lowe/keypoints/>>. Acesso em 10 Jan 2006.
 - 63 DUDA, R. O.; HART, P. E.; STORK; D. G. **Pattern Classification**. 2.ed. New York: Wiley-Interscience, 2001.

-
- 64 KOVESI, P. Surface Normals to Surfaces via Shapelets, **Proceedings Australia-Japan Advanced Workshop on Computer Vision**, Adelaide, 9-11 September 2003.
- 65 KOVESI, P. D. **MATLAB and Octave Functions for Computer Vision and Image Processing. School of Computer Science & Software Engineering.** Disponível em <http://www.csse.uwa.edu.au/~pk/research/matlabfns/>. Acesso em 28 Jan 2006.
- 66 BRODATZ, P. **Textures: A photographic album for artists and designers.** Dover Publications, New York, 1966.
- 67 GRAEFE, F.; SCHUMACHER, W; FEITOSA, R. Q.; DUARTE, D. M. Filled: video data based fill level detection of agricultural bulk freight. **ICINCO**, p. 439-442, 2005.
- 68 LUCCHESI, L. A Frequency Domain Technique Based on Energy Radial Projections for Robust Estimation of Global 2D Affine Transformations. **Computer Vision and Image Understanding**, Vol. 81, p. 72-116, Feb. 2001.

Apêndice I

Tabelas extras dos experimentos da estimação da transformação afim

variância	theta		erro médio	erro máx.	erro mín.	desvio-padrão do erro
0.00001	0°	<i>slant</i>	0,5	4,4	0,1	0,9
		<i>tilt</i>	3,8	51,0	0,1	8,1
0.00001	30°	<i>slant</i>	0,6	4,4	0,1	0,9
		<i>tilt</i>	4,2	56,2	0,1	9,5
0.00001	45°	<i>slant</i>	0,6	4,7	0,1	1,0
		<i>tilt</i>	4,0	49,7	0,2	8,4
0.00001	60°	<i>slant</i>	0,6	4,4	0,1	1,0
		<i>tilt</i>	4,0	52,2	0,1	8,8
0.00001	90°	<i>slant</i>	0,6	5,1	0,1	1,0
		<i>tilt</i>	4,1	52,6	0,1	9,0
0.0001	0°	<i>slant</i>	1,3	8,2	0,3	1,7
		<i>tilt</i>	6,8	58,7	0,3	10,9
0.0001	30°	<i>slant</i>	1,6	7,9	0,3	1,8
		<i>tilt</i>	6,3	46,8	0,3	9,9
0.0001	45°	<i>slant</i>	1,4	8,0	0,4	1,8
		<i>tilt</i>	5,8	51,0	0,4	9,4
0.0001	60°	<i>slant</i>	1,3	7,8	0,3	1,7
		<i>tilt</i>	5,9	48,8	0,5	9,2
0.0001	90°	<i>slant</i>	1,3	7,7	0,2	1,8
		<i>tilt</i>	6,2	51,4	0,4	10,0
0.001	0°	<i>slant</i>	3,6	14,4	1,2	3,0
		<i>tilt</i>	8,9	53,6	0,9	10,4
0.001	30°	<i>slant</i>	3,5	13,7	1,3	3,1
		<i>tilt</i>	9,0	45,8	1,2	10,0
0.001	45°	<i>slant</i>	3,9	14,0	1,4	3,1
		<i>tilt</i>	9,9	53,3	1,2	11,3
0.001	60°	<i>slant</i>	4,4	15,3	1,4	3,4
		<i>tilt</i>	10,3	45,6	1,1	10,5
0.001	90°	<i>slant</i>	3,4	13,7	1,1	3,1
		<i>tilt</i>	9,5	49,8	1,3	10,6

Tabela 5 - Erros da decomposição da transformação afim por *SVD* na presença de ruído.

Versão completa da Tabela 2.

variância	theta		erro médio	erro máx.	erro mín.	desvio-padrão do erro
0.01	0°	<i>slant</i>	9,1	25,8	3,6	6,1
		<i>tilt</i>	17,6	47,3	3,2	10,7
0.01	30°	<i>slant</i>	11,1	27,8	4,2	6,2
		<i>tilt</i>	15,8	53,3	3,0	10,6
0.01	45°	<i>slant</i>	10,1	25,4	3,9	5,9
		<i>tilt</i>	13,9	48,0	3,6	9,6
0.01	60°	<i>slant</i>	10,6	26,8	4,1	6,1
		<i>tilt</i>	15,8	49,4	3,3	10,1
0.01	90°	<i>slant</i>	9,3	26,6	3,5	6,3
		<i>tilt</i>	15,2	53,0	3,2	10,2
0.1	0°	<i>slant</i>	19,1	41,9	5,2	9,1
		<i>tilt</i>	21,6	48,7	7,8	7,7
0.1	30°	<i>slant</i>	25,8	48,7	9,1	10,6
		<i>tilt</i>	22,6	48,5	9,6	8,1
0.1	45°	<i>slant</i>	19,9	42,4	6,8	10,0
		<i>tilt</i>	23,7	48,3	7,4	8,2
0.1	60°	<i>slant</i>	25,1	49,6	7,2	11,7
		<i>tilt</i>	25,1	51,1	9,3	8,5
0.1	90°	<i>slant</i>	24,1	46,1	9,8	9,8
		<i>tilt</i>	23,1	54,3	10,9	8,4
1	0°	<i>slant</i>	37,4	62,5	24,1	10,1
		<i>tilt</i>	29,6	54,2	15,6	9,2
1	30°	<i>slant</i>	38,0	59,3	21,3	9,3
		<i>tilt</i>	30,4	49,0	18,8	6,4
1	45°	<i>slant</i>	39,6	58,4	25,8	8,4
		<i>tilt</i>	30,4	51,5	21,1	6,6
1	60°	<i>slant</i>	35,2	66,0	16,7	12,5
		<i>tilt</i>	29,7	55,5	13,8	9,1
1	90°	<i>slant</i>	38,1	56,4	27,2	6,7
		<i>tilt</i>	31,5	53,6	25,0	5,8

Tabela 6 – Continuação da Tabela 5.

Imagem	theta		erro médio	erro máx.	erro mín.	desvio-padrão do erro
D20	0°	<i>slant</i>	29,6	68,9	0,0	20,4
		<i>tilt</i>	38,7	90,0	0,0	22,3
D20	30°	<i>slant</i>	30,4	71,1	0,0	21,0
		<i>tilt</i>	34,0	88,4	0,0	25,5
D20	45°	<i>slant</i>	30,8	70,9	0,0	21,2
		<i>tilt</i>	33,3	89,8	0,0	26,1
D20	60°	<i>slant</i>	30,4	72,5	0,0	21,0
		<i>tilt</i>	33,1	86,9	0,0	24,9
D20	90°	<i>slant</i>	29,4	72,0	0,0	20,3
		<i>tilt</i>	35,3	90,0	0,0	19,6
D18	0°	<i>slant</i>	22,1	72,0	0,0	16,4
		<i>tilt</i>	34,7	81,0	0,0	19,7
D18	30°	<i>slant</i>	21,9	72,0	0,0	16,1
		<i>tilt</i>	28,8	89,6	0,1	19,4
D18	45°	<i>slant</i>	21,6	72,0	0,0	16,1
		<i>tilt</i>	25,6	89,7	0,0	19,8
D18	60°	<i>slant</i>	22,8	72,6	0,0	16,4
		<i>tilt</i>	20,3	74,8	0,0	17,3
D18	90°	<i>slant</i>	24,5	72,6	0,0	16,8
		<i>tilt</i>	24,5	90,0	0,0	21,2
D95	0°	<i>slant</i>	21,9	66,4	0,0	17,7
		<i>tilt</i>	33,2	89,2	0,0	25,6
D95	30°	<i>slant</i>	23,2	66,4	0,0	17,6
		<i>tilt</i>	31,8	86,9	0,0	20,2
D95	45°	<i>slant</i>	24,5	66,3	0,0	17,0
		<i>tilt</i>	29,1	89,9	0,0	18,0
D95	60°	<i>slant</i>	23,6	66,3	0,0	17,8
		<i>tilt</i>	27,0	88,5	0,0	20,3
D95	90°	<i>slant</i>	23,8	66,3	0,0	17,9
		<i>tilt</i>	27,7	90,0	0,1	20,8
D101	0°	<i>slant</i>	25,7	73,0	0,0	20,5
		<i>tilt</i>	46,5	89,9	0,0	25,4
D101	30°	<i>slant</i>	27,5	71,2	0,0	20,4
		<i>tilt</i>	34,9	89,9	0,1	22,0
D101	45°	<i>slant</i>	27,3	77,1	0,0	20,1
		<i>tilt</i>	28,3	89,5	0,0	23,1
D101	60°	<i>slant</i>	27,4	71,4	0,0	20,2
		<i>tilt</i>	31,2	88,5	0,0	22,8
D101	90°	<i>slant</i>	26,0	70,7	0,0	20,0
		<i>tilt</i>	33,8	90,0	0,1	22,6
D103	0°	<i>slant</i>	22,9	66,4	0,0	16,0
		<i>tilt</i>	39,3	89,9	0,0	26,3
D103	30°	<i>slant</i>	23,3	66,7	0,0	15,8
		<i>tilt</i>	32,8	89,8	0,0	19,8
D103	45°	<i>slant</i>	22,9	69,4	0,0	14,7
		<i>tilt</i>	27,4	89,3	0,1	22,9
D103	60°	<i>slant</i>	23,3	70,4	0,0	15,6
		<i>tilt</i>	27,9	88,7	0,1	19,8
D103	90°	<i>slant</i>	23,7	71,0	0,0	15,8
		<i>tilt</i>	27,9	90,0	0,0	25,5

Tabela 7 - Erros calculados (em graus) para o método de Ruth Rosenholtz.

Imagem	theta		erro médio	erro máx.	erro mín.	desvio-padrão do erro
D20	0°	<i>slant</i>	14,2	50,8	0,0	7,5
		<i>tilt</i>	15,5	87,7	0,0	21,6
D20	30°	<i>slant</i>	9,7	43,4	0,0	8,0
		<i>tilt</i>	17,7	57,9	0,0	14,3
D20	45°	<i>slant</i>	10,0	60,8	0,0	9,7
		<i>tilt</i>	16,7	63,5	0,0	12,6
D20	60°	<i>slant</i>	10,9	62,3	0,0	10,7
		<i>tilt</i>	15,2	75,4	0,0	12,9
D20	90°	<i>slant</i>	11,0	54,5	0,0	9,8
		<i>tilt</i>	14,0	90,0	0,0	15,4
D18	0°	<i>slant</i>	17,7	66,3	0,0	15,6
		<i>tilt</i>	19,5	75,5	0,0	15,4
D18	30°	<i>slant</i>	19,0	66,3	0,0	15,8
		<i>tilt</i>	14,0	84,4	0,0	21,4
D18	45°	<i>slant</i>	18,8	66,3	0,0	15,3
		<i>tilt</i>	17,8	71,1	0,1	20,9
D18	60°	<i>slant</i>	17,6	66,3	0,0	14,8
		<i>tilt</i>	23,2	74,4	0,0	20,0
D18	90°	<i>slant</i>	12,7	66,3	0,0	14,4
		<i>tilt</i>	20,0	90,0	0,0	17,8
D95	0°	<i>slant</i>	9,7	48,1	0,0	8,2
		<i>tilt</i>	23,0	89,7	0,0	18,5
D95	30°	<i>slant</i>	10,1	38,8	0,0	7,8
		<i>tilt</i>	17,6	74,0	0,0	18,5
D95	45°	<i>slant</i>	9,0	28,0	0,0	6,7
		<i>tilt</i>	13,6	69,7	0,0	16,9
D95	60°	<i>slant</i>	7,8	28,0	0,0	6,1
		<i>tilt</i>	13,4	60,0	0,2	12,5
D95	90°	<i>slant</i>	12,2	35,7	0,0	7,0
		<i>tilt</i>	10,4	90,0	0,0	16,0
D101	0°	<i>slant</i>	8,3	32,8	0,0	7,2
		<i>tilt</i>	5,8	89,4	0,0	10,5
D101	30°	<i>slant</i>	8,1	29,2	0,0	6,8
		<i>tilt</i>	9,6	42,8	0,0	9,5
D101	45°	<i>slant</i>	8,5	27,4	0,0	7,0
		<i>tilt</i>	11,7	52,6	0,0	12,7
D101	60°	<i>slant</i>	9,0	28,6	0,0	6,7
		<i>tilt</i>	13,6	67,5	0,0	16,8
D101	90°	<i>slant</i>	11,2	30,6	0,0	6,7
		<i>tilt</i>	16,8	90,0	0,0	22,7
D103	0°	<i>slant</i>	14,0	54,7	0,0	12,1
		<i>tilt</i>	25,1	85,7	0,0	18,3
D103	30°	<i>slant</i>	10,6	65,7	0,0	10,7
		<i>tilt</i>	14,7	64,9	0,0	13,0
D103	45°	<i>slant</i>	9,1	49,2	0,0	10,2
		<i>tilt</i>	10,6	61,6	0,0	11,5
D103	60°	<i>slant</i>	9,2	54,0	0,0	10,6
		<i>tilt</i>	11,3	61,3	0,0	11,3
D103	90°	<i>slant</i>	16,6	62,6	0,0	13,3
		<i>tilt</i>	20,8	90,0	0,0	24,1

Tabela 8 - Erros calculados (em graus) para o método de Angeline Loh.

Anexo I

Derivação das equações de Super e Bovik

Neste anexo são apresentadas as derivações para obtermos a equação (22) do método de Super e Bovik apresentado no Capítulo 4. As derivações aqui encontradas são oriundas do artigo e apresentadas para uma melhor compreensão do texto.

Através das equações dos momentos de segunda ordem e de um modelo de projeção ortográfico alcança-se a equação que relaciona os momentos de segunda ordem na superfície (no espaço objeto, sem deformação) e no plano imagem.

O primeiro passo para obter a eq. (22) é definir o modelo de projeção como uma transformação afim:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos \sigma \cos \tau & -\sin \tau \\ \cos \sigma \sin \tau & \cos \tau \end{bmatrix} \begin{bmatrix} x_s \\ y_s \end{bmatrix} \quad (38)$$

onde x e y são as coordenadas na imagem e x_s e y_s na superfície, σ é o *slant* e τ é o *tilt*. Essa notação com o índice s será empregada em todas as fórmulas aqui apresentadas.

Estendendo esta projeção para o domínio da frequência, onde temos a inversa transposta, como na eq. (26), temos:

$$\begin{bmatrix} u_s \\ v_s \end{bmatrix} = \begin{bmatrix} \cos \sigma \cos \tau & \cos \sigma \sin \tau \\ -\sin \tau & \cos \sigma \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} \quad (39)$$

Que nos resulta:

$$\begin{aligned} u_s &= u \cos \sigma \cos \tau + v \cos \sigma \sin \tau \\ v_s &= -u \sin \tau + v \cos \sigma \end{aligned} \quad (40)$$

A projeção da convolução da imagem com um filtro, $A_i(x,y)$ na eq. (16), é particularmente simples neste trabalho pois são empregados filtros com

gaussianas isotrópicas, o que faz com que a projeção sofra apenas efeito de compressão e não de rotação, resultando:

$$A_{si}(x_s, y_s) = \frac{1}{\cos \sigma} A_i(x, y) \quad (41)$$

Reescrevendo a eq. (17) para a superfície, incluindo a normalização para efeitos de contraste e iluminação, dividindo por $\sum_i A_i^2(x, y)$ temos:

$$\begin{aligned} a_s(x_s, y_s) &= \sum_i U_{si}^2 A_{si}^2(x_s, y_s) / \sum_i A_{si}^2(x_s, y_s) \\ b_s(x_s, y_s) &= 2 \sum_i U_{si} V_{si} A_{si}^2(x_s, y_s) / \sum_i A_{si}^2(x_s, y_s) \\ c_s(x_s, y_s) &= \sum_i V_{si}^2 A_{si}^2(x_s, y_s) / \sum_i A_{si}^2(x_s, y_s) \end{aligned} \quad (42)$$

Realizando uma troca de variáveis, fazendo U_{si} e V_{si} iguais a u_s e v_s da eq.(40), e substituindo A_{si} por A_i da eq. (41), temos:

$$a_s(x_s, y_s) = \sum_i (u \cos \sigma \cos \tau + v \cos \sigma \sin \tau)^2 (\cos \sigma)^{-2} A_i^2(x, y) / \sum_i (\cos \sigma)^{-2} A_i^2(x, y)$$

$$a_s(x_s, y_s) = \sum_i \left(\frac{u^2 \cos^2 \sigma \cos^2 \tau + 2uv \cos^2 \sigma \cos \tau \sin \tau + v^2 \cos^2 \sigma \sin^2 \tau}{(\cos \sigma)^{-2} A_i^2(x, y)} \right) / \sum_i (\cos \sigma)^{-2} A_i^2(x, y)$$

$$\begin{aligned} a_s(x_s, y_s) &= \cos^2 \sigma \cos^2 \tau \left[\sum_i u^2 (\cos \sigma)^{-2} A_i^2(x, y) / \sum_i (\cos \sigma)^{-2} A_i^2(x, y) \right] + \\ &\cos^2 \sigma \cos \tau \sin \tau \left[\sum_i 2uv (\cos \sigma)^{-2} A_i^2(x, y) / \sum_i (\cos \sigma)^{-2} A_i^2(x, y) \right] + \\ &\cos^2 \sigma \sin^2 \tau \left[\sum_i v^2 (\cos \sigma)^{-2} A_i^2(x, y) / \sum_i (\cos \sigma)^{-2} A_i^2(x, y) \right] \end{aligned}$$

$$a_s(x_s, y_s) = a(x, y) \cos^2 \sigma \cos^2 \tau + b(x, y) \cos^2 \sigma \cos \tau \sin \tau + c(x, y) \cos^2 \sigma \sin^2 \tau$$

De forma semelhante são calculados b_s e c_s , que resultam na eq. (22).

Livros Grátis

(<http://www.livrosgratis.com.br>)

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)
[Baixar livros de Literatura de Cordel](#)
[Baixar livros de Literatura Infantil](#)
[Baixar livros de Matemática](#)
[Baixar livros de Medicina](#)
[Baixar livros de Medicina Veterinária](#)
[Baixar livros de Meio Ambiente](#)
[Baixar livros de Meteorologia](#)
[Baixar Monografias e TCC](#)
[Baixar livros Multidisciplinar](#)
[Baixar livros de Música](#)
[Baixar livros de Psicologia](#)
[Baixar livros de Química](#)
[Baixar livros de Saúde Coletiva](#)
[Baixar livros de Serviço Social](#)
[Baixar livros de Sociologia](#)
[Baixar livros de Teologia](#)
[Baixar livros de Trabalho](#)
[Baixar livros de Turismo](#)