

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.

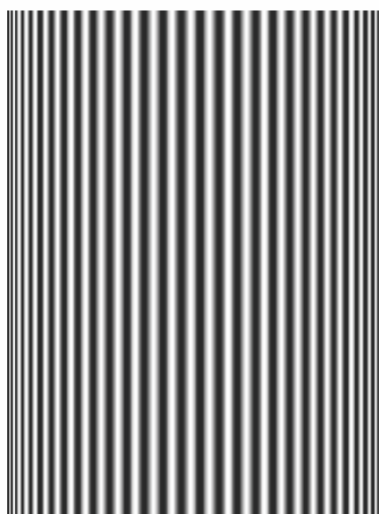


Figura 12 - Cilindro sintético com textura senoidal aplicada.

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:

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

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.

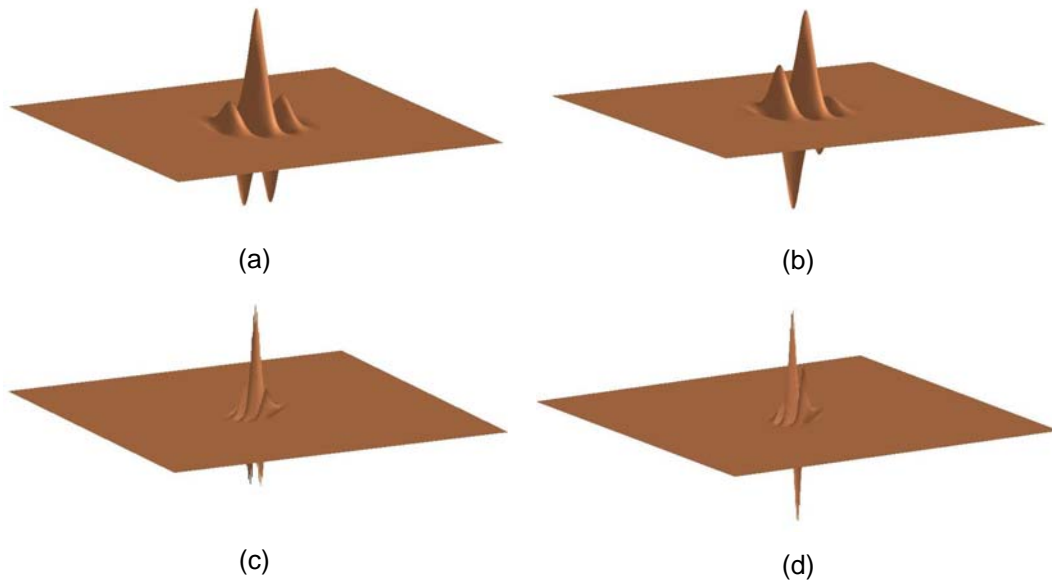


Figura 13 - Exemplos de Filtros de Gabor no domínio espacial. (a) e (b), parte real e imaginária, respectivamente, de um filtro tendo $F=12$ ciclos/imagem, $\kappa=1$ e $\Phi=0^\circ$. (c) e (d), parte real e imaginária, respectivamente, de um filtro tendo $F=32$ ciclos/imagem, $\kappa=1/2$ e $\Phi=45^\circ$.

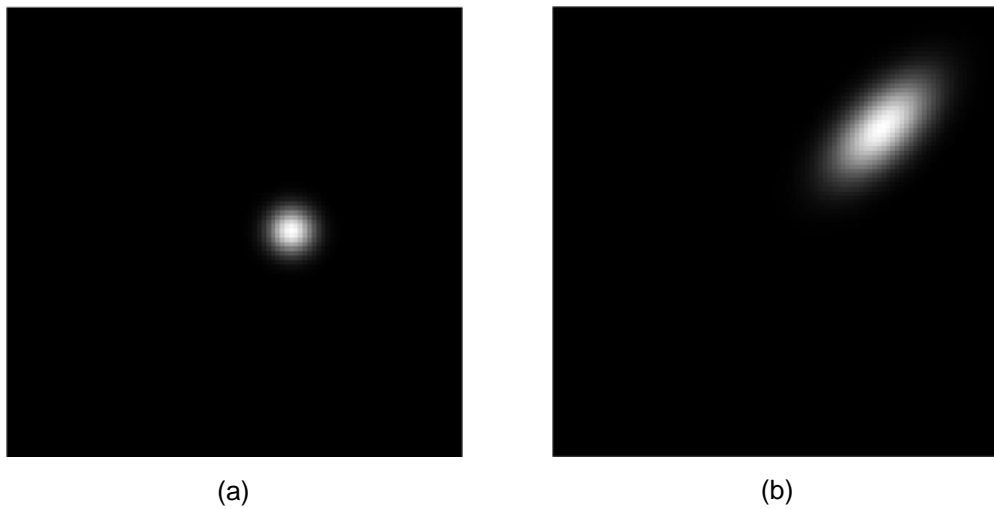


Figura 14 - Transformadas de Fourier dos Filtros de Gabor. (a) $F=12$ ciclos/imagem, $\kappa=1$ e $\Phi=0^\circ$. (b) $F=32$ ciclos/imagem, $\kappa=1/2$ e $\Phi=45^\circ$.

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]:

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

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 $1/F_1$, 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.

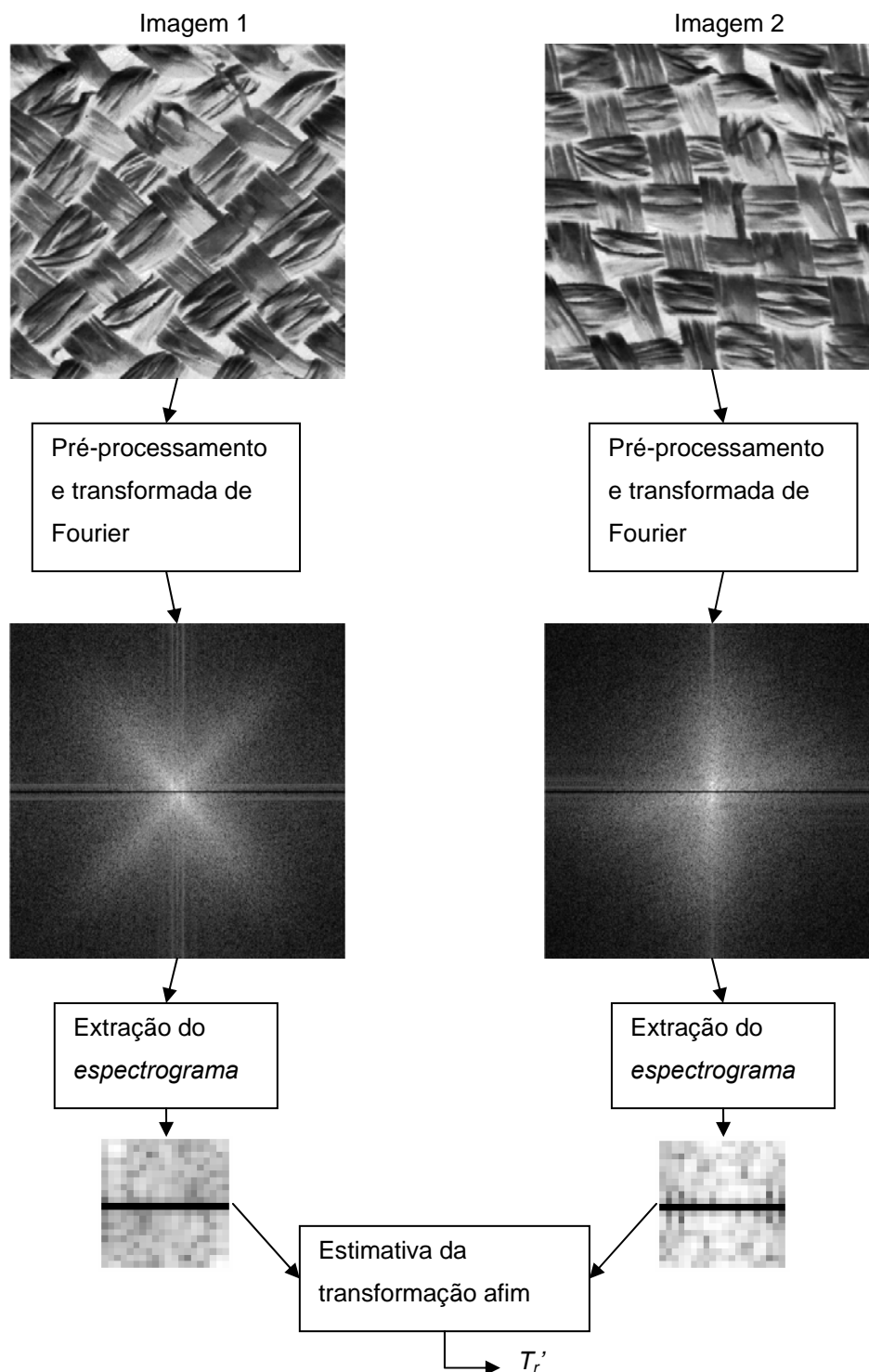


Figura 15 - Diagrama do método de Rosenholtz e Malik.

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:

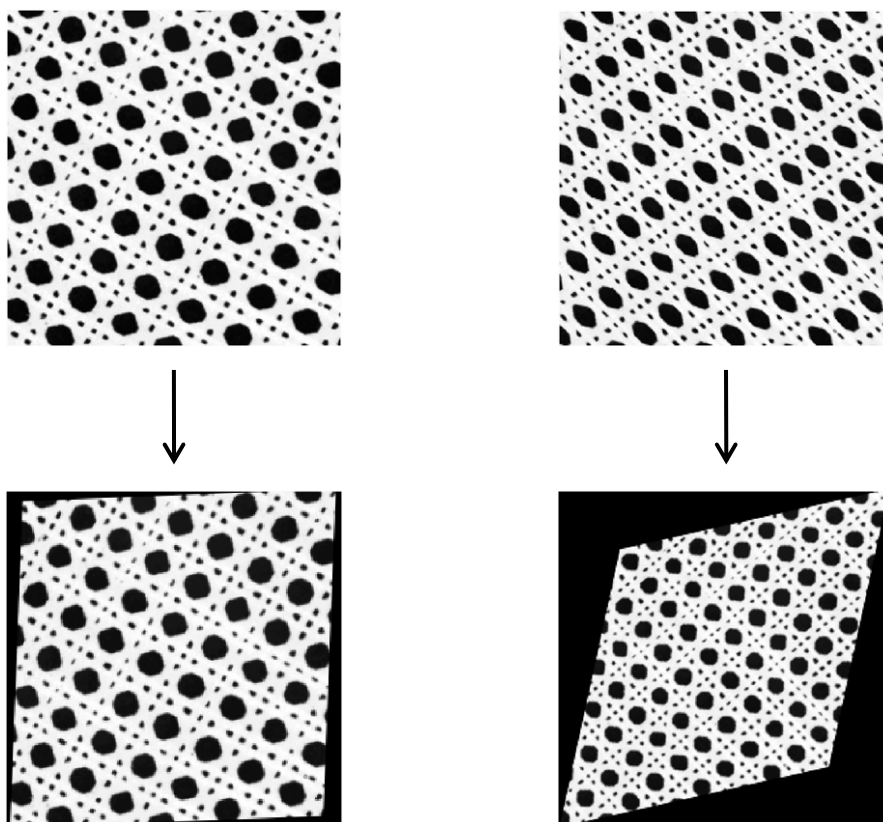


Figura 16 - Transformação de uma textura em sua versão isotrópica.

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.

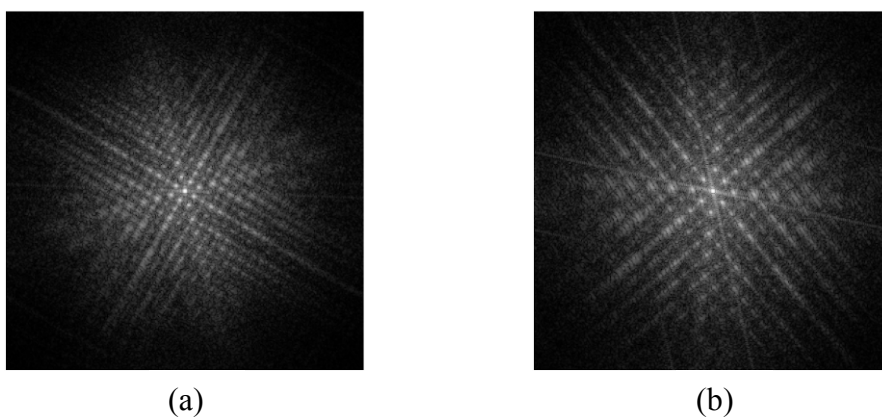
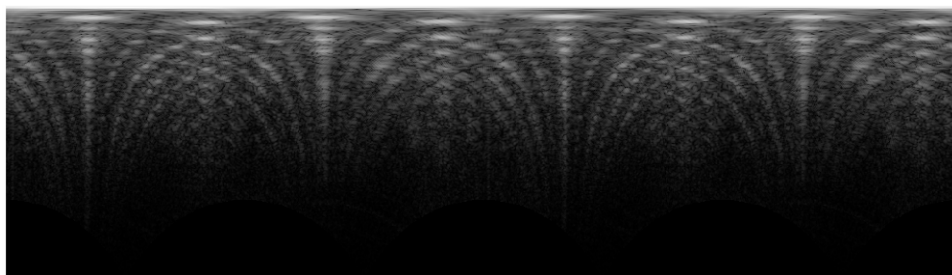
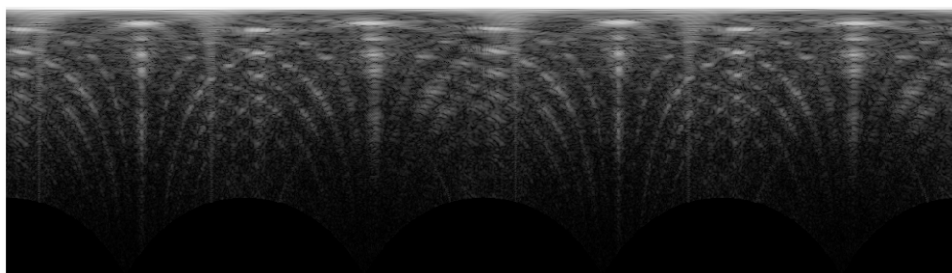


Figura 17 - Transformada de Fourier da versão isotrópica da primeira imagem (a) e da segunda (b).

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.



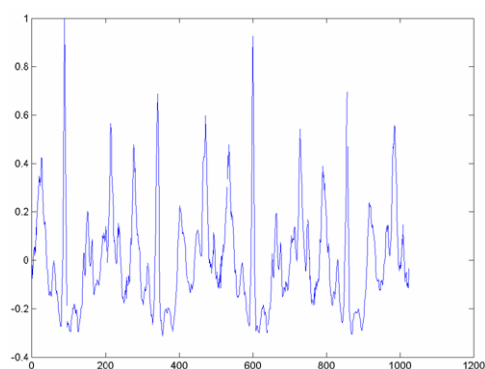
(a)



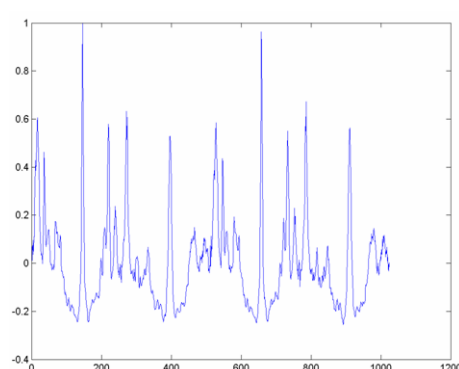
(b)

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.

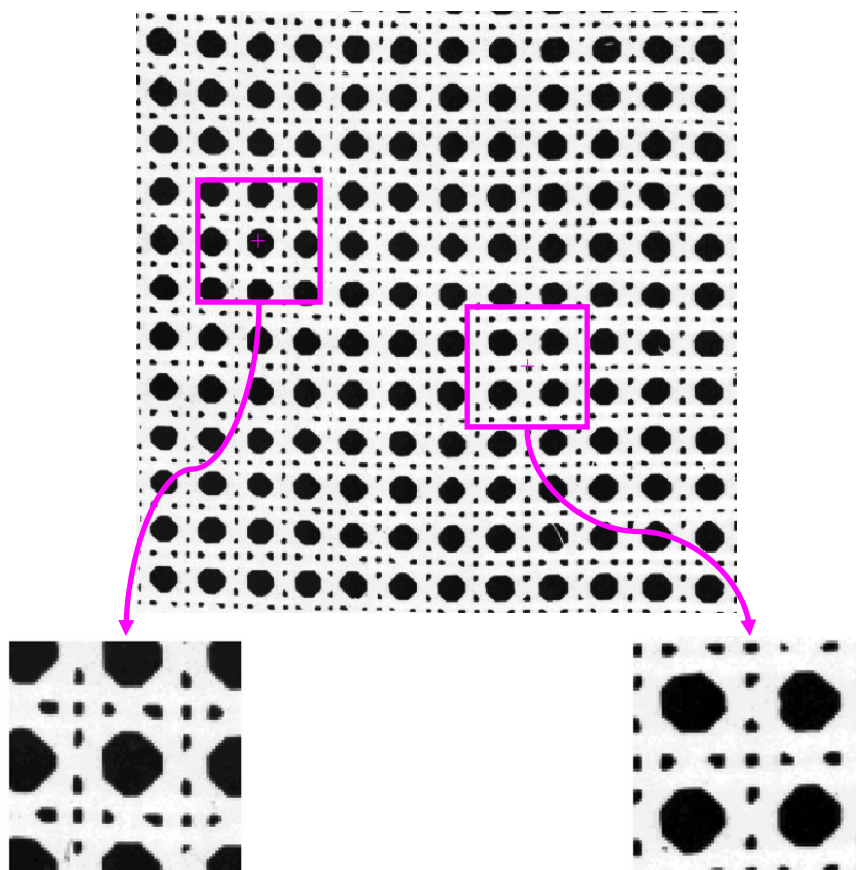


Figura 20 – Exemplo de extração de recortes de uma imagem.

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.

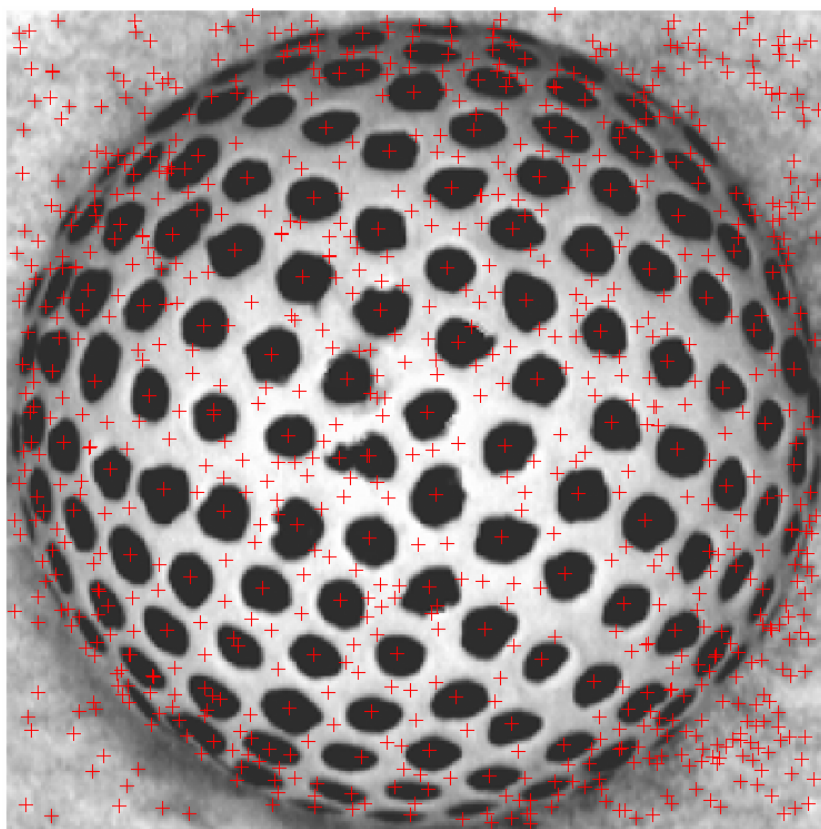


Figura 21 – Exemplo de detecção de pontos de interesse. As marcações “+” representam os pontos de interesse detectados pelo método de Lowe.

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.

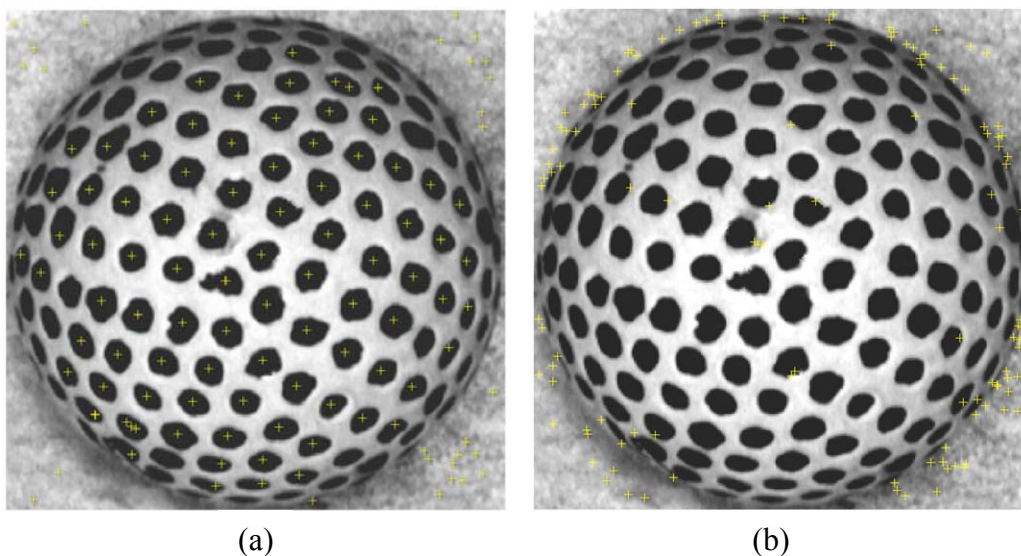


Figura 22 – Grupos de pontos de interesse encontrados pelo método de Lowe e segmentados por *K-means*. (a) grupo de pontos que representam bem os elementos de textura. (b) grupo que representa mal os elementos de textura (concentrados nas bordas da bola de golfe).

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.

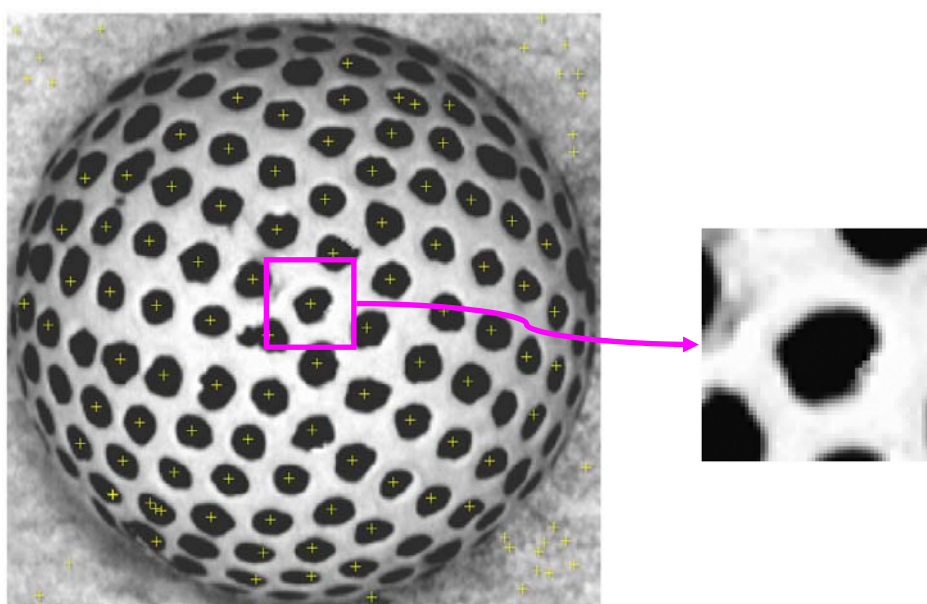


Figura 23 – Exemplo de modelo frontal extraído da imagem.

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.

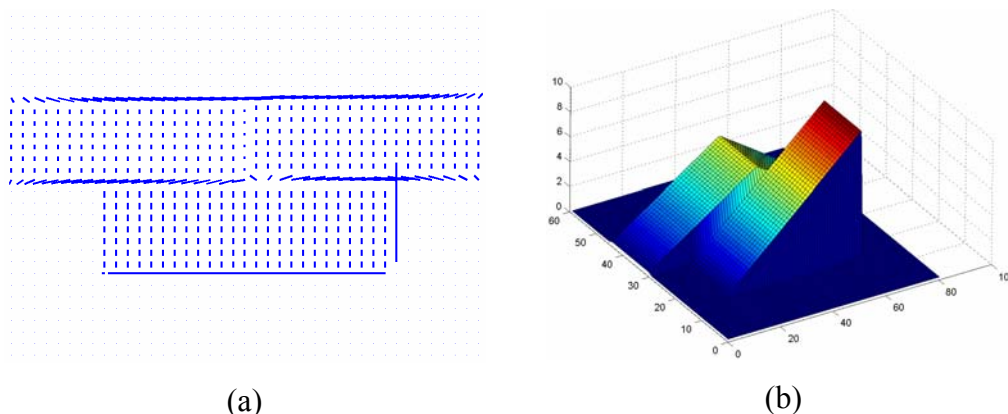


Figura 24 - (a) Mapa de agulhas. (b) Modelo 3D reconstruído usando método de Kovesi.