

**Universidade de São Paulo
Instituto de Física de São Carlos**

Wesley Nunes Gonçalves

**Análise de texturas estáticas e dinâmicas e suas
aplicações em biologia e nanotecnologia**

São Carlos

2013

Wesley Nunes Gonçalves

Análise de texturas estáticas e dinâmicas e suas aplicações em biologia e nanotecnologia

Tese apresentada ao Programa de Pós-Graduação em Física do Instituto de Física de São Carlos da Universidade de São Paulo, para obtenção do título de Doutor em Ciências.

Área de Concentração: Física Aplicada
Opção: Física Computacional

Orientador: Prof. Dr. Odemir Martinez Bruno

Versão Corrigida
(Versão original disponível na Unidade que aloja o Programa)

São Carlos

2013

AUTORIZO A REPRODUÇÃO E DIVULGAÇÃO TOTAL OU PARCIAL DESTE TRABALHO, POR QUALQUER MEIO CONVENCIONAL OU ELETRÔNICO PARA FINS DE ESTUDO E PESQUISA, DESDE QUE CITADA A FONTE.

Ficha catalográfica elaborada pelo Serviço de Biblioteca e Informação do IFSC,
com os dados fornecidos pelo(a) autor(a)

Nunes Gonçalves, Wesley

Análise de texturas estáticas e dinâmicas e suas aplicações em biologia e nanotecnologia / Wesley Nunes Gonçalves; orientador Odemir Martinez Bruno - versão corrigida -- São Carlos, 2013.
187 p.

Tese (Doutorado - Programa de Pós-Graduação em Física Aplicada Computacional) -- Instituto de Física de São Carlos, Universidade de São Paulo, 2013.

1. Análise de texturas. 2. Texturas dinâmicas. 3. Caminhadas determinísticas. 4. Dimensão fractal. I. Martinez Bruno, Odemir, orient. II. Título.

AGRADECIMENTOS

A Deus por tudo que tens feito na minha vida.

Aos meus pais Osmar e Marlene, e aos meus irmãos Diogo e Pâmela, pelo incentivo e amor que manifestaram ao longo da minha vida.

À Jociane pelo apoio e amor incondicional.

Ao meu orientador Odemir Martinez Bruno, por participar ativamente na minha formação acadêmica, pela orientação e amizade.

Ao Ernesto Chaves Pereira de Souza e Aleksandro Zimer pelos experimentos e imagens de nanotecnologia.

À Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) e ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) pelo apoio financeiro.

Aos amigos de grupo e república pelos momentos de alegria, discussões e trocas de conhecimentos.

Ao Instituto de Física de São Carlos (IFSC-USP), aos professores e funcionários e a todos que, direta ou indiretamente, colaboraram neste trabalho.

RESUMO

GONÇALVES, W. N. *Análise de texturas estáticas e dinâmicas e suas aplicações em biologia e nanotecnologia*. 2013. 187 p. Tese (Doutorado em Ciências) – Instituto de Física de São Carlos, Universidade de São Paulo, São Carlos, 2013.

A análise de texturas tem atraído um crescente interesse em visão computacional devido a sua importância na caracterização de imagens. Basicamente, as pesquisas em texturas podem ser divididas em duas categorias: texturas estáticas e texturas dinâmicas. As texturas estáticas são caracterizadas por variações de intensidades que formam um determinado padrão repetido espacialmente na imagem. Por outro lado, as texturas dinâmicas são padrões de texturas presentes em uma sequência de imagens. Embora muitas pesquisas tenham sido realizadas, essa área ainda se encontra aberta a estudos, principalmente em texturas dinâmicas por se tratar de um assunto recente e pouco explorado. Este trabalho tem como objetivo o desenvolvimento de pesquisas que abrangem ambos os tipos de texturas nos âmbitos teórico e prático. Em texturas estáticas, foram propostos dois métodos: (i) baseado em caminhadas determinísticas parcialmente auto-repulsivas e dimensão fractal – (ii) baseado em atividade em redes direcionadas. Em texturas dinâmicas, as caminhadas determinísticas parcialmente auto-repulsivas foram estendidas para sequências de imagens e obtiveram resultados interessantes em reconhecimento e segmentação. Os métodos propostos foram aplicados em problemas da biologia e nanotecnologia, apresentando resultados interessantes para o desenvolvimento de ambas as áreas.

PALAVRAS-CHAVE: Análise de texturas. Texturas dinâmicas. Caminhadas determinísticas. Dimensão fractal.

ABSTRACT

GONÇALVES, W. N. *Static and dynamic texture analysis and their applications in biology and nanotechnology*. 2013. 187 p. Tese (Doutorado em Ciências) – Instituto de Física de São Carlos, Universidade de São Paulo, São Carlos, 2013.

Texture analysis has attracted an increasing interest in computer vision due to its importance in describing images. Basically, research on textures can be divided into two categories: static and dynamic textures. Static textures are characterized by intensity variations which form a pattern repeated in the image spatially. On the other hand, dynamic textures are patterns of textures present in a sequence of images. Although many studies have been carried out, this area is still open to study, especially in dynamic textures since it is a recent and little-explored subject. This study aims to develop research covering both types of textures in theoretical and practical fields. In static textures, two methods were proposed: (i) based on deterministic partially self-avoiding walks and fractal dimension – (ii) based on activity in directed networks. In dynamic textures, deterministic partially self-avoiding walks were extended to sequences of images and obtained interesting results in recognition and segmentation. The proposed methods were applied to problems of biology and nanotechnology, presenting interesting results in the development of both areas.

KEYWORDS: Texture analysis. Dynamic textures. Deterministic walks. Fractal dimension.

LISTA DE FIGURAS

- 2.1 Exemplos de seis texturas dinâmicas: (a) fumaça, (b) cachoeira, (c) passagem, (d) água com chuva, (e) escada rolante, e (f) aro de bicicleta em movimento. p. 38
- 3.1 Exemplo de três CDPARs iniciadas em diferentes cidades. As cidades pertencentes à parte transiente e ao atrator estão em verde e azul, respectivamente. p. 43
- 3.2 Distribuição conjunta do tempo de transiente τ e período do atrator ρ obtidos em uma configuração com $N = 1000$ cidades e dimensão $d = 2$. Embora diferentes valores de memórias tenham sido utilizados, o aspecto geral de ambas as distribuições são similares. p. 44
- 3.3 Exemplo de caminhada determinística parcialmente auto-repulsiva utilizando a regra *min* e tamanho de memória $\mu = 2$. Cada pixel da imagem é representado por um círculo e a intensidade de cada pixel é dado dentro do círculo. O peso de conexão é dado pelo módulo da diferença entre intensidades. O transiente é dado pelos círculos em laranja, enquanto que o atrator é dado pelos círculos em verde. p. 46
- 3.4 Exemplos de distribuições conjuntas obtidas para três padrões de texturas utilizando memória $\mu = 1$ e regra de movimentação $r = \text{min}$ p. 48
- 3.5 Atratores gerados em uma imagem de textura utilizando a regra *min* e diferentes valores de μ . (a) Imagem Original; (b) $\mu = 1$; (c) $\mu = 3$; (d) $\mu = 7$. p. 49
- 3.6 Atratores gerados em uma imagem de textura utilizando a regra *max* e diferentes valores de μ . (a) Imagem Original; (b) $\mu = 1$; (c) $\mu = 3$; (d) $\mu = 7$. p. 49
- 3.7 Exemplos de CDPARs em que a parte cinza representa o transiente, enquanto que a parte listrada representa o atrator. p. 50
- 3.8 Uma lista encadeada é utilizada para armazenar uma CDPAR. Cada posição da lista contém o pixel visitado. p. 50
- 3.9 Comparação entre diferentes partes de uma lista encadeada. Um pixel repetido ao menos três vezes na lista pode configurar a existência de uma sequência de pixels repetidos. p. 50

3.10	Lista encadeada associando as posições em que cada pixel aparece na caminhada, tornando possível o cálculo eficiente do número de vezes que um determinado pixel aparece na caminhada.	p. 51
3.11	Gráfico do tamanho médio das caminhadas para diferentes tamanhos de memória μ . O comportamento da curva é linear.	p. 52
3.12	Sequência de conjuntos E_j demonstrando o conjunto de Cantor. O conjunto inicia com um intervalo unitário e, em seguida, retira-se o terço do meio.	p. 54
3.13	Sequência de conjuntos E_j demonstrando a curva de Koch. Inicia-se com um intervalo unitário, remove-se o terço do meio e substitui-se por dois lados de um triângulo.	p. 54
3.14	Exemplo de uma imagem em níveis de cinza e a sua conversão para o espaço tridimensional. Cada pixel $p_i = (x_i, y_i)$ com intensidade $I(p_i)$ é convertido em um ponto tridimensional $s_i = (x_i, y_i, I(p_i))$	p. 57
3.15	Ilustração do método BoxCounting sobre pontos de uma superfície obtida de uma imagem.	p. 57
3.16	A dimensão fractal é computada utilizando-se regressão linear do gráfico $\log r \times \log A(r)$. Dado o coeficiente angular da reta α , a dimensão fractal é estimada por $D_M = 3 - \alpha$	p. 58
3.17	Ilustração do processo de dilatação de pontos de uma superfície obtida de uma imagem. Cada ponto é dilatado por uma esfera com um determinado raio r	p. 59
4.1	Exemplo de duas CDPARs utilizando $\mu = 2$ e regra de movimentação <i>min</i> . Os pixels são representados por círculos e a intensidade de cada pixel é dada dentro do círculo. Embora os tempos de transiente $\tau = 3$ e $\tau = 4$ (em laranja) sejam diferentes, as duas caminhadas produzem o mesmo atrator de tamanho $\rho = 3$ (em verde).	p. 63
4.2	Exemplo do cálculo da relevância de cada pixel. Cada círculo representa um pixel da imagem e a intensidade é dada dentro dele (número na parte inferior). Considerando os pixels em pretos como atratores, a relevância de cada pixel é dada na sua parte superior.	p. 66

4.3	O novo processo de dilatação utilizando a relevância de cada pixel aos atratores. A cada passo, os pixels são adicionados aos atratores de acordo com a sua relevância.	p. 67
4.4	Atratores obtidos com tamanho de memória $\mu = 2$ e duas regras de movimentação, $r = max$ e $r = min$. Os atratores estão desenhados em verde.	p. 70
4.5	Exemplos de atratores (pixels em verde) para $r = max$ e (a) $\mu = 1$, (b) $\mu = 2$, (c) $\mu = 3$, (d) $\mu = 4$	p. 71
4.6	Exemplos de dilatação de (a) atratores obtidos por caminhadas utilizando $\mu = 5$ e $r = min$ e valores de raio (b) $\epsilon = 5$, (c) $\epsilon = 10$, (d) $\epsilon = 15$	p. 73
4.7	Exemplos de dilatação para diferentes classes de texturas e valores de μ e r . O eixo x representa o raio de dilatação e o eixo y representa o número de pixels pertencentes ao conjunto de atratores. Ambos os eixos estão em escala logarítmica.	p. 74
4.8	Exemplos de algumas classes da base de dados Brodatz. Cada uma das 111 classes possui 10 amostras de 200×200 pixels.	p. 76
4.9	Exemplos de classes da base de dados vistex. Cada uma das 54 classes possui 16 amostras de 128×128 pixels.	p. 77
4.10	Taxa de classificação correta para valores máximos de dilatação n . Foram utilizados tamanhos de memórias 1, 2, 3 e concatenação entre as regras de movimentação min e max	p. 79
5.1	Exemplo de uma imagem mapeada em uma rede regular. Cada nó é conectado com os demais nós dentro de um raio $r = 2$	p. 84
5.2	Exemplo da função ϕ aplicada em uma rede regular para valores de limiar: (a) 10, (b) 50, e (c) 150.	p. 85
5.3	Atividade de cada nó/pixel para valores de limiar $t = 30, 100, 240$ e raios $r = \sqrt{2}, \sqrt{5}, \sqrt{9}$ em uma imagem de textura.	p. 88
5.4	Atividade de cada nó/pixel para diferentes texturas. A rede direcionada foi obtida para $t = 100$ e $r = \sqrt{2}$	p. 89
5.5	Os histogramas para três classes de textura. A rede direcionada foi obtida com parâmetros $r = \sqrt{8}$ e $t = 80$	p. 90

5.6	Tamanho médio de caminhadas aleatórias executadas em 2292 imagens de tamanho 128×128 pixels. As curvas correspondem à diferentes valores de raios r	p. 92
5.7	Histogramas para imagens de texturas com duas transformações afins: (a) rotação e (b) translação. As redes foram obtidas usando-se $r = \sqrt{8}$ e $t = 80$	p. 93
5.8	Histogramas para imagens de texturas em diferentes perspectivas. As redes foram obtidas usando-se $r = \sqrt{8}$ e $t = 80$	p. 94
5.9	Histograma para sete imagens de textura com diferentes luminâncias. A imagem original é transformada pela adição da determinada luminância em todos os pixels.	p. 95
5.10	Exemplos de amostras da base de dados usptex. Esta base é composta por 191 classes, cada uma contendo 12 amostras de 128×128 pixels.	p. 97
5.11	Avaliação do limiar inicial t_0 para as três base de dados. O gráfico corresponde a taxa de classificação correta para t_0 variando de 1 a 100.	p. 99
5.12	Avaliação do limiar de incremento para as três base de dados. O gráfico corresponde a taxa de classificação correta para t_i variando de 10 à 100.	p. 100
5.13	Avaliação do limiar final para as três base de dados. O gráfico corresponde a taxa de classificação correta para x variando de 1 à 8.	p. 101
5.14	Avaliação do conjunto de raios para as três base de dados. O gráfico corresponde a taxa de classificação correta para a concatenação de $\sqrt{1}$ a $\sqrt{1} \dots \sqrt{10}$	p. 102
5.15	Comparação entre o tamanho médio das caminhadas em redes direcionadas e não-direcionadas com raio $r = \sqrt{1}$	p. 103
5.16	Gráfico das duas primeiras componentes da análise de componentes principais na base de dados Brodatz. A variância acumulada das duas componentes é 0.59.	p. 106
6.1	Sumarização do método proposto. Passo 1: o vídeo contendo as texturas dinâmicas é dividido em três planos ortogonais. Passo 2: para cada plano, são executadas CDPARs com diferentes tamanhos de memória. Passo 3: um vetor de características para cada distribuição conjunta é calculado. Passo 4: os vetores são concatenados para compor o vetor de características final do método proposto.	p. 108

6.2	Exemplo dos três planos ortogonais obtidos dos vídeos: (a) galhos balançando devido ao vento e (b) fonte de água.	p. 109
6.3	As três classes utilizadas para avaliar características extraídas do plano XY : (a) flores, (b) escada rolante, (c) grama. Cada classe contém 8 amostras com diferentes padrões de movimento. Nesses vídeos sintéticos, uma textura dinâmica é utilizada como fundo e a outra é utilizada para simular o padrão de movimento.	p. 114
6.4	As quatro classes utilizadas para avaliar características extraídas dos planos XT e YT : (a) movimento circular, (b) movimento linear, (c) movimento aleatório, (d) sem movimento. Cada classe contém seis amostras com diferentes aparências. Nos vídeos sintéticos, uma textura dinâmica é utilizada como fundo e a outra é utilizada para simular o padrão de movimento.	p. 116
6.5	Vetores projetados pela análise de componentes principais utilizando diferentes planos e configurações de classes: (a) XY , (b) XT , (c) YT , (d) XY , XT , YT . As amostras são divididas em classes apenas para fins de visualização, dado que a análise de componentes principais não utiliza qualquer informação das classes.	p. 117
6.6	Os quatro padrões de movimento rotacionados utilizados para avaliar a invariância à rotação do método proposto. O movimento linear é realizado em quatro direções: $0^\circ, 90^\circ, 180^\circ, 270^\circ$	p. 117
6.7	Características extraídas dos planos XT e YT e, então, projetadas pela análise de componentes principais utilizando classes de texturas dinâmicas rotacionadas.	p. 118
6.8	Quatro classes utilizadas para avaliar a capacidade do método em caracterizar múltiplas texturas dinâmicas. Cada classe contém três amostras.	p. 119
6.9	Características extraídas dos planos XY , XT e YT e, então, projetadas pela análise de componentes principais, utilizando vídeos compostos de múltiplas texturas dinâmicas.	p. 119
6.10	Exemplos de texturas dinâmicas da base de dados dyntex. Essa base é composta por 123 classes, cada uma contendo 10 amostras. Os vídeos possuem pelo menos 250 quadros com dimensão de 400×300 pixels. Todos os quadros foram convertidos para níveis de cinza.	p. 120

6.11	Taxa de classificação correta para tamanhos de memória μ e regras de movimentação r . Cada gráfico apresenta os resultados para os planos na base de dados dyntex.	p. 122
6.12	Taxa de classificação correta para diferentes conjuntos de tamanho de memória obtida pelo método proposto na base de dados dyntex. Cada gráfico corresponde a um plano ortogonal.	p. 123
6.13	Taxa de classificação correta para diferentes concatenações de planos na base de dados dyntex.	p. 124
6.14	Passos do método proposto de segmentação de texturas dinâmicas. (1) O vídeo é dividido em blocos sobrepostos; (2) um vetor de características é obtido para cada bloco por meio do método de caracterização de texturas dinâmicas; (3) o algoritmo k-means agrupa os vetores e então a segmentação do vídeo é obtida.	p. 126
6.15	Estratégias para extração de blocos de um vídeo. (a) Exemplos de blocos com dimensões $p \times p \times q$ pixels. (b) Exemplos de blocos com dimensão temporal $q = T$	p. 127
6.16	A primeira coluna apresenta a partição real, a segunda coluna apresenta a partição inicial para o método de inicialização do k-means, e a terceira coluna apresenta exemplos dos vídeos. As linhas representam os vídeos compostos por 2, 3 ou 4 texturas dinâmicas, respectivamente.	p. 128
6.17	Exemplos de texturas dinâmicas reais da base de dados dyntex. Todos os vídeos possuem 50 quadros com dimensão de 720×576 pixels.	p. 129
6.18	Número de iterações M versus a média do <i>rand index</i> para a base dados com $k = 4$. A média do <i>rand index</i> estabiliza para $M > 30$	p. 130
6.19	Segmentação final obtida por diferentes métodos de inicialização. A primeira linha corresponde aos primeiros quadros dos vídeos e as demais linhas seguem: inicialização manual, múltiplas execuções, PCA, KKZ, Astrahan e Aleatório.	p. 133
6.20	Segmentação final obtida por diferentes métodos de segmentação de texturas dinâmicas. Cada coluna corresponde a um método de segmentação na seguinte ordem: método proposto, MDT e LDT. As linhas correspondem a diferentes vídeos.	p. 135

6.21	Segmentação de textura dinâmica real utilizando o método proposto para diferentes valores de k	p. 136
6.22	Comparação de segmentação de texturas dinâmicas reais. A coluna à esquerda apresenta os resultados de segmentação para o método proposto, enquanto que a coluna à direita apresenta os resultados obtidos por Fazekas.	p. 138
6.23	Comparação de segmentação de texturas dinâmicas reais. A coluna à esquerda apresenta os resultados de segmentação para o método proposto, enquanto que a coluna à direita apresenta os resultados obtidos por Fazekas.	p. 139
6.24	Histograma para amostras de texturas dinâmicas reais com diferentes padrões de movimento.	p. 140
6.25	Exemplo de regiões descartadas com base na entropia do histograma $\psi_{\mu=1, r=max}^{XT}$. Os pixels cuja entropia é maior que 1,42 foram descartados.	p. 140
7.1	Processo de extração de amostras de texturas de uma folha. Este processo consiste em extrair janelas aleatórias e sem sobreposição.	p. 145
7.2	Amostra de textura para cada uma das 40 espécies de plantas. Na base de dados, cada espécie contém 30 amostras.	p. 146
7.3	Variação de algumas amostras de seis espécies de plantas. A variação depende das condições em que a folha foi submetida.	p. 147
7.4	Matriz de confusão do método proposto no Capítulo 4 para a base de dados de folhas. A porcentagem de classificação correta é de 92.08%.	p. 148
7.5	Matriz de confusão do método proposto no Capítulo 5 para a base de dados de folhas. A porcentagem de classificação correta é de 96.58%.	p. 148
7.6	Exemplos de vídeos da base de dados de tráfego. Esta base é composta por 254 vídeos divididos em três classes. Cada vídeo possui entre 42 e 52 quadros com resolução de 320×240	p. 150
7.7	Taxa de classificação correta para diferentes concatenações de tamanhos de memória. Cada gráfico corresponde aos resultados obtidos por cada plano ortogonal.	p. 152
7.8	Taxa de classificação correta para diferentes concatenações de planos ortogonais na base de tráfego de carros.	p. 152

7.9	Exemplos de vídeos dos três grupos obtidos pelo k-means usando características extraídas pelo método proposto. Cada linha corresponde a um grupo.	p. 154
7.10	Comparação usando o índice do grupo e a velocidade medida na rodovia. Existe uma correspondência entre ambas as medidas.	p. 154
7.11	Primeiro quadro de cada condição experimental para verificação da corrosão na mistura.	p. 157
7.12	Ilustração do esquema para extrair amostras da sequência de imagens. Cada amostra possui 200×200 pixels com os mesmos 20.000 quadros da sequência original.	p. 157
7.13	Projeção das duas primeiras componentes obtidas pela aplicação do PCA nos vetores de características das amostras de quatro tipos de corrosão. . .	p. 158

LISTA DE TABELAS

4.1	Taxa de classificação correta para diferentes tamanhos de memória μ e regras de movimentação r utilizados nas CDPARs. A dilatação máxima n foi 17.	p. 78
4.2	Taxa de classificação correta para diferentes tamanhos de memória utilizados nas CDPARs. A dilatação máxima n é igual a 17.	p. 79
4.3	Resultados comparativos para diferentes métodos de caracterização de texturas na base de dados Brodatz.	p. 80
4.4	Resultados comparativos para diferentes métodos de caracterização de texturas na base de dados Vistex.	p. 81
5.1	Resultados experimentais para métodos de texturas na base de dados Brodatz. A tabela apresenta o número de características e a taxa de classificação correta.	p. 104
5.2	Resultados experimentais para métodos de texturas na base de dados vistex. A tabela apresenta o número de características e a taxa de classificação correta.	p. 105
5.3	Resultados experimentais para métodos de texturas na base de dados usptex. A tabela apresenta o número de características e a taxa de classificação correta.	p. 105
6.1	Resultados comparativos para métodos da literatura em classificação de texturas dinâmicas da base de dados dyntex.	p. 124
6.2	Média e desvio padrão do número de iterações que o k-means precisa para convergir quando inicializado por diferentes métodos.	p. 130
6.3	Média e desvio padrão do <i>rand index</i> para diferentes métodos de inicialização do k-means.	p. 131
6.4	Comparações entre métodos de inicialização do k-means para vídeos com $k = 3$ utilizando testes estatísticos.	p. 132
6.5	Comparações entre métodos de inicialização do k-means para vídeos com $k = 4$ utilizando testes estatísticos.	p. 132
6.6	Comparação entre os métodos de segmentação de texturas dinâmicas para bases com $k = 2, 3, 4$	p. 134

7.1	Resultados experimentais para os métodos de textura utilizando a base de dados de folhas.	p.146
7.2	Comparação de métodos existentes na base de tráfego de carros. A tabela apresenta o número de características, a taxa de classificação correta para os classificadores KNN e SVM e a média da taxa.	p.153

LISTA DE ALGORITMOS

1	Algoritmo para encontrar os atratores $A_{\mu,r}^{p_i}$ na imagem.	p. 64
2	Algoritmo básico para calcular a relevância de cada pixel da imagem de textura.	p. 68
3	Algoritmo de classificação k-vizinhos mais próximos.	p. 177
4	Algoritmo k-means.	p. 181
5	Método aleatório.	p. 182
6	Método Astrahan.	p. 182
7	Método KKZ proposto por Katsavounidis.	p. 183
8	Múltiplas execuções do k-means.	p. 183
9	Método de inicialização baseado em PCA.	p. 184
10	Método de inicialização manual.	p. 184

SUMÁRIO

1	Introdução	p. 29
1.1	Contextualização e Motivação	p. 29
1.2	Objetivos	p. 31
1.3	Organização do Texto	p. 31
2	Análise de Texturas	p. 33
2.1	Aplicações de Texturas	p. 34
2.2	Tipos de Texturas	p. 35
2.2.1	Texturas Estáticas	p. 35
2.2.2	Texturas Dinâmicas	p. 37
3	Metodologias	p. 41
3.1	Caminhadas Determinísticas Parcialmente Auto-repulsivas - CDPAR	p. 41
3.1.1	CDPARs em imagens	p. 45
3.1.2	Dinâmica da CDPAR	p. 47
3.1.3	Algoritmo para Detecção de Atratores	p. 49
3.1.4	Complexidade Computacional	p. 51
3.2	Dimensão Fractal	p. 52
3.2.1	Dimensão de Hausdoff	p. 55
3.2.2	Dimensão Fractal em Imagens	p. 56
3.2.2.1	BoxCounting	p. 56
3.2.2.2	Bouligand-Minkowski	p. 58
4	Combinando Dimensão Fractal e CDPAR para Análise de Texturas Estáticas	p. 61

4.1	Motivação	p. 61
4.2	Método Proposto	p. 62
4.2.1	Atratores em Imagens de Textura	p. 62
4.2.2	Dimensão Fractal de Atratores	p. 64
4.2.3	Vetor de Características	p. 68
4.3	Análise do Método Proposto	p. 69
4.3.1	Atratores em Imagens de Textura	p. 70
4.3.2	Dilatação de Atratores em Imagens de Textura	p. 72
4.4	Experimentos e Resultados	p. 72
4.4.1	Configuração dos Experimentos	p. 75
4.4.2	Avaliação dos Parâmetros	p. 78
4.4.3	Comparação com Métodos da Literatura	p. 80
4.5	Conclusão	p. 81

5 Atividade em Redes Direcionadas aplicada ao Reconhecimento de Texturas Estáticas p. 83

5.1	Método Proposto	p. 83
5.1.1	Modelando Imagens em Redes Direcionadas	p. 84
5.1.2	Atividade em Imagens de Textura	p. 86
5.1.3	Vetor de Características	p. 87
5.1.4	Complexidade Computacional	p. 91
5.2	Análise de Invariância	p. 92
5.2.1	Invariância a Transformações Afins	p. 93
5.2.2	Invariância à Mudança de Contraste Global	p. 94
5.3	Experimentos	p. 96
5.4	Resultados Experimentais	p. 98
5.4.1	Análise de Parâmetros	p. 98

5.4.2	Redes Direcionadas vs Não-direcionadas	p. 103
5.4.3	Comparação com Métodos da Literatura	p. 104
5.5	Conclusão	p. 105
6	Caracterização de Texturas Dinâmicas	p. 107
6.1	Método proposto	p. 107
6.1.1	CDPARs em Vídeo	p. 108
6.1.2	Vetor de Características	p. 111
6.2	Análise do Método Proposto	p. 112
6.2.1	Influência dos Planos Ortogonais	p. 113
6.2.2	Análise de Invariância à Rotação	p. 115
6.2.3	Caracterização de Múltiplas Texturas Dinâmicas	p. 118
6.3	Reconhecimento de Texturas Dinâmicas	p. 120
6.3.1	Avaliação dos Parâmetros	p. 121
6.3.2	Comparação com Métodos da Literatura	p. 122
6.4	Segmentação de Texturas Dinâmicas	p. 125
6.4.1	Método Proposto para Segmentação de Texturas Dinâmicas	p. 125
6.4.2	Configuração dos Experimentos	p. 127
6.4.3	Análise dos Métodos de Inicialização do K-means	p. 128
6.4.4	Comparação com Métodos de Segmentação	p. 134
6.4.5	Resultados em Texturas Dinâmicas Reais	p. 136
6.4.6	Partes não-informativas do Vídeo	p. 137
6.5	Complexidade Computacional	p. 141
6.6	Conclusão	p. 141
7	Aplicações	p. 143
7.1	Aplicação I: Reconhecimento de Plantas	p. 143
7.1.1	Base de Dados	p. 144

7.1.2	Classificação de Espécies de Plantas	p. 144
7.2	Aplicação II: Análise do Tráfego de Veículos	p. 149
7.2.1	Base de Vídeos	p. 150
7.2.2	Classificação do Tráfego de Veículos	p. 151
7.2.2.1	Comparação entre Métodos	p. 151
7.2.3	Agrupamento de Tráfego de Carros	p. 153
7.3	Aplicação III: Classificação das condições da corrosão de aço	p. 155
7.3.1	Base de dados	p. 156
7.3.2	Classificação do tipo de corrosão	p. 156
8	Conclusão	p. 159
8.1	Contribuições	p. 160
8.2	Trabalhos Futuros	p. 161
8.3	Trabalhos Gerados	p. 161
	REFERÊNCIAS	p. 165
	Apêndice A – Fundamentação Teórica	p. 175
A.1	Classificação de Padrões	p. 175
A.1.1	Validação-cruzada	p. 176
A.1.2	K-Vizinhos mais Próximos	p. 177
A.1.3	Máquina de Vetores de Suporte	p. 178
A.1.3.1	Fronteira de Decisão e Margem	p. 178
A.1.3.2	Estimando uma SVM Linear	p. 179
A.2	Agrupamento de Dados	p. 180
A.2.1	K-means	p. 180
A.2.2	Métodos de Inicialização do K-means	p. 181
A.2.3	Validação do Agrupamento	p. 185

A.2.3.1	Rand index	p.185
A.2.3.2	Coeficiente de Silhueta	p.186

Introdução

1.1 Contextualização e Motivação

Análise de textura é uma importante área de pesquisa em visão computacional com diversas aplicações potenciais, tais como análise de imagens de satélite (1), inspeção industrial (2), análise de imagens médicas (3), biologia (4) e geologia (5). Apesar dessa área ter sido estudada por décadas, não existe uma definição formal de textura que seja clara, não ambígua e amplamente aceita na literatura. A textura está presente na maioria dos objetos do nosso cotidiano e pode ser facilmente compreendida, pelo sistema de visão natural, como um padrão repetido sobre uma superfície em sua forma exata ou com pequenas variações.

Embora a textura seja geralmente descrita como um padrão repetitivo, ela é muito mais complexa do que essa definição. A textura poderia ser formada por um conjunto de padrões combinados de uma maneira específica que pode ou não estar em escalas diferentes. De fato, a ausência total de padrões pode ser considerada uma textura, como os diferentes tipos de ruídos. Apesar da dificuldade na definição do termo textura, as suas características estão diretamente relacionadas com as propriedades físicas dos objetos, tornando-a um atributo importante para uma grande gama de aplicações em visão computacional.

As pesquisas em análise de texturas podem ser divididas, de acordo com o tipo de textura analisada, em duas modalidades: texturas estáticas e texturas dinâmicas. As texturas estáticas compreendem a repetição de padrões em uma única imagem. Recentemente, devido aos avanços dos equipamentos de captura e do processamento computacional, o termo textura dinâmica têm emergido na comunidade de visão computacional. As texturas dinâmicas correspondem aos padrões que apresentam repetição no espaço e no tempo em sequências de imagens (vídeos). Este trabalho aborda ambos os tipos de texturas.

Particularmente, são abordados os métodos de texturas estáticas das categorias baseadas em modelos e em agentes. Os métodos baseados em modelos, em específico os modelos

fractais, extraem suas características projetando a imagem em um espaço tridimensional (mapeando os níveis de cinza no eixo z). Essa superfície é dilatada por uma esfera com um determinado raio. A área da superfície após a dilatação para diferentes raios é utilizada para descrever a imagem. Embora essa estratégia tenha obtido resultados interessantes, ela não considera explicitamente a relação entre níveis de cinza da textura. Por outro lado, os métodos baseados em agentes geralmente utilizam o tamanho da caminhada na imagem como a principal característica para descrever a textura. Certamente, o tamanho da caminhada ou partes dela não extrai toda a riqueza de detalhes de uma imagem de textura.

Para superar as limitações acima, este trabalho propôs dois métodos para análise de texturas estáticas. O primeiro método é baseado na dimensão fractal de caminhadas determinísticas. A grande vantagem desse método é a capacidade de descrever as caminhadas realizadas pelos agentes por meio da dimensão fractal. O segundo método caracteriza a textura estática por meio da estimação da atividade em redes direcionadas. Os experimentos demonstraram que a atividade extrai as principais características da textura. Ambos os métodos apresentaram resultados promissores quando comparados com métodos da literatura em bases de imagens amplamente aceitas na literatura.

Em texturas dinâmicas, grandes avanços têm sido reportados durante os últimos anos (6–7). Entretanto, a maioria dos métodos da literatura apresenta pelo menos uma das seguintes desvantagens: (i) não fornecem uma combinação explícita entre características espaciais e temporais, isto é, alguns métodos de texturas dinâmicas (e.g. métodos baseados em movimento) estimam apenas características baseado em movimento e analisam essas sequências de padrões – (ii) não fornecem características invariantes a transformações na imagem, como a rotação – (iii) não modelam múltiplas texturas dinâmicas, pois em aplicações reais, texturas dinâmicas com diferentes propriedades podem estar presentes no mesmo quadro, por exemplo, fonte de água e árvores ao fundo. Entretanto, alguns métodos não modelam essa situação adequadamente (e.g. modelos que utilizam um único sistema dinâmico linear) – (iv) são computacionalmente inviáveis, isto é, alguns métodos, como os métodos que utilizam sistemas dinâmicos lineares, são impraticáveis para vídeos com mais de 100 quadros.

Para superar essas questões, um método baseado em agentes foi proposto para análise de texturas dinâmicas. Esse método estende os agentes para que eles sejam aplicados em três planos ortogonais do vídeo: XY, XT e YT. Assim, os agentes aplicados no plano XY descrevem características espaciais, enquanto que os agentes aplicados nos planos XT e YT extraem características temporais. Como apresentado nos experimentos, esse método apresenta resultados interessantes no reconhecimento e segmentação de texturas dinâmicas, superando as limitações citadas acima.

1.2 Objetivos

O objetivo principal deste projeto é a caracterização de texturas estáticas e dinâmicas. Para essa caracterização, propôs-se o estudo e a implementação de novos métodos baseados em modelos e agentes. Os métodos propostos foram avaliados em aplicações de biologia vegetal e nanotecnologia, tais como a identificação automática de espécies de plantas e a classificação do tipo de corrosão em aço. A aplicação dos métodos em ambas as áreas tem como motivação a validação em casos reais e ao mesmo tempo auxiliar o desenvolvimento de duas importantes áreas da ciência.

Resumidamente, os objetivos deste trabalho foram:

- Investigar e analisar os métodos de texturas estáticas e dinâmicas.
- Propor formalmente métodos para texturas estáticas baseado em modelos e agentes.
- Propor um método para texturas dinâmicas tendo como base as caminhadas determinísticas parcialmente auto-repulsivas.
- Comparar e avaliar os métodos propostos com o estado da arte.
- Aplicar os métodos em problemas reais das áreas de biologia vegetal e nanotecnologia.

1.3 Organização do Texto

Este texto está organizado da seguinte maneira:

Capítulo 2: este capítulo discute a análise de texturas, os dois tipos (texturas estáticas e dinâmicas) e suas aplicações.

neste capítulo são descritos os principais conceitos necessários para o entendimento desse trabalho. Em resumo, são apresentados os métodos de reconhecimento de padrões e agrupamento de dados.

Capítulo 3: as caminhadas determinísticas parcialmente auto-repulsivas e a dimensão fractal, duas metodologias que servem como base dos métodos propostos, são apresentadas em detalhes neste capítulo.

Capítulos 4 e 5: nestes capítulos são apresentados os dois métodos para caracterização de texturas estáticas: um método baseado em dimensão fractal de caminhadas determinísticas e outro método baseado em atividade em redes direcionadas. Experimentos foram realizados em base de dados amplamente aceitas na literatura e seus resultados foram comparados com o estado da arte.

Capítulo 6: o método proposto para texturas dinâmicas é apresentado em detalhes neste capítulo. Após a sua análise em problemas comumente encontrados em texturas dinâmicas, ele foi avaliado em duas aplicações: reconhecimento e segmentação de texturas dinâmicas. Em ambas as aplicações, o método é comparado com o estado da arte por meio de base de dados amplamente utilizadas na literatura.

Capítulos 7: cada seção deste capítulo descreve uma aplicação dos métodos propostos: reconhecimento de espécies de plantas, análise do tráfego de veículos e classificação das condições da corrosão de aço.

Capítulo 8: por fim, as conclusões deste trabalho são discutidas neste capítulo. Além disso, são apresentados os trabalhos futuros e os artigos publicados e em redação.

Apêndice A: este apêndice apresenta a fundamentação teórica necessária para o entendimento deste trabalho, incluindo os temas de classificação de padrões e agrupamento de dados.

Análise de Texturas

A textura é um termo muito utilizado em visão computacional e implicitamente compreensível pelos seres humanos. Embora a textura seja naturalmente compreendida, sua definição formal é uma tarefa difícil. Essa dificuldade reflete em um grande número de definições encontradas na literatura. Embora algumas definições sejam perceptivelmente motivadas, a maioria das definições descreve a textura com base na aplicação particular e, portanto, não há uma definição única aceita pela comunidade.

Julesz foi um dos primeiros pesquisadores a fornecer uma definição de textura nos anos 60 e 70 sua textura (8–10). Sua conjectura era que duas texturas fornecem padrões visuais parecidos se elas possuem estatísticas de primeira e segunda ordem idênticas. Entretanto, contraexemplos foram, mais tarde, construídos pelo próprio Julesz e demais pesquisadores (10). Tais contraexemplos possuem estatísticas de segunda ordem idênticas, mas são distinguíveis entre si.

Outra definição, talvez a mais aceita hoje em dia, é inspirada na resposta das células do córtex visual. Essas células respondem aos estímulos primitivos, como barras, bordas, pontos, etc, em diferentes frequências e orientações. De fato, os experimentos psicofísicos sugerem que o cérebro realiza uma análise espacial da frequência e orientação das imagens formadas na retina. Valois et al (11) estudaram o cérebro de macacos e concluíram que as células do córtex visual respondem às faixas com certa frequência e orientação. Hubel e Wiesel (12) demonstraram que essas células se desenvolvem após o nascimento em resposta aos estímulos encontrados no mundo real. Estas e outras descobertas inspiraram a descrição da textura com base na frequência e na resposta dos filtros (13).

Além das definições acima, a literatura apresenta muitas outras de definições e as mais clássicas são listadas abaixo:

- Tamura et al (14): "A estrutura da textura é simplesmente atribuída aos padrões repetitivos em que os elementos ou primitivas são dispostos com uma regra de posicionamento."
- Sklansky (15): "Uma região em uma imagem possui textura constante se um conjunto de estatísticas locais ou outras propriedades locais são constantes, variando suavemente

ou aproximadamente periódico.”

- Haralick (16): "Uma imagem de textura é descrita pelo número e tipos de suas primitivas e a organização espacial ou disposição de suas primitivas."
- Hawkins (17): "A textura depende de três ingredientes: (i) uma ordem local repetida sobre uma região — (ii) a ordem consiste no arranjo não aleatório de partes elementares — (iii) as partes são entidades uniformes com aproximadamente as mesmas dimensões em qualquer lugar dentro da região da textura."

2.1 Aplicações de Texturas

A textura tem sido utilizada em uma grande variedade de aplicações, uma vez que a maioria das imagens reais contém texturas e, portanto, a sua análise se torna um componente principal em tarefas de visão computacional. Devido a sua importância, as texturas se tornaram uma subárea da visão computacional, sendo possível identificar três problemas principais: classificação, segmentação e síntese de texturas. Esses três problemas são discutidos abaixo.

- **Classificação:** uma aplicação imediata é o reconhecimento de imagens utilizando propriedades das texturas. O objetivo da classificação é produzir um mapeamento que receba uma imagem de textura de entrada e retorne, como saída, uma das categorias de textura predefinidas. A classificação de texturas possui uma ampla gama de aplicações, que inclui controle de qualidade de defeitos, diagnósticos de imagens médicas e sensoriamento remoto.
- **Segmentação:** esse problema consiste em dividir uma imagem em regiões de acordo com sua textura. Portanto, o objetivo da segmentação é produzir um mapa de regiões similares da imagem. Em geral, assume-se que nenhuma informação sobre as categorias das texturas esteja disponível.
- **Síntese:** esse problema é frequentemente enfrentado em aplicações de compressão de imagens e em computação gráfica. Dada uma amostra finita de uma textura, o objetivo

é extrair as principais características e, então, sintetizar outras amostras com a mesma textura. Os algoritmos de síntese de texturas são utilizados para obter uma imagem de saída que atende a dois requisitos principais: (i) a imagem de saída deve ter o tamanho definido pelo usuário – (ii) a imagem de saída deve ser similar à amostra de entrada.

2.2 Tipos de Texturas

As pesquisas em análise de texturas podem ser divididas basicamente em duas modalidades: texturas estáticas e texturas dinâmicas. Ambas as texturas, incluindo os trabalhos correlatos, são descritas abaixo em detalhes.

2.2.1 Texturas Estáticas

As texturas estáticas são tradicionalmente utilizadas como descritores e compreendem a repetição de padrões em uma única imagem. A principal característica que descreve esse tipo de textura é a regularidade estatística de regiões presentes na imagem. Tais regiões são compostas por padrões com diferentes características que dependem das propriedades da superfície dos objetos e das condições do ambiente. Exemplos de texturas estáticas incluem imagens de pele de zebras, conjunto de pedras, casca das árvores, tecido, parede, tabuleiro de xadrez, mármore, entre outros.

A literatura apresenta uma grande quantidade de abordagens para análise de texturas estáticas. Essas abordagens, descritas abaixo, podem ser divididas em cinco categorias: (i) estrutural – (ii) estatística – (iii) espectral – (iv) baseada em modelos – (v) baseada em

agentes.

- **Métodos estruturais:** esses métodos consideram a textura como arranjos hierárquicos de elementos bem definidos, provendo uma descrição simbólica da imagem (18–19). Geralmente, as operações morfológicas (e.g. abertura e fechamento) são sucessivamente aplicadas nas imagens para caracterizar a evolução dos elementos de textura, como apresentado em (19–20) e em um método invariante à iluminação (21). Ainda nessa categoria, detectores e descritores de pontos de interesse têm sido propostos para encontrar e descrever os elementos de textura (22). Um método com bons resultados foi proposto por Lazebnik et al (23), o qual utiliza um conjunto de regiões obtidas pelos detectores Harris e Laplacian como elementos de textura. Esses elementos são então caracterizados pelos descritores RIFT e spin.
- **Métodos estatísticos:** os métodos desta categoria representam a textura por meio das propriedades estatísticas que governam a relação entre os níveis de cinza da imagem. De fato, essa ideia também pode ser estendida para imagens coloridas. Os primeiros métodos propostos baseiam-se nos momentos de primeira e segunda ordem calculados do histograma da imagem (24). Por ser uma abordagem limitada para texturas irregulares ou estruturais, métodos mais eficientes foram propostos, como a matriz de co-ocorrência (16) e sua versão invariante à rotação proposta por Davis (25). Um dos métodos mais eficientes desta categoria, em termos de classificação, é o padrão local binário (26–27). Esse método fornece resultados interessantes através do cálculo da co-ocorrência dos níveis de cinza em vizinhanças circulares.
- **Métodos espectrais:** esses métodos representam a textura por meio de informações espectrais da imagem. A ideia principal é estimar a frequência espacial da intensidade dos pixels e correlacionar as texturas mais finas com as altas frequências e as texturas mais rugosas às baixas frequências (28). Portanto, os métodos dessa categoria focam no desenvolvimento de bancos de filtros, tais como os filtros de Gabor (29–31), e na decomposição em sub-bandas (e.g. wavelets (32–33)) para descrever as texturas eficientemente.
- **Métodos baseados em modelos:** os métodos buscam representar a textura por meio da construção de um modelo e a obtenção dos seus parâmetros. Os modelos geralmente empregados em texturas são os modelos fractais (34–35) e estocásticos (36). Os modelos autoregressivos invariantes à rotação foram, inicialmente, propostos por Kashyap e Khotanzad (37). Em (38), os campos aleatórios de Markov foram introduzidos para o reconhecimento de texturas. Além disso, os métodos baseados em geometria fractal

têm apresentado resultados interessantes na análise de texturas. Por exemplo, é possível citar a dimensão fractal multiescala (4), o espectro multifractal (39) e dimensão fractal local (40).

- **Métodos baseados em agentes:** recentemente, os métodos desta categoria têm sido propostos por meio da utilização de entidades autônomas que observam e agem em uma imagem de textura. Em geral, os métodos extraem informações do relacionamento dos agentes ou de suas caminhadas na imagem para caracterizar as texturas. O sistema de formigas (41), as caminhadas aleatórias (42) e as caminhadas determinísticas (43–45) são metodologias baseadas em agentes aplicadas com sucesso no reconhecimento de texturas.

2.2.2 Texturas Dinâmicas

Diferente das texturas estáticas, as texturas dinâmicas são padrões de texturas em movimento, como sequência de imagens das ondas do mar, fogo, fumaça, escada rolante, folhas em crescimento, bandeira balançando devido ao vento e crescimento de colônias de bactérias. Exemplos de texturas dinâmicas podem ser visualizados na Figura 2.1. Uma definição concisa define textura dinâmica como um movimento de padrões que apresenta regularidade estatística no espaço e no tempo (46). Em texturas dinâmicas, a ideia de auto-similaridade percebida em texturas estáticas é estendida para o domínio espaço-temporal. Portanto, a mudança contínua na forma e na aparência das texturas dinâmicas torna a sua aplicação muito desafiadora.

As aplicações potenciais de texturas dinâmicas incluem sistemas automáticos de vigilância em ambientes desafiadores, como o monitoramento de incêndios florestais para prevenção de desastres naturais, monitoramento de tráfego, aplicações de segurança interna, além da síntese, segmentação e classificação de vídeos.

Por ser um campo de pesquisa ainda recente, poucas abordagens foram propostas na literatura. Os métodos existentes para modelagem de texturas dinâmicas podem ser divididos em quatro categorias: (i) baseado em movimento – (ii) baseado em filtragem – (iii) baseado

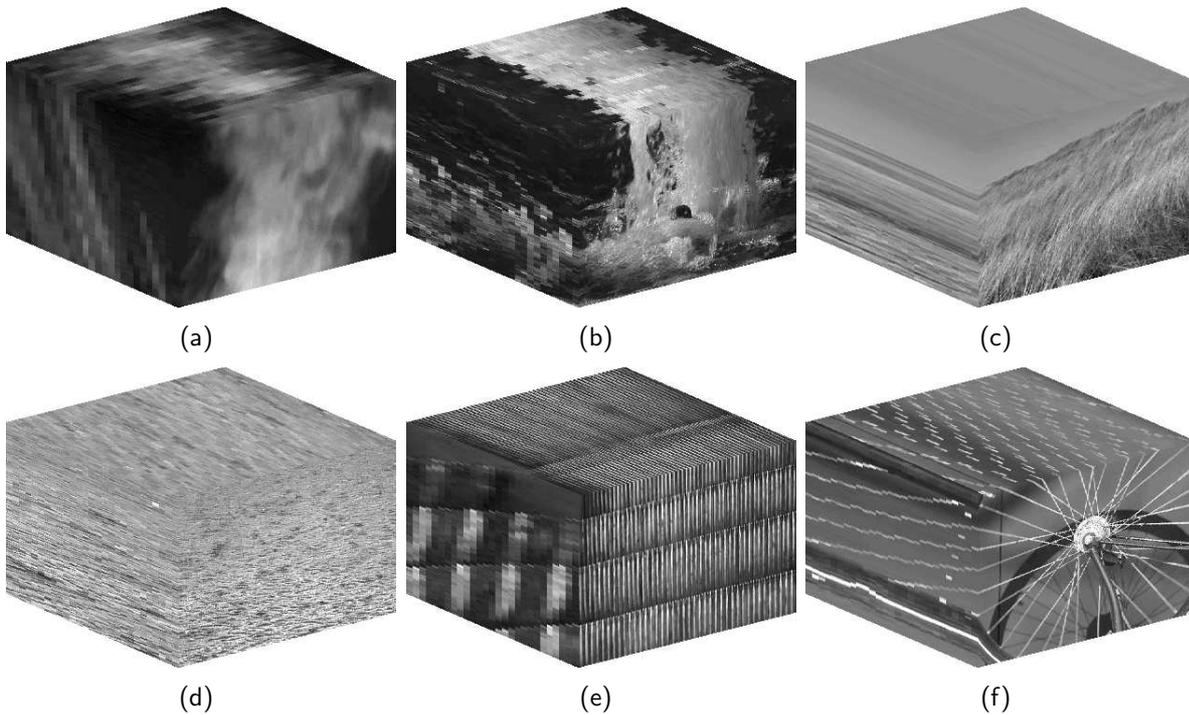


Figura 2.1 – Exemplos de seis texturas dinâmicas: (a) fumaça, (b) cachoeira, (c) pastagem, (d) água com chuva, (e) escada rolante, e (f) aro de bicicleta em movimento.

em modelos – (iv) baseado em propriedades geométricas. Cada categoria é descrita abaixo.

- Métodos baseados em movimento:** esses métodos são os mais populares devido a sua eficiência em estimar medidas baseadas em movimento, tais como o fluxo óptico (6, 47, 48). Portanto, esses métodos reduzem a modelagem de texturas dinâmicas em uma análise de sequências de padrões em movimento. Péteri e Chetverikov (49) propuseram um método que combina características do fluxo normal com características de periodicidade, com o objetivo de caracterizar a magnitude, direcionalidade e periodicidade do movimento. Lu et al (50) propuseram um método usando histogramas espaço-temporal baseado nos campos de velocidade e aceleração. Fazekas e Chetverikov (6) desenvolveram um método promissor com base no fluxo óptico para segmentação de texturas dinâmicas. Por fim, Chen et al (51) caracterizam os padrões de movimento de uma textura dinâmica por meio do histograma orientado do fluxo óptico.
- Métodos baseados em filtragem:** os métodos desta categoria descrevem as texturas dinâmicas em diferentes escalas no espaço e no tempo através da filtragem espaço-temporal. A transformada wavelet (52–54) foi utilizada com o intuito de decompor o movimento da textura dinâmica em movimento local e global, de acordo com o grau de detalhe desejado. A periodicidade destes movimentos resulta em energia máxima em di-

ferentes escalas e, portanto, pode ser utilizada para descrever as texturas dinâmicas. Os filtros de Gabor, bastante conhecidos e utilizados para texturas estáticas, foram estendidos para texturas dinâmicas em (55). Para modelar uma textura dinâmica, a sequência de imagens é convoluída com um banco de filtros de Gabor espaço-temporal. Para cada resposta da convolução, a energia é calculada para compor o vetor de características.

- **Métodos baseados em modelos:** esses métodos foram propostos para descrever as texturas dinâmicas por meio de um processo generativo (e.g. sistema dinâmico linear) (7, 56–58). Os processos ou modelos generativos são modelos para gerar aleatoriamente dados observáveis, normalmente com alguns parâmetros ocultos. Doretto et al (56) modelaram as texturas dinâmicas por meio do modelo Gauss-Markov e, então, estimaram os seus parâmetros. Chan e Vasconcelos (59) derivaram o algoritmo *expectation-maximization* para aprender os seus parâmetros com base em texturas dinâmicas. Depois disso, eles propuseram a textura dinâmica em camadas para representar um vídeo como um conjunto de camadas estocásticas. Ravichandran et al (60) propuseram um método que caracterizam adequadamente texturas dinâmicas obtidas em diferentes escalas e condições por meio do modelo *bag-of-system*. Esse modelo é análogo ao conhecido modelo *bag-of-features* para reconhecimento de objetos, exceto por utilizar sistemas dinâmicos lineares como descritores de características. Após a obtenção dos parâmetros, uma distância ou *kernel* entre os parâmetros dos modelos é definida. Uma vez definida a distância, os classificadores, como os k-vizinhos mais próximos (KNN) ou máquinas de vetores de suporte (SVM), podem ser utilizados para categorizar as texturas dinâmicas.
- **Métodos baseados em propriedades geométricas:** os métodos desta categoria são menos populares, sendo dois, os trabalhos principais: o primeiro proposto por Otsuka et al (61) e o segundo proposto por Zhong e Scarlaroff (62). Ambos os métodos são baseados nas propriedades das superfícies de trajetórias em movimento. A partir destas superfícies, é possível extrair características de movimento e aparência com base na distribuição do plano tangente (61).

Metodologias

Neste capítulo, são apresentadas e discutidas as metodologias utilizadas no desenvolvimento deste trabalho: as caminhadas determinísticas parcialmente auto-repulsivas e a dimensão fractal. Essas duas metodologias são alvos deste trabalho devido principalmente aos resultados promissores obtidos em aplicações de visão computacional.

3.1 Caminhadas Determinísticas Parcialmente Auto-repulsivas - CDPAR

As caminhadas aleatórias em ambientes regulares ou desordenados têm sido frequentemente estudadas e apresentam uma grande gama de aplicações (63–64). Apesar de não serem exaustivamente estudadas, as caminhadas determinísticas também apresentam diversos resultados interessantes em ambientes regulares (65–66) e desordenados (67).

Em particular, as caminhadas determinísticas parcialmente auto-repulsivas (CDPAR) são capazes de extrair propriedades ou informações relevantes de uma grande variedade de problemas. Em visão computacional, essas caminhadas têm apresentado resultados interessantes na análise de texturas (43–44) e no reconhecimento de folhas de plantas através de imagens (43). O estudo inicial dessas caminhadas em imagens foi realizado em (68). Nesse estudo, o período do atrator e o tempo de transiente foram utilizados para caracterizar diferentes classes de texturas. Os experimentos, realizados em texturas do banco de imagens Brodatz, apresentaram resultados promissores. Posteriormente, a formalização do método para a análise de texturas foi proposto em (43), apresentando aplicações em reconhecimento de texturas sintéticas e reconhecimento de espécies de plantas por meio da sua imagem. Além da formalização do

método, os autores propuseram a combinação entre dois critérios de caminhada e uma nova medida extraída da distribuição do período do atrator e do tempo de transiente.

Após esses dois trabalhos precursores, surgiram outras variantes do método original. Em (45) foi proposto um novo critério de caminhada baseado em vetores resultantes. A grande vantagem desse critério é a inexistência de empates na caminhada. Uma variante proposta por Backes (44) combinou as caminhadas determinísticas parcialmente auto-repulsivas e os grafos. Nesse método, as caminhadas resultantes são convertidas em um grafo e, então, medidas são extraídas desse grafo para caracterizar a textura na imagem. Além da utilização dessas caminhadas em análise de texturas, foram reportados trabalhos em reconhecimento de padrões (69–70) e análise topológica de dicionários (71).

A caminhada determinística parcialmente auto-repulsiva pode ser compreendida como um viajante que deseja visitar N pontos distribuídos em um mapa de d dimensões. Considere esses pontos como cidades e um turista viajando entre elas seguindo a regra determinística: vá para a cidade mais próxima que não tenha sido visitada nos últimos μ passos. O turista realiza uma caminhada determinística parcialmente auto-repulsiva limitada a uma janela $\mu - 1$. Assim, a memória de tamanho μ é vista como o tempo necessário para uma cidade já visitada tornar-se atrativa novamente ao turista. Os passos são baseados em uma tabela de distância entre as cidades. A distância é, simplesmente, uma maneira de ranquear as cidades vizinhas, tornando essa caminhada invariante a transformações de escala. Essa característica permite a sua aplicação em diversos problemas, principalmente em imagens onde a invariância à escala é uma característica desejável. Embora as CDPARs possuam regras simples, os resultados obtidos apresentam uma dinâmica bastante complexa.

Após alguns passos, o turista produz uma caminhada que pode ser dividida em duas partes: uma parte inicial de tamanho τ chamada transiente e uma parte final, composta por ciclo de período $\rho \geq \mu + 1$, chamada atrator. O transiente é o número de cidades vistas pelo turista, desde a cidade inicial até a primeira cidade do atrator. O atrator é um conjunto de cidades no qual o turista não consegue escapar, sendo o período do atrator ρ , a cardinalidade desse conjunto. Após ingressar em um atrator, as mesmas cidades sempre serão visitadas, independente do tamanho da caminhada, como pode ser visto na Figura 3.1. Apesar de raro, existem alguns casos em que o turista não é capaz de encontrar um atrator, sendo a caminhada composta apenas do tempo de transiente τ . Além disso, a cada passo do turista, podem existir duas ou mais direções que satisfaçam a regra de movimentação da caminhada. Neste caso de empate, a primeira cidade que satisfaz a regra é selecionada, preservando a natureza determinística da caminhada (43).

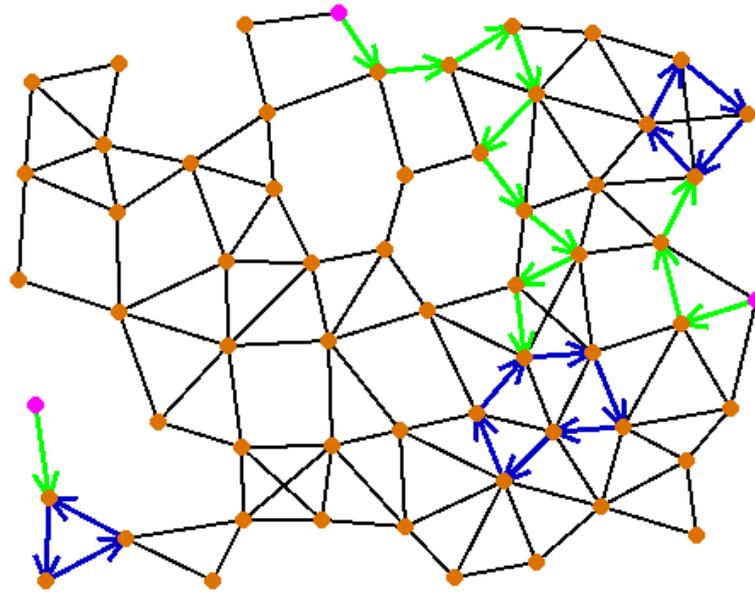


Figura 3.1 – Exemplo de três CDPARs iniciadas em diferentes cidades. As cidades pertencentes à parte transiente e ao atrator estão em verde e azul, respectivamente.

Para cada situação inicial, uma caminhada diferente é produzida pelo turista. O comportamento depende, estritamente, da configuração do conjunto de cidades, da memória e da cidade inicial. Assim, considera-se cada cidade do mapa como uma condição inicial de uma caminhada. Portanto, se o ambiente em que o turista será aplicado possuir N cidades, existirão N caminhadas diferentes, cada uma iniciada em uma das cidades. Ambos, tempo de transiente τ e período do atrator ρ de cada caminhada são combinados em uma distribuição conjunta $S_{\mu,d}(\tau, \rho)$. Cada posição da distribuição conjunta armazena a quantidade de caminhadas que obtiveram tempo de transiente igual a τ e período do atrator igual a ρ , de acordo com a Equação 3.1.1. A Figura 3.2 apresenta a distribuição conjunta para $\mu = 9$ e $\mu = 25$ para cidades geradas aleatoriamente em duas dimensões.

$$S_{\mu,d}(\tau, \rho) = \frac{1}{N} \sum_{i=1}^N \begin{cases} 1, & \text{se } \tau_i = \tau, \rho_i = \rho \\ 0, & \text{caso contrário} \end{cases} \quad (3.1.1)$$

onde τ_i e ρ_i são, respectivamente, o tempo de transiente e o período do atrator obtidos pela caminhada iniciada na cidade i .

A seguir, são discutidos alguns casos especiais da distribuição conjunta obtidos para diferentes valores de μ . O caso mais trivial considera $\mu = 0$, i.e., o turista não possui memória. Neste caso, o turista permanece na mesma cidade e a trajetória tem transiente nulo e um

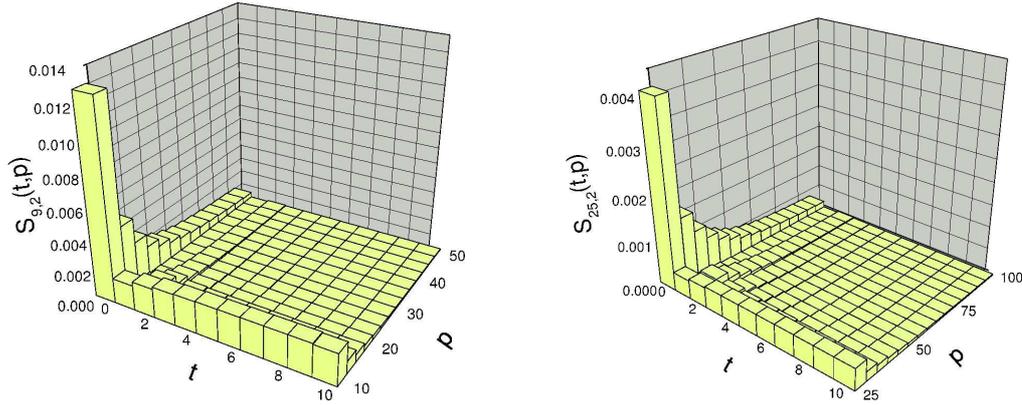


Figura 3.2 – Distribuição conjunta do tempo de transiente τ e período do atrator ρ obtidos em uma configuração com $N = 1000$ cidades e dimensão $d = 2$. Embora diferentes valores de memórias tenham sido utilizados, o aspecto geral de ambas as distribuições são similares.

atrator de período $\rho = 1$. Apesar da sua simplicidade e da sua inaplicabilidade em situações reais, este caso é interessante porque é a situação mais simples de caminhada estocástica (72). A distribuição conjunta é dada por:

$$S_{0,d}(\tau, \rho) = \delta_{\tau,0} \delta_{\rho,1} , \quad (3.1.2)$$

onde $\delta_{i,j}$ é o delta de Kronecker.

Para memória de tamanho mínimo $\mu = 1$, o turista armazena somente a cidade atual. Assim, a cada passo, o turista deve deixar a cidade atual e ir para a cidade vizinha mais próxima. Este tamanho de memória não conduz à exploração do ambiente, uma vez que depois de um pequeno tempo de transiente, o turista é aprisionado por uma dupla de cidades vizinhas mais próximas entre si. A distribuição conjunta do tempo de transiente e período do atrator pode ser analiticamente obtida para um número de cidades $N \gg 1$ (73):

$$S_{1,d}(\tau, \rho) = \frac{\Gamma(1 + I_d^{-1})(\tau + I_d^{-1})}{\Gamma(\tau + \rho + I_d^{-1})} \delta_{\rho,2} , \quad (3.1.3)$$

onde $\Gamma(z)$ é a função gamma e $I_d = I_{\frac{1}{4}}(\frac{1}{2}, \frac{(d+1)}{2})$ é a função beta normalizada e incompleta.

Eventos interessantes ocorrem para grandes valores de μ . Neste caso, a distribuição conjunta não apresenta um pico em $\rho_{min} = \mu + 1$, mas apresenta uma completa gama de atratores com períodos $\rho \geq \rho_{min}$ e uma possível lei de decaimento (74–75).

3.1.1 CDPARs em imagens

Considere uma imagem que consiste em um par (\mathcal{I}, I) – um conjunto finito \mathcal{I} de pixels e um mapeamento I que atribui para cada pixel $p_i = (x_i, y_i)$ em \mathcal{I} uma intensidade $I(p_i) \in [0, 255]$. A vizinhança $\eta(p_i)$ do pixel p_i consiste em um conjunto de pixels p_j cuja distância Euclidiana entre os pixels p_i e p_j é menor que ou igual a $\sqrt{2}$:

$$\eta(p_i) = \{p_j \mid d(p_i, p_j) \leq \sqrt{2}\} \quad (3.1.4)$$

$$d(p_i, p_j) = \sqrt{(x_{p_i} - x_{p_j})^2 + (y_{p_i} - y_{p_j})^2}$$

O peso da conexão entre dois pixels é dado pelo módulo da diferença entre suas intensidades:

$$\omega(p_i, p_j) = \begin{cases} |I(p_i) - I(p_j)|, & \text{se } p_j \in \eta(i) \\ -1, & \text{caso contrário} \end{cases} \quad (3.1.5)$$

Dadas as definições acima, o turista inicia sua caminhada $W_t = \{w_0, w_1, \dots, w_t\}$ em um pixel p_i (t é o índice de tempo):

$$w_0 = p_i; \quad (3.1.6)$$

O turista movimentar-se de acordo com a seguinte regra determinística: vá para o pixel mais próximo (pixel cujo peso de conexão é mínimo) que não esteja na memória (Equação 3.1.7). Esta regra determinística é baseada em uma memória M_t^μ (Equação 3.1.8) que contém os últimos μ pixels da caminhada W_t . Outra regra proposta em (43) guia o turista na direção do pixel cujo peso de conexão é máximo (Equação 3.1.9). A primeira e a segunda regras serão referenciadas através desse trabalho como regras *min* e *max*, respectivamente.

$$w_{t+1} = \underset{p_j \in \eta(w_t), p_j \notin M_t^\mu}{\operatorname{argmin}} \omega(w_t, p_j) \quad (3.1.7)$$

$$M_t^\mu = \{X \subset W_t \mid X = \bigcup_{k=t-\mu}^t w_k\} \quad (3.1.8)$$

$$w_{t+1} = \underset{p_j \in \eta(w_t), p_j \notin M_t^\mu}{\operatorname{argmax}} \omega(w_t, p_j) \quad (3.1.9)$$

Estas duas regras produzem CDPARs que podem ser divididas em transiente e atrator, conforme descrito anteriormente. Em imagens, o atrator consiste em um conjunto de pixels cujas intensidades formam um padrão do qual o turista não consegue escapar. Um atrator de tamanho ρ considerando um transiente de tamanho τ é encontrado se:

$$(w_{\tau+j} = w_{\tau+j+\rho}), \forall j \in [0, \rho - 1] \quad (3.1.10)$$

Dado o tempo de transiente τ e período do atrator ρ , o conjunto de pixels pertencentes ao transiente T e o conjunto de pixels pertencentes ao atrator A podem ser definidos como:

$$T = \{w_0, \dots, w_{\tau-1}\}, A = \{w_{\tau}, \dots, w_{\tau+\rho-1}\} \quad (3.1.11)$$

Um exemplo de CDPAR utilizando a regra *min* e tamanho de memória $\mu = 2$ (armazena o pixel atual e o anterior) pode ser visualizado na Figura 3.3. Os círculos representam os pixels da imagem e a intensidade de cada pixel é dada dentro do seu círculo. O peso de conexão, como descrito na Equação 3.1.5, é dado pelo módulo da diferença entre as intensidades. O transiente ($\tau = 3$) é composto pelos círculos em laranja, enquanto que o atrator ($\rho = 3$) é composto pelos círculos em verde.

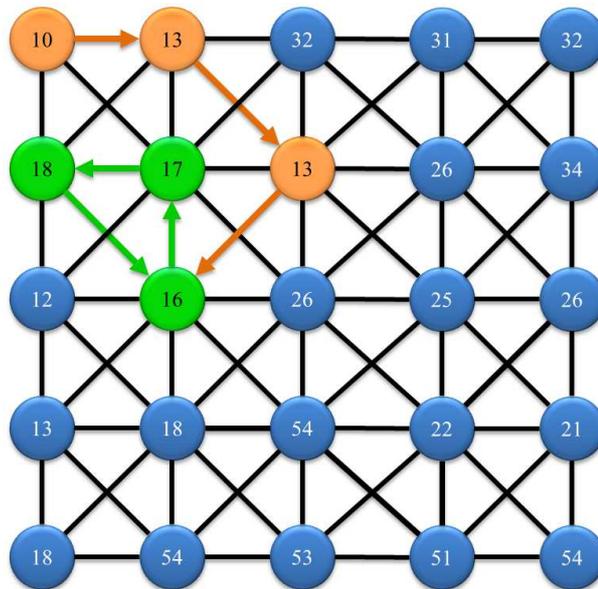


Figura 3.3 – Exemplo de caminhada determinística parcialmente auto-repulsiva utilizando a regra *min* e tamanho de memória $\mu = 2$. Cada pixel da imagem é representado por um círculo e a intensidade de cada pixel é dado dentro do círculo. O peso de conexão é dado pelo módulo da diferença entre intensidades. O transiente é dado pelos círculos em laranja, enquanto que o atrator é dado pelos círculos em verde.

Para cada condição inicial, uma caminhada diferente é produzida. Assim, dada uma imagem I , cada pixel p_i é utilizado como uma condição inicial de uma caminhada $W_{\mu,r}^{p_i}$, onde μ e r são, respectivamente, o tamanho da memória e a regra (*min* ou *max*). Se a imagem possuir N pixels, serão obtidas N caminhadas iniciadas em cada um dos pixels ($W_{\mu,r}^{p_0}, W_{\mu,r}^{p_1}, \dots, W_{\mu,r}^{p_N}$). O conjunto do atrator e o conjunto do transiente de uma caminhada $W_{\mu,r}^{p_i}$ serão referenciados como $T_{\mu,r}^{p_i}$ e $A_{\mu,r}^{p_i}$, respectivamente. Combinando o tamanho do transiente $|T_{\mu,r}^{p_i}|$ e do atrator $|A_{\mu,r}^{p_i}|$, uma distribuição conjunta é obtida (Equação 3.1.12) e utilizada para caracterizar a imagem em problemas de classificação (43).

$$S_{\mu,d}(\tau, \rho) = \frac{1}{N} \sum_{i=1}^N \begin{cases} 1, & \text{se } |T_{\mu,r}^{p_i}| = \tau, |A_{\mu,r}^{p_i}| = \rho \\ 0, & \text{caso contrário} \end{cases} \quad (3.1.12)$$

A distribuição conjunta tem demonstrado um grande potencial em classificação de texturas. A Figura 3.4 apresenta as distribuições conjuntas de três padrões de textura. Podemos notar que existe uma discriminação entre os padrões pela forma das superfícies das distribuições.

3.1.2 Dinâmica da CDPAR

A CDPAR, quando aplicada em um mapa de cidades geradas de forma aleatória, é caracterizada por um comportamento caótico. Devido à não correlação entre as cidades, o turista muda de direção com a mesma frequência em que o ambiente muda. Por outro lado, ambientes regulares apresentam um contexto que correlacionam os vizinhos (e.g. pixels de uma imagem possuem intensidades similares em regiões próximas). Com isso, a correlação entre os vizinhos influencia a escolha da direção do turista ao longo de sua caminhada.

A regra de movimentação do turista em uma CDPAR tem grande importância na caracterização do ambiente. As caminhadas guiadas por diferentes regras (e.g. *min* ou *max*) geram atratores com padrões distintos para o mesmo ambiente. As caminhadas guiadas pela regra *min* são apropriadas para encontrar atratores em regiões que apresentam alta homogeneidade. Por outro lado, as caminhadas guiadas pela regra *max* encontram atratores localizados em

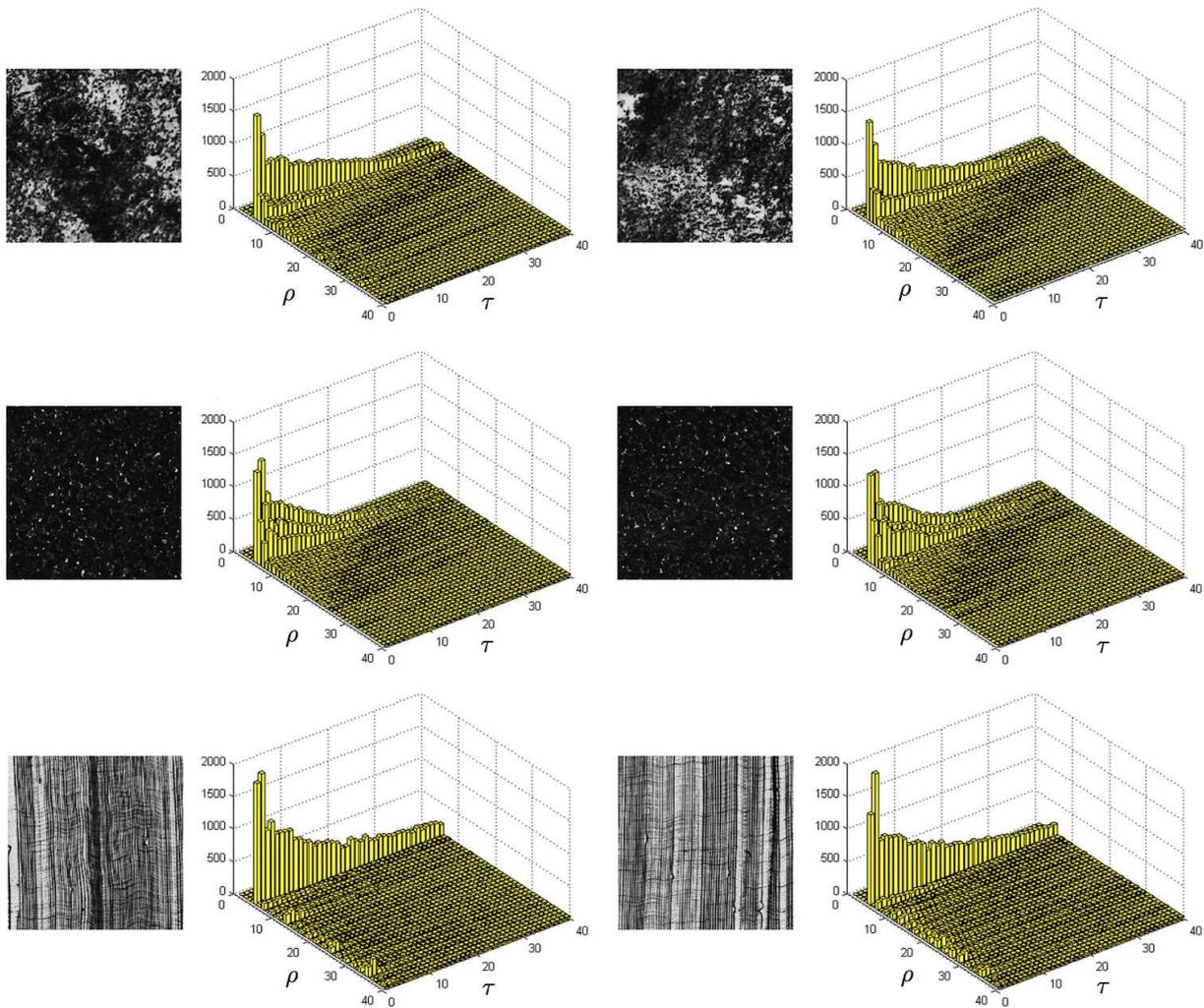


Figura 3.4 – Exemplos de distribuições conjuntas obtidas para três padrões de texturas utilizando memória $\mu = 1$ e regra de movimentação $r = \min$.

regiões de alta heterogeneidade, como regiões de bordas em uma imagem.

A utilização de diferentes regras de movimentação reflete em diferentes distribuições conjuntas. Este interessante resultado permite utilizar informações de diferentes fontes na caracterização de um ambiente, como pode ser visto nas Figuras 3.5 e 3.6 para atratores gerados pelas regras *min* e *max*, respectivamente.

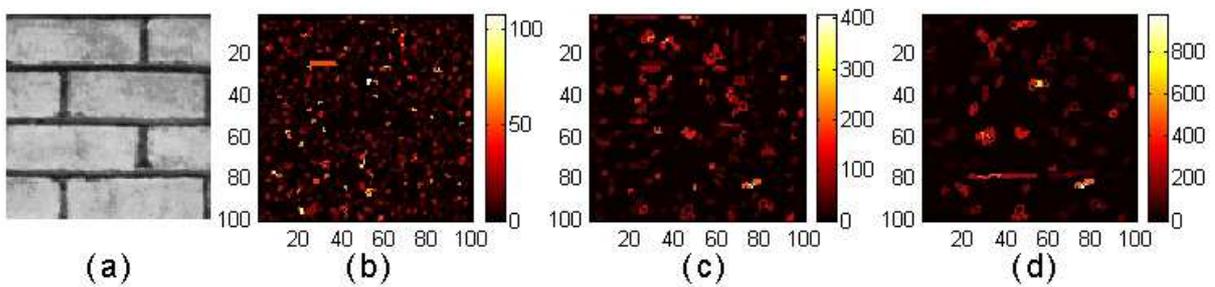


Figura 3.5 – Atratores gerados em uma imagem de textura utilizando a regra *min* e diferentes valores de μ . (a) Imagem Original; (b) $\mu = 1$; (c) $\mu = 3$; (d) $\mu = 7$.

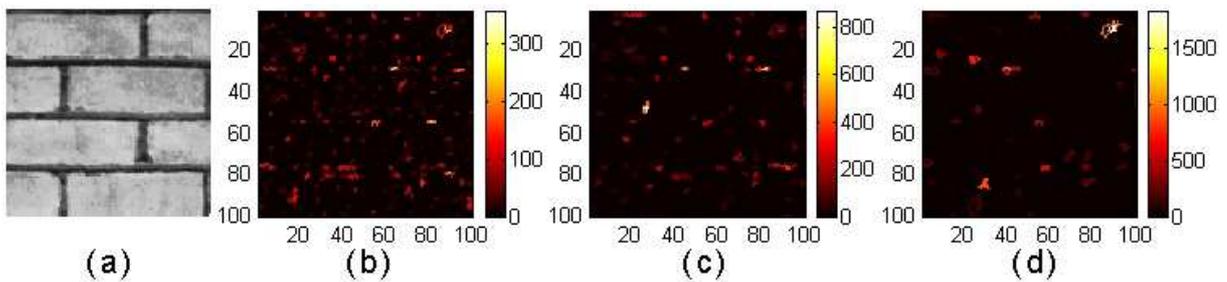


Figura 3.6 – Atratores gerados em uma imagem de textura utilizando a regra *max* e diferentes valores de μ . (a) Imagem Original; (b) $\mu = 1$; (c) $\mu = 3$; (d) $\mu = 7$.

3.1.3 Algoritmo para Detecção de Atratores

O atrator de uma CDPAR é caracterizado por iniciar e terminar no mesmo pixel. Além disso, os pixels intermediários devem se repetir ao longo de todo o atrator. Entretanto, um pixel pode ser revisitado sem se configurar um atrator. Esta propriedade possibilita a construção de caminhadas complexas, ao mesmo tempo em que demanda uma metodologia sofisticada para detecção do atrator. A Figura 3.7 apresenta alguns exemplos de CDPARs.

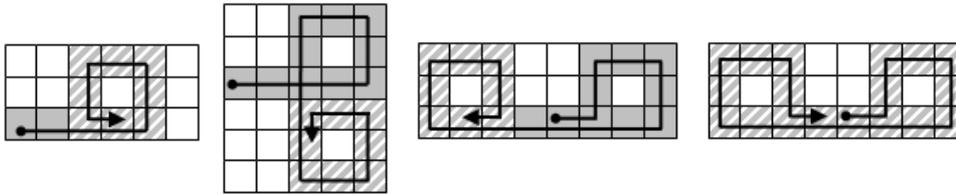


Figura 3.7 – Exemplos de CDPARs em que a parte cinza representa o transiente, enquanto que a parte listrada representa o atrator.

Para a detecção do atrator, diferentes partes da caminhada devem ser comparadas. Para possibilitar essa comparação, a caminhada é armazenada em uma lista encadeada conforme exemplo da Figura 3.8. A caminhada é iniciada no pixel $C1$, em seguida o pixel $C2$ é visitado e assim por diante.



Figura 3.8 – Uma lista encadeada é utilizada para armazenar uma CDPAR. Cada posição da lista contém o pixel visitado.

A comparação entre seções de uma lista encadeada consome um alto custo computacional. A cada novo pixel visitado, é necessário checar a existência de uma sequência de pixels $\rho \geq \mu + 1$ que se repita ao longo da lista. Quando um atrator é encontrado durante a caminhada, existem pelo menos três repetições de um mesmo pixel na lista. Dado esse pixel com três repetições, a detecção é obtida através da comparação entre as seções desse pixel na lista (Figura 3.9). Esta estratégia permite acelerar a busca na lista encadeada e executar a detecção do atrator de forma simples e rápida.

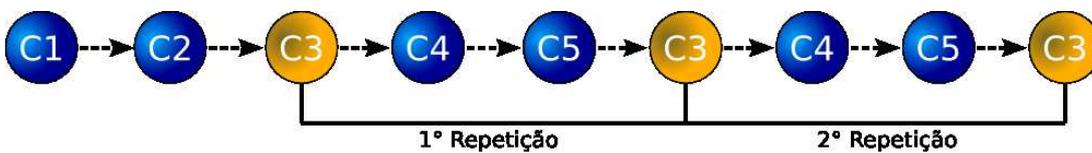


Figura 3.9 – Comparação entre diferentes partes de uma lista encadeada. Um pixel repetido ao menos três vezes na lista pode configurar a existência de uma sequência de pixels repetidos.

Para checar o número de repetições de pixels, uma nova lista é associada a cada pixel. Essa lista encadeada é responsável por manter a posição em que cada pixel aparece na lista encadeada da caminhada (Figura 3.10). Assim, é possível checar rapidamente a existência de um conjunto de pixels repetidos, bem como, o número de repetições de cada pixel. Por exemplo, o pixel $C3$ é visitado três vezes e essa informação é rapidamente obtida por meio

dessa lista. Por meio dessa estratégia, o custo computacional envolvido na detecção do atrator é drasticamente reduzido.

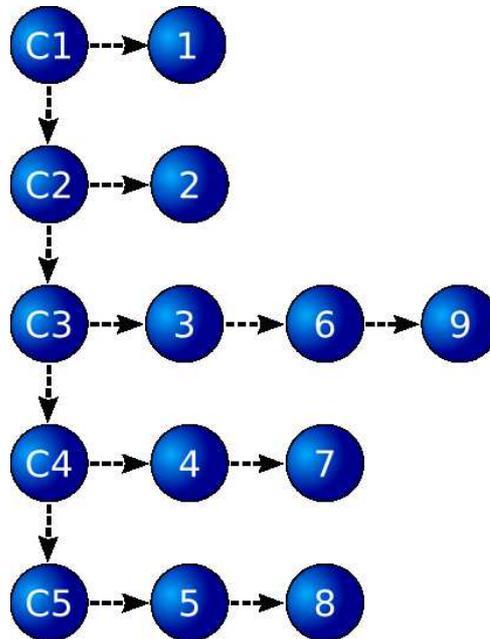


Figura 3.10 – Lista encadeada associando as posições em que cada pixel aparece na caminhada, tornando possível o cálculo eficiente do número de vezes que um determinado pixel aparece na caminhada.

3.1.4 Complexidade Computacional

No método descrito anteriormente, uma CDPAR é iniciada em cada pixel de uma imagem. Considerando que a imagem possui N pixels, N caminhadas serão iniciadas. Cada caminhada consiste em uma parte transiente de tamanho τ e um atrator de tamanho $\rho \geq \mu + 1$. Em casos onde o atrator não é encontrado, a caminhada é interrompida quando o número de passos atinge o número de pixels N , assim $\tau = N$ e $\rho = 0$.

A complexidade computacional das CDPARs em imagens é determinada pelo número de pixels N e pelo tamanho das caminhadas $M = \tau + \rho$. Desse modo, o método possui uma complexidade $O(NM)$. A seguir discutiremos o melhor caso, caso médio e pior caso.

O melhor caso é obtido quando todas as caminhadas iniciam em um atrator, ou seja, o transiente é nulo ($\tau = 0$) e o atrator apresenta tamanho mínimo ($\rho = 1$). No melhor caso

$M = 1$, resultando em uma complexidade $O(N)$.

O pior caso é obtido quando nenhum atrator é encontrado durante todas as N caminhadas. Neste caso, a caminhada apresenta um tamanho $M = \tau + \rho = N$, sendo a complexidade $O(N^2)$. É importante enfatizar que o pior caso é muito raro e mesmo uma imagem gerada de forma aleatória não produz este caso especial.

Para o caso médio, a Figura 3.11 apresenta o tamanho médio das caminhadas para diferentes tamanhos de memória μ . Para uma memória $\mu = 5$ (maior tamanho utilizado pelo método), a média do tamanho da caminhada é igual a $M = 23$. Isto é, o agente realiza, em média, 23 passos até encontrar o atrator. Este resultado está bem próximo da complexidade do turista no melhor caso, $O(N)$, dado que $N \gg 23$. Este é um excelente resultado quando comparado com a complexidade de alguns métodos de reconhecimento de texturas como os filtros de Gabor ($O(N \log N)$) e as matrizes de co-ocorrência ($O(N)$), considerando N como o número de pixels.

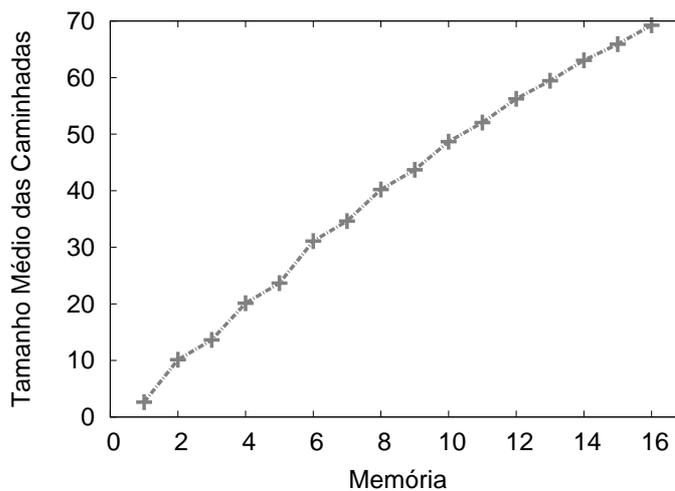


Figura 3.11 – Gráfico do tamanho médio das caminhadas para diferentes tamanhos de memória μ . O comportamento da curva é linear.

3.2 Dimensão Fractal

Na geometria clássica, os conceitos e métodos estão interessados em descrever objetos regulares, enquanto que os objetos irregulares são desconsiderados. Por exemplo, as pro-

priedades de curvas e superfícies suaves podem ser caracterizadas por meio do conceito de derivada. Entretanto, muitos fenômenos naturais (e.g. crescimento e declínio de populações, distribuição da chuva) são irregulares, sendo assim, eles não são modelados adequadamente utilizando os conceitos da geometria clássica.

Nas últimas décadas, trabalhos têm reportado que estes conjuntos irregulares podem ser considerados pela geometria fractal. Diversas definições do termo geometria fractal estão disponíveis na literatura. Mandelbrot foi o primeiro a descrever o conceito em 1975, dando o nome de fractal para conjuntos irregulares cuja dimensão de Hausdorff é estritamente maior que a dimensão topológica. Entretanto, existem alguns conjuntos que deveriam ser considerados como fractais, mas não se adequam aos critérios definidos por Mandelbrot. Outras definições foram dadas para fractais, mas apesar disso, não existe uma definição única que se adequa a todos os conjuntos irregulares (76).

O principal indício da geometria fractal é que a natureza exibe alguma forma de auto-similaridade em diferentes escalas. Apesar de considerar formas ou comportamentos dinâmicos extremamente complexos, ainda assim, é possível encontrar características em uma escala que se assemelham a características de outra escala. O ramo de uma árvore possui estrutura similar com a árvore inteira, mas em uma escala menor. O mesmo comportamento estrutural pode ser observado em cascas de árvores, nuvens, montanhas, vegetais como couve flor e brócolis, entre outros. Entretanto, os fractais observados na natureza não são auto-similares em todas as escalas. Existem um limite inferior e um limite superior, a partir do qual, a estrutura não apresenta características fractais. Abaixo são descritos dois exemplos comuns de fractais: o conjunto de Cantor e a curva de Koch.

Exemplo 3.2.1 *Um dos fractais mais simples é o conjunto de Cantor, ilustrado na Figura 3.12. Esse conjunto inicia com um intervalo unitário $E_0 = [0, 1]$, i.e., números reais entre $0 \leq x \leq 1$. No próximo passo, retira-se o terço do meio do intervalo E_0 . Assim, E_1 é o conjunto contendo os dois intervalos restantes $[0, \frac{1}{3}] \cup [\frac{2}{3}, 1]$. Em seguida, E_2 é obtido removendo-se o terço do meio de cada um dos dois intervalos criados pelo passo anterior, i.e., $[0, \frac{1}{9}] \cup [\frac{2}{9}, \frac{3}{9}] \cup [\frac{6}{9}, \frac{7}{9}] \cup [\frac{8}{9}, 1]$. Aplicando-se essa ideia recursivamente, o conjunto de Cantor é definido como a intersecção dos conjuntos E_j , i.e., $C_{1/3} = \bigcap_{j=1}^{\infty} E_j$.*

Exemplo 3.2.2 *A curva de Koch (Figura 3.13) foi um dos primeiros fractais a ser descrito. Considerando o mesmo intervalo unitário $E_0 = [0, 1]$, E_1 é obtido removendo-se o terço do meio de E_0 e substituindo-o pelos dois lados de um triângulo baseado no segmento removido. Em seguida, E_2 é obtido aplicando-se o mesmo procedimento para os quatros segmentos de*

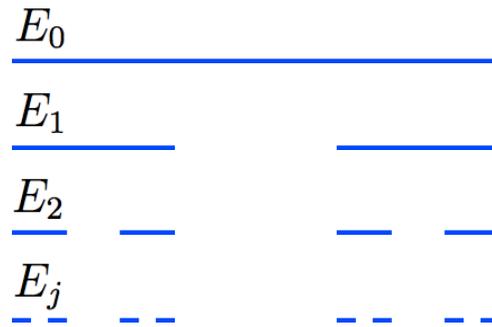


Figura 3.12 – Sequência de conjuntos E_j demonstrando o conjunto de Cantor. O conjunto inicia com um intervalo unitário e, em seguida, retira-se o terço do meio.

E_1 . Aplicando-se recursivamente a ideia anterior, a sequência de curvas poligonais E_j para $j \rightarrow \infty$ aproxima-se da curva limite (curva de Koch).

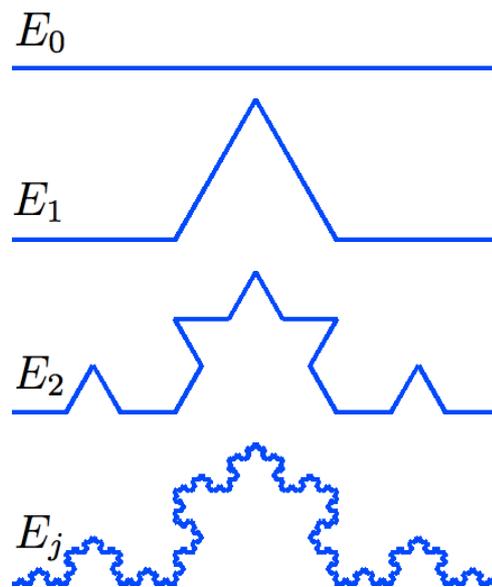


Figura 3.13 – Sequência de conjuntos E_j demonstrando a curva de Koch. Inicia-se com um intervalo unitário, remove-se o terço do meio e substitui-se por dois lados de um triângulo.

Na curva de Koch, um conjunto E_j possui um tamanho $(4/3)^n$. Para $n \rightarrow \infty$, isto implica que esse fractal possui tamanho infinito. Ainda para $n \rightarrow \infty$, a curva de Koch ocupa uma área nula no plano e, portanto, o tamanho e a área não apresentam informações úteis para descrever esse conjunto. Para solucionar essa questão e descrever tais conjuntos, Hausdorff propôs uma dimensão que permite que objetos possuam dimensão não inteiras, visto que a curva de Koch parece possuir dimensão maior do que 1 e menor do que 2. Assim, a dimensão fractal é um número real associado a um fractal que quantifica quão densamente o fractal ocupa o espaço.

A dimensão D para fractais auto-similares em \mathfrak{R}^n é dada pela união de N cópias não sobrepostas do fractal, sendo cada cópia dimensionada por uma razão $r < 1$. A relação entre r e N é uma lei de potência:

$$N = r^{-D} \quad (3.2.1)$$

Assim, a dimensão fractal é dada por:

$$D = \frac{-\ln N}{\ln r} \quad (3.2.2)$$

Como exemplo, o conjunto de Cantor é gerado para $N = 2$ partes redimensionada por um escalar $r = 1/3$. Assim, a dimensão do conjunto de Cantor é dada por $D = -\ln 2 / \ln(1/3) = 0.631$. Esta dimensão sugere que o conjunto de Cantor deve possuir dimensão maior que a dimensão de uma coleção de pontos, mas dimensão menor que a dimensão de uma reta. A curva de Koch é composta por quatro cópias $N = 4$ dimensionada por um fator $r = 1/3$. Portanto, a dimensão da curva de Koch pode ser obtida por $D = -\ln 4 / \ln(1/3) = 1.262$.

3.2.1 Dimensão de Hausdorff

Para fractais que não possuem um conjunto de regras precisas, como os dos fractais citados acima, uma definição que podemos utilizar é a dimensão de Hausdorff-Besicovitch. Dado um conjunto de pontos $X \in \mathfrak{R}^d$ e dois reais não-negativos, s e δ , temos que:

$$H_\delta^s(X) = \inf \left\{ \sum |U_i|^s \right\} \quad (3.2.3)$$

em que U_i é uma δ -cobertura de X e $|\cdot|$ é o diâmetro em \mathfrak{R}^d , i.e., $|U| = \sup |x - y|$ sendo $x, y \in U$.

Com base na definição anterior, a s -medida de Hausdorff H^s é obtida por:

$$H^s(X) = \lim_{\delta \rightarrow 0} H_\delta^s(X) \quad (3.2.4)$$

Portanto, a dimensão Hausdoff-Besicovitch $D_H(X)$ é dada pela Equação 3.2.5.

$$D_H(X) = \inf\{s \mid H^s(X) = \infty\} = \sup\{s \mid H^s(X) = 0\} \quad (3.2.5)$$

3.2.2 Dimensão Fractal em Imagens

Para a descrição dos métodos de dimensão fractal em imagens, considere uma imagem composta por pixels $p_i = (x_i, y_i)$, onde cada pixel possui uma intensidade $I(p_i) \in [0, 255]$. Essa imagem pode ser convertida em uma superfície tridimensional mantendo-se os eixos x, y da imagem e tornando o eixo z igual a intensidade de cada pixel. Por exemplo, o pixel $p_i = (x_i, y_i)$ com intensidade $I(p_i)$ é convertido em um ponto $(x_i, y_i, I(p_i))$ no espaço 3D. Convertendo-se todos os pixels da imagem, um conjunto $S = \{s_i \mid s_i = (x_i, y_i, I(p_i))\}$ de pontos tridimensionais é obtido.

A Figura 3.14 mostra um exemplo de uma imagem em níveis de cinza convertida para um conjunto de pontos tridimensionais. Os métodos de dimensão fractal poderão, portanto, calcular a dimensão fractal desse conjunto de pontos obtidos a partir de informações da imagem. A literatura apresenta diversos métodos para estimativa da dimensão fractal de imagens. Nas seções abaixo, são apresentados os dois principais métodos.

3.2.2.1 BoxCounting

Devido à facilidade de implementação e baixo custo computacional, o método BoxCounting é um dos mais utilizados para estimar a dimensão fractal. O cálculo da dimensão baseia-se na sobreposição de uma malha sobre o espaço tridimensional e na respectiva contagem do número de quadrados necessários para cobrir toda a superfície S (Figura 3.15).

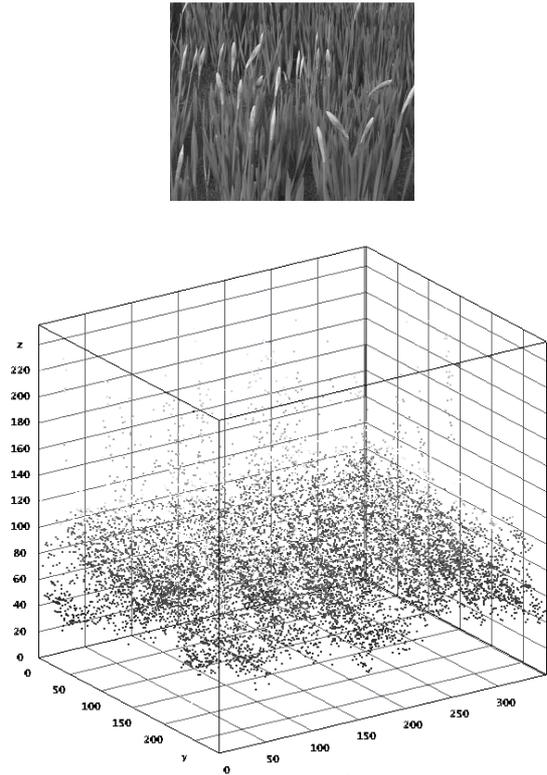


Figura 3.14 – Exemplo de uma imagem em níveis de cinza e a sua conversão para o espaço tridimensional. Cada pixel $p_i = (x_i, y_i)$ com intensidade $I(p_i)$ é convertido em um ponto tridimensional $s_i = (x_i, y_i, I(p_i))$.

A dimensão fractal por BoxCounting é obtida por meio da relação entre o tamanho da malha r e o número de quadrados que coincidem com a superfície S , $N_S(r)$:

$$D_B = 3 - \lim_{r \rightarrow 0} \frac{\log N_S(r)}{\log r} \quad (3.2.6)$$

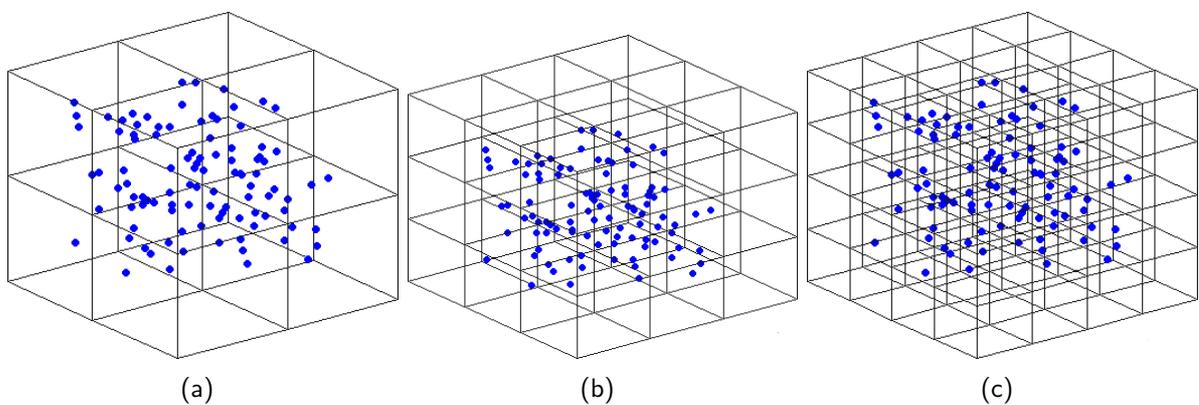


Figura 3.15 – Ilustração do método BoxCounting sobre pontos de uma superfície obtida de uma imagem.

A partir do gráfico $\log r \times \log N_S(r)$ (exemplo na Figura 3.16), a dimensão fractal por BoxCounting é obtida por:

$$D_M = 3 - \alpha \quad (3.2.7)$$

onde α é o coeficiente angular da reta.

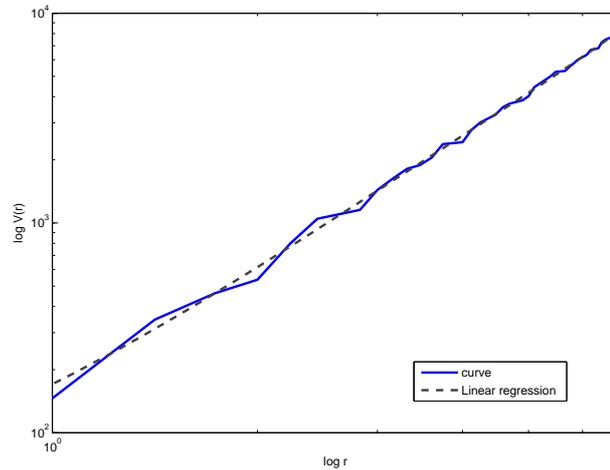


Figura 3.16 – A dimensão fractal é computada utilizando-se regressão linear do gráfico $\log r \times \log A(r)$. Dado o coeficiente angular da reta α , a dimensão fractal é estimada por $D_M = 3 - \alpha$.

3.2.2.2 Bouligand-Minkowski

Um dos principais métodos de estimativa da dimensão fractal em imagens é o Bouligand-Minkowski (77). Esse método fornece resultados precisos devido a sua sensibilidade em detectar pequenas mudanças estruturais presentes na imagem.

A dimensão fractal de Bouligand-Minkowski é obtida por meio da área de influência $A(r)$ obtida através da dilatação dos pontos da imagem por uma esfera de raio r . Portanto, a área de influência $A(r)$ é dada pelo número de pontos que se encontram a uma distância de S menor ou igual a r :

$$A(r) = \bigcup_{s_i \in S} B_r(s_i) \quad (3.2.8)$$

onde $B_r(s_i)$ é uma esfera de raio r centrada no ponto $s_i \in S$.

A dimensão fractal é obtida variando-se o raio r da esfera:

$$D_M = 3 - \lim_{r \rightarrow 0} \frac{\log A(r)}{\log r} \quad (3.2.9)$$

O processo é ilustrado na Figura 3.17. Dado uma imagem, ela é mapeada no espaço 3D para se obter o conjunto de pontos S . Em seguida, cada ponto é dilatado por uma esfera de raio r (Figura 3.17 (a), (b) and (c)). Ao aumentar-se o valor do raio r , mais colisões são observadas entre as esferas dilatadas. Estas colisões perturbam a área de influência $A(r)$, o qual está diretamente relacionada com a rugosidade da superfície e, conseqüentemente, da imagem.

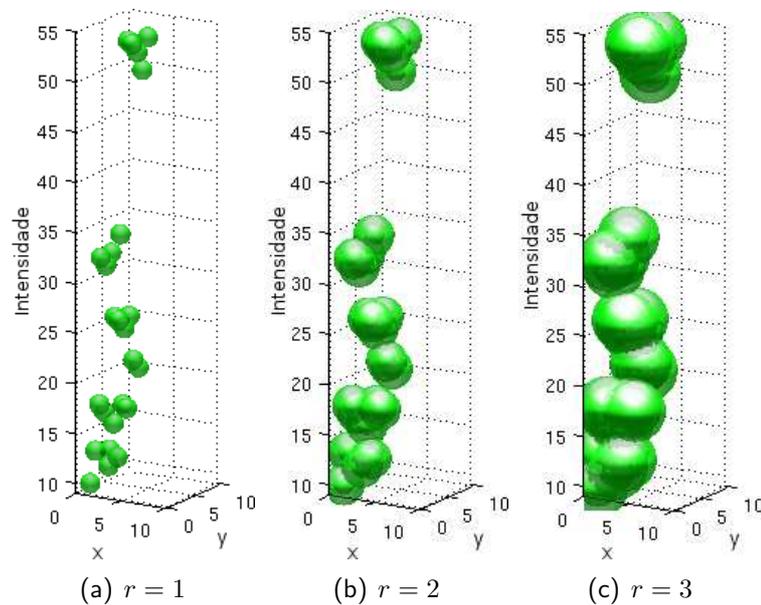


Figura 3.17 – Ilustração do processo de dilatação de pontos de uma superfície obtida de uma imagem. Cada ponto é dilatado por uma esfera com um determinado raio r .

Da mesma maneira do método BoxCounting, a dimensão fractal de Bouligand-Minkowski é obtida a partir do coeficiente angular α da regressão linear dos pontos $\log r \times \log A(r)$:

$$D_M = 3 - \alpha \quad (3.2.10)$$

Combinando Dimensão Fractal e CDPAR para Análise de Texturas Estáticas

No capítulo 3, as caminhadas determinísticas parcialmente auto-repulsivas (CDPARs) foram descritas e, baseado nelas, foi apresentado o método tradicional para caracterização de texturas. Neste capítulo, é proposto e avaliado um novo método de reconhecimento de texturas que, basicamente, consiste em estimar a dimensão fractal dos atratores obtidos por CDPARs. Esse método foi publicado em (78).

4.1 Motivação

O método original para reconhecimento de texturas baseado em CDPARs obteve bons resultados, superando alguns métodos da literatura (43). Apesar dos resultados promissores, este método considera somente o tamanho dos atratores e o tamanho do tempo de transiente como informações relevantes em sua distribuição conjunta. Certamente, os atratores fornecem muito mais informações além do seu tamanho.

Em vez de utilizar somente o tamanho e a frequência, o método apresentado aqui considera a incidência, a distribuição e o arranjo espacial dos atratores por meio dos descritores fractais. De acordo com trabalhos anteriores (77, 79), os descritores fractais permitem extrair tais informações de arranjos complexos de pixels, informações estas que demonstraram ser importantes na caracterização dos atratores e, conseqüentemente, na caracterização de imagens de texturas.

4.2 Método Proposto

A grande vantagem do método proposto é a sua habilidade em descrever os atratores da imagem. Basicamente, este método pode ser descrito em três passos:

- **Atratores da imagem:** o primeiro passo é iniciar uma CDPAR em cada pixel com o intuito de encontrar os atratores da imagem de textura.
- **Dimensão fractal dos atratores:** os atratores são dilatados até incorporarem todos os pixels da imagem. Para isso, os pixels que não pertencem a qualquer atrator são adicionados, de acordo com suas relevâncias, a um dos atratores. A relevância de um pixel é calculada como o caminho mínimo entre ele e os pixels que pertencem aos atratores.
- **Vetor de características:** o vetor é composto pelo número de pixels adicionados aos atratores em cada passo de dilatação.

Nas seções seguintes, os três passos do método são descritos em detalhes, seguido de uma análise das principais vantagens.

4.2.1 Atratores em Imagens de Textura

Diferentemente de outros trabalhos que utilizam CDPARs (e.g. (43–45, 68, 70)), o tempo de transiente não é utilizado no método proposto. Este método considera que o tempo de transiente não é uma parte útil da caminhada, pois o mesmo padrão (atrator) pode ser gerado por diferentes tempos de transiente. Este comportamento pode ser observado na Figura 4.1. Nessa figura, são apresentadas duas CDPARs utilizando $\mu = 2$ (a memória armazena o pixel atual e o anterior) e regra de movimentação *min*. Cada círculo representa um pixel da imagem

bem próxima ao melhor caso, dado que $23 \ll N$.

Algoritmo 1: Algoritmo para encontrar os atratores $A_{\mu,r}^{p_i}$ na imagem.

Data: Imagem I , vizinhança η , peso de conexão ω , tamanho de memória μ e regra de movimentação r .

Result: Atratores $A_{\mu,r}^{p_i}$

for $i \leftarrow 1$ **to** N **do**

$w_0 = p_i, t = 0, M_t^\mu = w_0;$

while *atrator não for encontrado* **do**

if $r = \min$ **then**

$w_{t+1} = \operatorname{argmin}_{p_j \in \eta(w_t), p_j \notin M_t^\mu} \omega(w_t, p_j);$

else

$w_{t+1} = \operatorname{argmax}_{p_j \in \eta(w_t), p_j \notin M_t^\mu} \omega(w_t, p_j);$

end

$M_{t+1}^\mu = \bigcup_{k=t-\mu}^t w_k;$

$t = t + 1;$

if $t > N$ **then**

break;

end

end

if *atrator foi encontrado* **then**

$A_{\mu,r}^{p_i} = \{w_\tau, \dots, w_{\tau+\rho-1}\}$

end

end

4.2.2 Dimensão Fractal de Atratores

A dimensão fractal de atratores tem como objetivo dilatar os atratores através da adição de pixels que não pertencem a qualquer atrator. Para obter uma dilatação gradual, alguns pixels precisam ser adicionados antes de outros pixels devido à sua relevância aos atratores. Neste trabalho, a relevância $\alpha(p_j)$ de um pixel p_j para o conjunto de atratores $A_{\mu,r}^{p_i}$ é dada

pelo caminho mínimo:

$$\alpha(p_j) = \min_{p_i} \{\phi(p_j, A_{\mu,r}^{p_i})\} \quad (4.2.1)$$

em que $\phi(p_j, A_{\mu,r}^{p_i})$ é a distância do pixel p_j ao atrator $A_{\mu,r}^{p_i}$ dada por:

$$\phi(p_j, A_{\mu,r}^{p_i}) = \min_{a \in A_{\mu,r}^{p_i}} \{d(p_j, a)\} \quad (4.2.2)$$

onde $d(p_j, a)$ é o caminho mínimo entre dois pixels na imagem considerando a vizinhança η e peso de conexão ω , como descrito no Capítulo 3 pelas Equações 3.1.4 e 3.1.5. Além da relevância, o atrator mais próximo do pixel p_j é armazenado em $\varepsilon(p_j)$ e será útil no processo de dilatação.

Como exemplo do cálculo da relevância, considere uma imagem de 5×5 pixels como mostrado na Figura 4.2. A intensidade de um pixel é dada dentro do círculo (número na parte inferior). Cada pixel é conectado com seus 8 vizinhos e o peso da conexão é obtido pelo módulo da diferença entre suas intensidades. Após executar as CDPARs, foram encontrados os atratores correspondentes aos pixels em preto. A relevância de cada pixel é dada dentro do círculo em sua parte superior. Por exemplo, a relevância do primeiro pixel é $\alpha(p_1) = 7$, o qual corresponde ao caminho mínimo para um dos pixels em preto (neste caso o pixel da segunda linha e segunda coluna).

A estratégia proposta aqui examina o relacionamento entre os pixels que pertencem aos atratores e os demais pixels da imagem. Por meio da relevância, cada atrator $A_{\mu,r}^{p_i}$ pode ser dilatado e sua dimensão fractal estimada. A cada nível de dilatação, pixels cuja relevância é igual ao raio ϵ são adicionados aos atratores correspondentes:

$$\delta_\epsilon(A_{\mu,r}^{p_i}) = \{p_j \mid \alpha(p_j) = \epsilon, \varepsilon(p_j) = p_i\} \quad (4.2.3)$$

Assim, a área de influência $V(\epsilon)$ do conjunto de atratores pode ser calculada como:

$$V(\epsilon) = \cup_{e=1}^{\epsilon} Q(e), \quad Q(e) = \cup_{p_i} \delta_e(A_{\mu,r}^{p_i}) \quad (4.2.4)$$

A partir do incremento do raio ϵ , mais pixels são adicionados aos atratores. O número de pixels adicionados influencia na área total $V(\epsilon)$, o qual está diretamente relacionado com a rugosidade do padrão de textura na imagem. Assim, a dimensão fractal do conjunto de atratores na imagem é dada por:

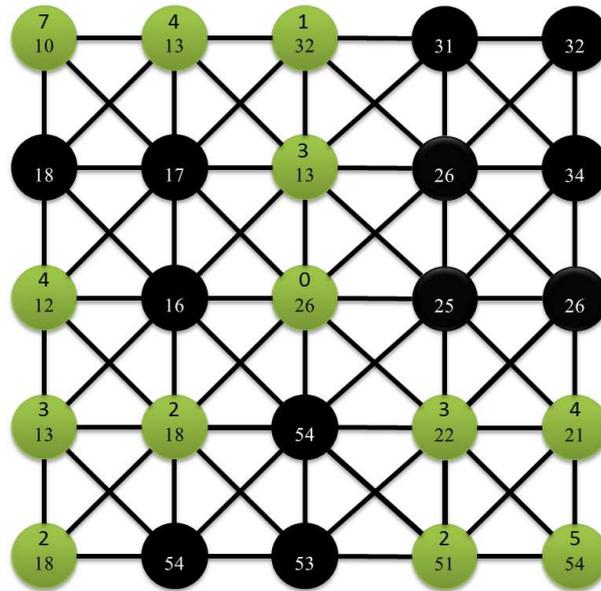


Figura 4.2 – Exemplo do cálculo da relevância de cada pixel. Cada círculo representa um pixel da imagem e a intensidade é dada dentro dele (número na parte inferior). Considerando os pixels em pretos como atratores, a relevância de cada pixel é dada na sua parte superior.

$$D_A = \lim_{\epsilon \rightarrow 0} \frac{\log V(\epsilon)}{\log \epsilon} \quad (4.2.5)$$

Um exemplo do novo processo de dilatação é dado na Figura 4.3. Dado o conjunto de atratores (Figura 4.3 (a)), a relevância de cada pixel pode ser calculada utilizando as Equações 4.2.1 e 4.2.2. Utilizando a relevância como o tempo em que cada pixel é adicionado aos atratores, o processo de dilatação para diferentes valores de raio ϵ é ilustrado nas Figuras 4.3 (b) - (g). Por exemplo, quando $\epsilon = 0$, os pixels cuja relevância é 0 são adicionados aos atratores.

A essência da dilatação dos atratores, proposta neste trabalho, pode ser computada pelo Algoritmo 2, o qual é fundamentalmente o algoritmo de Dijkstra estendido para permitir múltiplas fontes. Esse algoritmo termina em $O(NM)$ passos, onde N é o número de pixels e $M = 8N$ é o número de arestas (8 vizinhos para cada pixel). Entretanto, o tempo de execução pode ser reduzido para $O(M + N \log N)$ utilizando uma estrutura de dados heap. Em nosso algoritmo, o peso de conexão $\omega(p_i, p_j)$ são inteiros no intervalo $[0, 255]$. Para este caso especial, Dial (80) propôs um algoritmo que utiliza ordenação para obter um tempo de

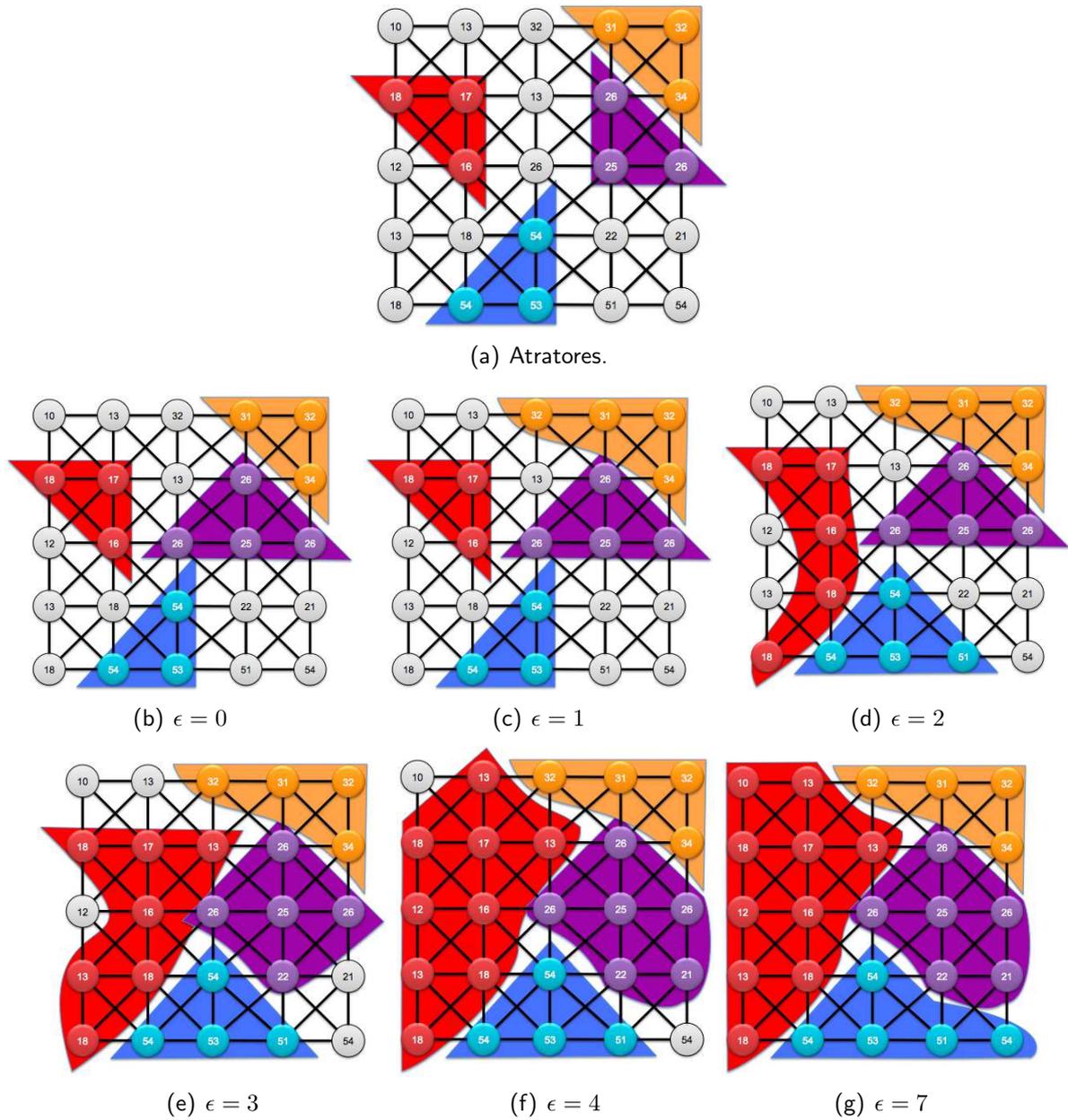


Figura 4.3 – O novo processo de dilatação utilizando a relevância de cada pixel aos atratores. A cada passo, os pixels são adicionados aos atratores de acordo com a sua relevância.

execução $O(M + NK)$.

Algoritmo 2: Algoritmo básico para calcular a relevância de cada pixel da imagem de textura.

Data: Imagem de textura I , vizinhança η , peso de conexão ω , atratores $A_{u,r}^{p_i}$.

Result: Relevância de cada pixel $\alpha(p_i)$

forall $p_j \in \bigcup_{p_i \in I} A_{u,r}^{p_i}$ **do** $\alpha(p_j) = 0$;

forall $p_j \in (I - \bigcup_{p_i \in I} A_{u,r}^{p_i})$ **do** $\alpha(p_j) = \infty$;

$\mathcal{F} = I$;

while $\mathcal{F} \neq \{\}$ **do**

 remove de \mathcal{F} um pixel p_j cujo $\alpha(p_j)$ é mínimo;

foreach $p_k \in \eta(p_j)$ **do**

if $\alpha(p_j) + \omega(p_j, p_k) < \alpha(p_k)$ **then**

$\alpha(p_k) = \alpha(p_j) + \omega(p_j, p_k)$;

end

end

end

4.2.3 Vetor de Características

Embora a dimensão fractal seja amplamente utilizada em diferentes áreas, um objeto (e.g. atratores na imagem) é representado por apenas um único número real, sua dimensão. Isso se torna uma desvantagem para fins de classificação, pois podem ser encontrados diversos objetos com a mesma dimensão fractal, mas aparências completamente diferentes. Para superar essa limitação, um vetor de características baseado no processo de dilatação do atrator é apresentado e utilizado para representar a imagem de textura.

Inicialmente, um histograma contendo o número de pixels adicionados em cada nível de dilatação ϵ é calculado de acordo com a Equação 4.2.6.

$$h_{\mu,r}(\epsilon) = \sum_{p_i} |\delta_\epsilon(A_{\mu,r}^{p_i})| \quad (4.2.6)$$

onde $|\cdot|$ representa a cardinalidade de um conjunto. Análises estatísticas têm mostrado que o histograma concentra as mais importantes informações da imagem nos seus primeiros elementos, indicando uma análise local da textura (81). Portanto, os primeiros n componentes deste histograma são selecionados para compor o vetor:

$$\psi_{\mu,r} = [h_{\mu,r}(0), h_{\mu,r}(1), \dots, h_{\mu,r}(n-1)] \quad (4.2.7)$$

O vetor de características $\psi_{\mu,r}$ depende do tamanho da memória μ e da regra de movimentação r utilizados para encontrar os atratores que foram dilatados. Alguns trabalhos anteriores (43, 45) discutiram a influência de diferentes valores destes parâmetros. Por exemplo, baixos valores de μ obtêm uma melhor análise local da imagem. Além disso, a regra de movimentação *max* tem demonstrado ser importante na caracterização de regiões heterogêneas da imagem, enquanto que a regra de movimentação *min* tem sido importante na caracterização de regiões homogêneas (43). Para caracterizar diferentes propriedades da imagem, um vetor de características φ , considerando diferentes valores de memória μ e duas regras de movimentação r , é dado pela concatenação:

$$\varphi = [\psi_{\mu_1,max}, \psi_{\mu_1,min}, \psi_{\mu_2,max}, \psi_{\mu_2,min}, \dots, \psi_{\mu_M,max}, \psi_{\mu_M,min}] , \quad (4.2.8)$$

onde *max* é a regra de movimentação em que o agente caminha para a máxima distância e *min* para a mínima distância.

4.3 Análise do Método Proposto

Nesta seção, é apresentada uma análise do método proposto utilizando classes de texturas da base de dados Brodatz (82). Essa base é amplamente utilizada na avaliação de métodos de texturas. As próximas duas seções analisam os atratores obtidos pelas CDPARs e a dilatação dos mesmos. Essa análise é importante para verificar o comportamento do método na descrição de diferentes classes de texturas.

4.3.1 Atratores em Imagens de Textura

Inicialmente, são analisados os atratores em imagens de texturas obtidos por diferentes parâmetros das CDPARs. A Figura 4.4 apresenta os atratores obtidos pelas duas regras de movimentação, $r = max$ e $r = min$. Nesta figura, os pixels pertencentes aos atratores estão desenhados em verde. Para $r = max$, os atratores são compostos por pixels de regiões heterogêneas, os quais estão relacionados à máxima diferença na intensidade dos pixels (Figura 4.4 (a)). Por outro lado, para $r = min$, os atratores são encontrados em regiões homogêneas da imagem, como pode ser visto na Figura 4.4 (b). Com o uso de ambas as regras, o método proposto é capaz de caracterizar ambas as propriedades da imagem de textura.

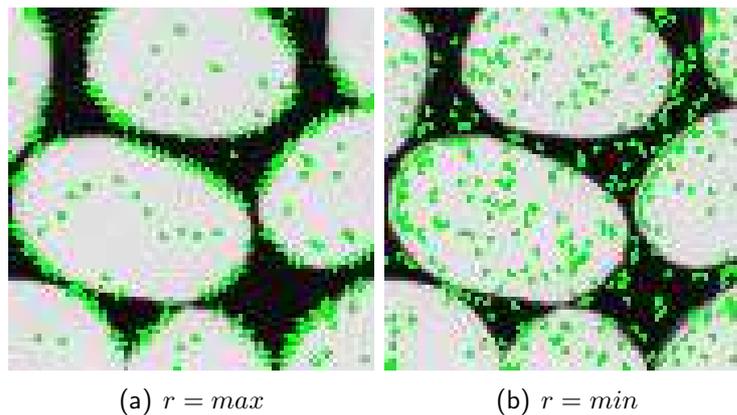


Figura 4.4 – Atratores obtidos com tamanho de memória $\mu = 2$ e duas regras de movimentação, $r = max$ e $r = min$. Os atratores estão desenhados em verde.

A Figura 4.5 ilustra os atratores obtidos com o uso de tamanhos de memória (a) $\mu = 1$, (b) $\mu = 2$, (c) $\mu = 3$, (d) $\mu = 4$. Nesta figura, são apresentados os atratores para três classes de texturas utilizando regra de movimentação $r = max$. De acordo com as figuras, o número de atratores diminui ao passo que, o tamanho de memória μ aumenta. Por exemplo, existem muito mais atratores para $\mu = 1$ (Figura 4.5 (a)) do que para $\mu = 4$ (Figura 4.5 (d)). Para valores altos de memória, a caminhada executa uma análise global da imagem, o que dificulta a descoberta de novos atratores. Nessa análise, geralmente as caminhadas tendem a ser maiores e, com isso, os mesmos atratores são encontrados por diferentes caminhadas. Por outro lado, o tamanho dos atratores (cardinalidade do conjunto de pixels) possui o comportamento oposto. Com o aumento do tamanho da memória μ , o tamanho de um atrator tende a ser maior devido ao tempo refratário $\mu - 1$.

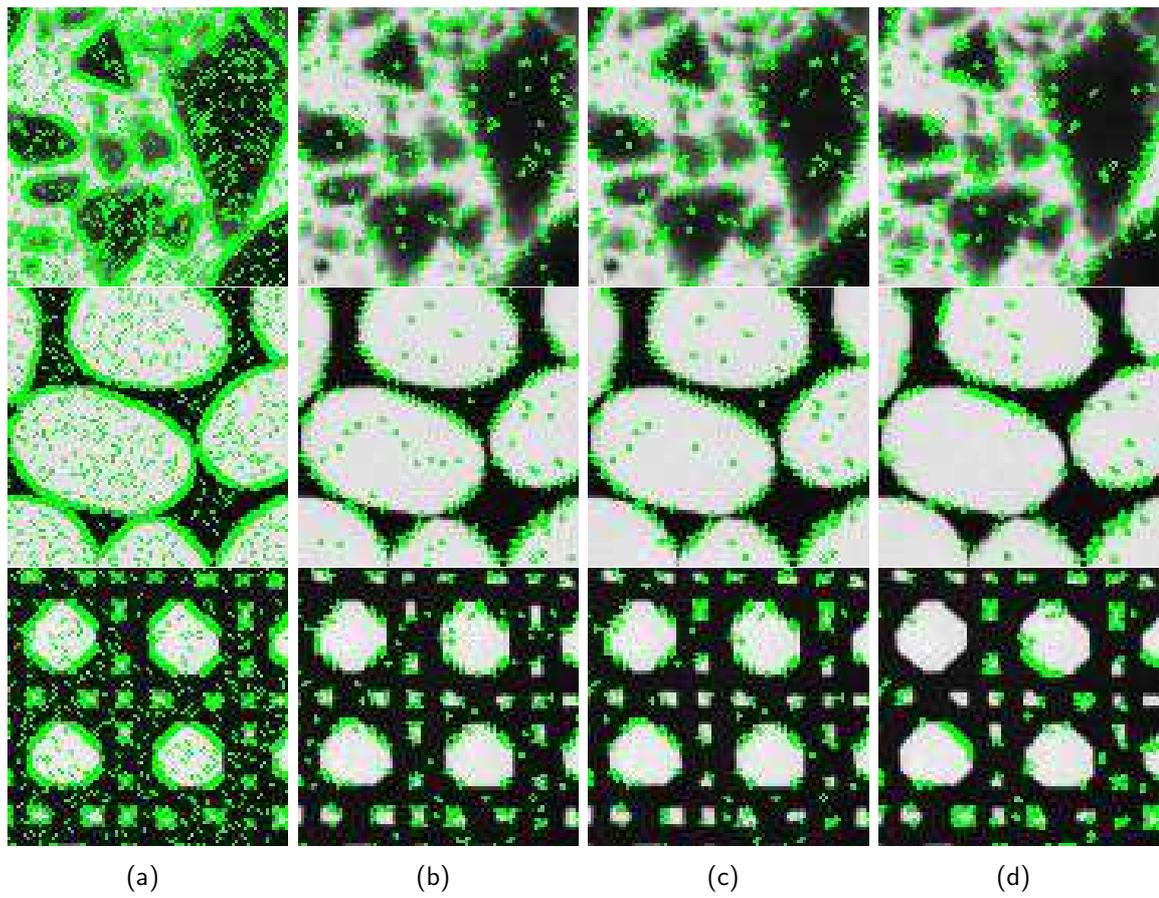


Figura 4.5 – Exemplos de atratores (pixels em verde) para $r = max$ e (a) $\mu = 1$, (b) $\mu = 2$, (c) $\mu = 3$, (d) $\mu = 4$.

4.3.2 Dilatação de Atratores em Imagens de Textura

A ideia do método proposto é gradualmente dilatar os atratores utilizando a relevância de cada pixel. Nesta seção, é analisado o procedimento de dilatação para diferentes classes de textura. A Figura 4.6 ilustra a dilatação de atratores para valores de raio $\epsilon = 5$, $\epsilon = 10$, $\epsilon = 15$. Com o incremento no valor do raio, mais pixels são adicionados aos atratores. É importante notar que o comportamento de dilatação segue as propriedades da imagem, assim preservando as características de cada textura.

Para baixos valores de dilatação (e.g. Figura 4.6(d) para raio $\epsilon = 15$), a maioria dos pixels da imagem já está adicionada aos atratores. Assim, poucos valores de raio ϵ são necessários no processo de dilatação, o que define uma importante questão relacionada ao desempenho computacional. De fato, os melhores resultados, descritos nas próximas seções, são obtidos para raio $\epsilon = 17$.

Como ilustração, na Figura 4.7 são plotados os gráficos para diferentes classes de textura, tamanhos de memória μ e regras de movimentação r . O eixo x representa os raios de dilatação e o eixo y representa o número de pixels pertencentes aos atratores. Ambos os eixos neste gráfico estão em escala logarítmica. Como pode ser visto, cada classe de textura apresenta um comportamento de dilatação diferente. Embora a textura das classes D111 e D112 sejam similares, o comportamento obtido por cada classe pode ser separado utilizando esta estratégia.

4.4 Experimentos e Resultados

Nesta seção, são apresentados os resultados do método proposto em dois experimentos: análise de parâmetros e comparação com métodos da literatura. No primeiro experimento, uma análise dos parâmetros do método foi realizada para entender o seu comportamento em reconhecimento de texturas. Em seguida, no segundo experimento, foi realizada uma

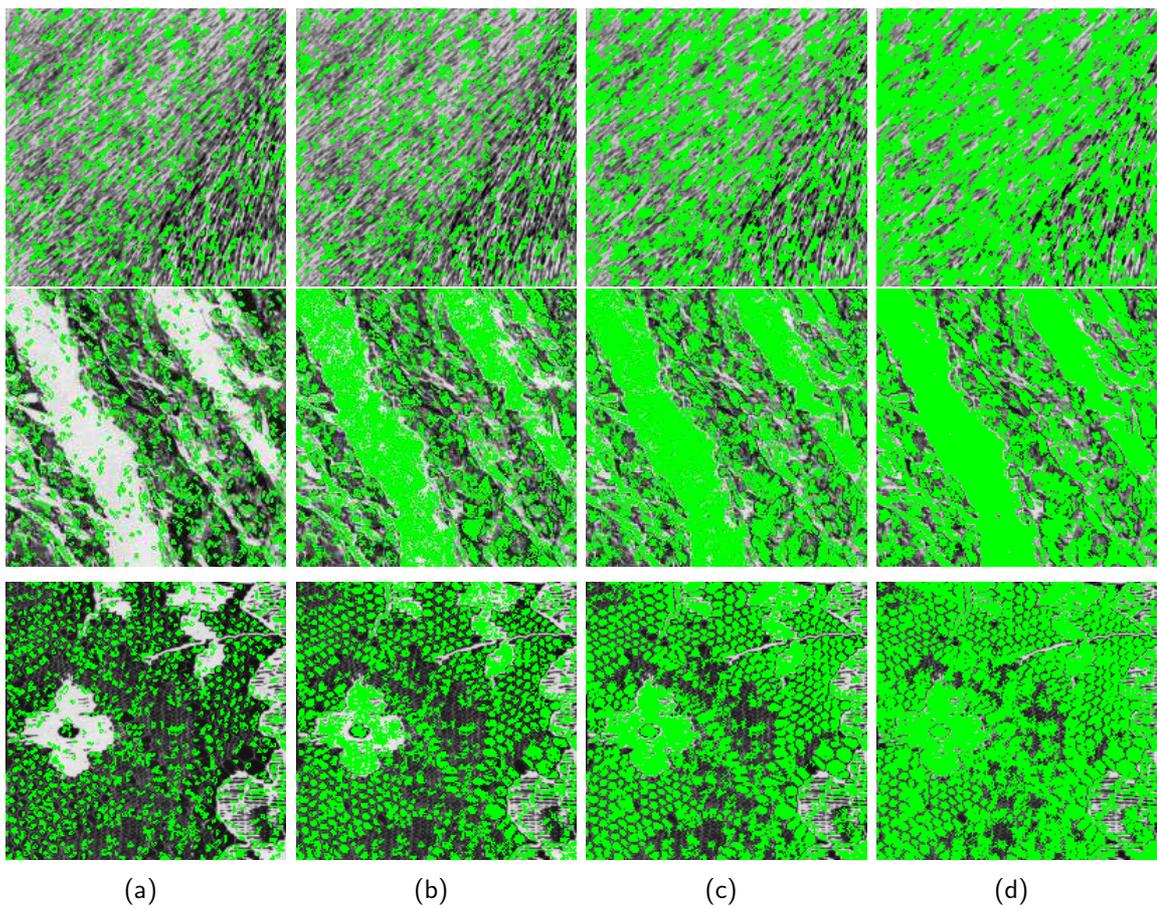
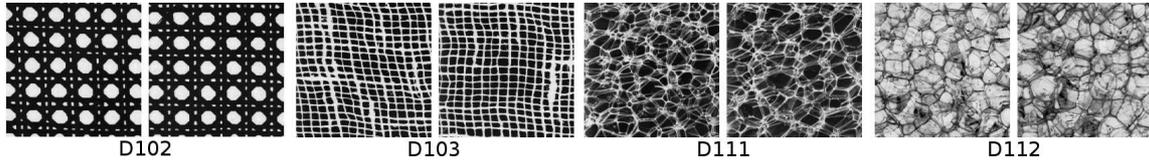


Figura 4.6 – Exemplos de dilatação de (a) atratores obtidos por caminhadas utilizando $\mu = 5$ e $r = \min$ e valores de raio (b) $\epsilon = 5$, (c) $\epsilon = 10$, (d) $\epsilon = 15$.



(a) Classes de textura.

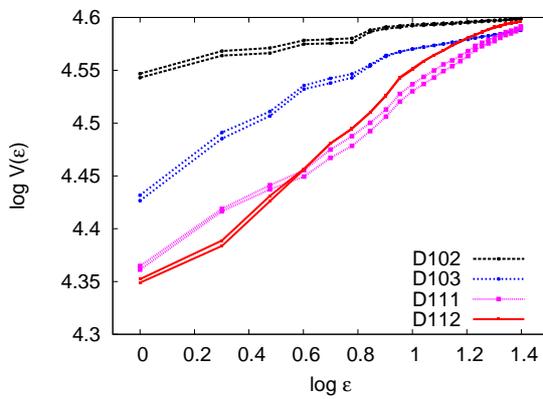
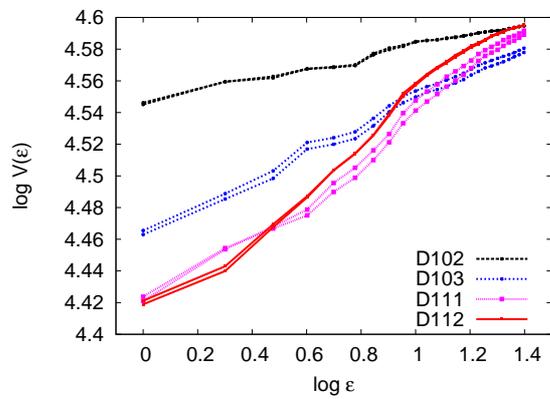
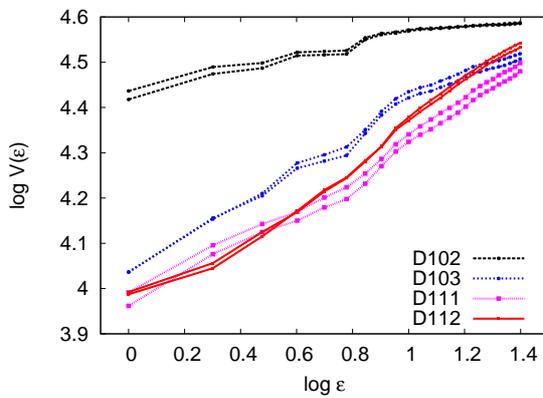
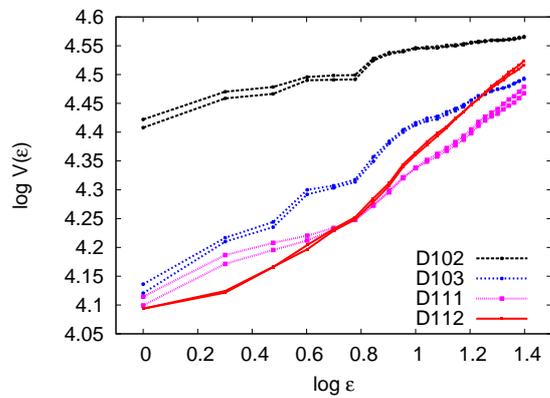
(b) $\mu = 1$ e $r = max.$ (c) $\mu = 1$ e $r = min.$ (d) $\mu = 3$ e $r = max.$ (e) $\mu = 3$ e $r = min.$

Figura 4.7 – Exemplos de dilatação para diferentes classes de texturas e valores de μ e r . O eixo x representa o raio de dilatação e o eixo y representa o número de pixels pertencentes ao conjunto de atratores. Ambos os eixos estão em escala logarítmica.

comparação entre o método proposto e as principais abordagens presentes na literatura.

4.4.1 Configuração dos Experimentos

Os vetores de características são classificados utilizando o classificador k-vizinhos mais próximos e validação cruzada em 10 pastas. Este classificador foi escolhido devido à sua simplicidade, assim enfatizando as características extraídas por cada método.

O método proposto foi avaliado em duas bases de dados: Brodatz (82) e Vistex (83). A base de dados Brodatz, composta por 111 classes, é referência para avaliar métodos de texturas. Cada classe possui 10 amostras de 200×200 pixels em níveis de cinza. Exemplos de algumas classes são apresentadas na Figura 4.8.

A base de dados Vision Texture - Vistex fornece texturas reais obtidas em condições desafiadores (e.g. diferentes iluminações e perspectivas). Um total de 54 classes de texturas está disponível – cada classe contém 16 amostras. As amostras possuem 128×128 pixels com 256 níveis de cinza. A Figura 4.9 apresenta exemplos de classes contidas nessa base.

Para avaliar o método proposto e comparar seu desempenho, os seguintes métodos da literatura foram utilizados: matriz de co-ocorrência (84), filtros de Gabor (31), descritores de Fourier (85), caminhada determinística do turista (43), dimensão fractal multiescala (4), espectro multifractal (39), dimensão fractal local (40), comprimento fractal local (40) e padrões locais binários (26). Foram consideradas as implementações tradicionais de cada método.

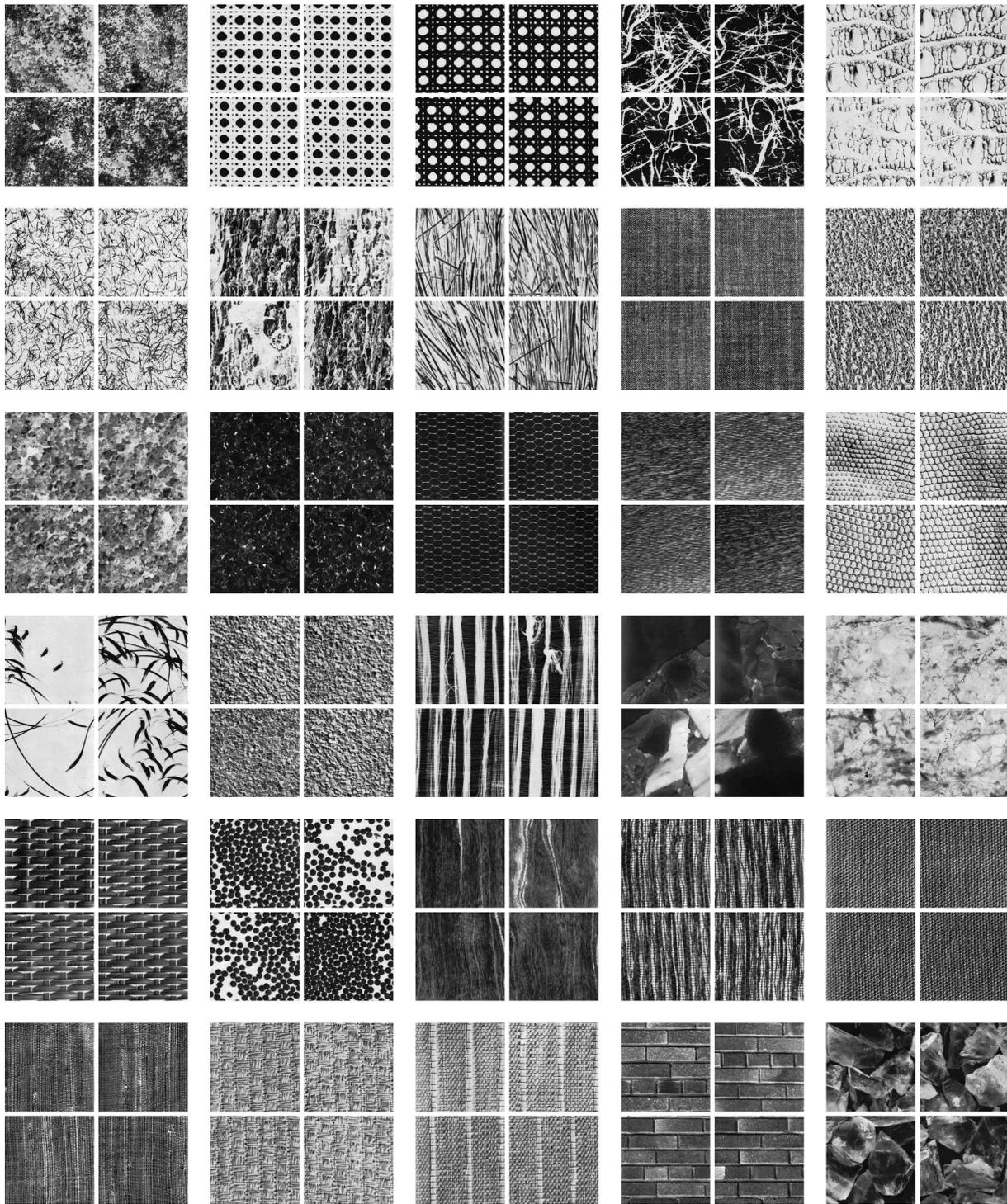


Figura 4.8 – Exemplos de algumas classes da base de dados Brodatz. Cada uma das 111 classes possui 10 amostras de 200×200 pixels.

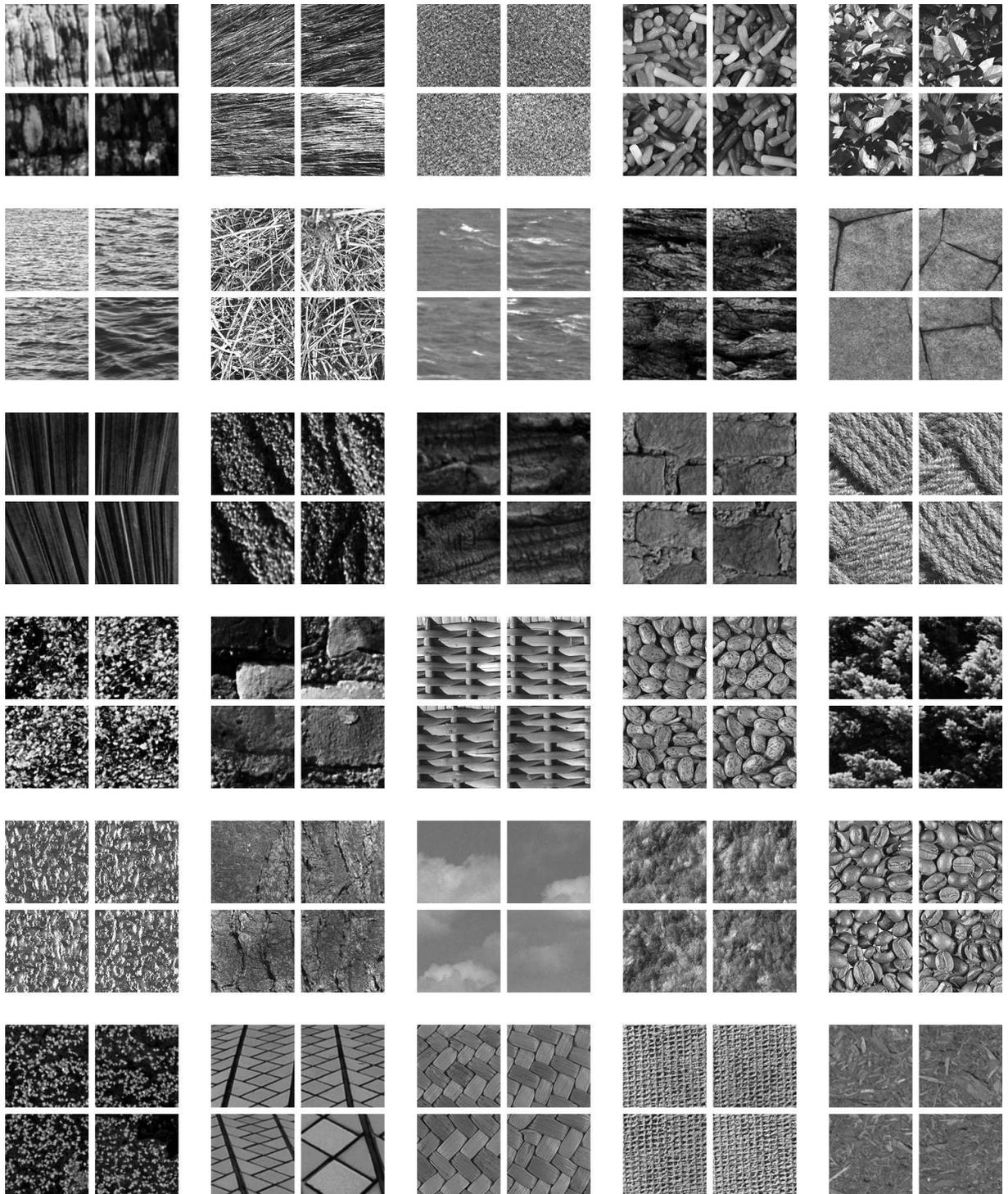


Figura 4.9 – Exemplos de classes da base de dados vistex. Cada uma das 54 classes possui 16 amostras de 128×128 pixels.

4.4.2 Avaliação dos Parâmetros

Nesta seção, diferentes configurações dos parâmetros são avaliadas em termos da taxa de classificação correta. Este experimento tem como objetivo analisar a influência de cada parâmetro e, assim, obter a melhor configuração para a classificação de texturas. A Tabela 4.1 apresenta os resultados para diferentes tamanhos de memória e regras de movimentação utilizados nas CDPARs. A partir da tabela, é observado que a taxa de classificação correta decreta ao passo que, o tamanho da memória μ é incrementado. Os baixos valores de μ forneceram os melhores resultados devido à análise local obtida pelas caminhadas (43).

Tabela 4.1 – Taxa de classificação correta para diferentes tamanhos de memória μ e regras de movimentação r utilizados nas CDPARs. A dilatação máxima n foi 17.

	Tamanho de Memória (μ)					
	1	2	3	4	5	6
<i>max</i>	93.69	90.99	90.45	89.01	88.74	86.13
<i>min</i>	96.22	94.23	93.15	91.17	89.64	89.19
$[min, max]$	96.40	94.59	93.87	94.41	93.96	91.89

Além disso, verificou-se que a regra de movimentação $r = min$ obtém resultados superiores aos resultados obtidos pela regra $r = max$. Neste caso, os atratores são encontrados em regiões homogêneas da imagem. Embora resultados interessantes tenham sido obtidos por cada regra de movimentação, os melhores resultados foram obtidos pela concatenação de ambas as regras $[min, max]$, a qual combina diferentes propriedades da imagem e fornece um método poderoso para caracterizar e classificar texturas.

Na Tabela 4.2, são apresentados os resultados obtidos através da concatenação de tamanhos de memória. É observado que os resultados para concatenação de memórias sempre superam os resultados para memórias individuais. Entretanto, aumentar o número de memórias não fornece, necessariamente, melhores resultados. Como apresentado na tabela, uma taxa de classificação correta de 97.30% foi obtida pela concatenação de tamanhos de memória $[1, 2, 3]$ e regras de movimentação $[min, max]$.

Finalmente, o gráfico da Figura 4.10 apresenta a avaliação da dilatação máxima n . Os melhores resultados foram obtidos para $n > 10$, pois para altos valores de n , o método explora uma melhor gama de dilatação. Por outro lado, para pequenos valores de n (menores que 10), o método não caracteriza todas as características da textura presentes na imagem. É

Tabela 4.2 – Taxa de classificação correta para diferentes tamanhos de memória utilizados nas CD-PARs. A dilatação máxima n é igual a 17.

Tamanhos de memória (μ)	max	min	$[min, max]$
[1]	93.69	96.22	96.40
[1, 2]	95.77	96.58	96.81
[1, 2, 3]	95.68	96.49	97.30
[1, 2, 3, 4]	95.68	96.49	97.21
[1, 2, 3, 4, 5]	95.59	96.22	97.21
[1, 2, 3, 4, 5, 6]	95.41	96.31	97.12

possível notar que, a taxa de classificação correta alcançou o máximo de 97.30% para $n = 17$ e concatenação das regras de movimentação $[min, max]$. Neste gráfico, foram utilizados tamanhos de memória que fornecem os melhores resultados, como descrito anteriormente.

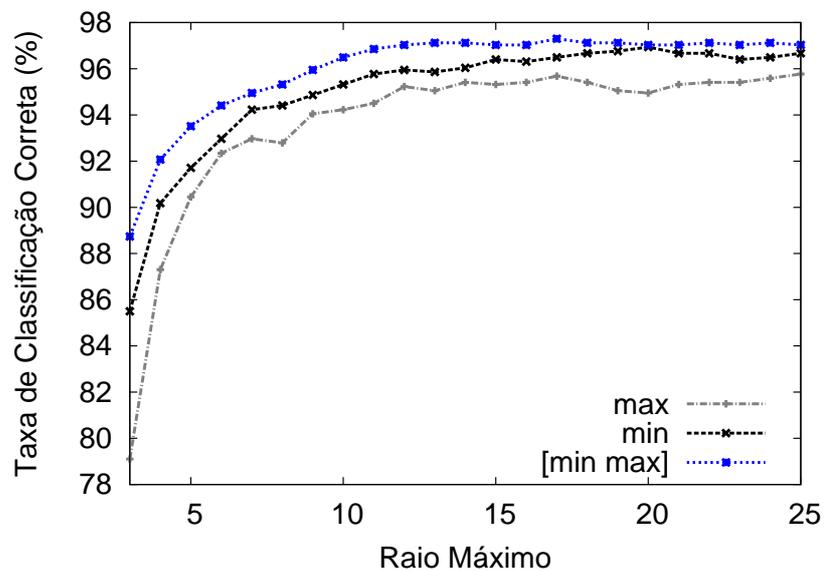


Figura 4.10 – Taxa de classificação correta para valores máximos de dilatação n . Foram utilizados tamanhos de memórias 1, 2, 3 e concatenação entre as regras de movimentação min e max .

4.4.3 Comparação com Métodos da Literatura

Nesta seção, é apresentada uma comparação do método proposto com alguns dos principais métodos da literatura. É importante comentar que o método proposto é exatamente o mesmo dos experimentos anteriores com parâmetros $\mu = [1, 2, 3]$, $r = [min, max]$ e $n = 17$. A comparação é realizada com métodos tradicionais da literatura, além das caminhadas determinísticas do turista e métodos de fractal. Os resultados experimentais são reportados em termos da taxa de classificação correta.

A Tabela 4.3 apresenta os resultados obtidos na base de dados Brodatz. Para essa base, a taxa de classificação correta de 97.30% foi obtida pelo método proposto, o que claramente demonstra a robustez da estratégia de dilatação dos atratores. A segunda maior taxa de classificação foi obtida pelos padrões locais binários, um importante método de reconhecimento de texturas. Os resultados experimentais indicam que, o método proposto incrementa a taxa de classificação correta sobre os métodos nos quais ele é baseado, por exemplo, de 89.37% para 97.30% sobre as caminhadas determinísticas do turista e de 94.05% para 97.30% sobre a dimensão fractal local.

Tabela 4.3 – Resultados comparativos para diferentes métodos de caracterização de texturas na base de dados Brodatz.

Método	Taxa de Classificação Correta
Descritores de Fourier	70.67(± 3.39)
Matrizes de co-ocorrência	86.49(± 2.69)
Filtros de Gabor	88.99(± 2.03)
Caminhada determinística do turista	89.37(± 1.96)
Caminhadas-Grafos	85.52(± 2.70)
Dimensão fractal multiescala	89.11(± 2.80)
Espectro multifractal	75.05(± 3.41)
Dimensão fractal local	94.05(± 2.07)
Comprimento fractal local	91.62(± 2.21)
Padrões locais binários	96.22(± 1.48)
Método proposto	97.30(± 1.36)

Comparações entre os métodos de texturas utilizando a base de dados Vistex são reportadas na Tabela 4.4. De acordo com os resultados, o método proposto fornece uma taxa de classificação correta de 85.33%. Como comparação, a caminhada determinística do turista e a dimensão fractal local obtiveram taxas de classificação correta de 82.48% e 80.09%,

respectivamente. Entretanto, deve ser notado que, para esta base de dados, a melhor taxa de classificação correta foi de 93.75%, obtida pelos padrões locais binários. Todavia, nas duas bases de dados, o método proposto apresentou bons resultados comparado com métodos tradicionais da literatura.

Tabela 4.4 – Resultados comparativos para diferentes métodos de caracterização de texturas na base de dados Vistex.

Método	Taxa de Classificação Correta
Descritores de Fourier	72.25(\pm 4.41)
Matrizes de co-ocorrência	80.51(\pm 3.49)
Filtros de Gabor	84.86(\pm 3.05)
Caminhada determinística do turista	82.48(\pm 3.67)
Caminhadas-Grafos	78.28(\pm 3.81)
Dimensão fractal multiescala	78.52(\pm 3.84)
Espectro multifractal	65.97(\pm 3.92)
Dimensão fractal local	80.09(\pm 3.55)
Comprimento fractal local	67.48(\pm 4.14)
Padrões locais binários	93.75(\pm 2.20)
Método proposto	85.33(\pm 3.81)

4.5 Conclusão

Neste capítulo, foi proposto um método para caracterizar imagens de texturas estáticas baseado em caminhadas determinísticas parcialmente auto-repulsivas e dimensão fractal. Por meio da relevância de cada pixel, foi apresentada uma estratégia de dilatação de atratores que permite compor um vetor de características para a classificação de texturas. Resultados promissores, em duas bases de dados, demonstraram a eficácia e robustez do método proposto em caracterização de texturas.

Atividade em Redes Direcionadas aplicada ao Reconhecimento de Texturas Estáticas

Neste capítulo, é proposto e avaliado um método para reconhecimento de texturas baseado em atividade em redes direcionadas. A atividade é, basicamente, a frequência relativa de visitas em equilíbrio de cada nó da rede. Esse método foi submetido para revista *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

5.1 Método Proposto

O método proposto pode ser descrito em três etapas:

- **Construção da rede a partir da imagem:** a imagem é mapeada em uma rede regular direcionada (i.e., todos os nós possuem o mesmo número de conexões). Aplicando-se uma função sobre as conexões (e.g. remover conexões cujo peso é maior que um determinado limiar), é possível transformar essa rede em uma determinada rede que apresenta diferentes propriedades e transitividade (86).
- **Estimação da atividade:** após a construção da rede direcionada transformada por um limiar, são iniciadas caminhadas aleatórias em cada nó. Essas caminhadas são utilizadas para estimar a atividade, isto é, a frequência relativa de visitas em equilíbrio de cada nó.

- **Construção do vetor de características:** a atividade de um nó e o seu grau de entrada são, então, utilizados para compor o vetor de características que descreve a textura presente na imagem.

Os três passos, descritos acima, são detalhados nas próximas seções.

5.1.1 Modelando Imagens em Redes Direcionadas

Considere uma imagem que consiste em um par (\mathcal{I}, I) , onde \mathcal{I} é um conjunto finito de pixels e I atribui uma intensidade $I(p_i) \in [0, 255]$ para cada pixel $p_i = (x_{p_i}, y_{p_i}) \in \mathcal{I}$. Para modelar uma imagem como uma rede $N = (V, E)$, cada pixel é mapeado em um nó do conjunto V . Inicialmente, a rede regular é obtida conectando-se dois nós, v_i e v_j , se a distância Euclidiana entre seus pixels correspondentes, p_i e p_j , for igual ou menor que um determinado raio r . Um exemplo de rede regular para $r = 2$ é apresentado na Figura 5.1.

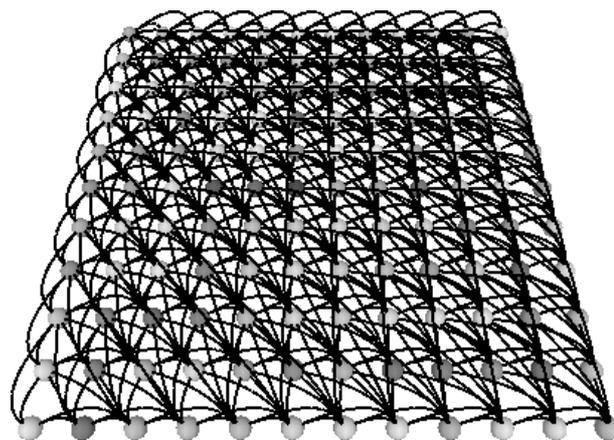


Figura 5.1 – Exemplo de uma imagem mapeada em uma rede regular. Cada nó é conectado com os demais nós dentro de um raio $r = 2$.

Para cada conexão, um peso, definido pela diferença entre as intensidades dos pixels, é atribuído de acordo com:

$$e_{v_i, v_j} = \begin{cases} I(p_i) - I(p_j), & \text{se } \text{dist}(p_i, p_j) \leq r \\ \text{NaN}, & \text{caso contrário} \end{cases} \quad (5.1.1)$$

onde $\text{dist}(p_i, p_j) = \sqrt{(x_{p_i} - x_{p_j})^2 + (y_{p_i} - y_{p_j})^2}$ é a distância Euclidiana entre os pixels p_i e p_j . A diferença entre intensidades, $I(p_i) - I(p_j)$, não é afetada por mudança na luminosidade e, portanto, os pesos da rede são invariantes com relação à mudanças de iluminação.

Analisando-se a Equação 5.1.1, é possível notar que pesos negativos são obtidos para conexões onde $I(p_j) > I(p_i)$. Para se obter uma rede direcionada, essas conexões com pesos negativos são descartadas. Desta maneira, uma conexão é considerada dirigida de v_i para v_j se seu peso $e_{v_i, v_j} > 0$. Além da dinâmica de difusão, a principal diferença dessa modelagem e as demais propostas anteriormente (86–87), é o uso de uma rede direcionada. Como apresentado nos experimentos a seguir, a rede direcionada modela adequadamente a textura além de ser computacionalmente mais eficiente. Isso reflete a capacidade de se identificar agrupamentos no espaço estrutura \times dinâmica da rede (88).

Dada a rede regular direcionada, é possível aplicar uma função nas conexões para obter-se uma rede que revela diferentes propriedades da textura. Essa função, definida como $\phi(t, N)$, consiste em remover conexões da rede de acordo com seus pesos (86–87). Como apresentado na Equação 5.1.2, as conexões com pesos menores ou iguais à t são mantidas. Exemplos da aplicação dessa função sobre a rede regular da Figura 5.1 são ilustrados na Figura 5.2.

$$e_{v_i, v_j} = \begin{cases} e_{v_i, v_j}, & \text{se } 0 < e_{v_i, v_j} \leq t \\ \text{NaN}, & \text{caso contrário} \end{cases} \quad (5.1.2)$$

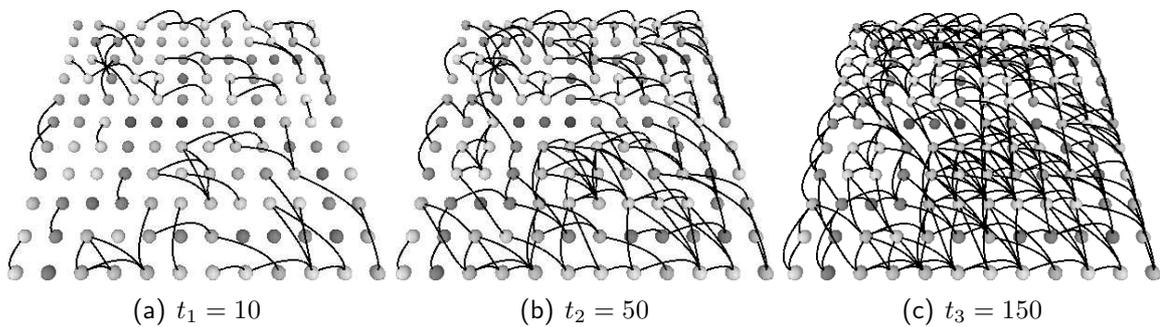


Figura 5.2 – Exemplo da função ϕ aplicada em uma rede regular para valores de limiar: (a) 10, (b) 50, e (c) 150.

A aplicação da função $\phi(t, N)$ pode ser considerada como uma análise multi-escala da rede (86). Para cada limiar t , a rede regular é transformada em uma rede com escala t . Comparada com a rede regular, essa rede apresenta diferentes propriedades, revelando a estrutura e a

topologia relacionada com a sua escala. Para pequenos valores de t , a rede realça os detalhes da imagem por meio de pequenos conjuntos de pixels conectados (e.g. Figura 5.2(a)). A medida que o limiar t é incrementado, a rede modela informações globais da imagem, como as bordas dos objetos. Assim, o método proposto combina medidas (atividade dos nós) extraídas de redes transformadas com diferentes valores de t para obter uma análise multi-escala da rede.

5.1.2 Atividade em Imagens de Textura

A atividade de cada nó é estimada por caminhadas aleatórias. Para descrever essas caminhadas, suponha que o agente está em um nó v_i . O próximo passo é ir para um nó vizinho v_j com probabilidade:

$$p(v_i, v_j) = \frac{e_{v_i, v_j}}{\sum_{v_k \in V} e_{v_i, v_k}} \quad (5.1.3)$$

O agente realiza sucessivos passos aleatórios de acordo com a Equação 5.1.3. Assim, a atividade $\alpha(v_i)$ é dada pelo número de vezes que o nó v_i foi visitado pelo agente. Para estimar a atividade com mais precisão, M caminhadas aleatórias são iniciadas em cada nó da rede. O agente estabiliza quando o nó em que ele se encontra não possui conexões de saída ou o comprimento da sua caminhada é maior que um dado limiar.

Na Figura 5.3, é demonstrada a atividade em imagens de textura para diferentes valores de limiar t e raio r utilizados para obter a rede direcionada. É possível notar que a atividade de cada pixel mantém as principais características da textura. Os pixels próximos das bordas tendem a ter atividade elevada, uma vez que os pesos de suas conexões são maiores do que os pesos das conexões de pixels em regiões homogêneas. Por exemplo, considere um pixel p_i com intensidade $I(p_i) = 255$ (pixel branco) e um pixel vizinhos p_j com intensidade $I(p_j) = 0$ (pixel preto). O peso dessa conexão é $e_{v_i, v_j} = 255 - 0 = 255$ (o peso máximo possível), o que tende a atrair os agentes aleatórios para o pixel p_j .

Também pode ser observado da Figura 5.3 que, a medida que o limiar t é incrementado,

os valores de atividade altos estão concentrados em poucos pixels. Isso ocorre porque quando t aumenta, mais conexões são mantidas na rede e, portanto, o agente se move rapidamente para os pixels da borda. O mesmo comportamento é observado para os valores de raio r . A medida que r aumenta, o agente alcança mais rápido os pixels das bordas, sem a necessidade de passar por pixels intermediários.

Para demonstrar a capacidade de caracterização de texturas, a Figura 5.4 apresenta a atividade dos pixels para diferentes imagens. A rede direcionada foi obtida para $t = 100$ e $r = \sqrt{2}$. Como pode ser observado, a atividade demonstra ser uma boa medida para reconhecer as classes de textura.

5.1.3 Vetor de Características

Como observado na seção anterior, a atividade incorpora importantes informações da textura. A fim de melhorar essa medida, é proposto um vetor de características composto pelo histograma da atividade. O histograma $h^{r,t}(j)$ correlaciona a atividade $\alpha(v_i)$ e o grau de entrada $k_{v_i}^{in}$. Esse histograma é obtido de acordo com:

$$h^{r,t}(j) = \sum_{v_i \in V} \begin{cases} \alpha^{r,t}(v_i), & \text{se } k_{v_i}^{in} == j \\ 0, & \text{caso contrário} \end{cases} \quad (5.1.4)$$

onde $\alpha^{r,t}(v_i)$ é a atividade do nó v_i obtida em uma rede com raio r e limiar t . O histograma também é normalizado pelo número de pixels para obter invariância com relação ao tamanho da imagem.

O histograma descrito acima pode ser interpretado da seguinte maneira. Quanto maior o grau de entrada $k_{v_i}^{in}$, maior a probabilidade do nó ser visitado. Entretanto, isto nem sempre ocorre devido ao peso das conexões, o que reflete diretamente as propriedades da textura na imagem. Assim, a atividade dos nós e o histograma contêm informações correlacionadas com a textura. Exemplos para histograma $h^{r,t}(j)$ com $t = 80$ e $r = \sqrt{8}$ são mostrados na Figura 5.5. Cada textura possui quatro amostras apresentadas na Figura 5.5 (a). Estes exemplos são desafiadores, pois as imagens foram capturadas em diferentes pontos de vista (e.g. diferentes

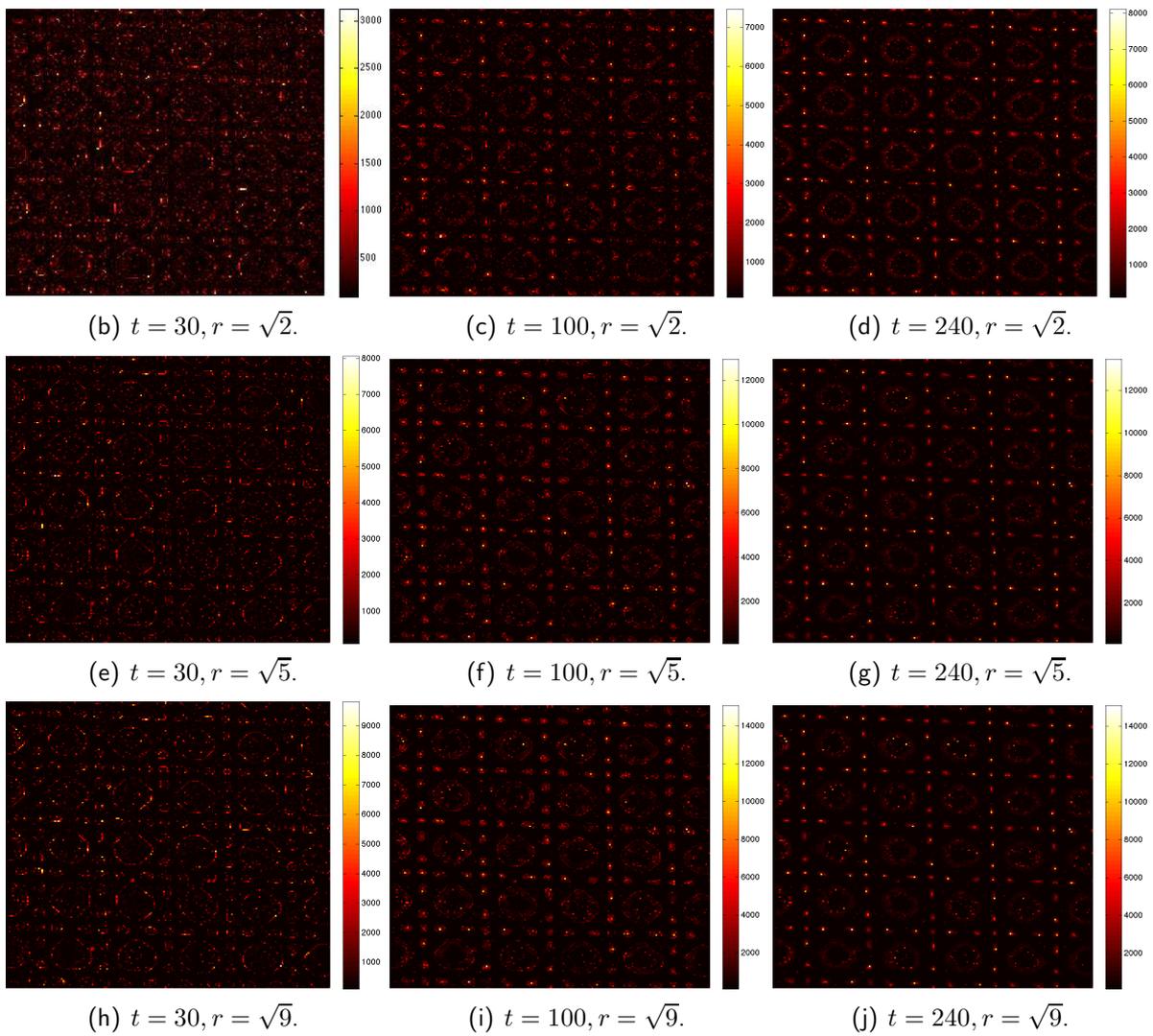
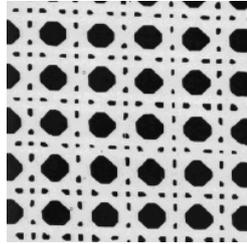


Figura 5.3 – Atividade de cada nó/pixel para valores de limiar $t = 30, 100, 240$ e raios $r = \sqrt{2}, \sqrt{5}, \sqrt{9}$ em uma imagem de textura.

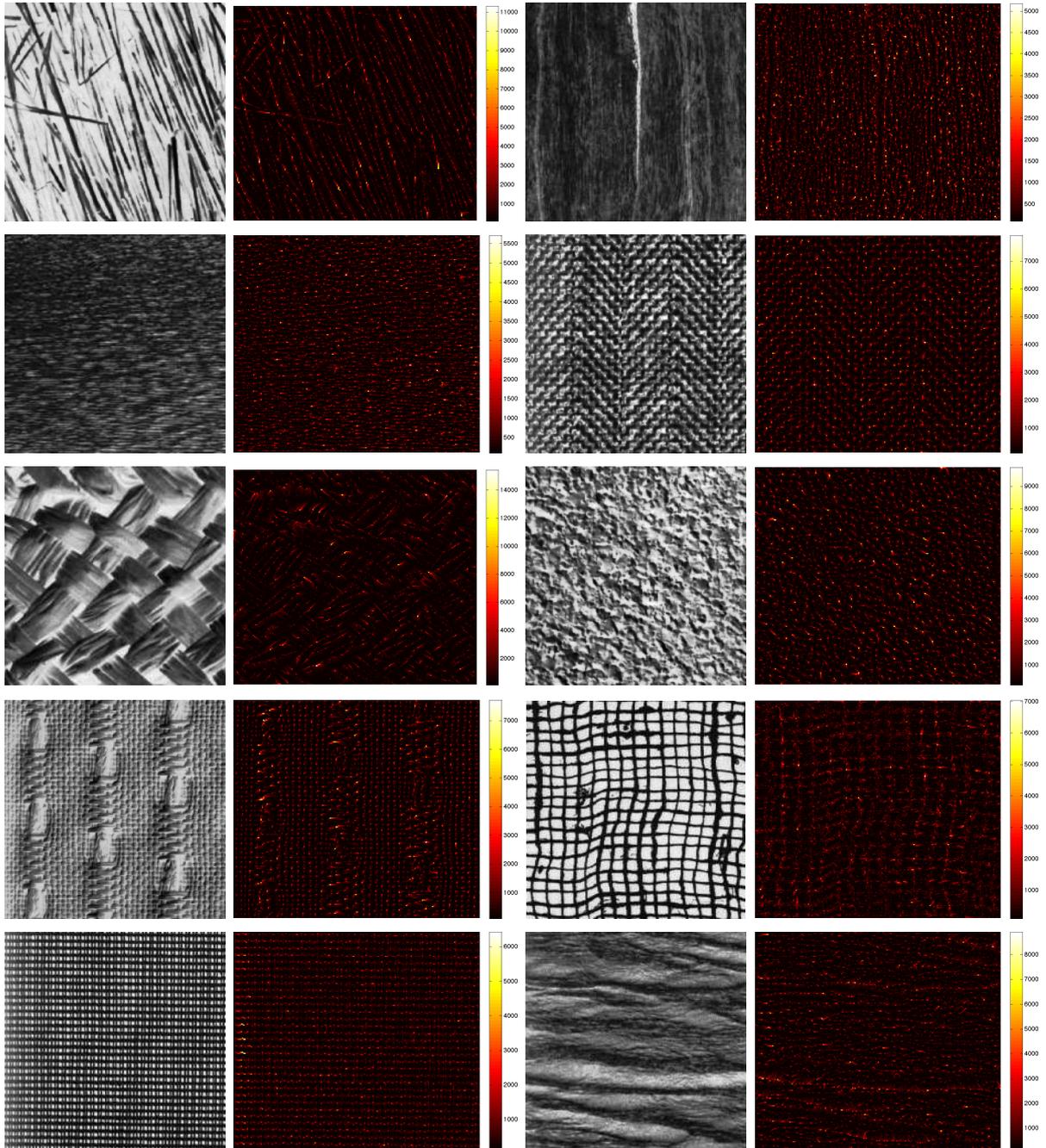
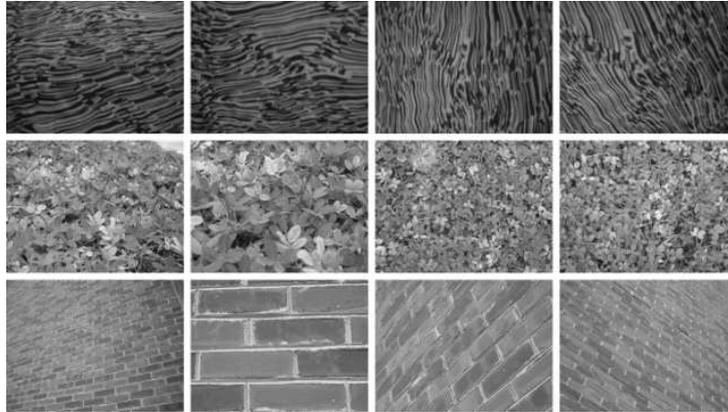
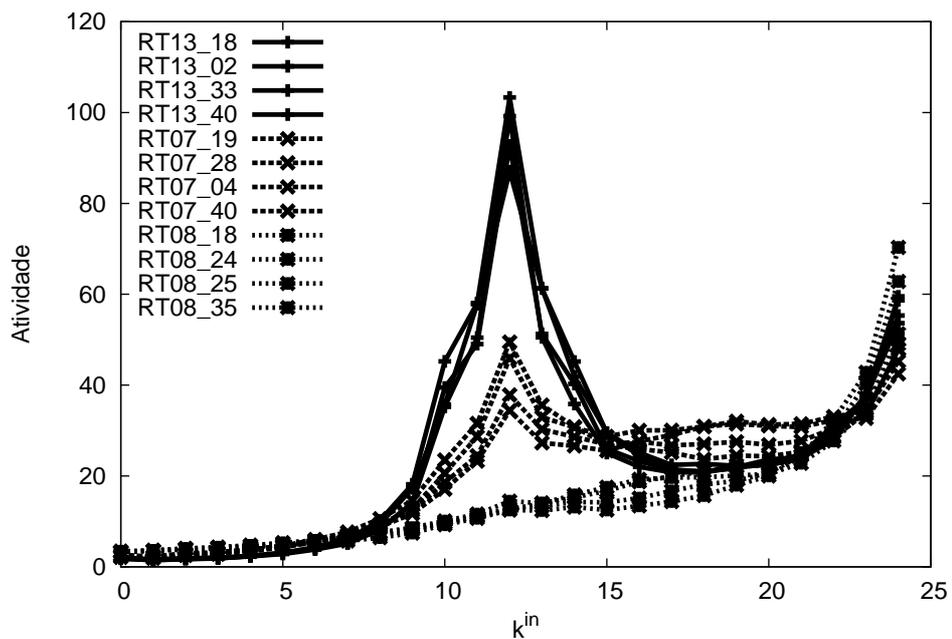


Figura 5.4 – Atividade de cada nó/pixel para diferentes texturas. A rede direcionada foi obtida para $t = 100$ e $r = \sqrt{2}$.

escalas, orientações e perspectivas). A partir do gráfico, pode-se observar que os histogramas da mesma classe são similares, o que corrobora a robustez sob mudanças em pontos de vista.



(a) Da esquerda para direita: RT13_18, RT13_02, RT13_33, RT13_40, RT07_19, RT07_28, RT07_4, RT07_40, RT08_18, RT08_24, RT08_25, RT08_35



(b)

Figura 5.5 – Os histogramas para três classes de textura. A rede direcionada foi obtida com parâmetros $r = \sqrt{8}$ e $t = 80$.

Para obter a análise multi-escala, os histogramas para diferentes valores de t são concatenados. Os valores de limiares estão no intervalo $t_0 \leq t \leq t_f$ e incrementados por uma constante t_i . Assim, o vetor de características φ^r , que considera informação de diferentes redes transformadas com mesmo raio r , é dado por:

$$\varphi^r = [h^{r,t_0}, h^{r,t_0+t_i}, \dots, h^{r,t_f}] \quad (5.1.5)$$

Por fim, o vetor de características φ é dado pela concatenação de vetores φ^r para diferentes valores de raio r :

$$\varphi = [\varphi^{r_0}, \dots, \varphi^{r_n}] \quad (5.1.6)$$

5.1.4 Complexidade Computacional

O método proposto modela uma imagem de $n \times n$ pixels como uma rede direcionada com n^2 nós. Em seguida, M caminhadas aleatórias são iniciadas em cada nó da rede. Portanto, a complexidade computacional é dada por $O(Mn^2l)$, a qual é baseada no número de nós n^2 e o tamanho de cada caminhada l . Como M é uma constante e geralmente $M \ll n^2$ (e.g. $M = 100$ e $n^2 = 40000$), pode-se suprimir essa constante multiplicativa, o que nos leva a uma complexidade $O(n^2l)$.

A principal variável que afeta a complexidade é o tamanho das caminhadas aleatórias. Abaixo é discutido o melhor, o pior e o caso médio da complexidade baseado no tamanho l . Apesar da sua não aplicabilidade em problemas reais, o melhor caso ocorre quando $l = 1$ para todas as caminhadas (i.e. os nós não estão conectados), o que leva a uma complexidade $O(n^2)$. O pior caso ocorre quando o tamanho de todas as caminhadas é o máximo possível. Neste caso, as caminhadas são paradas quando seus tamanhos são iguais ao limiar L (neste trabalho $L = 500$). O pior caso leva a uma complexidade $O(Ln^2)$.

Para avaliar o caso médio a Figura 5.6 apresenta o tamanho médio de caminhadas aleatórias executadas em 2292 imagens com diferentes valores de t e r para obter a rede direcionada. Basicamente, o tamanho das caminhadas depende do número de conexões saindo do nó. Se o agente está em um nó sem conexões de saída, sua caminhada é parada e o seu tamanho l tende a ser pequeno. Para $t = 10$ e $r = \sqrt{5}$, o tamanho médio l é igual a 4.92, levando a uma complexidade muito próximo ao melhor caso $O(n^2)$. O caso médio é então um excelente resultado em comparação com as complexidades de métodos tradicionais de textura,

como os filtros de Gabor ($O(n^2 \log n)$) e as matrizes de co-ocorrência ($O(n^2)$).

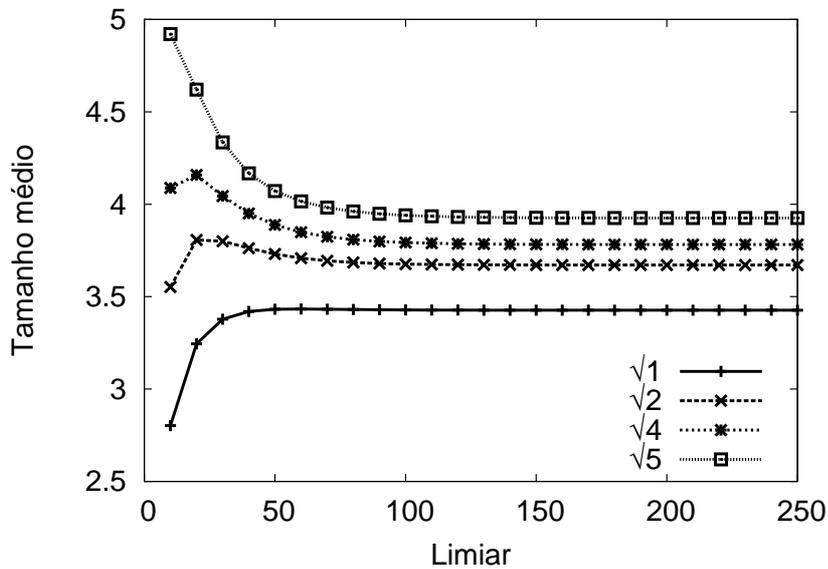


Figura 5.6 – Tamanho médio de caminhadas aleatórias executadas em 2292 imagens de tamanho 128×128 pixels. As curvas correspondem à diferentes valores de raios r .

5.2 Análise de Invariância

Nesta seção, é discutida a invariância do método proposto sob mudanças nos pontos de vista da imagem. Dado duas redes $N^I = (V^I, E^I)$ e $N^{I'} = (V^{I'}, E^{I'})$ obtidas respectivamente de duas imagens I e I' , N^I é dita ser isomórfica de $N^{I'}$ se existe um mapeamento um-para-um $f : V^I \rightarrow V^{I'}$ tal que $e_{v_i, v_j} \in E^I$ se e somente se $e_{f(v_i), f(v_j)} \in E^{I'}$. Considere duas redes isomórficas e suas matrizes de adjacência A , isto é, uma matriz $n \times n$ onde cada posição a_{ij} corresponde ao peso da conexão entre o vértice v_i e o vértice v_j . A atividade de cada nó pode ser analiticamente obtida por A^m , onde m é o número máximo de passos do agente. Dado o mapeamento f , é possível reorganizar as matrizes de adjacência de tal maneira que ambas as matrizes sejam idênticas e consequentemente a atividade de cada nó. Assim, é suficiente mostrar que existe um mapeamento f entre duas redes para provar que o método proposto é invariante a mudanças de pontos de vista.

5.2.1 Invariância a Transformações Afins

As transformações afins incluem todas as transformações que preservam colinearidade, razão e paralelismo (89). Entre as transformações afins estão a translação, a rotação, a reflexão, o cisalhamento, entre outras. Dada uma imagem I , a imagem transformada I' pode ser obtida por meio de um mapa afim A , tal que $I'(p) = I(Ap)$.

Para demonstrar que as duas redes N^I e $N^{I'}$ obtidas, respectivamente, das imagens I e I' são isomórficas, é suficiente demonstrar o mapeamento $f(v_i) = Av_i$, em que $v_i = \{x_i, y_i\}$ é a posição do pixel p_i na imagem. Por meio desse mapeamento, cada nó $v_i \in V^I$ é mapeado em um dos nós $v_j \in V^J$ e, conseqüentemente, as conexões entre eles são exatamente as mesmas em ambas as redes.

Para corroborar a invariância a transformações afins, os histogramas para imagens rotacionadas e transladas são apresentados na Figura 5.7. Uma imagem de textura foi rotacionada em quatro direções: $0^\circ, 90^\circ, 180^\circ, 270^\circ$, enquanto que a mesma imagem foi deslocada por seis fatores na direção x : 20, 40, 60, 80, 100, 120. Os resultados são ainda mais interessantes para imagens de texturas em diferentes escalas e perspectivas. A Figura 5.8 apresenta os histogramas para essas imagens de texturas.

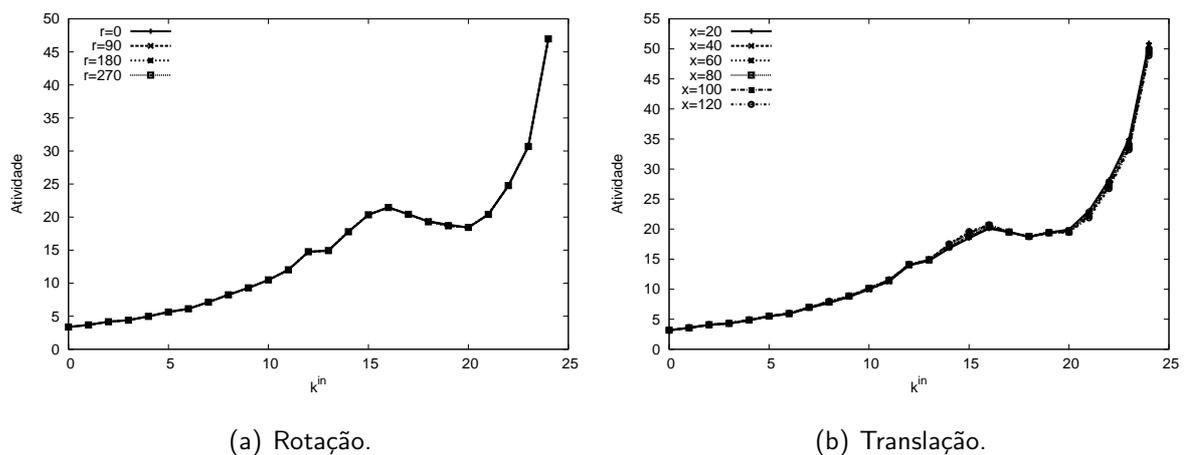


Figura 5.7 – Histogramas para imagens de texturas com duas transformações afins: (a) rotação e (b) translação. As redes foram obtidas usando-se $r = \sqrt{8}$ e $t = 80$.

Embora as imagens de texturas foram capturadas em transformações bastante desafiadoras, os histogramas são similares, confirmando a eficácia do método proposto na descrição de

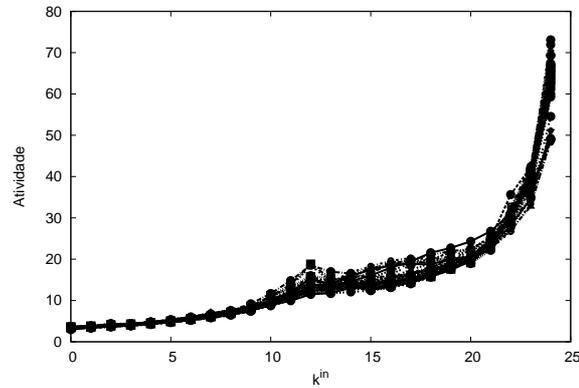
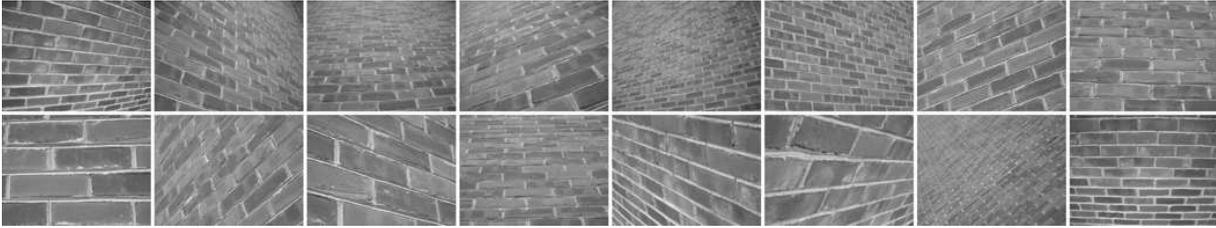


Figura 5.8 – Histogramas para imagens de texturas em diferentes perspectivas. As redes foram obtidas usando-se $r = \sqrt{8}$ e $t = 80$.

texturas com diferentes transformações.

5.2.2 Invariância à Mudança de Contraste Global

Duas imagens I e I' são perceptualmente equivalentes, se existe uma função aditiva g (mudança de contraste) tal que $I' = g(I)$. O contraste de uma imagem depende de diversos fatores, como condições de iluminação, propriedades de reflexão de objetos, etc. Como apresentado nas Equações 5.1.1 e 5.1.3, cada conexão possui uma probabilidade $p(v_i, v_j) = \frac{I(p_i) - I(p_j)}{\sum_{v_k \in V} I(p_i) - I(p_k)}$, onde p_i é o pixel correspondente ao nó v_i . Dada que a intensidade de cada pixel na imagem I' é transformada por uma função crescente, o peso da conexão entre nós nesta nova imagem é dado por:

$$\begin{aligned}
 p'(v_i, v_j) &= \frac{g(I(p_i)) - g(I(p_j))}{\sum_{v_k \in V} g(I(p_i)) - g(I(p_k))} \\
 &= \frac{g(I(p_i) - I(p_j))}{g(\sum_{v_k \in V} I(p_i) - I(p_k))} \\
 &= \frac{I(p_i) - I(p_j)}{\sum_{v_k \in V} I(p_i) - I(p_k)} = p(v_i, v_j)
 \end{aligned}$$

Portanto, a modelagem proposta por rede é independente do contraste da imagem, dado que ambas as imagens I e I' fornecem redes idênticas. Para corroborar essa invariância, a Figura 5.9 apresenta o histograma para diferentes imagens transformadas pela adição de luminância l à intensidade dos pixels.

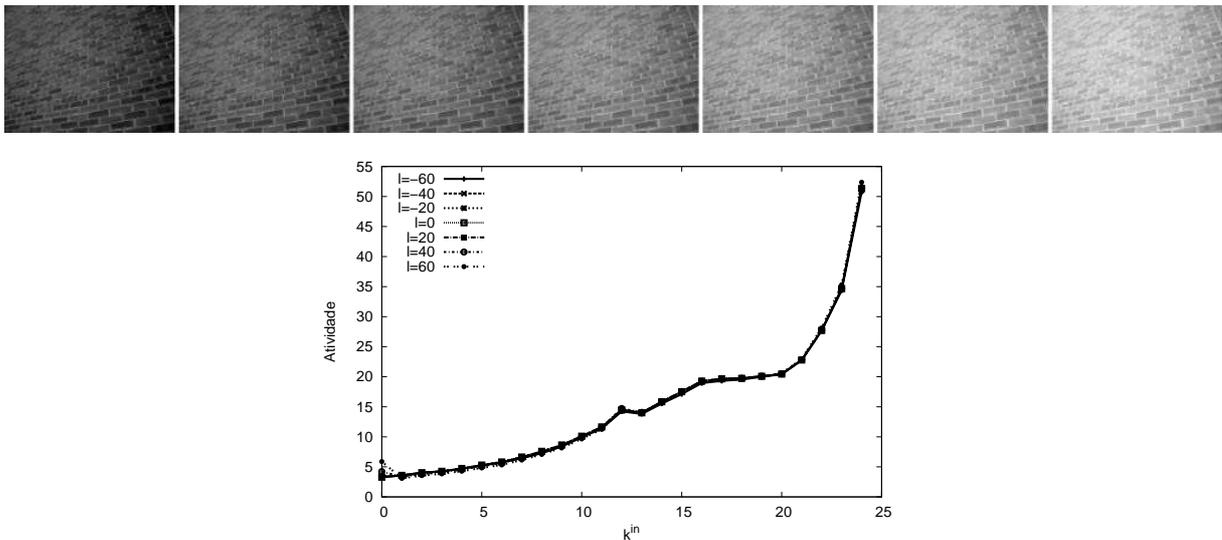


Figura 5.9 – Histograma para sete imagens de textura com diferentes luminâncias. A imagem original é transformada pela adição da determinada luminância em todos os pixels.

5.3 Experimentos

Nesta seção são descritas as três bases de dados que são amplamente utilizadas na literatura e o processo de classificação.

- Brodatz: A base de dados Brodatz (82) é amplamente utilizada pela comunidade de visão computacional como referência para avaliar métodos de reconhecimento de texturas. Esta base é composta por 111 classes, originalmente, extraídas do álbum de fotografias de Brodatz. Cada classe é composta por 10 amostras em níveis de cinza de 200×200 pixels. A Figura 4.8 apresenta quatro amostras para algumas das classes presentes na base de dados.
- Vistex: Embora o álbum de Brodatz seja muito utilizado, ele é um *benchmark* que tem recebido críticas em relação a baixa variação intra-classe e a falta de mudanças de escalas, condições de iluminação, perspectivas, etc. Para complementar os experimentos, é utilizada também a base de dados vistex (83). Esta base contém 54 classes com 16 amostras de 128×128 pixels. Quatro amostras para algumas texturas podem ser visualizadas na Figura 4.9.
- Usptex: Por fim, a última base de dados utilizada é a base usptex (81), desenvolvida na Universidade de São Paulo. Essa base fornece 191 classes, cada uma contendo 12 amostras de 128×128 pixels. Devido ao grande número de classes, esta base de dados é interessante para avaliar a generabilidade dos métodos. Exemplos de algumas amostras estão presentes na Figura 5.10.

Em todos os experimentos, o classificador k-vizinho mais próximo com $k = 1$ no esquema de validação cruzada foi utilizado para classificar as amostras. A escolha de um classificador simples foi feita a fim de evidenciar a importância das características na tarefa de classificação.

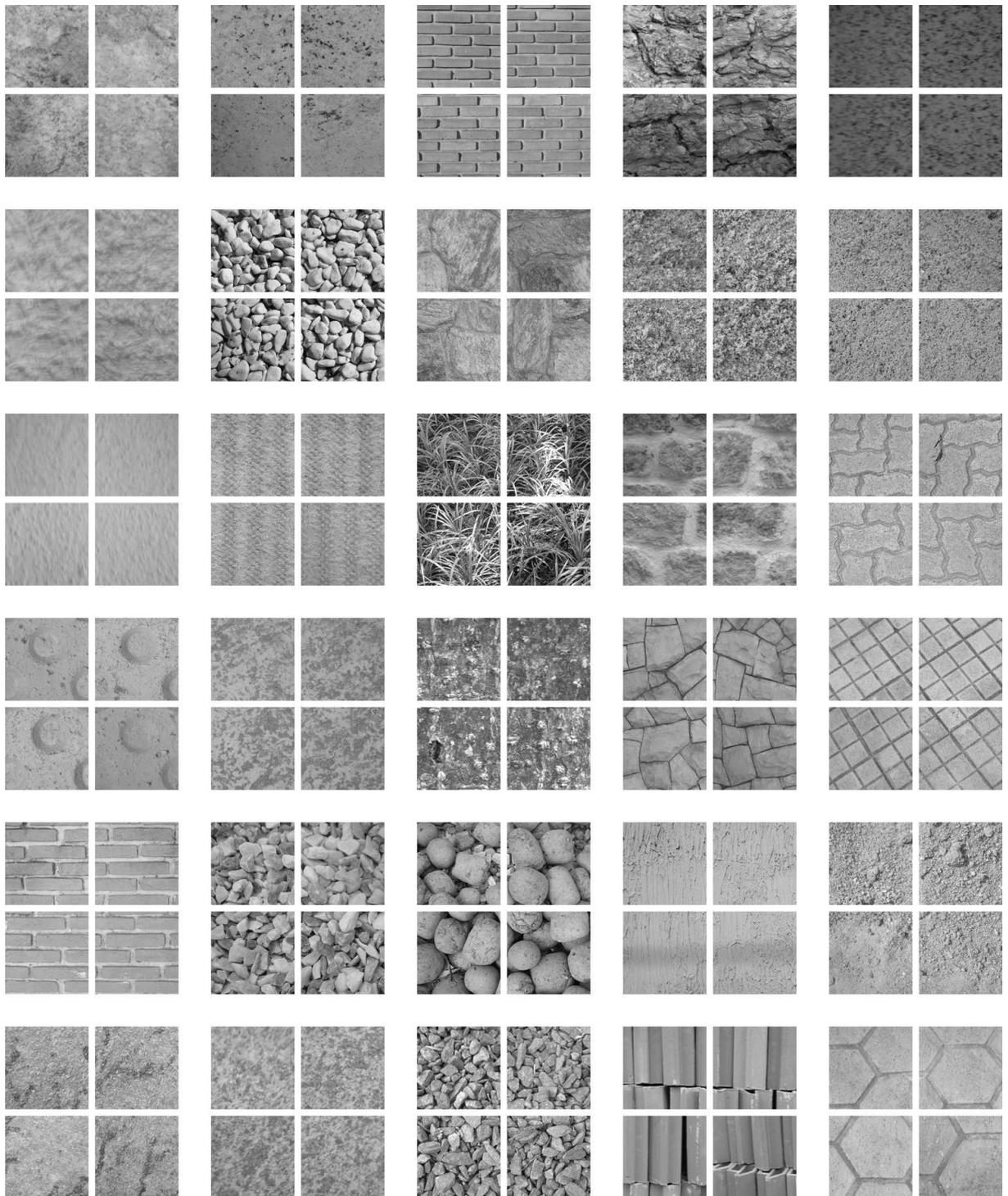


Figura 5.10 – Exemplos de amostras da base de dados usptex. Esta base é composta por 191 classes, cada uma contendo 12 amostras de 128×128 pixels.

5.4 Resultados Experimentais

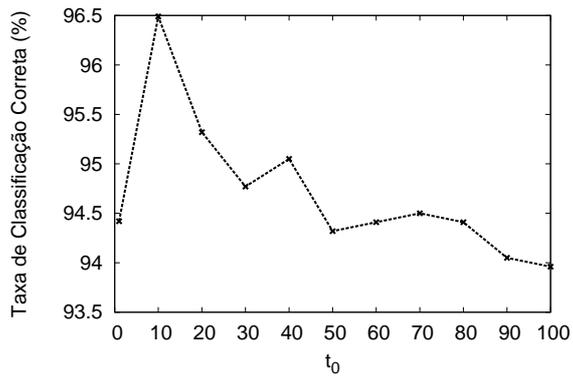
Para avaliar o desempenho do método proposto, foram analisados os seus parâmetros com o intuito de entender o seu comportamento na tarefa de classificação de texturas. Em seguida, os resultados obtidos no experimento anterior foram comparados com o estado-da-arte.

5.4.1 Análise de Parâmetros

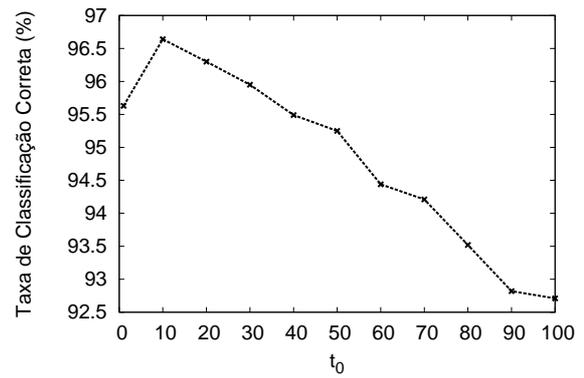
Os quatro parâmetros da modelagem da rede direcionada são: raios r , que convertem a imagem em uma rede regular, limiar inicial t_0 , limiar de incremento t_i e limiar final t_f . Nos experimentos abaixo, são apresentados os resultados em classificação usando diferentes configurações destes parâmetros.

Os limiares t_0 , t_i e t_f são utilizados na transformação $\phi(t, N)$ que converte a rede regular e são responsáveis pela análise multi-escala. O limiar inicial t_0 , cujos resultados para diferentes valores são apresentados na Figura 5.11, é responsável por limitar a escala da análise. Para $t_0 = 10$, o método proposto alcançou sua maior taxa de classificação correta nas três bases de dados. Para valores menores que 10, a rede não modela adequadamente todas as propriedades da textura e pode ser altamente perturbada por ruído. Por outro lado, para altos valores de t_0 , a análise multi-escala da rede não apresenta os detalhes da imagem e, com isso, perde-se informações importantes da textura.

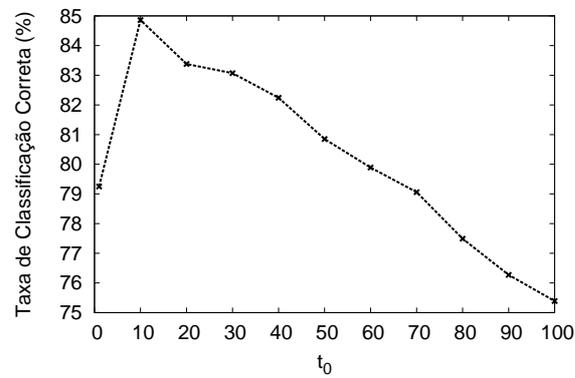
As Figuras 5.12 e 5.13 apresentam a taxa de classificação correta para os parâmetros t_i e t_f , respectivamente. O limiar final é plotado em função de x tal que $t_f = t_0 + x * t_i$. Portanto, x é o número de limiares utilizados na análise multi-escala. Como esperado, os melhores desempenhos foram obtidos para altos valores de t_i (Figura 5.12), pois eles permitem que o método explore uma maior gama de escalas. Para as bases de dados Brodatz e vistex, o melhor resultado foi obtido quando $t_i = 60$, enquanto que a base de dados usptex alcançou o



(a) Brodatz.



(b) Vistex.



(c) Usptex.

Figura 5.11 – Avaliação do limiar inicial t_0 para as três base de dados. O gráfico corresponde a taxa de classificação correta para t_0 variando de 1 a 100.

seu desempenho máximo utilizando $t_i = 30$.

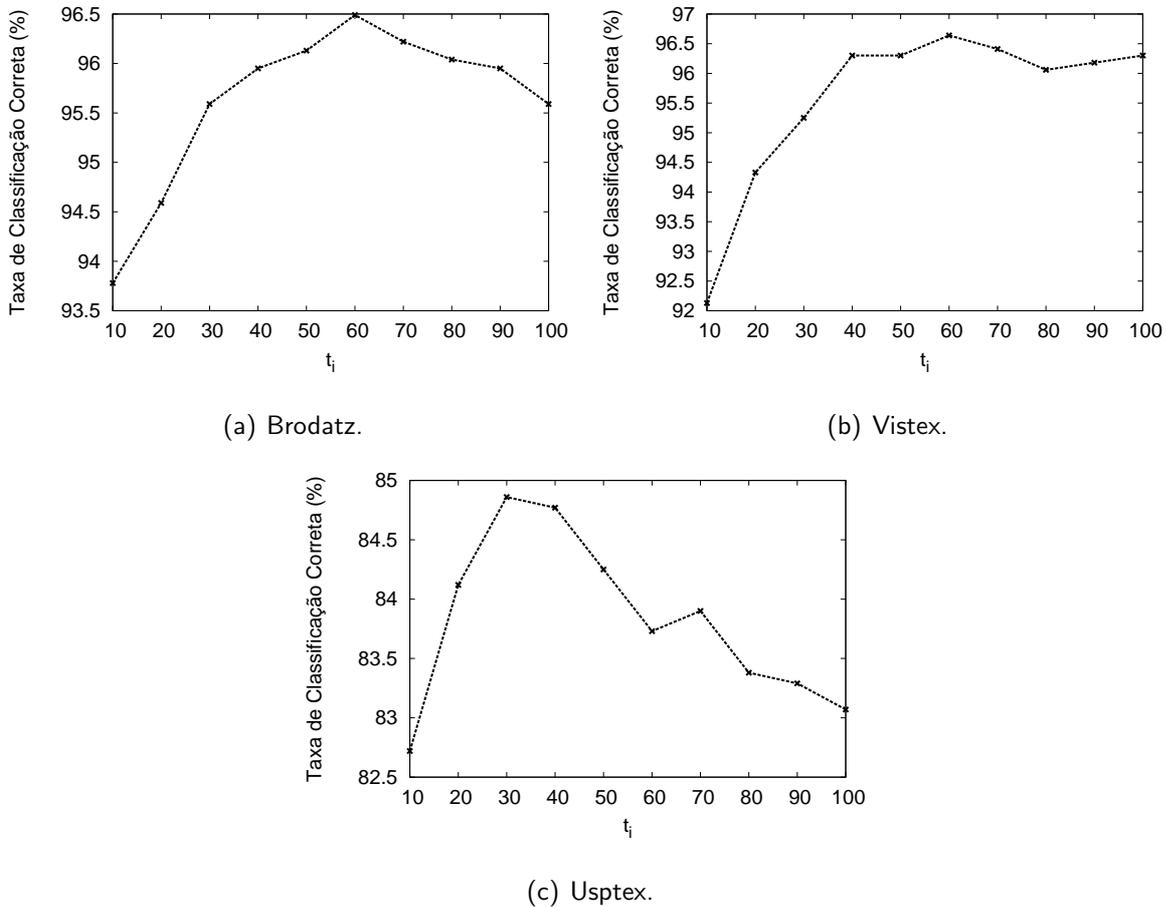


Figura 5.12 – Avaliação do limiar de incremento para as três base de dados. O gráfico corresponde a taxa de classificação correta para t_i variando de 10 à 100.

O número de limiares utilizado na análise multi-escala é avaliado no gráfico da Figura 5.13. Para a base de dados Brodatz, a taxa de classificação correta máxima foi obtida para $x = 5$. Por outro lado, $x = 4$ e $x = 6$ forneceram o melhor resultado para as bases vistex e usptex, respectivamente. Finalmente, usando esses três gráficos, foi possível determinar o conjunto de limiares adequados para cada base de dados.

A Figura 5.14 apresenta o gráfico da taxa de classificação correta para os valores de raios r . O raio r é o parâmetro que determina a zona de conexão entre nós na rede direcionada. Para as bases Brodatz e usptex, o melhor resultado foi obtido para a concatenação de raios $r = \sqrt{1}, \sqrt{2}, \sqrt{4}, \sqrt{5}$. Por outro lado, a concatenação de raios $r = \sqrt{1}, \sqrt{2}, \sqrt{4}$ forneceu a melhor taxa de classificação correta para a base vistex.

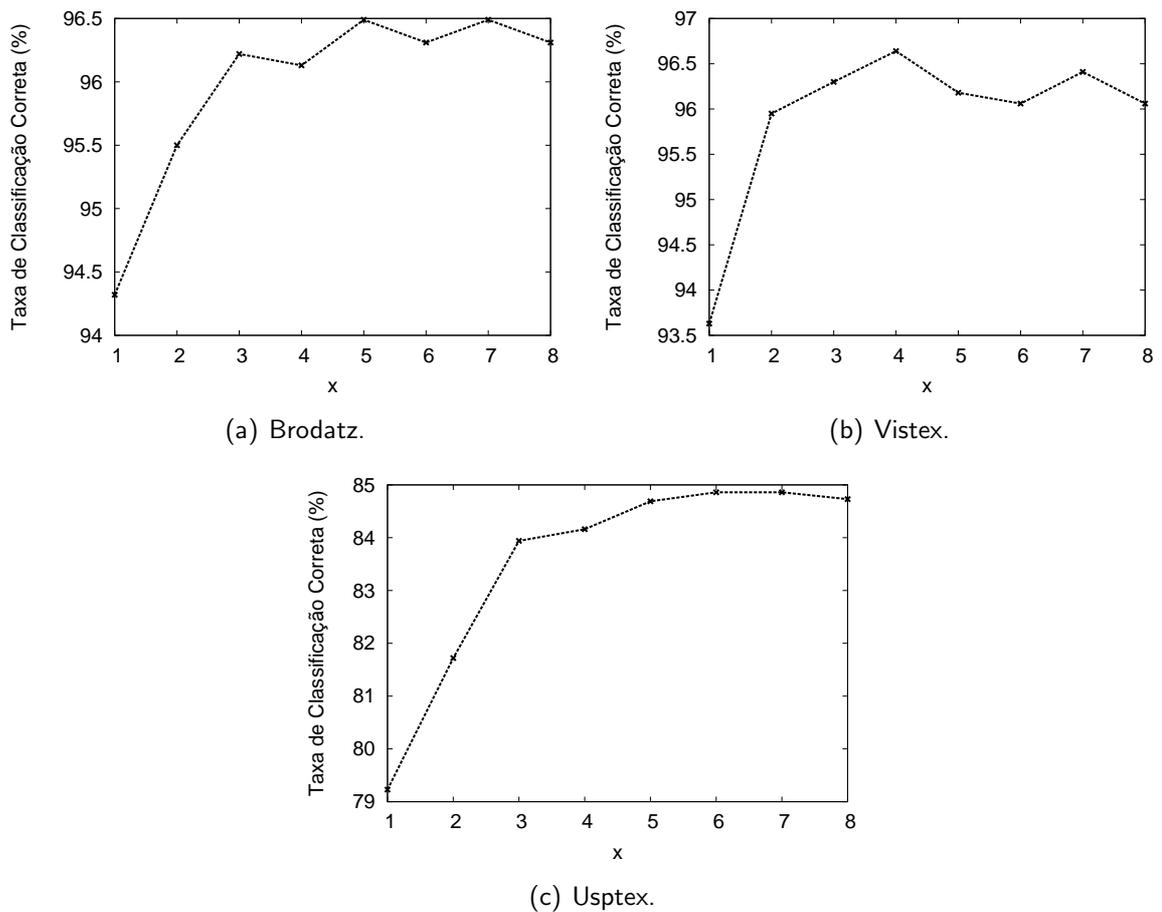


Figura 5.13 – Avaliação do limiar final para as três base de dados. O gráfico corresponde a taxa de classificação correta para x variando de 1 à 8.

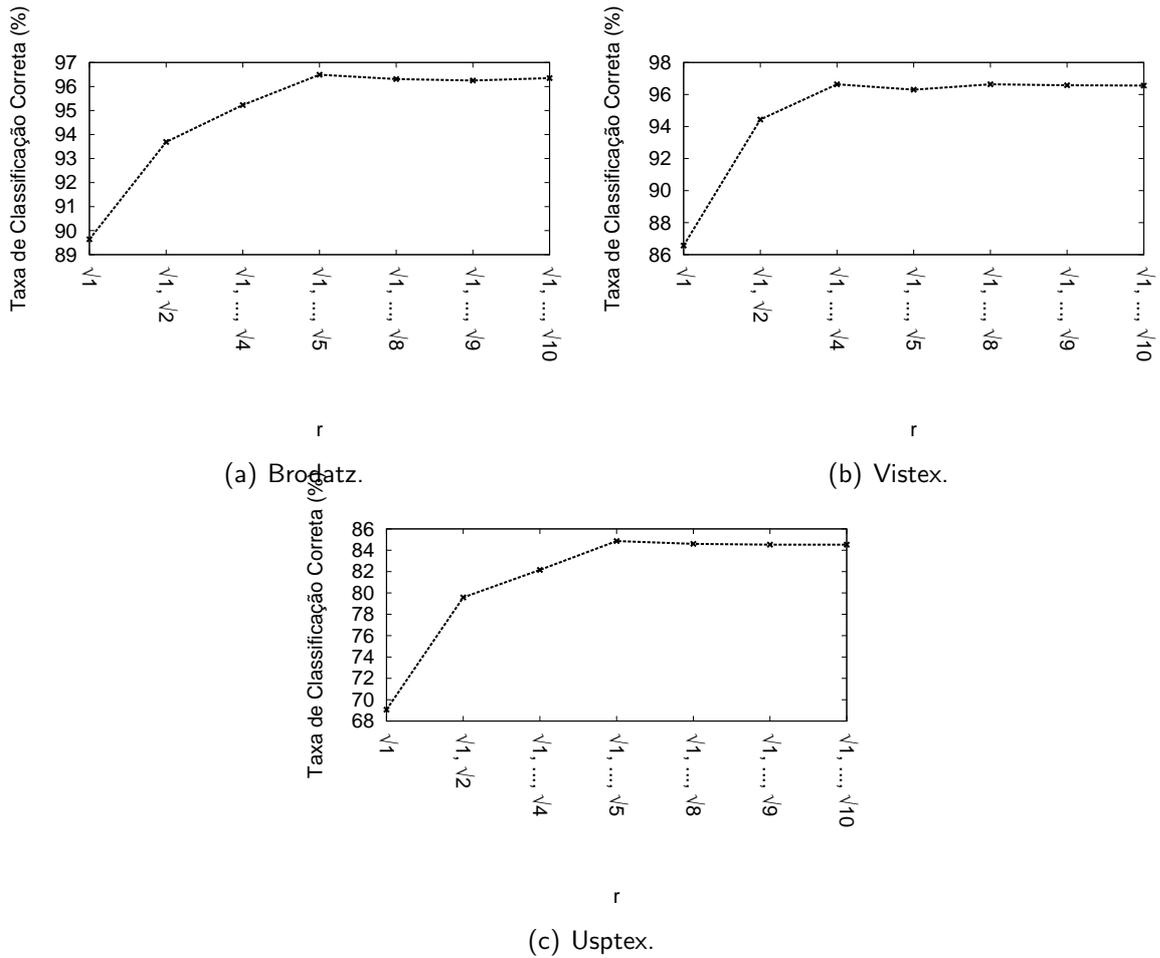


Figura 5.14 – Avaliação do conjunto de raios para as três base de dados. O gráfico corresponde a taxa de classificação correta para a concatenação de $\sqrt{1}$ a $\sqrt{1} \dots \sqrt{10}$.

5.4.2 Redes Direcionadas vs Não-direcionadas

O método proposto representa a imagem de textura como uma rede direcionada, enquanto que os outros trabalhos (43, 78, 87) utilizam redes não-direcionadas. Nesta seção, são apresentadas as vantagens na utilização de tais redes comparadas com as redes não-direcionadas. As redes não-direcionadas são obtidas tornando as conexões sem direção, ou seja, para cada conexão direcionada e_{v_i, v_j} , uma nova conexão e_{v_j, v_i} com o mesmo peso é adicionada na rede.

A primeira desvantagem é que as redes não-direcionadas possuem o dobro de conexões. Assim, o alto número de conexões afeta diretamente o custo computacional, como pode ser visto na Figura 5.15. Quando redes não-direcionadas são utilizadas, o tamanho médio das caminhadas aleatórias aumenta de 3.43 para 499.68. Portanto, o uso de redes direcionadas melhora significativamente o custo computacional.

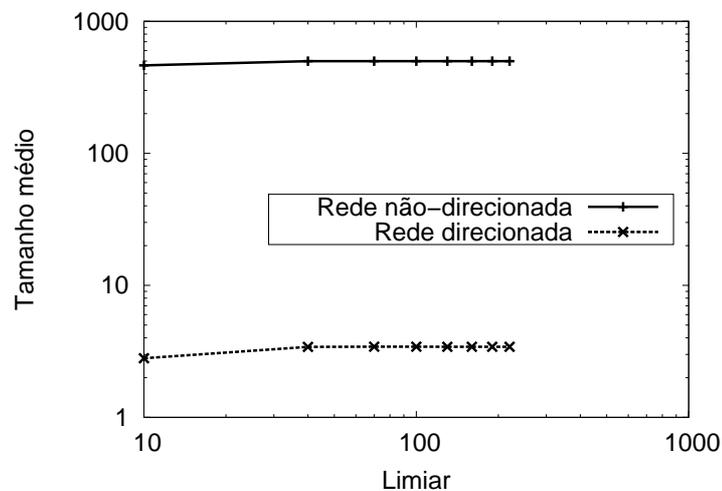


Figura 5.15 – Comparação entre o tamanho médio das caminhadas em redes direcionadas e não-direcionadas com raio $r = \sqrt{1}$.

Além disso, a taxa de classificação correta de redes não-direcionadas é inferior àquelas obtidas por redes direcionadas. Para a base de dados Brodatz, as redes direcionadas forneceram uma taxa de 96.49% (± 1.55) contra 90.36% (± 2.50) das redes não-direcionadas, um aumento de 6.13%. Esse comportamento ocorre, pois em redes não-direcionadas, o agente não explora suficientemente todos os nós. Por exemplo, dada uma conexão e_{v_i, v_j} com peso alto, o agente é geralmente aprisionado nesse laço e a atividade dos nós v_i e v_j são incorretamente estimadas. Dessa maneira, as redes não-direcionadas são altamente perturbadas por ruídos.

5.4.3 Comparação com Métodos da Literatura

Nas comparações dessa seção, os seguintes métodos da literatura foram utilizados: descritores de Fourier (85), matrizes de co-ocorrência (84), filtros de Gabor (29, 90), caminhada determinística do turista (43), espectro multifractal (39), dimensão fractal local (40) e padrões locais binários invariantes à rotação (26).

A Tabela 5.1 apresenta a comparação dos métodos na base de dados Brodatz. O método proposto obteve uma taxa de classificação correta comparável com os padrões locais binários, 96.49%(±1.55) e 96.22%(±1.48), respectivamente. Esses dois métodos foram superior aos demais, seguido pela dimensão fractal local com taxa de 94.05%(±2.07).

Tabela 5.1 – Resultados experimentais para métodos de texturas na base de dados Brodatz. A tabela apresenta o número de características e a taxa de classificação correta.

Método	N. de Caract.	Taxa de Classificação Correta
Descritores de Fourier	99	70.67(±3.39)
Matrizes de co-ocorrência	40	86.49(±2.69)
Filtros de Gabor	40	88.99(±2.03)
Caminhada determinística do turista	48	89.37(±1.96)
Espectro multifractal	78	75.05(±3.41)
Dimensão fractal local	5550	94.05(±2.07)
Padrões locais binários	54	96.22(±1.48)
Método proposto	156	96.49(±1.55)

Na Tabela 5.2 são apresentados os resultados para a base de dados vistex. A tarefa de reconhecer texturas desta base é um problema desafiador devido à mudanças nas condições de iluminação e perspectivas. Como apresentado na tabela, uma taxa de classificação correta de 96.64%(±1.75) foi obtida pelo método proposto. Isso demonstra claramente que, mesmo em imagens capturadas em diferentes condições, o método proposto caracteriza as texturas de forma eficaz. Estes resultados indicam que o método proposto melhora a taxa de classificação correta, e.g. de 84.86%(±3.05) para 96.64%(±1.75) sobre os filtros de Gabor e de 93.75%(±2.20) para 96.64%(±1.75) sobre os padrões locais binários.

Finalmente, os resultados experimentais para a base de dados usptex são apresentados na Tabela 5.3. Esta base fornece um grande número de classes, apresentando um desafio para os métodos de reconhecimento de texturas. A taxa de classificação correta de 84.86%(±2.10) foi obtida pelo método proposto, seguido pelos padrões locais binários com uma taxa de 80.80%(±2.10). Devido ao grande número de classes, estes resultados sugerem que esses dois

Tabela 5.2 – Resultados experimentais para métodos de texturas na base de dados vistex. A tabela apresenta o número de características e a taxa de classificação correta.

Método	N. de Caract.	Taxa de Classificação Correta
Descritores de Fourier	63	72.25(± 4.41)
Matrizes de co-ocorrência	40	80.51(± 3.49)
Filtros de Gabor	40	84.86(± 3.05)
Caminhada determinística do turista	30	82.48(± 3.67)
Espectro multifractal	78	65.97(± 3.92)
Dimensão fractal local	2700	80.09(± 3.55)
Padrões locais binários	54	93.75(± 2.20)
Método proposto	75	96.64(± 1.75)

métodos apresentam uma boa capacidade de generalização.

Tabela 5.3 – Resultados experimentais para métodos de texturas na base de dados usptex. A tabela apresenta o número de características e a taxa de classificação correta.

Método	N. de Caract.	Taxa de Classificação Correta
Descritores de Fourier	63	49.78(± 3.13)
Matrizes de co-ocorrência	40	63.73(± 2.71)
Filtros de Gabor	40	73.62(± 2.68)
Caminhada determinística do turista	20	55.28(± 2.40)
Espectro multifractal	78	31.72(± 3.08)
Dimensão fractal local	9550	30.28(± 1.85)
Padrões locais binários	54	80.80(± 2.10)
Método proposto	182	84.86(± 2.10)

De modo a visualizar o comportamento das classes de textura, os vetores de características foram projetados para um espaço bidimensional usando análise de componentes principais (PCA). O PCA foi aplicado nos vetores de 35 classes da base de dados Brodatz e o gráfico dessa projeção pode ser visualizado na Figura 5.16. Como pode ser visto, as classes estão organizadas em grupos distintos, ilustrando o poder de separação do método proposto.

5.5 Conclusão

Neste capítulo foi proposto um método para reconhecimento de texturas estáticas baseado em atividade em redes direcionadas. Esse método mapeia uma imagem em uma rede

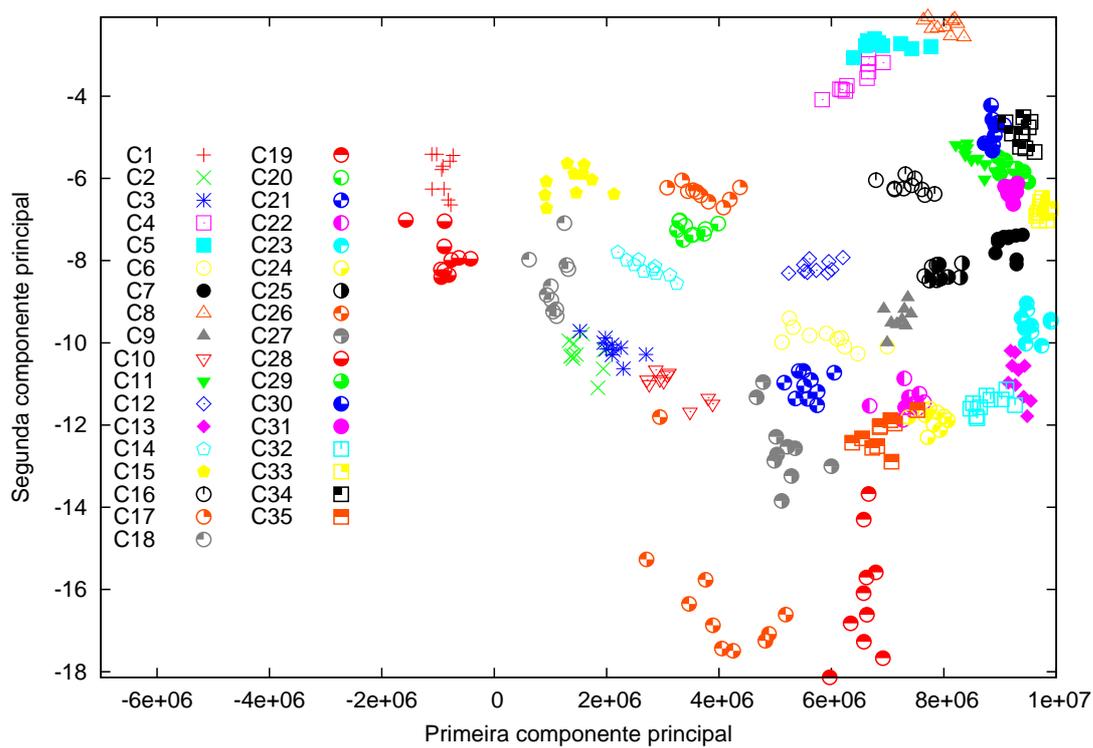


Figura 5.16 – Gráfico das duas primeiras componentes da análise de componentes principais na base de dados Brodatz. A variância acumulada das duas componentes é 0.59.

direcionada considerando cada nó como um pixel e conectando nós dentro de um determinado raio. A direcionalidade é obtida levando-se em conta somente conexões de pixels com maiores para menores intensidades. Então, a atividade de cada nó é estimada por meio de caminhadas aleatórias e utilizada para obter um vetor de características que descreve a textura a ser identificada. Parte do bom desempenho do método decorre do fato de que a difusão direcionada descreve a diversidade das dinâmicas na respectiva rede. Os resultados experimentais confirmaram a eficácia do método proposto comparado com vários métodos de identificação de textura em três bases de dados amplamente utilizadas como referência.

Caracterização de Texturas Dinâmicas

A tarefa de reconhecer as texturas dinâmicas de um vídeo é um problema desafiador devido à necessidade em descrever as características de aparência e movimento. Além disso, as texturas estão rotacionadas em muitas situações, sugerindo que características invariantes à rotação sejam utilizadas. Neste capítulo, é apresentado um método para caracterização de texturas dinâmicas baseado em CDPARs. Esse método foi publicado em (86).

6.1 Método proposto

O método proposto para caracterização de texturas dinâmicas em vídeos pode ser descrito em quatro passos (Figura 6.1).

- **Passo 1:** o vídeo é dividido em três planos ortogonais: XY, XT e YT. O plano XY é responsável pela descrição da aparência e os planos XT e YT descrevem as características de movimento.
- **Passo 2:** CDPARs são executadas em cada plano com diferentes tamanhos de memória (μ_1, \dots, μ_M) para a obtenção das distribuições conjuntas.
- **Passo 3:** para cada distribuição conjunta, um vetor de características é obtido e, então, eles são concatenados separados por plano ortogonal.
- **Passo 4:** os vetores são concatenados, obtendo o vetor final composto por características dos três planos ortogonais que caracteriza características de aparência e movimento.

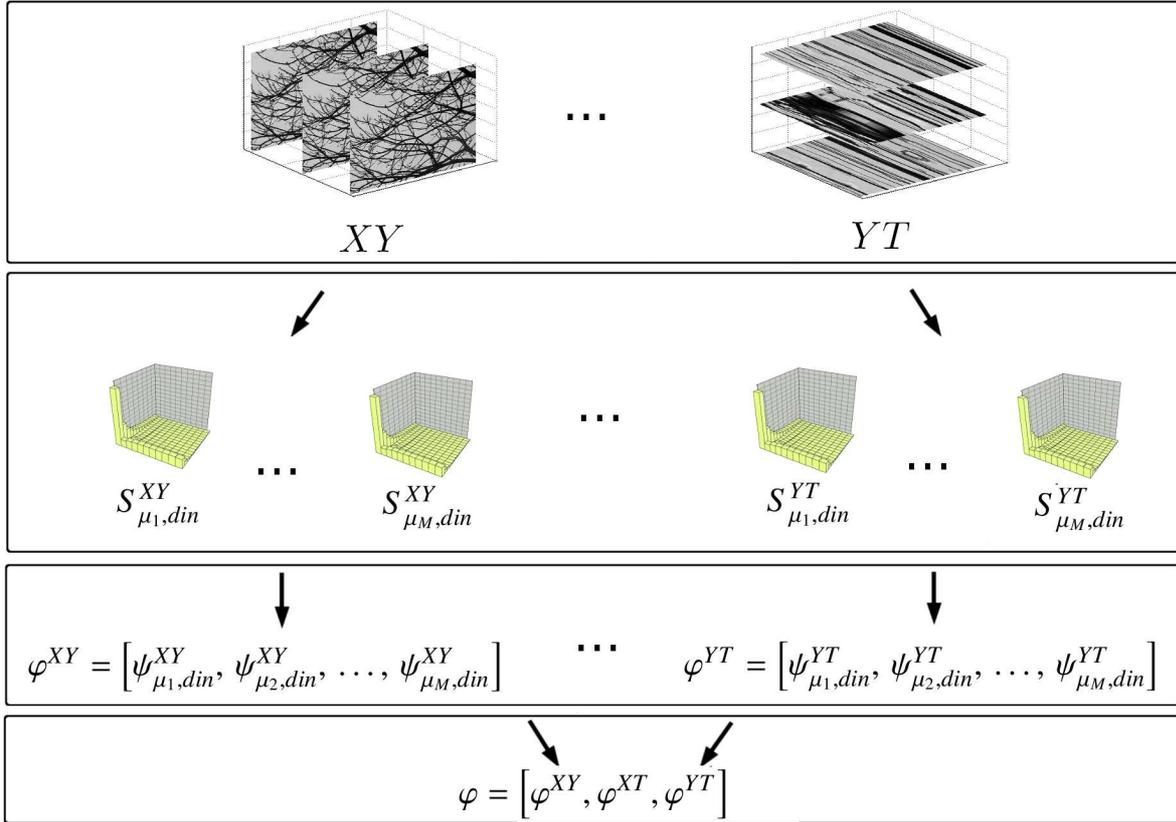


Figura 6.1 – Sumarização do método proposto. Passo 1: o vídeo contendo as texturas dinâmicas é dividido em três planos ortogonais. Passo 2: para cada plano, são executadas CDPARs com diferentes tamanhos de memória. Passo 3: um vetor de características para cada distribuição conjunta é calculado. Passo 4: os vetores são concatenados para compor o vetor de características final do método proposto.

Nas próximas seções, os passos do método são descritos em detalhes.

6.1.1 CDPARs em Vídeo

Considere um vídeo composto por texturas dinâmicas que consiste em um par (\mathcal{V}, V) – um conjunto finito \mathcal{V} de pixels e um mapeamento V que atribui para cada pixel $i = (x_i, y_i, t_i) \in \mathcal{V}$ uma intensidade $V(i) \in [0, 255]$. No vídeo, $x_i \in [0, W]$ e $y_i \in [0, H]$ são os índices espaciais e $t_i \in [0, T]$ é o índice temporal do vídeo. Considere também que cada pixel i possui uma vizinhança $\eta(i)$ composta por pixels vizinhos.

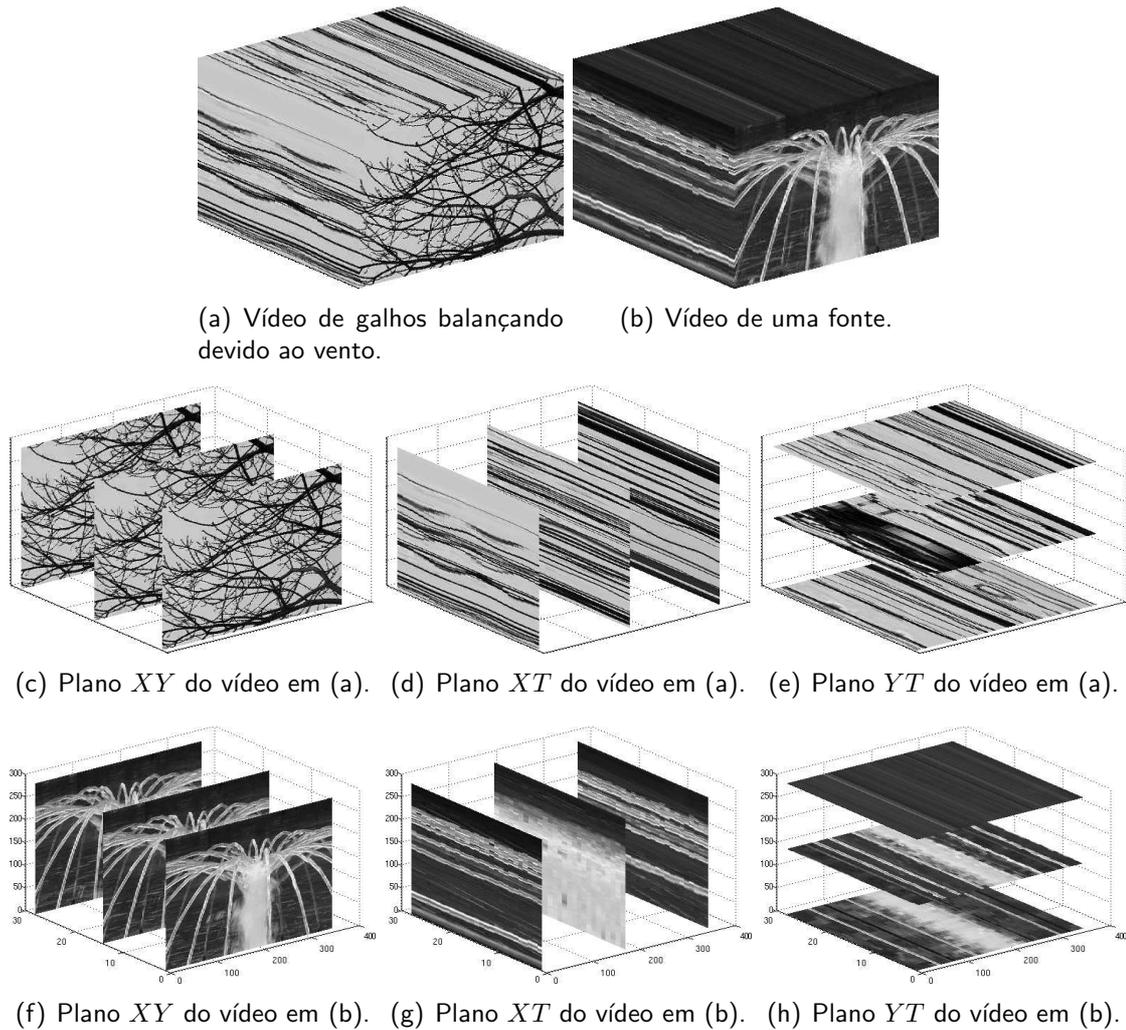


Figura 6.2 – Exemplo dos três planos ortogonais obtidos dos vídeos: (a) galhos balançando devido ao vento e (b) fonte de água.

Para descrever características de movimento e aparência de um vídeo, é proposta uma estratégia para executar CDPARs em três planos ortogonais do vídeo. Os três planos ortogonais, denominados XY , XT e YT , podem ser visualizados na Figura 6.2. Esses planos definem o conjunto de vizinhança $\eta(i)$ e, conseqüentemente, eles alteram e direcionam os passos do agente. Cada plano acentua diferentes propriedades das texturas dinâmicas presentes no vídeo.

O plano XY visa realçar as características de aparência presentes nas texturas dinâmicas do vídeo (Figuras 6.2 (c) e (f)). Neste plano, o agente caminha apenas em pixels pertencentes ao mesmo quadro. Conseqüentemente, o plano XY descreve a variância espacial das texturas dinâmicas observando cada quadro independentemente. Neste plano, pixels j pertencem ao conjunto de vizinhança $\eta(i)^{XY}$ se a distância Euclidiana entre os índices espaciais for menor ou igual a $\sqrt{2}$ (8-vizinhos) e o índice temporal t_i for igual a t_j , como definido na Equação (6.1.1).

$$j \in \eta(i)^{XY} \text{ se } \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \leq \sqrt{2} \text{ e } t_i = t_j \quad (6.1.1)$$

Por outro lado, os planos XT e YT evidenciam os padrões de movimento das texturas dinâmicas presentes no vídeo (Figuras 6.2 (d) e (g) para plano XT e Figuras 6.2 (e) e (h) para plano YT). Em ambos os planos, o agente caminha entre quadros e, portanto, ele caracteriza a variação temporal das texturas dinâmicas. No plano XT , o agente caminha em pixels pertencentes ao mesmo plano Y de acordo com a Equação (6.1.2). Para o plano YT , o agente caminha apenas em pixels que pertencem ao mesmo plano X (Equação 6.1.3).

$$j \in \eta(i)^{XT} \text{ se } \sqrt{(x_i - x_j)^2 + (t_i - t_j)^2} \leq \sqrt{2} \text{ e } y_i = y_j \quad (6.1.2)$$

$$j \in \eta(i)^{YT} \text{ se } \sqrt{(y_i - y_j)^2 + (t_i - t_j)^2} \leq \sqrt{2} \text{ e } x_i = x_j \quad (6.1.3)$$

Para combinar características de aparência e movimento, CDPARs são executadas independentemente em cada plano descrito acima. Para isso, considere que o agente está no pixel w_t e caminhando no plano XY . O próximo passo do agente é mover-se para um dos pixels vizinhos em $\eta^{XY}(w_t)$ que minimize o peso ω e que não esteja na memória M_t^μ . O peso de conexão entre dois pixels é dado por $\omega(i, j) = |V(i) - V(j)|$, onde $V(i)$ é a intensidade do pixel i e $|\cdot|$ é o valor absoluto. Os passos iterativos do agente caminhando no plano XY são descritos na Equação 6.1.4 para $r = \min$ (descrito analogamente na Equação 6.1.5 para $r = \max$).

$$w_{t+1} = \arg \min_{j \in \eta^{XY}(w_t)} \{\omega(w_t, j) \mid j \notin M_t^\mu\} \quad (6.1.4)$$

$$w_{t+1} = \arg \max_{j \in \eta^{XY}(w_t)} \{\omega(w_t, j) \mid j \notin M_t^\mu\} \quad (6.1.5)$$

Por meio da regra determinística acima, o agente é aprisionado em uma atrator ρ_i^{XY} após um tempo de transiente τ_i^{XY} . Considerando cada pixel i como uma situação inicial de uma caminhada, a distribuição conjunta $S_{\mu, \text{din}}^{XY}(\tau, \rho)$ é obtida como descrito na Equação 6.1.6, onde $N = W \times H \times T$ é o número de pixels do vídeo, μ é o tamanho da memória e r é a regra de movimentação do agente.

$$S_{\mu,r}^{XY}(\tau, \rho) = \frac{1}{N} \sum_{i=1}^N \begin{cases} 1, & \text{se } \tau_i^{XY} = \tau, \rho_i^{XY} = \rho \\ 0, & \text{caso contrário} \end{cases} \quad (6.1.6)$$

Como descrito anteriormente, CDPARs são executadas em cada plano separadamente. Portanto, três distribuições conjuntas são obtidas: $S_{\mu,r}^{XY}(\tau, \rho)$, $S_{\mu,r}^{XT}(\tau, \rho)$ e $S_{\mu,r}^{YT}(\tau, \rho)$. As distribuições conjuntas $S_{\mu,r}^{XT}(\tau, \rho)$ e $S_{\mu,r}^{YT}(\tau, \rho)$ são similarmente obtidas pela Equação 6.1.6. Por meio dessas três distribuições conjuntas, o método proposto é capaz de apropriadamente descrever as texturas dinâmicas presentes em um vídeo.

6.1.2 Vetor de Características

Devido ao grande número de informações em cada distribuição conjunta, um histograma $h_{\mu,r}^{\phi}$ é calculado para agregar as informações (Equação 6.1.7, em que ϕ corresponde a um dos três planos). O histograma contém a frequência de caminhadas que possuem tamanho igual a $(\tau + \rho)$ na distribuição conjunta, onde τ e ρ são, respectivamente, o tempo de transiente e o atrator.

$$h_{\mu,r}^{\phi}(l) = \sum_{k=0}^{l-1} S_{\mu,din}^{\phi}(k, l-k) \quad (6.1.7)$$

O histograma concentra as informações mais importantes da textura em seus primeiros elementos (43). Para compor o vetor de características $\psi_{\mu,r}^{\phi}$, os primeiros n elementos do histograma são utilizados. Dado que o agente executa uma caminhada auto-repulsiva limitada a uma janela μ , o primeiro elemento do histograma a ser considerado é o elemento na posição $\mu + 1$, pois não existem atratores com menor período. Assim, o vetor de características utilizando o histograma da caminhada é dado na Equação 6.1.8.

$$\psi_{\mu,r}^{\phi} = [h_{\mu,r}^{\phi}(\mu + 1), \dots, h_{\mu,r}^{\phi}(\mu + n)] \quad (6.1.8)$$

Para obter as distribuições conjuntas, os parâmetros correspondentes ao tamanho de memória μ e regra de movimentação r do agente precisam ser determinados. Trabalhos

anteriores (43–45) têm discutido a influência de diferentes tamanhos de memória e regras de movimentação dos agentes. Assim, para descrever diferentes características das texturas dinâmicas, um vetor de características φ^ϕ , considerando diferentes valores de μ , é apresentado na Equação 6.1.9. Esse vetor de características é composto pela concatenação dos vetores $\psi_{\mu_i,r}^\phi$.

$$\varphi^\phi = [\psi_{\mu_1,r}^\phi, \psi_{\mu_2,r}^\phi, \dots, \psi_{\mu_M,r}^\phi] \quad (6.1.9)$$

onde r é a regra de movimentação que pode ser *max* ou *min*.

O uso de múltiplos tamanhos de memória aprimora a caracterização de padrões de texturas e possui um grande potencial como método de caracterização (43). Para descrever características de aparência e movimento, a combinação de diferentes planos é apresentada abaixo. Assim, o vetor final φ , descrito na Equação 6.1.10, é composto por características extraídas dos três planos ortogonais: XY , XT e YT .

$$\varphi = [\varphi^{XY}, \varphi^{XT}, \varphi^{YT}] \quad (6.1.10)$$

6.2 Análise do Método Proposto

A caracterização de texturas dinâmicas é um problema desafiador devido, principalmente, a três questões chaves: (i) combinação de características de aparência e movimento – (ii) extração de características invariantes à rotação – (iii) caracterização de múltiplas texturas dinâmicas presentes no mesmo quadro. Para avaliar a robustez e eficácia do método proposto nessas questões, três experimentos foram conduzidos. O primeiro experimento foi realizado com o intuito de ilustrar o comportamento do método na caracterização de atributos de aparência e movimento. No segundo experimento, é discutida a robustez do método com relação a texturas dinâmicas rotacionadas. Por fim, a caracterização de múltiplas texturas dinâmicas é apresentada no terceiro e último experimento realizado na análise do método. Todos os experimentos dessa seção utilizaram texturas dinâmicas da base de dados Dyntex (91).

6.2.1 Influência dos Planos Ortogonais

Nesta seção, é avaliada a influência dos planos ortogonais na caracterização de características de aparência e movimento. Neste experimento, foram utilizados 24 vídeos com diferentes aparências e padrões de movimento. Em cada vídeo, duas texturas dinâmicas compõem cada quadro, sendo uma textura dinâmica utilizada como fundo e a outra utilizada para simular o padrão de movimento. Entre os 24 vídeos, existem três aparências (escada rolante, grama e flores) e quatro padrões de movimento (circular, linear, aleatório e sem movimento). Exemplos desses vídeos podem ser vistos na Figura 6.3, onde o padrão de movimento é indicado em verde.

Primeiramente, o plano XY é analisado em um conjunto de texturas dinâmicas sintéticas subdivididas em 3 classes: flores, escada rolante e grama. As três classes podem ser visualizadas na Figura 6.3. Nesse experimento, cada classe possui oito amostras. O plano XY tem como objetivo discriminar as texturas dinâmicas com base em características de aparência. Para fins de visualização, os vetores de características obtidos de cada amostra são projetados em um espaço bidimensional. Essa projeção é obtida através da aplicação da análise de componentes principais - PCA (92). É importante mencionar que o PCA não utiliza qualquer informação das classes para projetar os vetores. Portanto, as amostras são divididas em classes apenas para fins de visualização. O resultado da projeção, utilizando apenas características extraídas do plano XY , é ilustrado na Figura 6.5 (a). Por meio dessa figura, é possível notar que as amostras com a mesma aparência estão agrupadas no espaço projetado. Dessa maneira, sugere-se que o plano XY separa as três classes cuja principal característica depende da aparência. Como esperado, esse plano não pode caracterizar as amostras com mesmo padrão de movimento, dado que a variação entre cada classe contém diferentes padrões de movimento.

As projeções utilizando as características extraídas dos planos XT e YT são apresentadas nas Figuras 6.5 (b) e 6.5 (c), respectivamente. Nessas projeções, as classes são divididas de acordo com o padrão de movimento das texturas dinâmicas, como mostrado na Figura 6.4, em: movimento circular, movimento linear, movimento aleatório e sem movimento. Cada classe possui seis amostras. Em ambos os planos, as amostras com diferentes dinâmicas são agrupadas em diferentes regiões do espaço projetado. Esta projeção sugere que as características extraídas de amostras com o mesmo padrão de movimento resultam em agrupamentos simi-

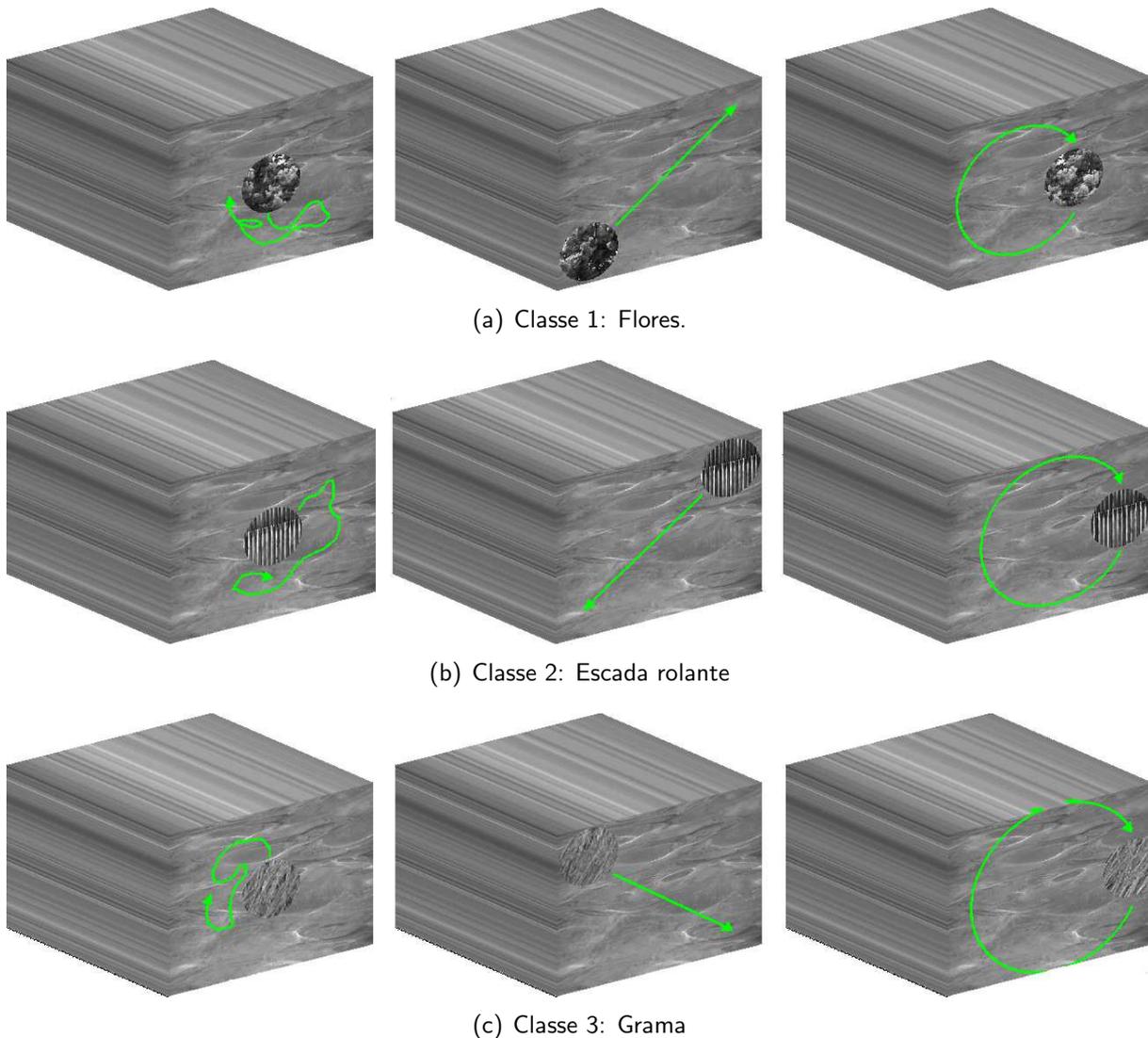


Figura 6.3 – As três classes utilizadas para avaliar características extraídas do plano XY : (a) flores, (b) escada rolante, (c) grama. Cada classe contém 8 amostras com diferentes padrões de movimento. Nesses vídeos sintéticos, uma textura dinâmica é utilizada como fundo e a outra é utilizada para simular o padrão de movimento.

lares. Os resultados são ainda mais interessantes para texturas dinâmicas sem movimento cujos os vetores são projetados quase que no mesmo ponto. As amostras dos demais tipos de movimentos também estão, de certa forma, agrupadas adequadamente.

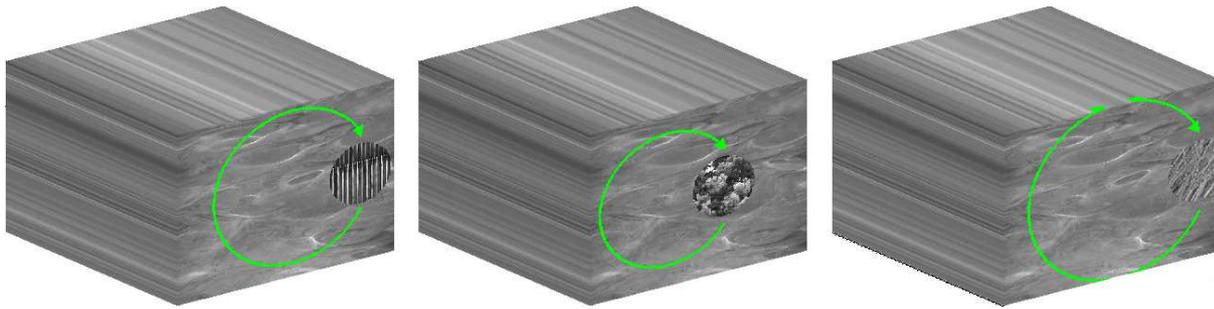
Por fim, para analisar o poder de classificação da combinação dos três planos, os 24 vídeos são divididos em 12 classes, em que cada classe possui duas amostras. As classes são combinações entre as três aparências da Figura 6.3 e os quatro padrões de movimento da Figura 6.4. Exemplos das classes utilizadas nessa análise são: flores em movimento aleatório, flores em movimento linear, escada rolante em movimento linear, escada rolante sem movimento, entre outras. Este cenário apresenta um interessante desafio para a proposta de caracterização

de texturas dinâmicas, uma vez que as classes são discriminadas somente se características de aparência e movimento forem caracterizadas. A Figura 6.5 (d) apresenta as amostras projetadas para este cenário utilizando PCA.

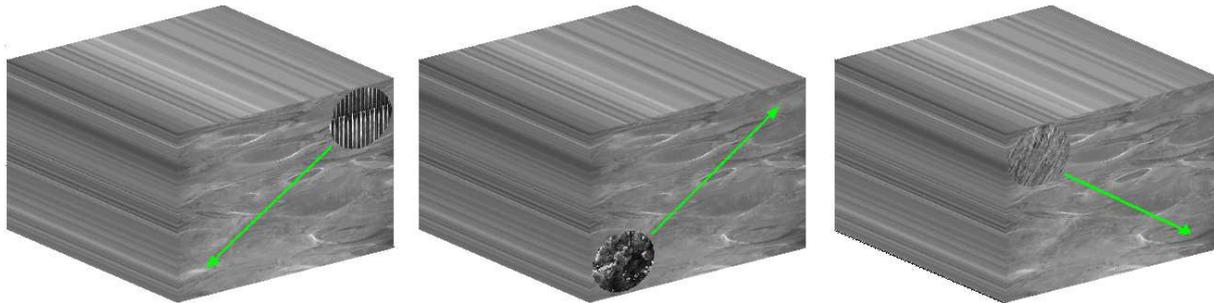
6.2.2 Análise de Invariância à Rotação

Uma importante propriedade de métodos de classificação de imagens ou vídeos é a sua capacidade em lidar com transformações na imagem, como a rotação. Como a base do método proposto são as CDPARs, ele já possui a capacidade de discriminar imagens de texturas com base na aparência, como descrito em (43). Entretanto, esta propriedade para padrões de movimento precisa ser avaliada. Para isso, o movimento linear é rotacionado em quatro direções como mostrado na Figura 6.6. Cada movimento linear rotacionado contém três amostras de diferentes aparências, as quais não podem influenciar na separação dado que características do plano XT e YT são utilizadas na projeção.

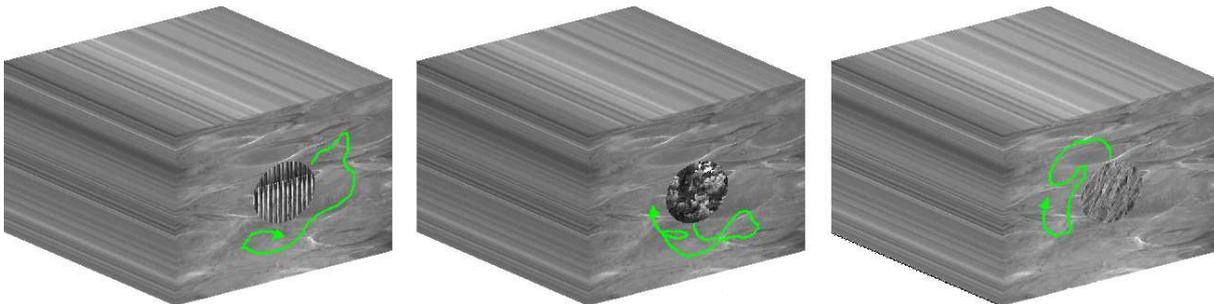
Seguindo a metodologia dos experimentos anteriores, os vetores são extraídos dos planos XT e YT e, então, eles são projetados pela análise de componentes principais. Os vetores projetados de cada amostra são apresentados na Figura 6.7. Para fins de comparação, são também apresentados os vetores extraídos de amostras com movimento aleatório e sem movimento. A Figura 6.7 indica que amostras de padrões de movimento rotacionados estão agrupadas e separadas de outros tipos de movimento. O método é invariante à rotação, pois o agente procura pelo próximo passo em todas as direções.



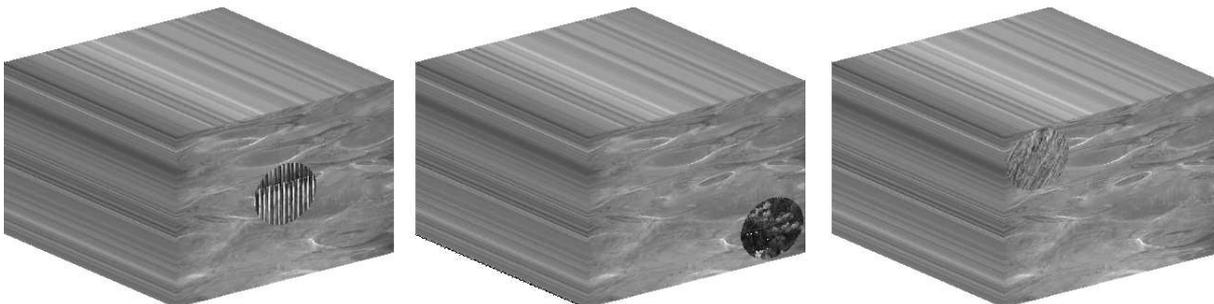
(a) Classe 1: Movimento circular.



(b) Classe 2: Movimento linear.

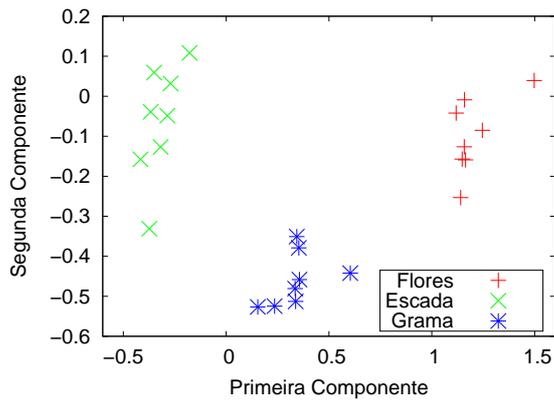


(c) Classe 3: Movimento aleatório.

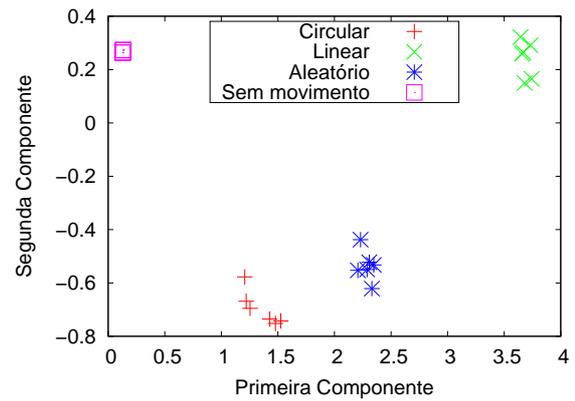


(d) Classe 4: Sem movimento.

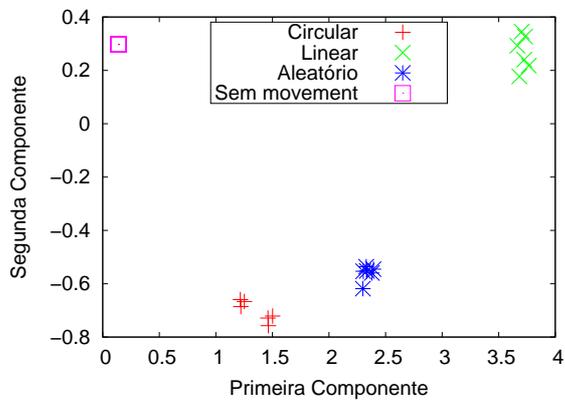
Figura 6.4 – As quatro classes utilizadas para avaliar características extraídas dos planos XT e YT : (a) movimento circular, (b) movimento linear, (c) movimento aleatório, (d) sem movimento. Cada classe contém seis amostras com diferentes aparências. Nos vídeos sintéticos, uma textura dinâmica é utilizada como fundo e a outra é utilizada para simular o padrão de movimento.



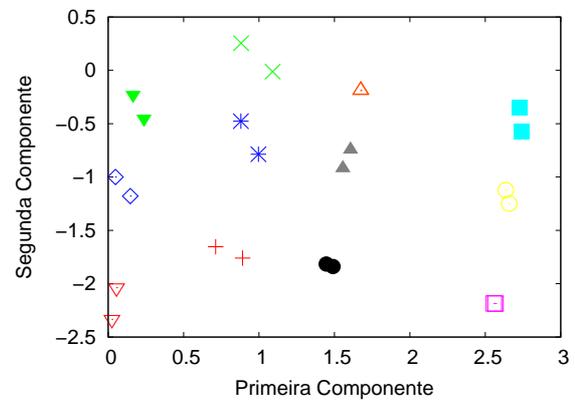
(a) Projeção de características extraídas do plano XY para avaliar a caracterização de aparência das texturas dinâmicas.



(b) Projeção de características extraídas do plano XT para avaliar a caracterização de padrões de movimento.



(c) Projeção de características extraídas do plano YT para avaliar a caracterização de padrões de movimento.



(d) Projeção de características obtidas dos três planos XY, XT, YT para avaliar ambas as características de aparência e padrões de movimento.

Figura 6.5 – Vetores projetados pela análise de componentes principais utilizando diferentes planos e configurações de classes: (a) XY, (b) XT, (c) YT, (d) XY, XT, YT. As amostras são divididas em classes apenas para fins de visualização, dado que a análise de componentes principais não utiliza qualquer informação das classes.

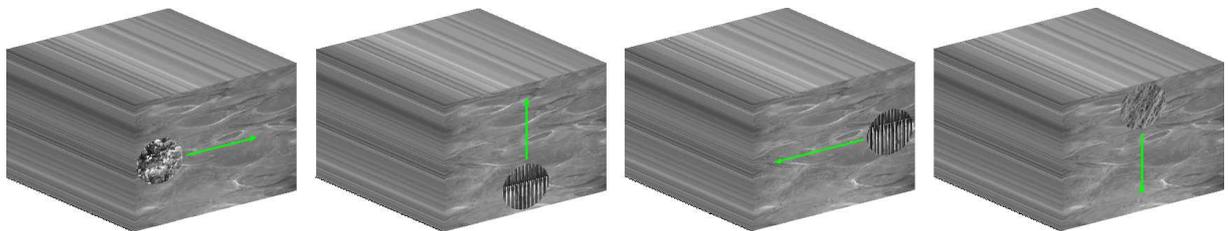


Figura 6.6 – Os quatro padrões de movimento rotacionados utilizados para avaliar a invariância à rotação do método proposto. O movimento linear é realizado em quatro direções: 0° , 90° , 180° , 270° .

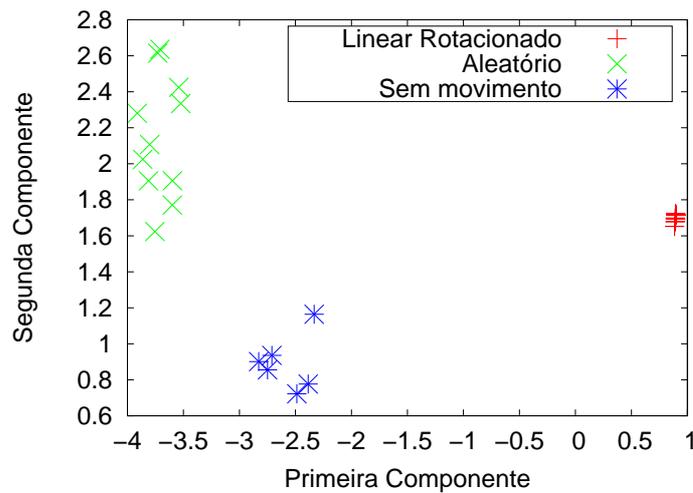
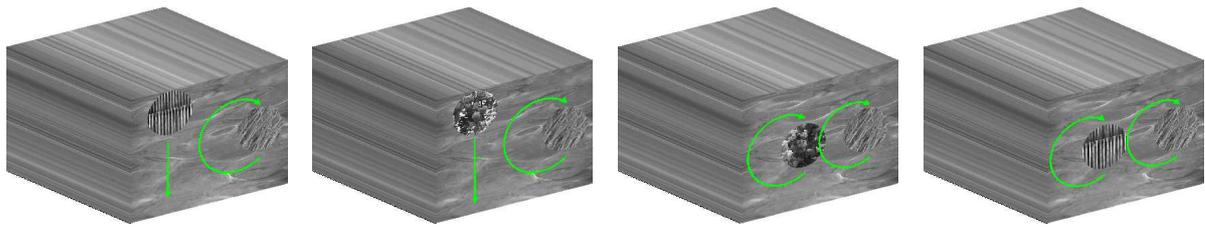


Figura 6.7 – Características extraídas dos planos XT e YT e, então, projetadas pela análise de componentes principais utilizando classes de texturas dinâmicas rotacionadas.

6.2.3 Caracterização de Múltiplas Texturas Dinâmicas

Considere um vídeo que contém uma fonte de água e no fundo uma bandeira balançando devido ao vento. Ambas as texturas dinâmicas (fonte e bandeira balançando) estão presentes no mesmo quadro do vídeo e essas texturas dinâmicas pertencem a processos visuais completamente diferentes. Esses tipos de vídeos são considerados múltiplas texturas dinâmicas, o qual é diferente, por exemplo, de um vídeo que contém apenas uma fonte ou n fontes. Alguns métodos tradicionais (e.g. aqueles baseados em um sistema dinâmico linear) não caracterizam múltiplas texturas dinâmicas adequadamente. Frequentemente, os métodos contornam esse problema aplicando-o localmente no vídeo. Esse procedimento consiste em dividir o vídeo em blocos não sobrepostos e, então, aplica-se o método em cada bloco separadamente. Os métodos supõem que cada bloco contém somente um processo visual e, assim, eles são capazes de caracterizar cada bloco.

Nesta seção é demonstrada a capacidade do método proposto em lidar com múltiplas texturas dinâmicas sem qualquer procedimento adicional. Para avaliar essa capacidade, experimentos foram realizados utilizando quatro classes, cada uma contendo três amostras (Figura 6.8). Como pode ser visto, as classes são discriminadas somente se características de ambas as texturas dinâmicas forem representadas no vetor de características. A Figura 6.9 apresenta



(a) Classe 1: Escada rolante em movimento linear e grama em movimento circular. (b) Classe 2: Flores em movimento linear e grama em movimento circular. (c) Classe 3: Flores em movimento circular e grama em movimento circular. (d) Classe 4: Escada rolante em movimento circular e grama em movimento circular.

Figura 6.8 – Quatro classes utilizadas para avaliar a capacidade do método em caracterizar múltiplas texturas dinâmicas. Cada classe contém três amostras.

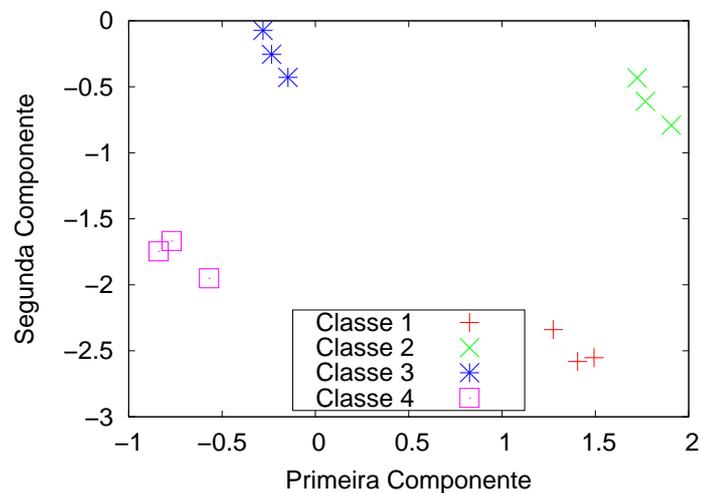


Figura 6.9 – Características extraídas dos planos XY , XT e YT e, então, projetadas pela análise de componentes principais, utilizando vídeos compostos de múltiplas texturas dinâmicas.

a projeção das amostras deste cenário. A partir da figura, é possível notar que o método proposto é capaz de lidar com múltiplas texturas dinâmicas. Essa caracterização se deve à inicialização de caminhadas em cada pixel do vídeo. Dessa maneira, algumas caminhadas caracterizam uma textura dinâmica enquanto outras caminhadas caracterizam a outra textura dinâmica. Assim, características de ambas as texturas dinâmicas estão presentes na distribuição conjunta e, conseqüentemente, no vetor de características. Na projeção, todas as amostras são linearmente separáveis e, portanto, classificadores simples podem ser utilizados na tarefa de classificação.



Figura 6.10 – Exemplos de texturas dinâmicas da base de dados dyntex. Essa base é composta por 123 classes, cada uma contendo 10 amostras. Os vídeos possuem pelo menos 250 quadros com dimensão de 400×300 pixels. Todos os quadros foram convertidos para níveis de cinza.

6.3 Reconhecimento de Texturas Dinâmicas

Para validar o método proposto e comparar a sua eficiência com outros métodos em tarefas de reconhecimento, dois conjuntos de experimentos foram realizados na base de dados dyntex (91). Essa base de dados consiste em 123 classes de texturas dinâmicas, cada uma contendo 10 amostras. Os vídeos contêm alta variabilidade, apresentando um desafio interessante para a proposta de caracterização e reconhecimento. Todos os vídeos possuem pelo menos 250 quadros com dimensão de 400×300 pixels. Cada quadro dos vídeos foi convertido para níveis de cinza. A Figura 6.10 apresenta exemplos de texturas dinâmicas da base de dados dyntex.

O primeiro experimento tem como objetivo determinar a melhor configuração de parâmetros por meio de uma análise detalhada. No segundo experimento, os resultados são comparados com métodos existentes na literatura. Em todos os experimentos, a classificação foi realizada por meio das máquinas de vetores de suporte (SVM).

6.3.1 Avaliação dos Parâmetros

A análise experimental foi conduzida para investigar os efeitos de cada parâmetro do método proposto no reconhecimento de texturas dinâmicas. O método assume os seguintes parâmetros: (i) conjunto de tamanho de memória – μ_1, \dots, μ_M – (ii) regras de movimentação – $r = \min$ ou $r = \max$ – (iii) planos ortogonais – $\phi \in \{XY, XT, YT\}$.

A Figura 6.11 sumariza a análise do tamanho de memória e regras de movimentação na base de dados dyntex em termos da taxa de classificação correta. Cada gráfico apresenta os resultados obtidos pelos planos XY , XT , e YT . Como pode ser visto, o tamanho da memória tem grande influência no desempenho de classificação. Na maioria dos casos, os resultados sugerem que a taxa de classificação correta é decrementada ao passo que o tamanho da memória é incrementado. Isso provavelmente ocorre devido ao agente possuir mais dificuldade em encontrar um novo atrator à medida que o tamanho da memória aumenta. Por outro lado, os tamanhos de memória menores fornecem uma melhor análise local refletindo em maiores taxas de acerto. Os resultados também mostram que a regra de movimentação \max obtém melhores resultados do que a regra \min . A regra \max está associada a regiões heterogêneas, onde a mudança nas intensidades ocorre de maneira abrupta. A combinação entre as regras de movimentação $[\min, \max]$ não apresentou resultados superiores e foram omitidos dos gráficos.

Resultados interessantes foram obtidos através da concatenação de tamanhos de memória, como apresentado na Figura 6.12. Como pode ser visto, a concatenação fornece melhores resultados comparado com tamanhos de memória individuais. Na base de dados dyntex, uma melhoria de 21.87% foi obtida quando a concatenação aumentou de $[1, 2]$ para $[1, 2, 3, 4, 5]$ (Figura 6.12(a)). Portanto, a concatenação de tamanhos de memória diminui a importância individual de cada valor de μ , fornecendo uma classificação de texturas dinâmicas mais eficaz.

Finalmente, a Figura 6.13 fornece uma comparação entre os planos ortogonais. A concatenação de planos sempre supera os resultados de planos individuais. Para a base de dados dyntex – Figura 6.13, é observado que a melhor taxa de classificação correta de 97.64% é obtida para a concatenação dos planos $[XY, XT, YT]$. Além disso, verificou-se que a concatenação dos planos $[XT, YT]$ fornece resultados piores do que as concatenações $[XY, XT]$ e $[XY, YT]$. Estes resultados sugerem que as características de aparência, obtidas pelo plano XY , são importantes para discriminar texturas dinâmicas na base de dados dyntex. Os resultados obtidos pela concatenação de planos corroboram com o fato de que os três planos

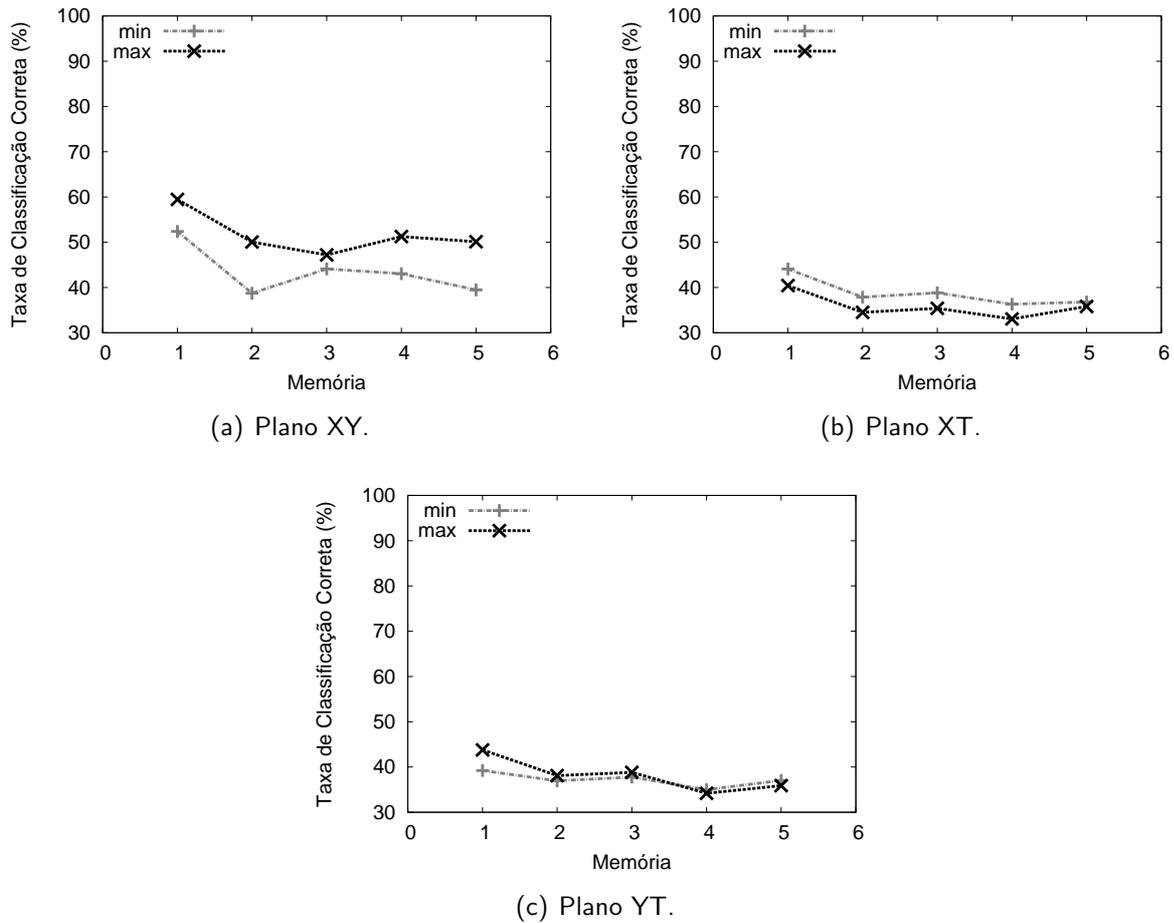


Figura 6.11 – Taxa de classificação correta para tamanhos de memória μ e regras de movimentação r . Cada gráfico apresenta os resultados para os planos na base de dados dyntex.

se completam de forma adequada e são importantes na classificação de texturas dinâmicas. Por meio dos experimentos acima, o conjunto adequado de parâmetros é definido como: $\mu = [1, 2, 3, 4, 5]$, $r = max$ e planos $[XY, XT, YT]$.

6.3.2 Comparação com Métodos da Literatura

Nesta seção, os resultados do método proposto são comparados com os métodos da literatura, indicados na Tabela 6.3.2, através da classificação de vídeos da base de dados dyntex. Na

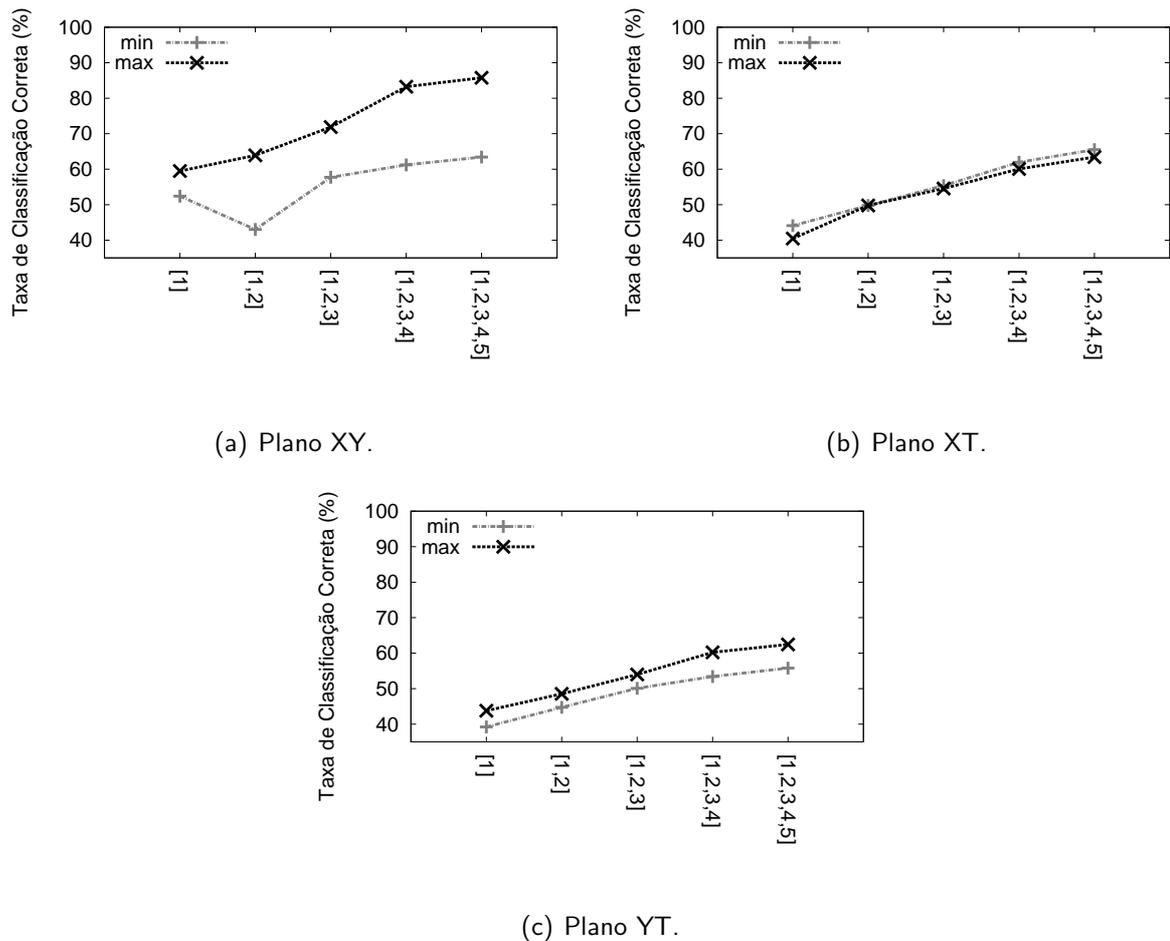


Figura 6.12 – Taxa de classificação correta para diferentes conjuntos de tamanho de memória obtida pelo método proposto na base de dados dyntex. Cada gráfico corresponde a um plano ortogonal.

tabela são apresentados a média e o desvio padrão em termos da taxa de classificação correta das pastas da validação cruzada. Além disso, a tabela inclui o número de características que cada método extrai dos vídeos. A última coluna apresenta a média das taxas de classificação correta obtida pelos classificadores k-vizinhos mais próximos (*k-nearest neighbor* – KNN) e máquinas de vetores de suporte (*Support Vector Machine* – SVM).

Os resultados da tabela indicam que o método proposto fornece uma taxa média de 97.60%, seguido pelos padrões locais binários (*Local Binary Patterns* – LBP-TOP) (93) com média de 96.34%. Por fim, uma taxa de classificação correta de 96.14% foi obtida pelos padrões locais binários volumétricos invariantes à rotação (*Rotation Invariant Volumetric Local Binary Patterns* – RI-VLBP) (94). Considerando os classificadores individuais, o método LBP-TOP obteve o melhor resultado no classificador KNN, seguido pelo método proposto com uma diferença de 1.46%. Por outro lado, no classificador SVM, o método proposto obteve um incremento de 3.98% comparado com o método LBP-TOP. O método RI-VLBP obteve a

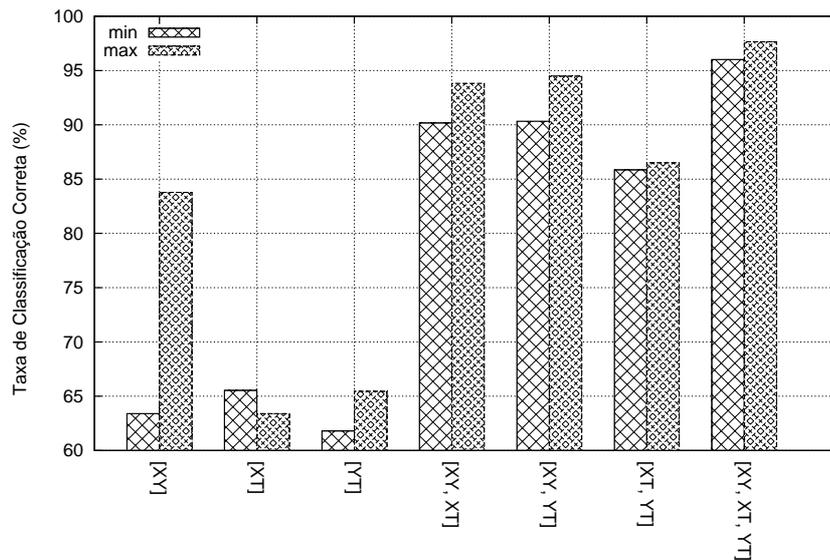


Figura 6.13 – Taxa de classificação correta para diferentes concatenações de planos na base de dados dyntex.

Tabela 6.1 – Resultados comparativos para métodos da literatura em classificação de texturas dinâmicas da base de dados dyntex.

Método	N. de Caract.	KNN	SVM	Média
RI-VLBP	4115	97.64 (± 1.32)	94.63 (± 2.68)	96.14
LBP-TOP	768	99.02 (± 0.77)	93.66 (± 2.29)	96.34
Método proposto	75	97.56 (± 1.20)	97.64 (± 1.34)	97.60

menor taxa em ambos os classificadores. Considerando o número de características, o método proposto extrai somente 75 características, enquanto que os demais métodos extraem no mínimo 768 características. Esse resultado é extremamente importante quando a complexidade do método é um componente crítico da aplicação.

6.4 Segmentação de Texturas Dinâmicas

A segmentação de imagens ou vídeos é um passo importante em muitos sistemas de visão computacional. Particularmente, segmentação de texturas dinâmicas consiste em dividir o vídeo em regiões homogêneas no espaço e no tempo. Desta maneira, a segmentação de vídeos, incluindo texturas dinâmicas, defronta um grande desafio devido à necessidade de coerência entre a segmentação de cada quadro, o qual é obtido através da combinação de informações espaço-temporal.

Nesta seção, é proposta uma abordagem para segmentação de texturas dinâmicas baseada em CDPARs e o algoritmo k-means. Inicialmente, vetores de características são extraídos para cada pixel do vídeo utilizando o método de caracterização de texturas dinâmicas descrito nas seções anteriores. Após a extração, os vetores obtidos para cada pixel são agrupados pelo algoritmo k-means. Embora o k-means forneça soluções robustas, é amplamente reportado na literatura que o seu desempenho depende da escolha inicial dos centroides. Para superar este problema, experimentos com seis métodos de inicialização do k-means são também reportados. Os resultados experimentais mostraram que o método de segmentação proposto supera os métodos existentes por dois motivos principais: a extração de características de texturas dinâmicas é mais eficaz e a escolha inicial dos centroides é mais robusta.

6.4.1 Método Proposto para Segmentação de Texturas Dinâmicas

A Figura 6.14 sumariza o método proposto: (1) o vídeo é dividido em blocos sobrepostos; (2) Para cada bloco, um vetor de características é obtido por meio do método de texturas dinâmicas; (3) O algoritmo k-means agrupa os vetores, resultando no vídeo segmentado.

- Blocos sobrepostos: dado um vídeo composto por T quadros com $w \times h$ pixels, ele é dividido em $N = (w \times h \times T)$ blocos. Para cada pixel j , um bloco β_j com $p \times p \times q$ pixels,

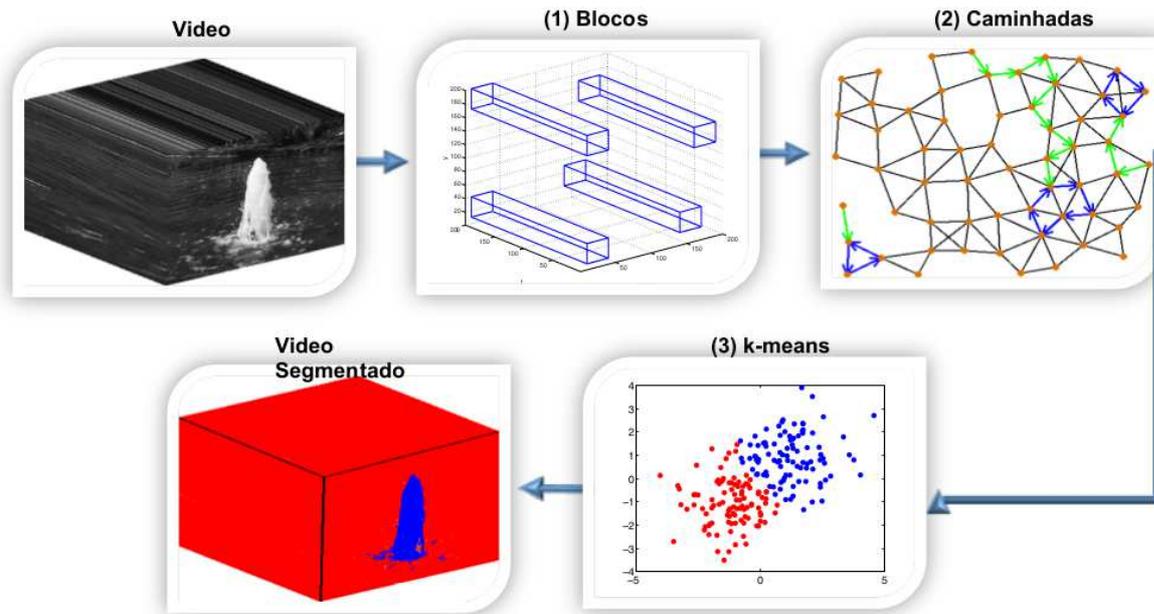


Figura 6.14 – Passos do método proposto de segmentação de texturas dinâmicas. (1) O vídeo é dividido em blocos sobrepostos; (2) um vetor de características é obtido para cada bloco por meio do método de caracterização de texturas dinâmicas; (3) o algoritmo k-means agrupa os vetores e então a segmentação do vídeo é obtida.

centrado no pixel j , é obtido para compor o conjunto de blocos. É importante notar que p e q devem ser grandes o suficiente para caracterizar a textura e, ao mesmo tempo, não englobar texturas diferentes. Estes valores dependem do tamanho das imagens e geralmente os valores para p variam de 5 a 10 ($p = 9$ neste trabalho, $p = 7$ em (95) e $p = 5$ ou 7 em (7, 59)) e os valores para q variam de 10 a T – número de quadros do vídeo.

A Figura 6.15 (a) apresenta alguns exemplos de blocos com dimensão $p \times p \times q$ extraídos de um vídeo. Se os limites de segmentação não mudarem ao longo do tempo, a dimensão temporal q pode ser igual ao número de quadros $q = T$ (Figura 6.15(b)). Neste caso, apenas o primeiro quadro do vídeo precisa ser segmentado, já que os limites das texturas dinâmicas não mudam ao longo do tempo.

- Extração de características: para cada bloco β_j , um vetor de características φ_j (Equação 6.1.10) é extraído utilizando o método de texturas dinâmicas descrito anteriormente. Para extrair o vetor de características, a definição de três parâmetros é necessária: regra de movimentação din , conjunto de memórias μ_1, \dots, μ_M e planos ortogonais. Os mesmos parâmetros dos experimentos de classificação foram utilizados.
- Algoritmo k-means: após a extração de características, os vetores φ_j de cada bloco j

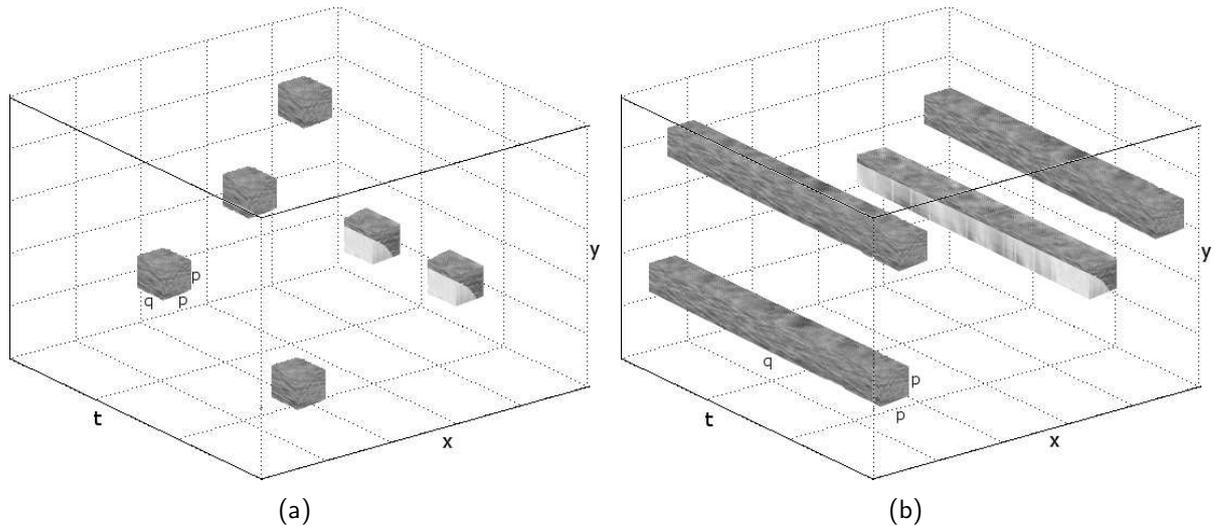


Figura 6.15 – Estratégias para extração de blocos de um vídeo. (a) Exemplos de blocos com dimensões $p \times p \times q$ pixels. (b) Exemplos de blocos com dimensão temporal $q = T$.

são agrupados por meio do algoritmo k-means. Este algoritmo foi escolhido, pois ele é bem estabelecido na área de agrupamento de dados, além de ser computacionalmente eficiente e não requerer a especificação de muitos parâmetros. O k-means é um algoritmo iterativo que minimiza a distância intra-cluster (distância entre os vetores do mesmo grupo) movendo o centroide \bar{x}_i para a média do seu conjunto Voronoi.

6.4.2 Configuração dos Experimentos

Experimentos utilizando vídeos sintéticos e reais foram executados para avaliar o método proposto de segmentação de texturas dinâmicas. Em todos os experimentos, blocos de dimensão $p = 9$ e $q = 60$ foram utilizados para extrair os vetores de características. É importante enfatizar que nenhuma estratégia adicional baseada no relacionamento espacial dos pixels foi utilizada para refinar a solução obtida pelo k-means.

Para a avaliação do método de segmentação, os experimentos foram conduzidos em 300 vídeos de texturas dinâmicas sintéticas (59). Devido à alta variabilidade, os vídeos apresentam

um desafio interessante e difícil para segmentação adequada das texturas dinâmicas. Os vídeos são compostos por $k = 2, 3, 4$ grupos (2, 3, 4 texturas dinâmicas diferentes). Cada grupo foi aleatoriamente escolhido de um conjunto de 12 texturas dinâmicas, as quais incluem grama, mar, água fervendo, escada rolante, fogo, fumaça, água e plantas. Os vídeos possuem 60 quadros com resolução de 160×110 pixels. Exemplos da partição real, da partição inicial utilizada como inicialização do k-means no método Manual, e vídeos com $k = 2, 3, 4$, podem ser visualizados na Figura 6.16.

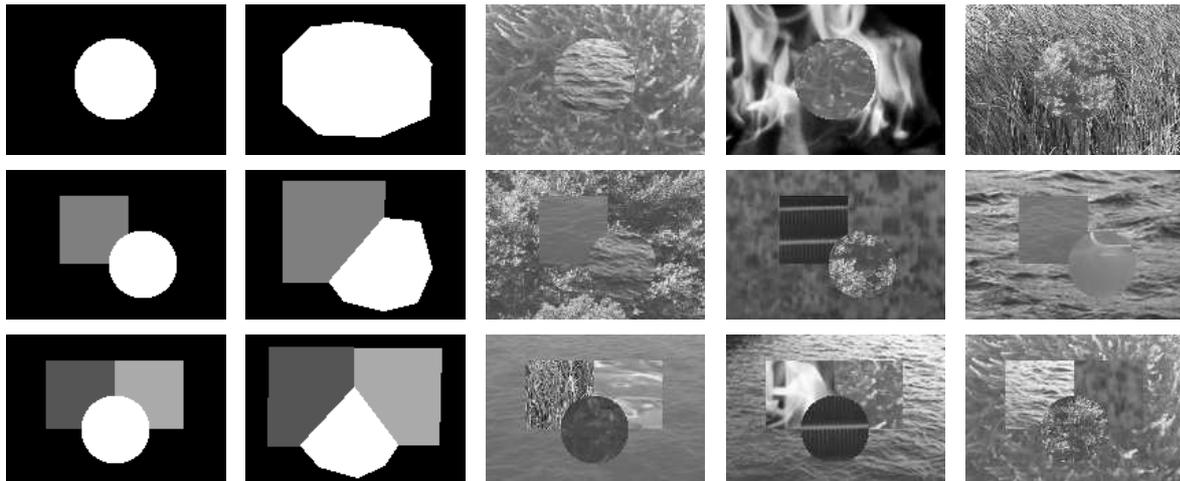


Figura 6.16 – A primeira coluna apresenta a partição real, a segunda coluna apresenta a partição inicial para o método de inicialização do k-means, e a terceira coluna apresenta exemplos dos vídeos. As linhas representam os vídeos compostos por 2, 3 ou 4 texturas dinâmicas, respectivamente.

Para avaliar o método proposto em um contexto real, foram executados experimentos em texturas dinâmicas da base de dados dyntex (91). Todos os vídeos utilizados nos experimentos possuem 50 quadros com dimensão de 720×576 pixels. Exemplos de texturas dinâmicas reais podem ser vistos na Figura 6.17.

6.4.3 Análise dos Métodos de Inicialização do K-means

Nesta seção, são avaliados os resultados obtidos pelo k-means quando inicializado por cada um dos seis métodos de inicialização descritos na Seção A.2.2. Para quantificar os resultados



Figura 6.17 – Exemplos de texturas dinâmicas reais da base de dados dyntex. Todos os vídeos possuem 50 quadros com dimensão de 720×576 pixels.

de segmentação, o critério de avaliação *rand index* (96) foi utilizado em todos os experimentos. Esse critério avalia o grau de concordância entre a solução do agrupamento e a solução ideal previamente conhecida. Para o método de múltiplas execuções do k-means, o número de iterações M é igual a 35 com base no gráfico da Figura 6.18. Esse gráfico apresenta o número de iterações M versus a média do *rand index* para a base de dados com $k = 4$. De acordo com o gráfico, a média do *rand index* estabiliza para $M > 30$.

A Tabela 6.2 apresenta a média e o desvio padrão do número de iterações que o k-means precisa para convergir. O k-means converge quando os centroides obtidos na iteração anterior não mudam significativamente na próxima iteração. Como esperado, a inicialização manual permite que o k-means execute menos iterações em todas as bases de dados, a qual é seguida pelo método de múltiplas execuções e pelo método PCA. É interessante notar que o número de grupos não influenciou o número de iterações para o método de inicialização manual, o que não acontece com os outros métodos.

Na Tabela 6.3, as partições finais obtidas por diferentes métodos de inicialização são quantitativamente avaliadas. Como pode ser observado, o método de inicialização manual obteve o maior valor de *rand index*. Desconsiderando-se o tempo computacional, o método de múltiplas execuções do k-means fornece resultados comparáveis ao método manual. Assim, esse método pode ser utilizado como uma alternativa automática para inicialização do k-means.

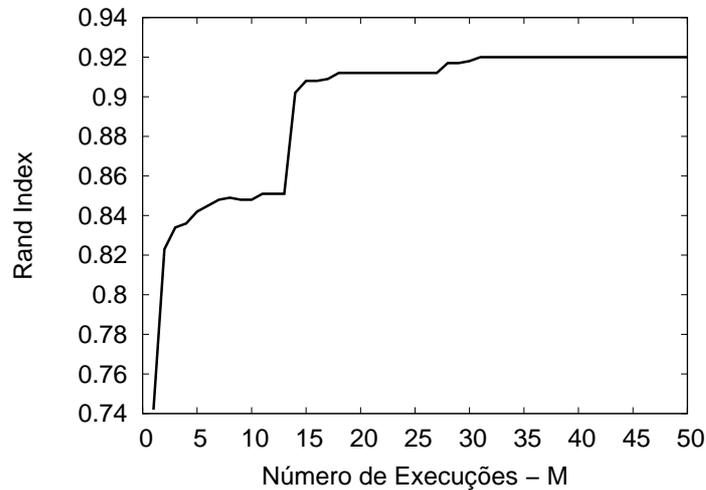


Figura 6.18 – Número de iterações M versus a média do *rand index* para a base dados com $k = 4$. A média do *rand index* estabiliza para $M > 30$.

Tabela 6.2 – Média e desvio padrão do número de iterações que o k-means precisa para convergir quando inicializado por diferentes métodos.

Métodos de Inicialização	$k = 2$	$k = 3$	$k = 4$
Manual	11.01(± 06.02)	12.90(± 08.13)	12.74(± 08.04)
Múltiplas execuções	12.03(± 08.02)	14.99(± 08.56)	18.52(± 13.21)
PCA	13.79(± 08.43)	19.44(± 14.26)	21.26(± 11.42)
KKZ	19.31(± 12.10)	25.41(± 16.29)	27.75(± 15.24)
Astrahan	18.22(± 06.17)	22.33(± 11.83)	25.94(± 14.40)
Aleatório	18.81(± 05.79)	24.50(± 14.28)	29.91(± 15.17)

Embora o método PCA não forneça resultados comparáveis aos dois melhores, ele pode ser considerado uma alternativa automática para segmentação em tempo real. Os demais métodos de inicialização apresentaram resultados similares ao método aleatório, com uma pequena vantagem para o método Astrahan.

Tabela 6.3 – Média e desvio padrão do *rand index* para diferentes métodos de inicialização do k-means.

Métodos de Inicialização	$k = 2$	$k = 3$	$k = 4$
Manual	0.97(± 0.06)	0.95(± 0.08)	0.94(± 0.07)
Múltiplas execuções	0.97(± 0.04)	0.93(± 0.10)	0.92(± 0.10)
PCA	0.96(± 0.06)	0.83(± 0.16)	0.85(± 0.11)
KKZ	0.95(± 0.09)	0.79(± 0.15)	0.78(± 0.11)
Astrahan	0.96(± 0.07)	0.79(± 0.15)	0.83(± 0.12)
Aleatório	0.95(± 0.10)	0.77(± 0.15)	0.77(± 0.10)

Para estatisticamente comparar os métodos de inicialização, o teste de Friedman (97) seguido pelo teste post-hoc Nemenyi foram executados no *rand index*. Basicamente, os valores de *rand index* para cada vídeo são ranqueados de maneira que o método com maior valor de *rand index* recebe o valor 1, o segundo maior *rand index* recebe o valor 2, e assim por diante. Após o ranqueamento, o teste de Friedman é aplicado para testar a veracidade da hipótese nula. A hipótese nula afirma que todos os métodos de inicialização fornecem desempenho estatisticamente similar. Se a hipótese nula é rejeitada, o teste post-hoc Nemenyi é aplicado para verificar quais pares de métodos apresentam desempenho significativamente diferentes. Para todos os testes, foi utilizado o valor padrão $\alpha = 0.05$. As Tabelas 6.4 e 6.5 apresentam as comparações entre pares de métodos para os vídeos com $k = 3$ e $k = 4$, respectivamente. O símbolo s_{ij} de cada posição da tabela possui os seguintes significados: \odot - não existe diferença estatística entre os métodos i e j , \triangleleft - existe diferença estatística entre os métodos i e j , sendo o método i superior ao método j , e \triangle - existe diferença estatística entre os métodos i e j , sendo o método i inferior ao método j .

Por meio das tabelas, os métodos manual e múltiplas execuções não apresentam resultados estatisticamente diferentes. Entretanto, eles demonstram ser estatisticamente superiores aos demais métodos. Para vídeos com $k = 3$, os métodos PCA, Astrahan, KKZ e Aleatório não apresentam resultados com diferenças significativas. Entretanto, para vídeos mais desafiadores com $k = 4$, o método PCA obteve resultados significativamente melhores que os métodos KKZ e Aleatório.

Como ilustração, a segmentação final obtida por diferentes métodos de inicialização é apresentada na Figura 6.19.

Tabela 6.4 – Comparações entre métodos de inicialização do k-means para vídeos com $k = 3$ utilizando testes estatísticos.

	Manual	Múltiplas execuções	PCA	Astrahan	KKZ	Aleatório
Manual	■	⊙	△	△	△	△
Múltiplas execuções	⊙	■	△	△	△	△
PCA	△	△	■	⊙	⊙	⊙
Astrahan	△	△	⊙	■	⊙	⊙
KKZ	△	△	⊙	⊙	■	⊙
Aleatório	△	△	⊙	⊙	⊙	■

Tabela 6.5 – Comparações entre métodos de inicialização do k-means para vídeos com $k = 4$ utilizando testes estatísticos.

	Manual	Múltiplas execuções	PCA	Astrahan	KKZ	Aleatório
Manual	■	⊙	△	△	△	△
Múltiplas execuções	⊙	■	△	△	△	△
PCA	△	△	■	⊙	△	△
Astrahan	△	△	⊙	■	⊙	⊙
KKZ	△	△	△	⊙	■	⊙
Aleatório	△	△	△	⊙	⊙	■



Figura 6.19 – Segmentação final obtida por diferentes métodos de inicialização. A primeira linha corresponde aos primeiros quadros dos vídeos e as demais linhas seguem: inicialização manual, múltiplas execuções, PCA, KKZ, Astrahan e Aleatório.

6.4.4 Comparação com Métodos de Segmentação

Além dos resultados referentes à inicialização do k-means, o método proposto é comparado com quatro abordagens da literatura: Layered Dynamic Texture - LDT (7), Ising (98), Multiple Dynamic Texture - MDT (59) e Generalized Principal Component Analysis - GPCA (99). Nessa comparação, estes métodos foram inicializados manualmente utilizando a segmentação inicial fornecida pela base de dados, exceto para o LDT que foi inicializado com os resultados da segmentação do método DTM (7). Os resultados dos métodos da literatura, incluindo as imagens, foram obtidos dos artigos originais e estão disponíveis no sítio ^{*}.

Na Tabela 6.6, são apresentados a média e o desvio padrão do *rand index* para segmentação de texturas dinâmicas. Como descrito no trabalho original, o Ising pode apenas segmentar vídeos compostos por duas texturas dinâmicas. Como pode ser observado, o método proposto utilizando inicialização manual fornece os melhores resultados. Melhorias de 0.03, 0.06 e 0.02 foram obtidas pelo método proposto comparados com o método LDT para vídeos com $k = 2, 3, 4$, respectivamente. Além disso, o método proposto, quando inicializado automaticamente com múltiplas execuções, obteve excelentes resultados comparado com os demais métodos inicializados manualmente.

Tabela 6.6 – Comparação entre os métodos de segmentação de texturas dinâmicas para bases com $k = 2, 3, 4$.

Método de Segmentação	$k = 2$	$k = 3$	$k = 4$
Método proposto - Manual	0.97(± 0.06)	0.95(± 0.08)	0.94(± 0.07)
Método proposto - Múltiplas execuções	0.97(± 0.04)	0.93(± 0.10)	0.92(± 0.10)
LDT	0.94(± 0.12)	0.89(± 0.13)	0.92(± 0.10)
MDT	0.91(± 0.13)	0.85(± 0.13)	0.87(± 0.11)
Ising	0.88 (± 0.14)	–	–
GPCA	0.54(± 0.11)	0.55(± 0.11)	0.55(± 0.07)

Na Figura 6.20 são apresentados resultados da segmentação de texturas dinâmicas para os diferentes métodos da literatura. Cada coluna da figura corresponde a um dos seguintes métodos: método proposto, MDT e LDT.

^{*}<http://www.svcl.ucsd.edu/projects/layerdytex/synthdb/>

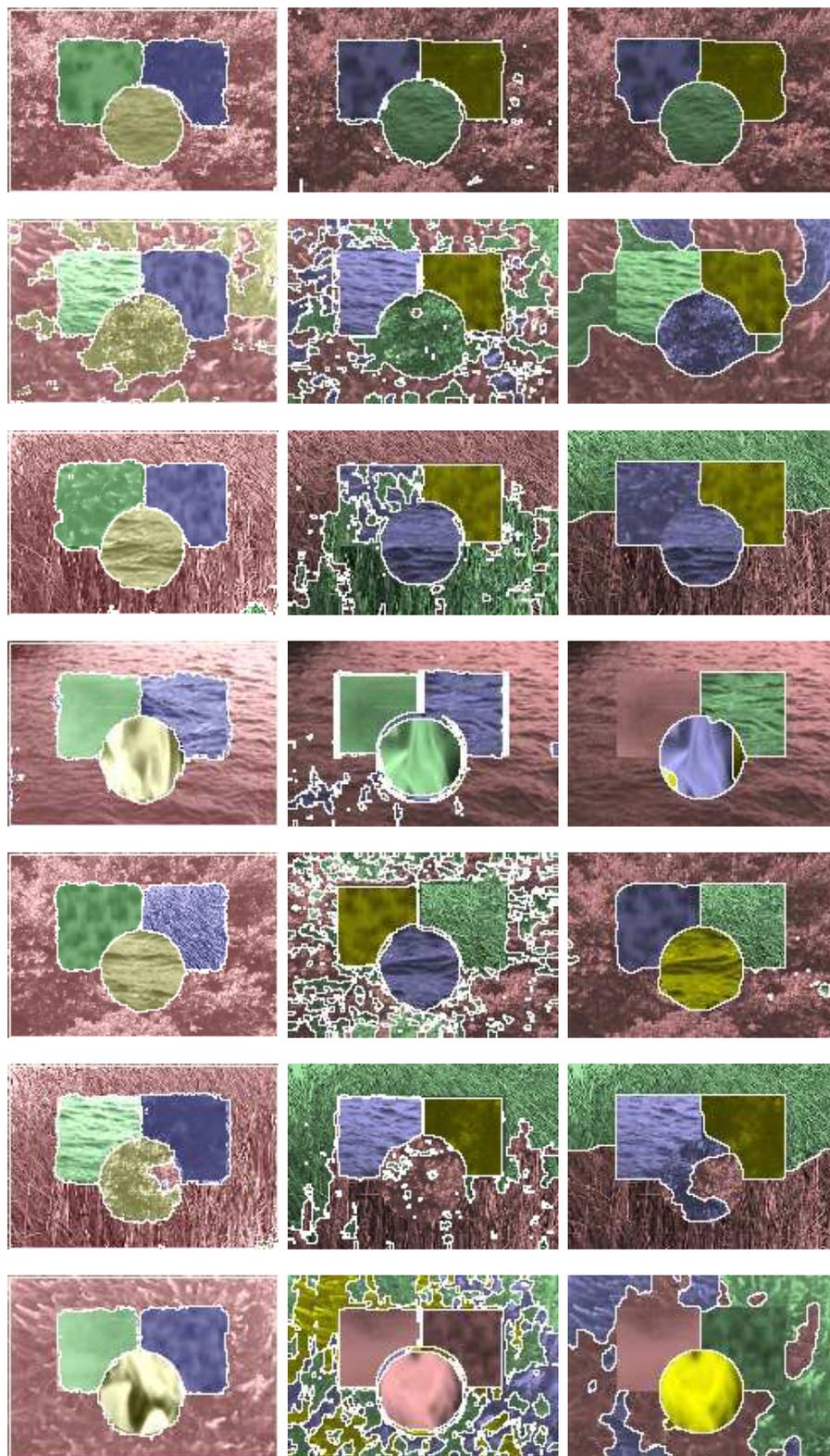


Figura 6.20 – Segmentação final obtida por diferentes métodos de segmentação de texturas dinâmicas. Cada coluna corresponde a um método de segmentação na seguinte ordem: método proposto, MDT e LDT. As linhas correspondem a diferentes vídeos.

6.4.5 Resultados em Texturas Dinâmicas Reais

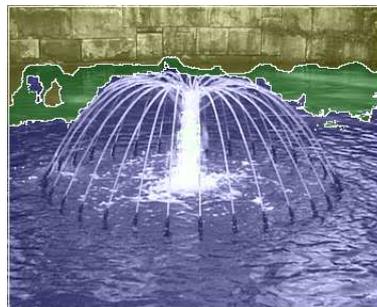
Os resultados em segmentação de texturas dinâmicas em um contexto real são apresentados nessa seção. O método proposto foi aplicado utilizando o método de múltiplas execuções para inicialização do k-means. Na Figura 6.21, são ilustrados os resultados de segmentação para diferentes valores de k na textura dinâmica da Figura 6.21(a). Como pode ser observado, o método é capaz de discriminar a parede da água utilizando $k = 2$ (Figura 6.21(b)). Para $k = 3$ (Figura 6.21(c)), as três classes são atribuídas à parede, água calma e água em movimento. Finalmente, para $k = 4$ (Figura 6.21(d)), o método proposto separa as quatro classes presentes no vídeo, denominadas parede, água calma, água em movimento e fonte. Dessa maneira, nosso método fornece uma segmentação hierárquica do vídeo mantendo a adequação das classes.



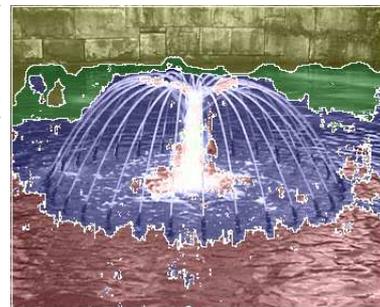
(a) Primeiro quadro do vídeo.



(b) $k = 2$.



(c) $k = 3$.



(d) $k = 4$.

Figura 6.21 – Segmentação de textura dinâmica real utilizando o método proposto para diferentes valores de k .

As Figuras 6.22 e 6.23 apresentam os resultados para vídeos de texturas dinâmicas reais. Os resultados do método proposto são apresentados na coluna à esquerda. Para comparação,

os resultados obtidos por Fazekas et al (100)[†] são apresentados na coluna à direita. Os resultados de segmentação demonstram que o método está de acordo com a percepção visual das texturas dinâmicas mesmo em um contexto desafiador.

6.4.6 Partes não-informativas do Vídeo

Para analisar partes não-informativas do vídeo, isto é, regiões sem variância no tempo, foram utilizadas amostras de texturas dinâmicas com diferentes padrões de movimento. As classes de texturas dinâmicas, mostradas na Figura 6.24 (a), (b) e (c), são: *chão*, *água 1*, *árvore*, *parede*, *água 2* e *fumaça*. Desse conjunto, as classes *água 1*, *água 2* e *fumaça* apresentam partes informativas, enquanto que as demais classes são estáticas no tempo. A Figura 6.24 (d) apresenta a curva do histograma $\psi_{\mu=1, r=max}^{XT}$ para cada amostra. Pode ser observado que existe uma relação entre o histograma e o padrão de movimento. Em texturas com movimento, atratores próximos uns aos outros são obtidos pelas CDPARs. Neste caso, o histograma apresenta um pico no começo da curva e decai rapidamente devido ao pequeno número de caminhadas longas. Por outro lado, para texturas estáticas no tempo, a probabilidade de um agente encontrar um atrator varia de acordo com a região de movimento, tornando o histograma mais uniforme.

Aplicando-se um limiar na entropia das curvas do histograma, as texturas estáticas no tempo podem ser descartadas. Geralmente, a entropia de histogramas uniformes é maior que a entropia de histogramas que apresentam um pico no início da curva. Assim, a Figura 6.25 apresenta um exemplo de regiões descartadas cuja entropia do histograma $\psi_{\mu=1, r=max}^{XT}$ é maior que 1, 42.

[†]As imagens foram obtidas do sítio <http://www.phy.bme.hu/fazekas/dtsegm/>

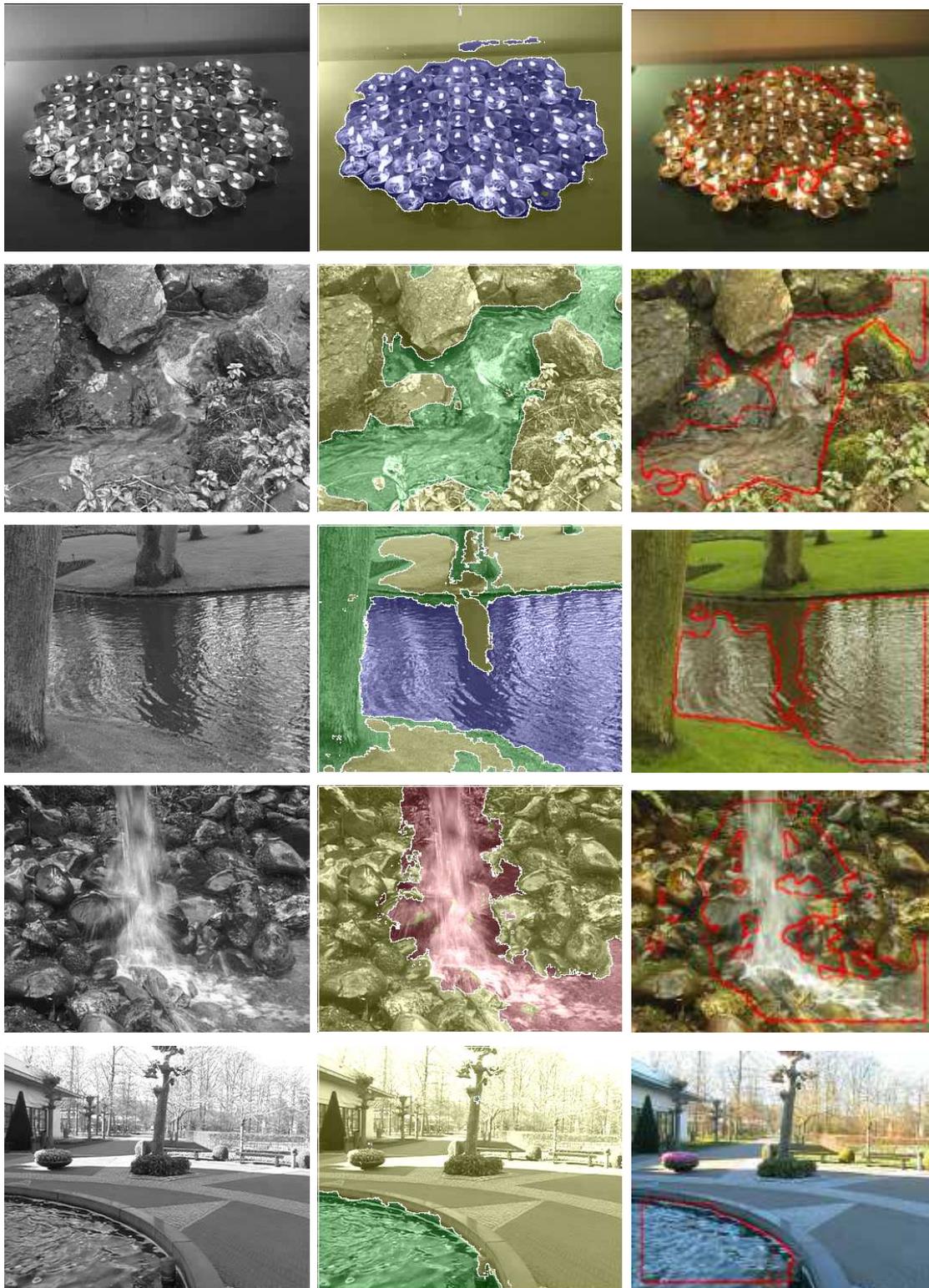


Figura 6.22 – Comparação de segmentação de texturas dinâmicas reais. A coluna à esquerda apresenta os resultados de segmentação para o método proposto, enquanto que a coluna à direita apresenta os resultados obtidos por Fazekas.

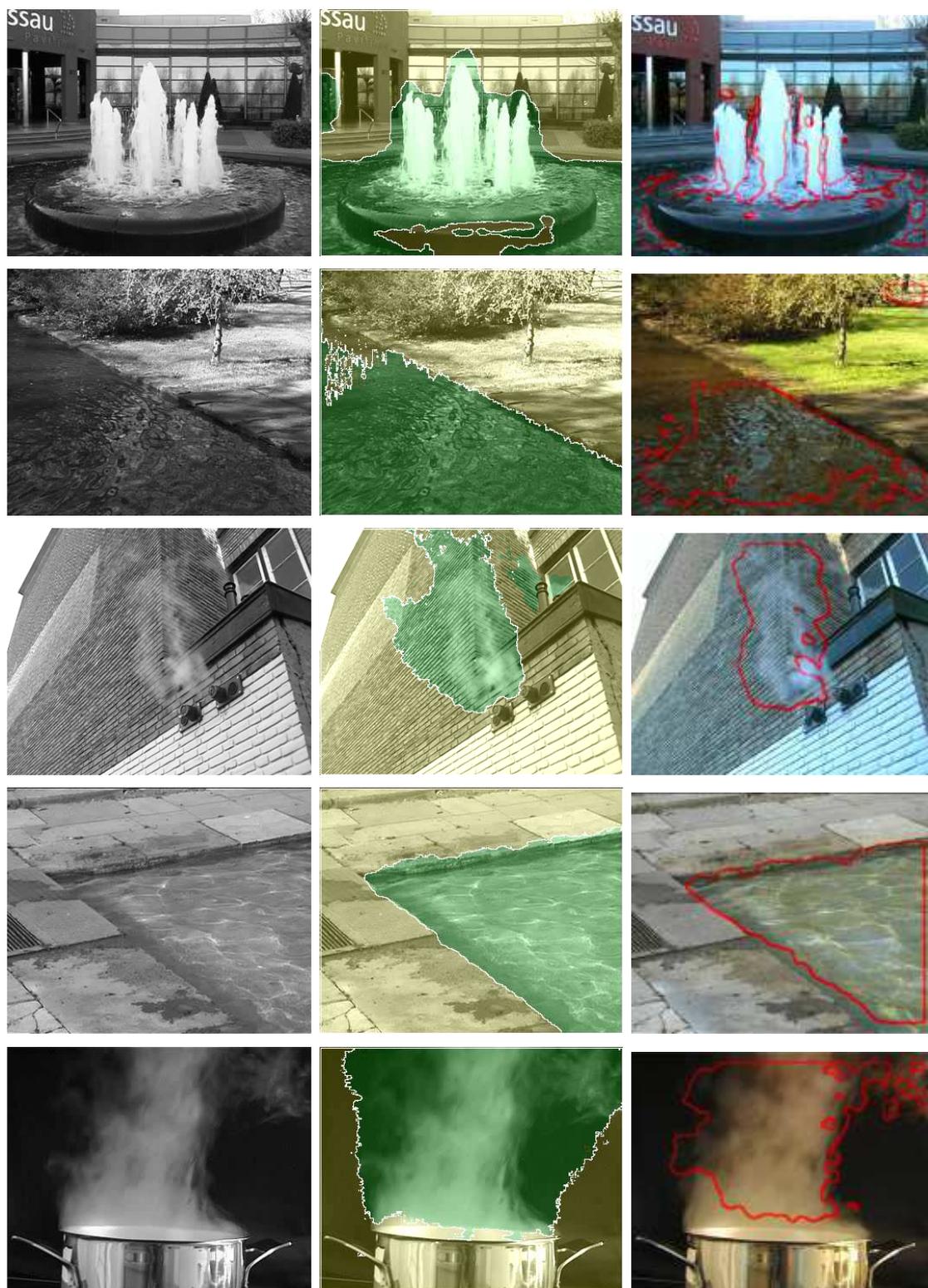
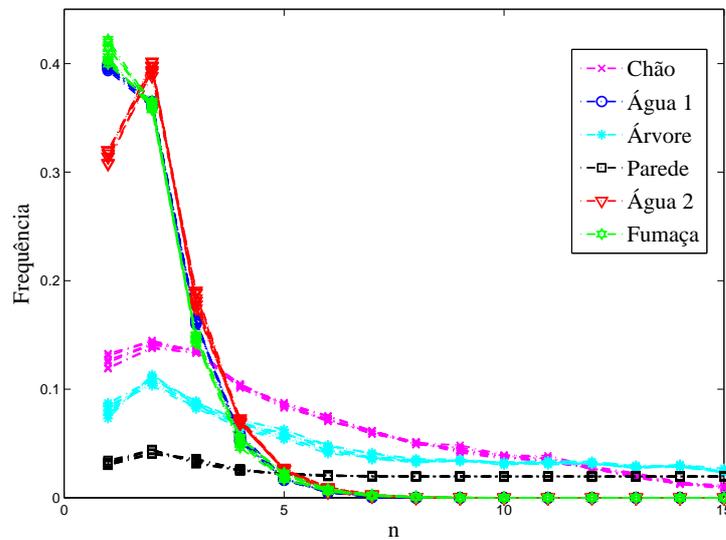


Figura 6.23 – Comparação de segmentação de texturas dinâmicas reais. A coluna à esquerda apresenta os resultados de segmentação para o método proposto, enquanto que a coluna à direita apresenta os resultados obtidos por Fazekas.

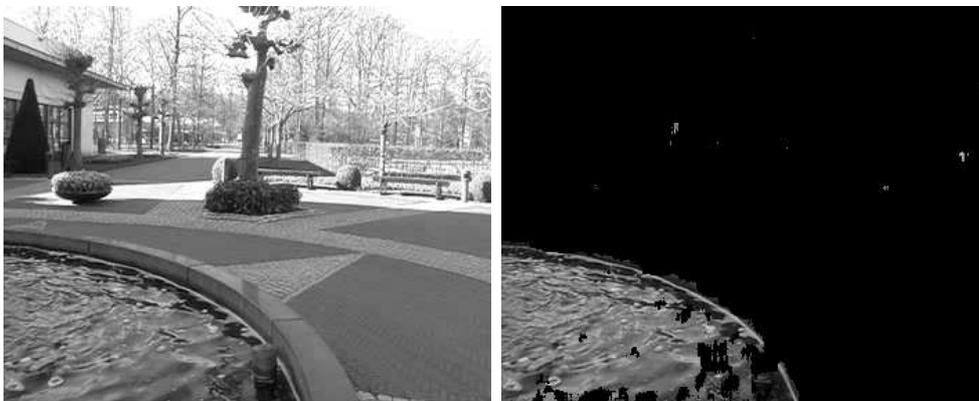


(a) Amostras das classes *chão*, (b) Amostras das classes *parede* e (c) Amostras das classes *fumaça*, *água 1* e *árvore*.



(d) Histograma para $\mu = 1$, $r = max$ e $\phi = XT$.

Figura 6.24 – Histograma para amostras de texturas dinâmicas reais com diferentes padrões de movimento.



(a) Primeiro quadro do vídeo.

(b) Resultado da segmentação.

Figura 6.25 – Exemplo de regiões descartadas com base na entropia do histograma $\psi_{\mu=1, r=max}^{XT}$. Os pixels cuja entropia é maior que 1,42 foram descartados.

6.5 Complexidade Computacional

Para compreender a complexidade computacional do método proposto, é preciso considerar que CDPARs são executadas em três planos ortogonais. Dessa maneira, a complexidade computacional é igual a três vezes a complexidade em se aplicar as caminhadas em cada plano. Para um vídeo com T quadros de N pixels cada, TN caminhadas são executadas considerando cada pixel como um ponto inicial de uma caminhada. Cada caminhada consiste de duas partes: tempo de transiente τ e atrator de tamanho $\rho \geq \mu + 1$. Assim, a complexidade computacional do método proposto é formulada como $O(3TN(\tau + \rho))$, onde $(\tau + \rho)$ é o tamanho da caminhada.

O tamanho da caminhada $(\tau + \rho)$ varia de acordo com a conectividade dos pixels e o tamanho da memória μ . O melhor caso é obtido quando todas as caminhadas iniciam em um atrator ($\tau = 0$) e esse atrator possui tamanho mínimo $\rho = 1$, o qual conduz a uma complexidade $O(3TN)$. Por outro lado, o pior caso ocorre quando um atrator nunca é encontrado durante todas as caminhadas. Neste caso, o tamanho da caminhada é $\tau + \rho = N$ e a complexidade é dada por $O(3TN^2)$. É importante enfatizar que ambos os casos são raros, requerendo uma configuração muito específica dos pixels na imagem. No caso médio, o tamanho de memória $\mu = 5$ resulta em caminhadas de tamanho $(\tau + \rho) = 23$, em média. Assim, o método proposto apresenta uma complexidade muito próxima do melhor caso $O(3TN)$, dado que $N \gg 23$.

6.6 Conclusão

Um novo método para caracterização de texturas dinâmicas foi apresentado por meio de CDPARs. Neste capítulo foi demonstrada como caminhadas podem ser aplicadas em três planos ortogonais derivados do vídeo de texturas dinâmicas. Essa estratégia é uma maneira eficaz

de salientar características de aparência e movimento como descrito na análise do método. Outra vantagem do método proposto é a sua capacidade de descrever múltiplas texturas dinâmicas sem qualquer procedimento adicional.

Resultados interessantes foram obtidos em duas aplicações: reconhecimento e segmentação de texturas dinâmicas. Em ambas as aplicações, os resultados obtidos foram comparados com alguns dos principais métodos da literatura e foi demonstrada a eficácia do método proposto.

Aplicações

Este capítulo descreve as aplicações dos métodos propostos, uma aplicação de texturas estáticas e duas aplicações de texturas dinâmicas.

7.1 Aplicação I: Reconhecimento de Plantas

A imensa variedade de plantas e suas características taxonômicas tornam a identificação de espécies um desafio científico e tecnológico. Em geral, a tarefa de catalogar plantas envolve: (i) coletar e desidratar um ramo da planta – (ii) processar o *voucher* (parte da planta processada) – (iii) compará-lo com outros *vouchers* nos herbários – (iv) enviar o material a especialistas para confirmação (101). Todo o processo descrito acima é manual e, portanto, considerado lento e suscetível a erros humanos. Além disso, o processo de desidratação pode causar perda de propriedades importantes para identificação da espécie, como a cor, a textura e o brilho.

A identificação de plantas envolve a análise de partes do espécime, tais como flores, sementes, folhas e partes lenhosas (102). Diferente das flores e frutos, as folhas são facilmente encontradas e coletadas praticamente em todas as estações do ano, simplificando o processo de aquisição de dados. Além disso, a análise de suas características morfológicas e anatômicas é considerada simples, fornecendo importantes informações para proposta de identificação.

Este capítulo discute a aplicação dos métodos de texturas estáticas para reconhecimento de plantas por meio da textura presente nas folhas. Essa tarefa é difícil e complexa devido à natureza das folhas. Embora elas apresentem características fundamentais da sua espécie, as folhas também apresentam uma ampla variedade de padrões. Essa variação pode ocorrer em folhas da mesma planta, em que as características como maturação, clima e exposição ao sol produzem variações no tamanho, na cor e na textura das folhas da mesma espécie.

As seções abaixo descrevem a base de dados utilizada no reconhecimento de espécie de plantas e os resultados obtidos pelos métodos propostos nos Capítulos 4 e 5.

7.1.1 Base de Dados

A base de folhas utilizada neste trabalho foi disponibilizada em (114). As folhas de 40 espécies de plantas foram manualmente coletadas e lavadas para remover impurezas. Cada folha foi ajustada ao seu eixo central (linha que conecta a base e a ponta) para ficar em posição vertical e, em seguida, foi digitalizada por um scanner (HP Scanjet 3800) com resolução de 1.200 dpi. A resolução alta permite que detalhes das folhas sejam capturados.

Para o experimento proposto, as amostras de textura foram obtidas das folhas. Esta tarefa consiste em extrair janelas aleatórias e sem sobreposição da imagem, como apresentado na Figura 7.1. Para cada folha, 30 amostras em níveis de cinza de 128×128 pixels foram extraídas. Uma amostra para cada espécie é apresentada na Figura 7.2.

A variação de textura ocorre entre folhas ou até mesmo dentro de uma mesma folha. Não são raros os casos onde a textura da parte inferior da folha difere bastante da textura de sua parte superior devido às condições em que a folha foi submetida. A Figura 7.3 apresenta a variação de texturas dentro da mesma espécie.

7.1.2 Classificação de Espécies de Plantas

De forma a avaliar os métodos de textura em um problema real e complexo, os resultados experimentais para a base de dados de folhas são apresentados na Tabela 7.1. O desenvol-

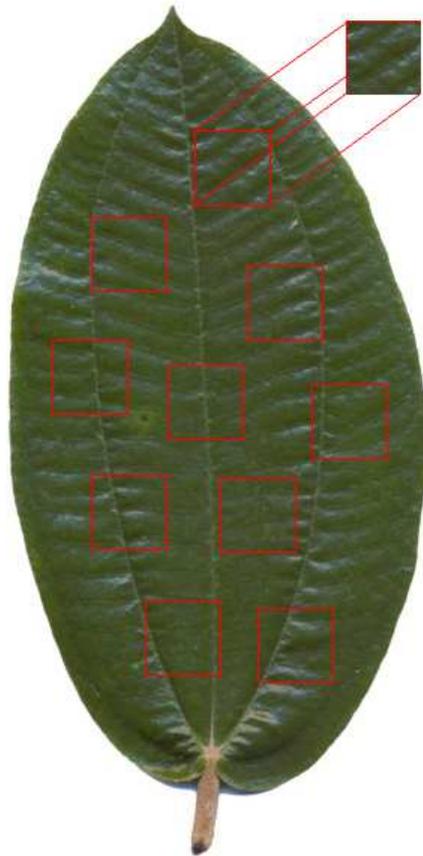


Figura 7.1 – Processo de extração de amostras de texturas de uma folha. Este processo consiste em extrair janelas aleatórias e sem sobreposição.

vimento de uma ferramenta automatizada para identificar espécies de plantas é um desafio científico principalmente devido à alta variabilidade entre folhas da mesma espécie (79). A partir da tabela, é observado que o método proposto no Capítulo 5 (*método proposto 2*) supera os demais métodos, incluindo o método proposto no Capítulo 4 (*método proposto 1*). O *método proposto 1* alcançou uma taxa de classificação correta de 92.08%, enquanto que o *método proposto 2* obteve uma taxa de 96.58%. Ambos os métodos obtiveram resultados melhores quando comparados com os métodos da literatura.

As Figuras 7.4 e 7.5 apresentam as matrizes de confusão dos métodos propostos nos Capítulos 4 e 5, respectivamente. Cada coluna da matriz representa as amostras de uma classe prevista e cada linha representa as amostras da classe real. Por meio dessas matrizes, é possível identificar quais classes estão sendo confundidas e, então, concluir importantes considerações sobre os métodos. Como apontado pela flecha em vermelho (Figura 7.4), podemos perceber que a espécie 22 é confundida com a espécie 16. De fato, as texturas são similares, como pode ser visto na Figura 7.2. Apesar disso, ambas as matrizes de confusão mostram que, em geral, as espécies são classificadas adequadamente.

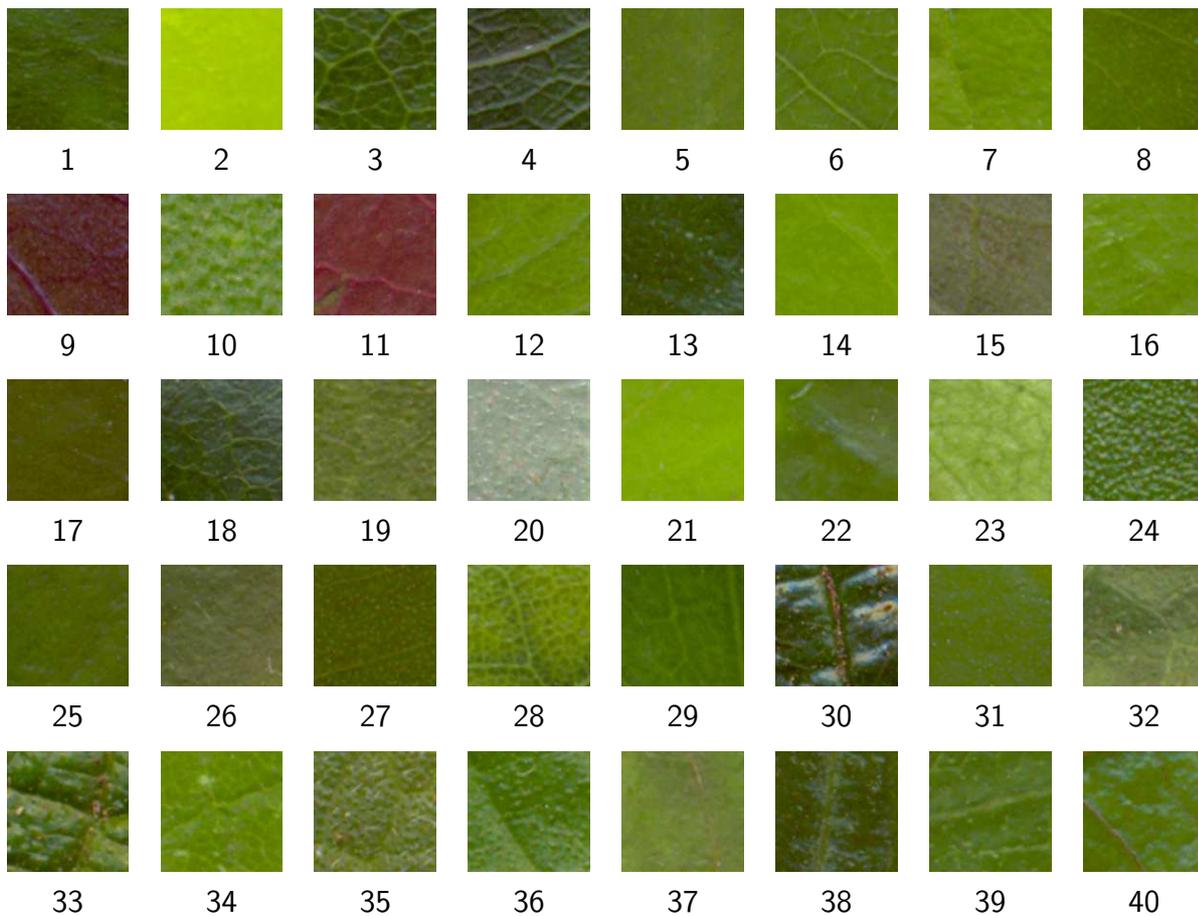


Figura 7.2 – Amostra de textura para cada uma das 40 espécies de plantas. Na base de dados, cada espécie contém 30 amostras.

Tabela 7.1 – Resultados experimentais para os métodos de textura utilizando a base de dados de folhas.

Método	Taxa de Classificação Correta
Descritores de Fourier	53.06(±3.87)
Matrizes de co-ocorrência	78.17(±3.38)
Filtros de Gabor	60.71(±4.11)
Caminhada determinística do turista	62.09(±3.68)
Dimensão fractal multiescala	79.90(±3.35)
Espectro multifractal	32.38(±3.64)
Dimensão fractal local	57.62(±3.48)
Padrões locais binários	90.28(±2.49)
Método proposto 1	92.08(±2.35)
Método proposto 2	96.58 (±1.52)

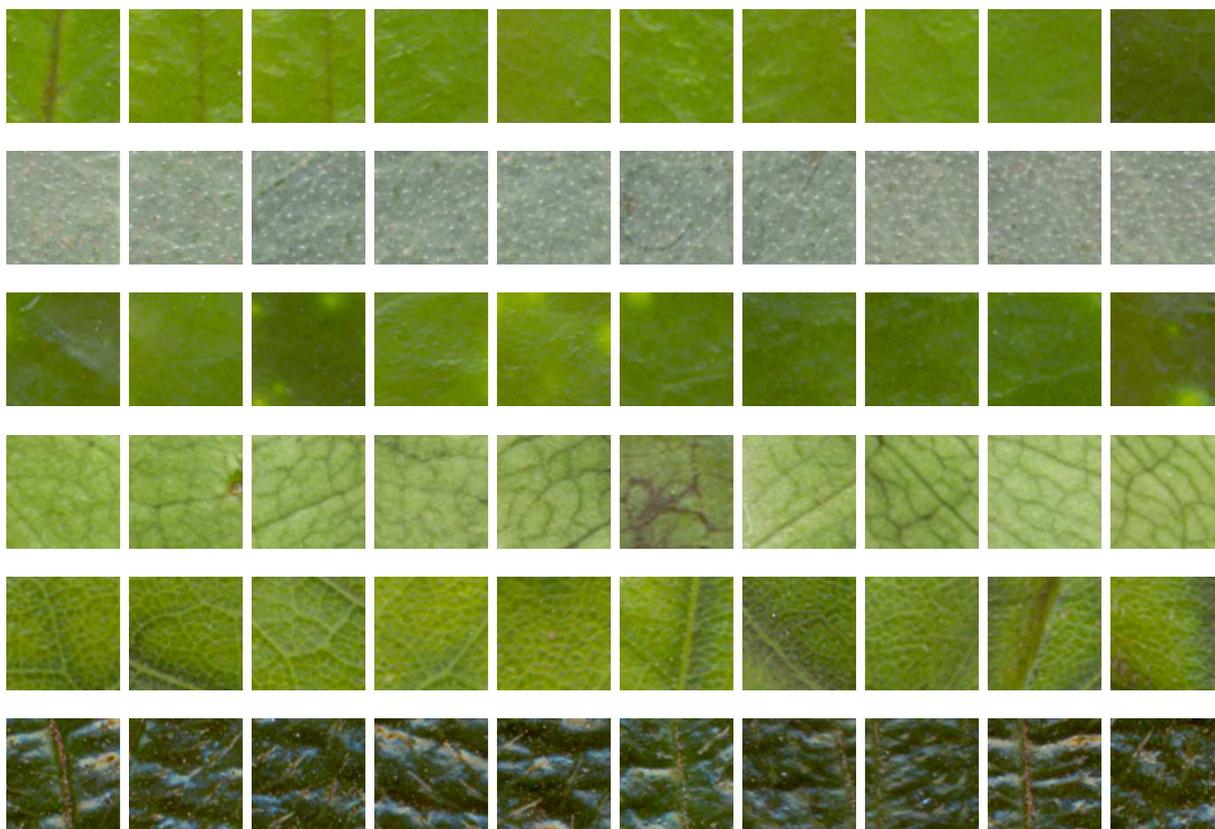


Figura 7.3 – Variação de algumas amostras de seis espécies de plantas. A variação depende das condições em que a folha foi submetida.

7.2 Aplicação II: Análise do Tráfego de Veículos

Nos últimos anos, a utilização de sistemas de vídeo para monitoramento do tráfego tem demonstrado ser uma alternativa promissora em relação aos sistemas tradicionais de monitoramento, como os radares de velocidade. O interesse na utilização de câmeras para monitorar diferentes aspectos do tráfego (e.g. congestionamento de veículos) têm crescido devido ao baixo custo e facilidade de instalação em relação aos radares tradicionais que precisam ser instalados sob o asfalto. Além da capacidade de analisar os veículos individualmente, a análise do tráfego por meio de vídeos ainda pode fornecer informações globais, tais como a velocidade do tráfego global e ocupação das pistas.

A maioria dos trabalhos existentes em monitoramento de tráfego por meio de vídeos utiliza uma combinação entre algoritmos de segmentação e algoritmos de rastreamento de veículos (103–105). Inicialmente, o veículo em potencial é segmentado na imagem utilizando características de movimento (103–104) ou através de métodos de subtração de fundo (105). Uma vez que a segmentação foi realizada, os veículos são rastreados nos quadros utilizando filtros preditivos (e.g. filtros de Kalman e filtros de partículas). Como desvantagem, a precisão desses sistemas é extremamente dependente da qualidade da segmentação da imagem. A precisão de tais algoritmos torna-se ainda mais crítica na presença de condições adversas, tais como mudanças climáticas (e.g. nublado, brilho, noite), sombras e oclusão parcial ou completa dos veículos. Além disso, a segmentação não pode ser obtida com confiabilidade em imagens de baixa resolução, pois os veículos ocupam poucos pixels na imagem. Os algoritmos de rastreamento também possuem diversos problemas quando existem muitos veículos nas imagens, o que tipicamente ocorre em congestionamentos (106).

Dado que a maioria das informações necessárias para classificação do tráfego está contida na interação entre os movimentos do vídeo, uma representação holística pode ser utilizada para modelar a variabilidade do campo de movimento, sem a necessidade de segmentar ou rastrear os veículos individualmente. Neste capítulo, o campo de movimento do tráfego é modelado como uma textura dinâmica por meio do método proposto no Capítulo 6, evitando assim, as limitações dos algoritmos de segmentação e rastreamento. Dados os vídeos de tráfego, eles podem ser classificados em categorias (e.g. tráfego leve ou intenso) utilizando classificadores como K-vizinhos mais próximo - KNN e máquinas de vetores de suporte - SVM.

Nas seções abaixo, a base de dados contendo os vídeos de tráfego é descrita e, em seguida,



Figura 7.6 – Exemplos de vídeos da base de dados de tráfego. Esta base é composta por 254 vídeos divididos em três classes. Cada vídeo possui entre 42 e 52 quadros com resolução de 320×240 .

os resultados experimentais em reconhecimento e agrupamento de tráfego são apresentados. Esses resultados mostram que o método proposto é útil em tarefas de classificação de congestionamento, pois o modelo é robusto às variações na condição de iluminação, oclusão e desfocagem das imagens.

7.2.1 Base de Vídeos

A base de dados, disponibilizada em (106), é composta por 254 vídeos de tráfego, coletados com uma câmera fixa por dois dias, de uma rodovia em Seattle. Os vídeos contêm diferentes intensidades de tráfego e condições climáticas (e.g. neblina, chuva e sol). Eles foram capturados com resolução de 320×240 pixels e possuem entre 42 e 52 quadros a 10 quadros por segundo. Cada vídeo foi rotulado manualmente em uma das seguintes classes: tráfego leve, tráfego médio e tráfego pesado. Esta base contém um total de 44 amostras de tráfego pesado, 45 de tráfego médio e 165 de tráfego leve. A variedade do tráfego pode ser vista na Figura 7.6.

7.2.2 Classificação do Tráfego de Veículos

Nesta seção, o método proposto no Capítulo 6 foi avaliado em um experimento de classificação cujo objetivo é identificar a intensidade do tráfego de um vídeo. Inicialmente, uma análise dos parâmetros foi realizada com o intuito de entender o comportamento do método na classificação de tráfego. Em seguida, os resultados obtidos são comparados com duas abordagens da literatura. Os vídeos foram classificados usando máquinas de vetores de suporte (SVM) na estratégia de validação cruzada com 10 pastas.

A Figura 7.7 apresenta os resultados de classificação para concatenação de tamanhos de memória. Cada gráfico corresponde aos resultados usando características extraídas dos planos XY , XT e YT . Como pode ser visto, a concatenação de tamanhos de memória fornece, na maioria dos casos, resultados melhores do que a utilização de um único tamanho de memória. Apesar da pequena melhoria, a concatenação de tamanhos de memória diminui a importância individual de cada μ , assim fornecendo uma classificação mais eficaz.

Por fim, a Figura 7.8 apresenta uma comparação entre concatenação de planos ortogonais. Como pode ser visto, a concatenação de planos sempre supera planos individuais. Para a base de tráfego de carros, observa-se que a maior taxa de classificação correta de 97.64% é obtida pela concatenação dos três planos $[XY, XT, YT]$ e regra de movimentação *max*. Como esperado, as características extraídas do plano XY (aparência dos carros) contribuem menos do que as características extraídas dos planos XT e YT (movimento).

7.2.2.1 Comparação entre Métodos

Nesta seção, são comparados os resultados do método proposto e abordagens da literatura. A Tabela 7.2.2.1 apresenta a taxa de classificação correta, desvio padrão e número de características para dois classificadores na base de tráfego de carros. A última coluna da tabela apresenta a média aritmética da taxa de classificação obtida pelos classificadores *k*-vizinhos mais próximos (*k-nearest neighbor* - KNN) e máquina de vetores de suporte (*Support Vector*

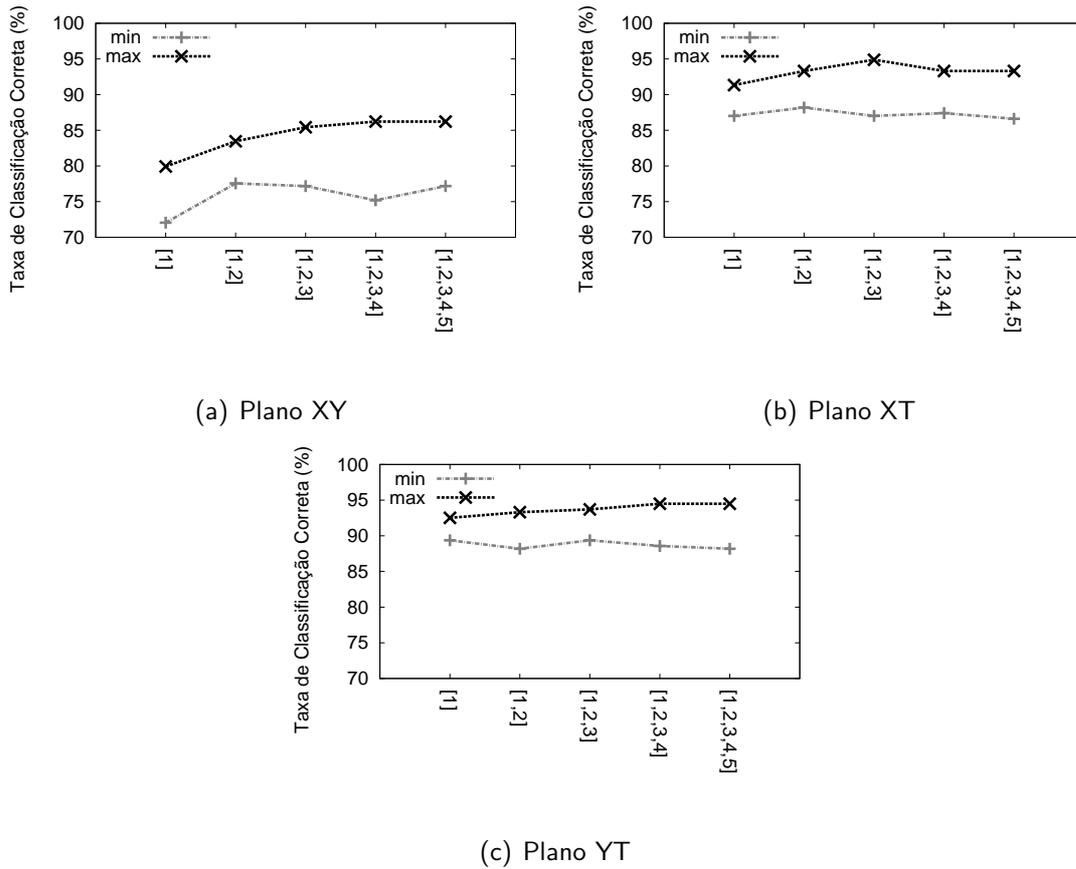


Figura 7.7 – Taxa de classificação correta para diferentes concatenações de tamanhos de memória. Cada gráfico corresponde aos resultados obtidos por cada plano ortogonal.

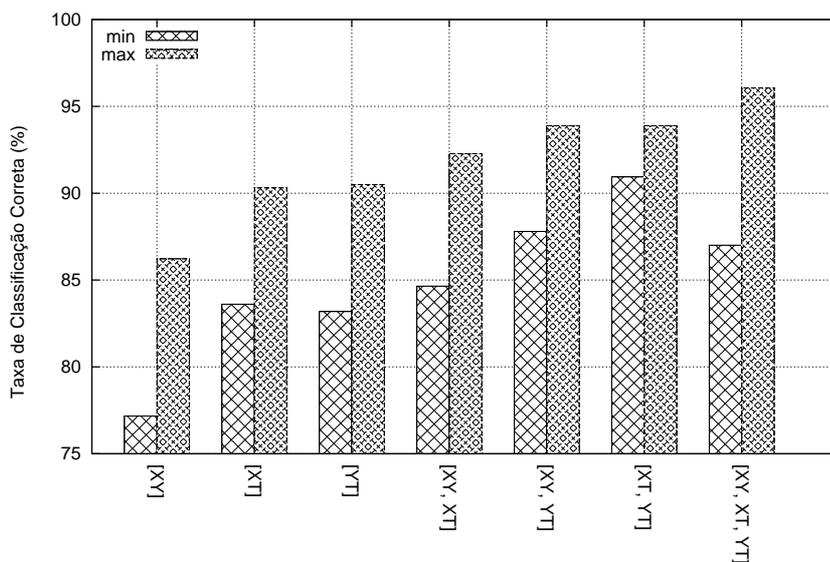


Figura 7.8 – Taxa de classificação correta para diferentes concatenações de planos ortogonais na base de tráfego de carros.

Tabela 7.2 – Comparação de métodos existentes na base de tráfego de carros. A tabela apresenta o número de características, a taxa de classificação correta para os classificadores KNN e SVM e a média da taxa.

Método	N. de Caract.	KNN	SVM	Média
RI-VLBP	4115	93.31 (± 4.34)	93.31 (± 4.10)	93.31
LBP-TOP	768	93.70 (± 4.70)	94.49 (± 4.23)	94.10
Método proposto	75	93.70 (± 5.05)	96.06 (± 3.92)	94.88

Machine - SVM).

É possível notar que o método proposto supera os padrões locais binários volumétricos invariantes à rotação (*Rotation Invariant Volumetric Local Binary Patterns* – RI-VLBP) (94) e apresenta desempenho similar aos padrões locais binários (*Local Binary Patterns* – LBP-TOP) (93). Entretanto, nota-se que o método proposto extrai vetores de características com apenas 75 dimensões, enquanto que o LBP-TOP extrai vetores com 768 dimensões. Comparado com os métodos existentes, os resultados demonstraram que o método proposto é uma representação efetiva para modelar texturas dinâmicas que possuem aparência similar (veículos), mas diferentes padrões de movimento (tráfego leve, médio e pesado).

7.2.3 Agrupamento de Tráfego de Carros

Nesta seção, o algoritmo k-means é utilizado para organizar os vídeos de tráfego de carros em k grupos. Para cada vídeo, um vetor de características é extraído pelo método proposto e utilizado pelo k-means no agrupamento. Vale ressaltar que os algoritmos de agrupamento não consideram as classes atribuídas aos vídeos.

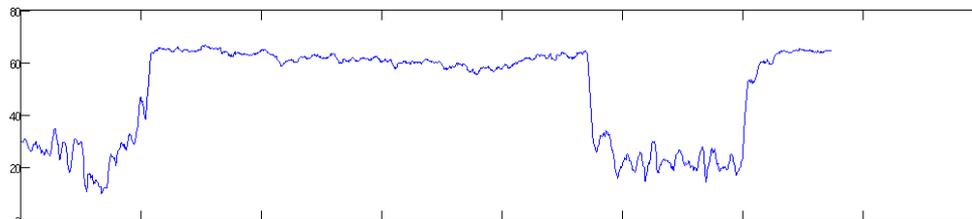
Inicialmente, os vídeos de tráfego foram agrupados em três grupos ($k = 3$). A Figura 7.9 mostra três vídeos típicos para cada um dos três grupos (cada linha corresponde a um grupo). Estes exemplos revelam que os grupos estão de acordo com a categorização perceptual de tráfego: leve, médio e pesado.

Aplicando-se o k-means com $k = 2$, dois grupos são formados, cada um relacionado com as velocidades baixas e altas dos veículos na rodovia. A Figura 7.10 apresenta a evolução

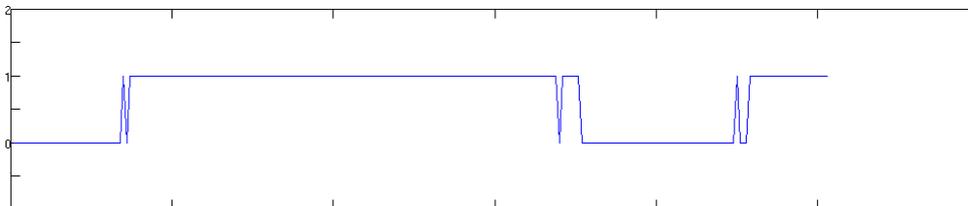


Figura 7.9 – Exemplos de vídeos dos três grupos obtidos pelo k-means usando características extraídas pelo método proposto. Cada linha corresponde a um grupo.

temporal do índice dos grupos e a velocidade medida na rodovia por um sensor. É possível notar uma correspondência entre a velocidade e o índice dos grupos.



(a) Velocidade medida pelo sensor localizado na rodovia.



(b) Índice do grupo obtido pelo k-means usando $k = 2$.

Figura 7.10 – Comparação usando o índice do grupo e a velocidade medida na rodovia. Existe uma correspondência entre ambas as medidas.

7.3 Aplicação III: Classificação das condições da corrosão de aço

O custo dos danos causados pela corrosão tem aumentado consideravelmente nos últimos anos. Além dos custos diretos (e.g. materiais, equipamentos, reparação e substituição), a corrosão gera custos indiretos relacionados com os danos ambientais, desperdício de recursos e perdas por parada de produção. Estes custos podem ser reduzidos com a aplicação de tecnologias de controle de corrosão.

Devido às inúmeras condições a serem avaliadas, os estudos do processo de corrosão são bastante complexos. As condições incluem o tipo de material (composição química e a sua microestrutura) e o meio ao qual ele está submetido (ar atmosférico, solo, produtos químicos). A corrosão ocorre através da interação do material com o meio corrosivo, e as condições dessa interação irá definir o tipo de corrosão.

Para o monitoramento da corrosão, as técnicas comumente utilizadas incluem a inspeção visual, medidas de perda de massa, sensores de hidrogênio e ultrassom (107). Embora elas sejam amplamente utilizadas, essas técnicas são de difícil implementação ou fornecem resultados não conclusivos para alguns tipos de corrosão. Por exemplo, podemos citar que a análise da corrosão por perda de massa não apresenta resultados confiáveis devido à pequena massa perdida durante alguns tipos de corrosão. A inspeção visual é bastante lenta e subjetiva quanto a pequenas mudanças na corrosão.

Tendo em vista estas considerações, esta seção apresenta o estudo do processo de corrosão do aço carbono em sequências de imagens. As imagens podem ser facilmente obtidas por microscopia de uma pequena parte do material e, então, todo o processo pode ser automatizado. O aço carbono é alvo de estudo, pois é amplamente utilizado na confecção de tubulações e máquinas que são utilizadas para extração de petróleo (108–109). No âmbito nacional, este estudo torna-se ainda mais importante, pois o petróleo brasileiro possui um alto teor de material corrosivo, como o gás sulfeto (H_2S). Desta forma, a principal contribuição deste capítulo é a aplicação do método de texturas dinâmicas na identificação do tipo de corrosão.

7.3.1 Base de dados

Para a base de dados, foram obtidas as micrografias dos eletrodos de aço carbono 1020 expostos a um meio corrosivo contendo a espécie H_2S em solução (faixa de pH 3-5) nas concentrações de 100 e 1000 ppm, na presença e ausência de cloreto. As soluções utilizadas nas medidas voltamétricas foram preparadas com água obtida pelo Sistema de Osmose Reversa da marca Gehaka - modelo OS- 20-LX- FARMA. O H_2S foi gerado pela dissociação do sal $Na_2S \cdot 9H_2O$ na concentração de $0,311 \text{ mol L}^{-1}$, obtendo-se a solução final de 100 mL com concentração de 10.000 ppm a qual foi usada como solução estoque para futuras diluições obtendo as concentrações de 100 e 1000 ppm.

Portanto, quatro condições experimentais, que influenciam o tipo de corrosão, foram realizadas: (i) H_2S 100 ppm e Cl – (ii) H_2S 100 ppm – (iii) H_2S 1000 ppm e Cl – (iv) H_2S 1000 ppm. A base de dados é composta por sequências de imagens in situ da superfície do eletrodo realizada com um tempo de aquisição a cada 0,1 s. Cada condição experimental resultou em uma sequência de 20.000 quadros com 1280×960 pixels. O primeiro quadro de cada condição experimental é apresentado na Figura 7.11.

Para realizar os experimentos, 30 amostras foram obtidas sem sobreposição como apresentado na Figura 7.12. Cada amostra possui 200×200 pixels com os mesmos 20.000 quadros da sequência original.

7.3.2 Classificação do tipo de corrosão

Nesta seção, foram analisadas as condições experimentais que resultam em diferentes tipos de corrosão. Para isso, o método proposto no Capítulo 6 foi aplicado em cada amostra da base de dados e o vetor de características foi obtido. Os parâmetros do método foram exatamente os mesmos dos experimentos realizados anteriormente. Os vetores foram utilizados no processo

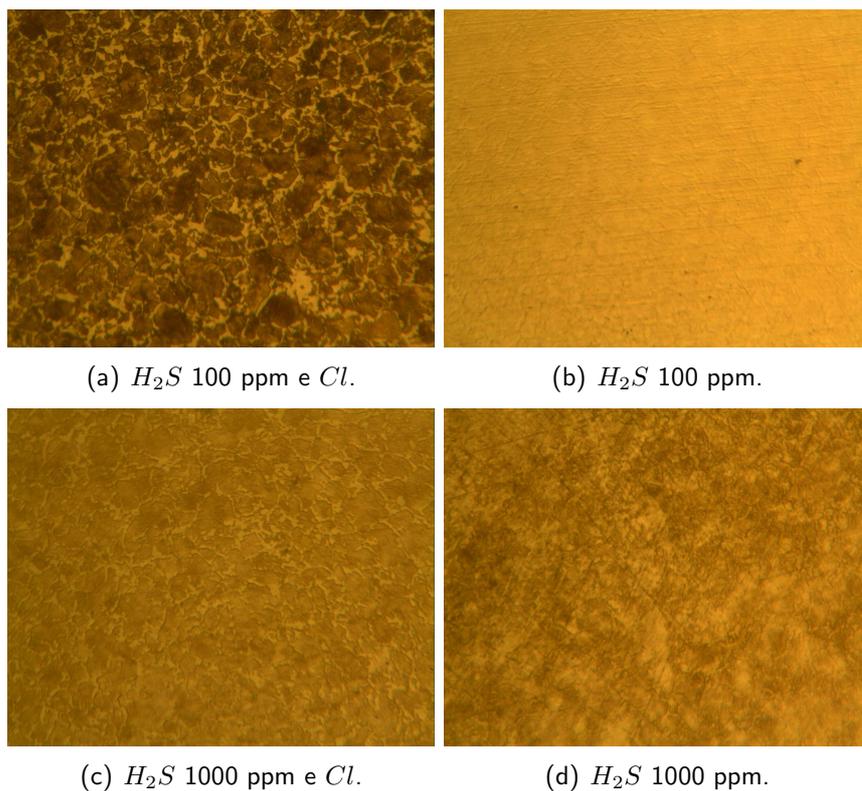


Figura 7.11 – Primeiro quadro de cada condição experimental para verificação da corrosão na mistura.

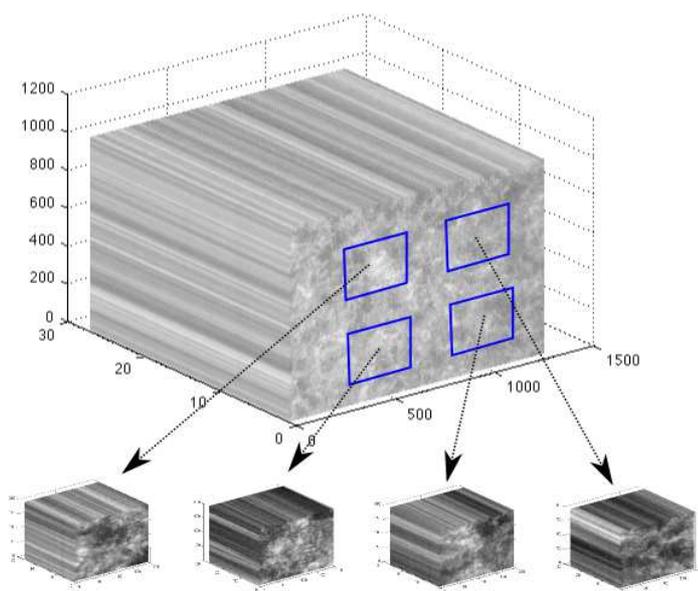


Figura 7.12 – Ilustração do esquema para extrair amostras da sequência de imagens. Cada amostra possui 200×200 pixels com os mesmos 20.000 quadros da sequência original.

de classificação e foi constatada uma taxa de classificação correta de 100% com o classificador vizinho mais próximo em validação cruzada de 10 pastas. Portanto, o método proposto foi

capaz de classificar adequadamente os tipos de corrosão.

Para visualizar o comportamento dos tipos de corrosão, os vetores extraídos foram projetados para duas dimensões por meio da análise de componentes principais. Essa projeção pode ser visualizada na Figura 7.13. É possível notar que, as amostras do mesmo tipo de corrosão são agrupadas, ilustrando o poder de discriminação do método proposto neste trabalho.

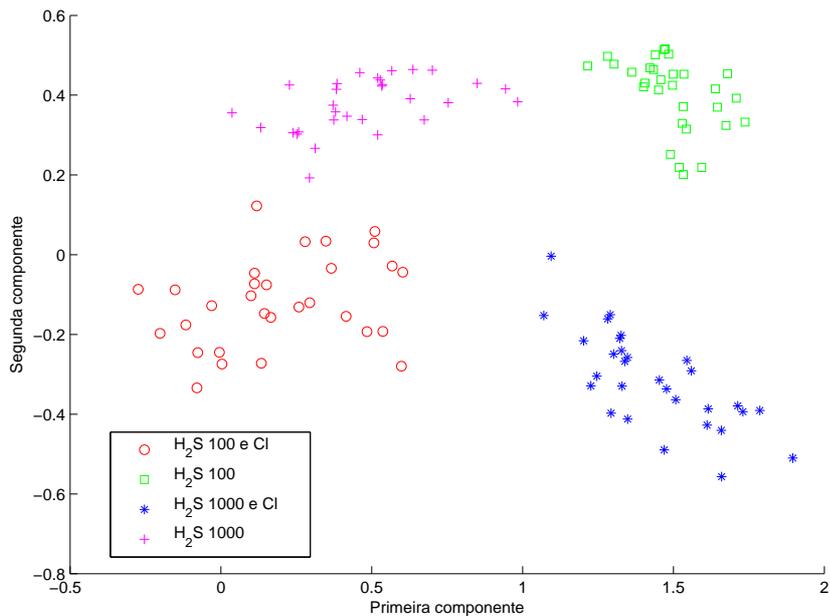


Figura 7.13 – Projeção das duas primeiras componentes obtidas pela aplicação do PCA nos vetores de características das amostras de quatro tipos de corrosão.

Conclusão

O problema de caracterização de texturas em imagens permanece um problema importante em visão computacional, apesar das diversas abordagens que têm sido reportadas ao longo de décadas. Uma das razões para tal situação é o fato de que a textura não foi objetivamente definida devido à sua natureza complexa. De fato, a maioria dos padrões presentes na imagem pode, eventualmente, ser considerada uma textura, e conseqüentemente, são necessários algoritmos mais gerais e robustos.

Este trabalho teve como objetivo estudar, avaliar e implementar métodos de texturas baseados em agentes e modelos. Em texturas estáticas, dois métodos foram propostos: o primeiro baseado em fractal/CDPAR e o segundo baseado em atividade em redes direcionadas. As características extraídas pela dimensão fractal dos atratores mostraram-se mais eficazes em relação ao método tradicional que considera somente o tamanho dos atratores. No segundo método, verificou-se que a atividade em redes direcionadas captura as principais características da textura presente na imagem. Em texturas dinâmicas, as CDPARs foram estendidas para vídeos de modo a descrever características de aparência e movimento. Os resultados experimentais mostraram que a aplicação de CDPARs em três planos do vídeo é adequada para a caracterização de ambas as características.

O foco principal em termos de aplicações foram problemas das áreas de biologia vegetal e nanotecnologia. Em biologia vegetal, os métodos de texturas estáticas foram aplicados no reconhecimento de espécies de plantas e mostraram ser eficientes nesta tarefa. Em nanotecnologia, a classificação de condições da corrosão de aço foi alvo do método de texturas dinâmicas. Nessa aplicação, o método foi capaz de classificar o tipo de corrosão adequadamente. A multidisciplinaridade inerente resultou em contribuições para as áreas envolvidas, o que demonstra a aplicabilidade dos métodos desenvolvidos em outras áreas.

8.1 Contribuições

As principais contribuições originárias deste trabalho são:

- **Texturas estáticas:** proposta de dois novos métodos, o primeiro baseado em CDPARs e dimensão fractal, e o segundo baseado em atividade em redes direcionadas.
- **Texturas dinâmicas:** proposta de um novo método tendo como base as CDPARs com maior poder de discriminação de texturas.
- **Segmentação:** método de segmentação de texturas dinâmicas a partir da aplicação do método proposto em blocos do vídeo. Além da segmentação de diferentes processos visuais, é possível identificar texturas com partes informativas no tempo, isto é, texturas dinâmicas com movimento.
- **Biologia:** aplicação dos métodos de texturas estáticas no reconhecimento de espécies de plantas, auxiliando na catalogação de plantas de maneira automática.
- **Nanotecnologia:** aplicação do método de texturas dinâmicas no reconhecimento dos tipos de corrosão de aço e na identificação do tempo decorrido.
- **Tráfego de carros:** aplicação na classificação da condição do tráfego de rodovias sem a necessidade de métodos de segmentação ou rastreamento de veículos.
- **Publicações:** este trabalho gerou publicações em periódicos e congressos relevantes no cenário internacional com o intuito de reportar as novas propostas para a comunidade científica.

8.2 Trabalhos Futuros

Os resultados obtidos pelos métodos baseados em agentes e modelos demonstraram seu potencial para caracterização de texturas. A lista abaixo apresenta os trabalhos futuros:

- Estender os métodos propostos para texturas coloridas.
- Estender o método baseado em atividade em redes direcionadas para texturas dinâmicas.
- Aprofundar o estudo da aplicação do reconhecimento de corrosão do aço.
- Aplicar o método de texturas dinâmicas no reconhecimento de corrosão de outros materiais.
- Realizar a modelagem e subtração de fundo usando o método de texturas dinâmicas.

8.3 Trabalhos Gerados

- Revistas
 1. GONÇALVES, W. N., BRUNO, O. M. Combining fractal and deterministic walkers to improve texture analysis and classification. *Pattern Recognition*, v. 46, n. 11, p. 2953–2968, 2013. ISSN 0031-3203.
 2. GONÇALVES, W. N., BRUNO, O. M. Dynamic texture analysis and segmentation using deterministic partially self-avoiding walks. *Expert Systems with Applications*, v. 40, n. 11, p. 4283–4300, 2013. ISSN 0957-4174.
 3. GONÇALVES, W. N., BACKES, A. R., MARTINEZ, A. S., BRUNO, O. M. Texture descriptor based on partially self-avoiding deterministic walker on networks. *Expert Systems with Applications*, v. 39, n. 15, p. 11818-11829, 2012. ISSN 0957-4174.

4. GONÇALVES, W. N., MARTINEZ, A. S., BRUNO, O. M. Complex network classification using partially self-avoiding deterministic walks, *Chaos*, v. 22, n. 3, p. 033139, 2012. ISSN 1054-1500.
5. FABBRI, R., GONÇALVES, W. N., LOPES, F. J. P., BRUNO, O. M. Multi-q pattern analysis: a case study in image classification, *Physica A: Statistical Mechanics and its Applications*, v. 391, n. 19, p. 4487-4496, 2012. ISSN 0378-4371.
6. ZIMER, A. M., RIOS, E. C., MENDES, P. C. D., GONÇALVES, W. N., BRUNO, O. M., PEREIRA, E. C., MASCARO, L. H. Investigation of AISI 1040 steel corrosion in H₂S solution containing chloride ions by digital image processing coupled with electrochemical techniques. *Corrosion Science*, v. 53, n. 10, p. 3193–3201, 2011. ISSN 0010-938X.

- Congressos

1. MACHADO, B. B., GONÇALVES, W. N., BRUNO, O. M. Material quality assessment of silk nanofibers based on swarm intelligence. *Journal of Physics: Conference Series*, v. 410, n. 1, p. 012163, 2013. ISSN 1742-6596.
2. MACHADO, B. B., CASANOVA, D., GONÇALVES, W. N., BRUNO, O. M. Partial differential equations and fractal analysis to plant leaf identification. *Journal of Physics: Conference Series*, v. 410, n. 1, p. 012066, 2013. ISSN 1742-6596.
3. GONÇALVES, W. N., BRUNO, O. M. Dynamic texture analysis and classification using deterministic partially self-avoiding walks. In: ADVANCES CONCEPTS FOR INTELLIGENT VISION SYSTEMS, 13., 2011, Ghent, Belgium. *Proceedings...* Ghent: Springer, 2011 (Lecture Notes in Computer Science, v. 6915). p. 349-359.
4. MACHADO, B. B., GONÇALVES, W. N., BRUNO, O. M. Enhancing the texture attribute with partial differential equations: a case of study with gabor filters. In: ADVANCES CONCEPTS FOR INTELLIGENT VISION SYSTEMS, 13., 2011, Ghent, Belgium. *Proceedings...* Ghent: Springer, 2011 (Lecture Notes in Computer Science, v. 6915). p. 337-348.
5. GONÇALVES, W. N., MACHADO, B. B., BRUNO, O. M. Spatiotemporal gabor filters: a new method for dynamic texture recognition. In: WORKSHOP DE VISÃO COMPUTACIONAL, 2011, Curitiba. *Proceedings ...* Curitiba: WVC, 2011. p. 184–189.
6. GONÇALVES, W. N., MACHADO, B. B., BRUNO, O. M. Segmentação de texturas dinâmicas: um novo método baseado em caminhadas determinísticas. In:

WORKSHOP DE VISÃO COMPUTACIONAL, 2011, Curitiba. *Proceedings ...* Curitiba: WVC, 2011. p. 73–78.

7. MACHADO, B. B., GONÇALVES, W. N., BRUNO, O. M. Image decomposition with anisotropic diffusion applied to leaf-texture analysis. In: WORKSHOP DE VISÃO COMPUTACIONAL, 2011, Curitiba. *Proceedings ...* Curitiba: WVC, 2011. p. 155–160.
8. GONÇALVES, W. N., SILVA, J. de A., BRUNO, O. M. A rotation invariant face recognition method based on complex network. In: IBEROAMERICAN CONGRESS ON PROGRESS IN PATTERN RECOGNITION, IMAGE ANALYSIS, COMPUTER VISION, AND APPLICATIONS, 2010, São Paulo. *Proceedings ...* São Paulo: Springer Berlin / Heidelberg, 2010, (Lecture Notes in Computer Science, v. 6419). p. 426–433. ISBN 978-3-642-16686-0.

- Submetidos

1. GONÇALVES, W. N., BRUNO, O. M. Activity of networks applied to texture analysis and recognition. *Pattern Analysis and Machine Intelligence*, 2012. Submetido.
2. GONÇALVES, W. N., MACHADO, B. B., BRUNO, O. M. Fractal dimension artificial crawler applied in texture analysis. *Physica A: Statistical Mechanics and its Applications*, 2013. Submetido.
3. FABBRI, R., BASTOS, I. N., NETO, F. D. M., LOPES, F. J. P., GONÇALVES, W. N., BRUNO, O. M. Multi-q pattern classification of polarization curves. *Physica A: Statistical Mechanics and its Applications*, 2013. Submetido.
4. GONÇALVES, W. N., MACHADO, B. B., BRUNO, O. M. Complex networks for dynamic texture recognition, *Pattern Recognition Letters*, 2013. Submetido.

REFERÊNCIAS

- 1 ZHAO, Y.; ZHANG, L.; LI, P. Texture feature fusion for high resolution satellite image classification. In: INTERNATIONAL CONFERENCE ON COMPUTER GRAPHICS, IMAGING AND VISION: NEW TRENDS, 2005, Washington. *Proceedings...* Washington, DC, USA: IEEE Computer Society, 2005. v. 1, p. 19 – 23.
- 2 XIE, X.; MIRMEHDI, M. Texems: texture exemplars for defect detection on random textured surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 29, n. 8, p. 1454–1464, 2007. ISSN 0162-8828.
- 3 GLATARD, T.; MONTAGNAT, J.; MAGNIN, I. E. Texture based medical image indexing and retrieval: application to cardiac imaging. In: INTERNATIONAL WORKSHOP ON MULTIMEDIA INFORMATION RETRIEVAL, 6, ACM SIGMM, 2004, New York. *Proceedings...* New York, NY, USA: ACM, 2004. p. 135–142. ISBN 1-58113-940-3.
- 4 BACKES, A. R.; BRUNO, O. M. Plant leaf identification using multi-scale fractal dimension. In: INTERNATIONAL CONFERENCE ON IMAGE ANALYSIS AND PROCESSING, 15, 2009, Berlin. *Proceedings...* Berlin, Heidelberg: Springer-Verlag, 2009. p. 143–150.
- 5 SHANKAR, V.; RODRIGUEZ, J.; GETTINGS, M. Texture analysis for automated classification of geologic structures. In: SOUTHWEST SYMPOSIUM ON IMAGE ANALYSIS AND INTERPRETATION, 2006, Washington. *Proceedings ...* Washington, DC, USA: IEEE Computer Society, 2006. p. 81–85.
- 6 FAZEKAS, S.; CHETVERIKOV, D. Analysis and performance evaluation of optical flow features for dynamic texture recognition. *Signal Processing: Image Communication*, v. 22, n. 7-8, p. 680–691, 2007.
- 7 CHAN, A. B.; VASCONCELOS, N. Layered dynamic textures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 31, n. 10, p. 1862–1879, 2009.
- 8 JULESZ, B. Visual pattern discrimination. *IRE Transactions on Information Theory*, v. 8, n. 2, p. 84–92, 1962. ISSN 0096-1000.
- 9 JULESZ, B. Textons, the elements of texture perception, and their interactions. *Nature*, v. 290, n. 5802, p. 91–97, 1981. ISSN 0028-0836.
- 10 JULESZ, B. et al. Inability of humans to discriminate between visual textures that agree in second-order statistics-revisited. *Perception*, v. 2, n. 4, p. 391–405, 1973.

- 11 VALOIS, R. L.; ALBRECHT, D. G.; THORELL, L. G. Spatial frequency selectivity of cells in macaque visual cortex. *Vision Research*, v. 22, n. 5, p. 545–559, 1982. ISSN 0042-6989.
- 12 HUBEL, D. H.; WIESEL, T. N. Binocular interaction in striate cortex of kittens reared with artificial squint. *Journal of Neurophysiology*, v. 28, n. 6, p. 1041–1059, 1965. ISSN 0022-3077.
- 13 MALIK, J.; PERONA, P. Preattentive texture discrimination with early vision mechanisms. *Journal of the Optical Society of America, OSA*, v. 7, n. 5, p. 923–932, 1990.
- 14 TAMURA, H.; MORI, S.; YAMAWAKI, T. Textural features corresponding to visual perception. *IEEE Transactions on Systems, Man and Cybernetics*, v. 8, n. 6, p. 460–473, 1978. ISSN 0018-9472.
- 15 SKLANSKY, J. Image segmentation and feature extraction. *IEEE Transactions on Systems, Man and Cybernetics*, v. 8, n. 4, p. 237–247, 1978. ISSN 0018-9472.
- 16 HARALICK, R. M. Statistical and structural approaches to texture. *Proceedings of the IEEE*, v. 67, n. 5, p. 768–804, 1979.
- 17 HAWKINS, J. K. Textural properties for pattern recognition. In: SYMPOSIUM ON PICTURE PROCESSING AND PSYCHOPICTORIS, 1969, Arlington, USA. *Proceedings ...* New York, USA: Academic Press, 1969. p. 347–370.
- 18 CHEN, Y.; DOUGHERTY, E. Gray-scale morphological granulometric texture classification. *Optical Engineering*, v. 33, n. 8, p. 2713–2722, 1994.
- 19 SERRA, J. *Image analysis and mathematical morphology*. Orlando, FL, USA: Academic Press, Inc., 1983. ISBN 0126372403.
- 20 ASANO, A.; MIYAGAWA, M.; FUJIO, M. Texture modelling by optimal gray scale structuring elements using morphological pattern spectrum. In: INTERNATIONAL CONFERENCE ON PATTERN RECOGNITION, 15, 2000, Barcelona. *Proceedings...* Barcelona, Spain: IEEE Computer Society, 2000. v. 3, p. 475–478.
- 21 HANBURY, A.; KANDASWAMY, U.; ADJEROH, D. A. Illumination-invariant morphological texture classification. In: INTERNATIONAL SYMPOSIUM ON MATHEMATICAL MORPHOLOGY: 40 YEARS ON, 7, 2005, Paris, FR. *Proceedings ...* Dordrecht: Springer-Verlag, 2005, (Computational Imaging and Vision, v. 30). p. 377–386.
- 22 ZHANG, J. et al. Local features and kernels for classification of texture and object categories: a comprehensive study. *International Journal of Computer Vision*, v. 73, n. 2, p. 213–238, 2007. ISSN 0920-5691.

- 23 LAZEBNIK, S.; SCHMID, C.; PONCE, J. A sparse texture representation using local affine regions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 27, n. 8, p. 1265–1278, 2005. ISSN 0162-8828.
- 24 TOMITA, F.; TSUJI, S. *Computer analysis of visual textures*. Norwell, MA, USA: Kluwer Academic Publishers, 1990. ISBN 0792391144.
- 25 DAVIS, L. S. Polarograms: a new tool for image texture analysis. *Pattern Recognition*, v. 13, n. 3, p. 219–223, 1981. ISSN 0031-3203.
- 26 OJALA, T.; PIETIKAINEN, M.; MAENPAA, T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 24, n. 7, p. 971–987, 2002. ISSN 0162-8828.
- 27 GUO, Y.; ZHAO, G.; PIETIKÄINEN, M. Discriminative features for texture description. *Pattern Recognition*, v. 45, n. 10, p. 3834 – 3843, 2012. ISSN 0031-3203.
- 28 LEE, T. S. Image representation using 2d gabor wavelets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 18, n. 10, p. 959–971, 1996. ISSN 0162-8828.
- 29 JAIN, A. K.; FARROKHNIYA, F. Unsupervised texture segmentation using gabor filters. *Pattern Recognition*, v. 24, n. 12, p. 1167–1186, 1991.
- 30 SANDLER, R.; LINDENBAUM, M. Optimizing gabor filter design for texture edge detection and classification. *International Journal of Computer Vision*, v. 84, p. 308–324, 2009. ISSN 0920-5691.
- 31 IDRISSE, M.; ACHEROY, M. Texture classification using gabor filters. *Pattern Recognition Letters*, v. 23, n. 9, p. 1095 – 1102, 2002. ISSN 0167-8655.
- 32 SIMONCELLI, E. P.; PORTILLA, J. Texture characterization via joint statistics of wavelet coefficient magnitudes. In: IEEE COMPUTER SOCIETY INTERNATIONAL CONFERENCE ON IMAGE PROCESSING, 1998, Chicago. *Proceedings ...* Chicago, Illinois: IEEE Computer Society, 1998. v. 1, p. 62–66.
- 33 PUN, C.-M.; LEE, M.-C. Log-polar wavelet energy signatures for rotation and scale invariant texture classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 25, p. 590–603, 2003. ISSN 0162-8828.
- 34 KASPARIS, T. et al. Segmentation of textured images based on fractals and image filtering. *Pattern Recognition*, v. 34, n. 10, p. 1963–1973, 2001.
- 35 BACKES, A. R.; BRUNO, O. M. A new approach to estimate fractal dimension of texture images. In: INTERNATIONAL CONFERENCE ON IMAGE AND SIGNAL PROCESSING, 3, 2008, Berlin. *Proceedings ...* Berlin, Heidelberg: Springer-Verlag, 2008. v. 5099, p. 136–143.

- 36 MOSQUERA, A.; CABELLO, D. The markov random fields in functional neighbors as a texture model: applications in texture classification. In: INTERNATIONAL CONFERENCE ON PATTERN RECOGNITION, 13, 1996, Vienna. *Proceedings ...* Vienna, Austria: IEEE Computer Society, 1996. v. 2, p. 815–819.
- 37 KASHYAP, R. L.; KHOTANZAD, A. A model-based method for rotation invariant texture classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 8, n. 4, p. 472–481, 1986. ISSN 0162-8828.
- 38 COHEN, F. S.; FAN, Z.; PATEL, M. A. Classification of rotated and scaled textured images using gaussian markov random field models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 13, n. 2, p. 192–202, 1991. ISSN 0162-8828.
- 39 XU, Y.; JI, H.; FERMÜLLER, C. Viewpoint invariant texture description using fractal analysis. *International Journal of Computer Vision*, v. 83, n. 1, p. 85–100, 2009. ISSN 0920-5691.
- 40 VARMA, M.; GARG, R. Locally invariant fractal features for statistical texture classification. In: IEEE COMPUTER SOCIETY INTERNATIONAL CONFERENCE ON COMPUTER VISION, 11, 2007, Rio de Janeiro. *Proceedings ...* Rio de Janeiro: IEEE Computer Society, 2007. p. 1–8.
- 41 LAI, M.-S.; HUANG, H.-C.; CHU, S.-C. Image texture segmentation with ant colony systems. In: INTERNATIONAL CONFERENCE ON INNOVATIVE COMPUTING, INFORMATION AND CONTROL, 2006, Washington. *Proceedings ...* Washington, DC, USA: IEEE Computer Society, 2006. v. 1, p. 652–656. ISBN 0-7695-2616-0.
- 42 GONÇALVES, W. N.; SILVA, N. R. da; BRUNO, L. da F. C. O. M. Texture recognition based on diffusion in network. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2013. *Submetido*.
- 43 BACKES, A. R. et al. Texture analysis and classification using deterministic tourist walk. *Pattern Recognition*, v. 43, n. 3, p. 685–694, 2010. ISSN 0031-3203.
- 44 BACKES, A. R.; MARTINEZ, A. S.; BRUNO, O. M. Texture analysis using graphs generated by deterministic partially self-avoiding walks. *Pattern Recognition*, v. 44, n. 8, p. 1684 – 1689, 2011. ISSN 0031-3203.
- 45 BACKES, A. R.; MARTINEZ, A. S.; BRUNO, O. M. Texture analysis based on maximum contrast walker. *Pattern Recognition Letters*, v. 31, n. 12, p. 1701–1707, 2010.
- 46 NELSON, R. C.; POLANA, R. Detecting activities. In: IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, 1993, New York. *Proceedings ...* New York, NY: IEEE Computer Society, 1993. p. 815–818.

- 47 POLANA, R.; NELSON, R. C. Temporal texture and activity recognition. In: MOTION-BASED RECOGNITION, 1997, Rochester. *Proceedings ...* Rochester, NY, USA: Springer Netherlands, 1997. p. 87–124.
- 48 FABLET, R.; BOUTHEMY, P. Motion recognition using nonparametric image motion models estimated from temporal and multiscale cooccurrence statistics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 25, n. 12, p. 1619–1624, 2003.
- 49 PÉTERI, R.; CHETVERIKOV, D. Dynamic texture recognition using normal flow and texture regularity. In: IBERIAN CONFERENCE ON PATTERN RECOGNITION AND IMAGE ANALYSIS, 2, 2005, Berlin. *Proceedings ...* Berlin, Heidelberg: Springer-Verlag, 2005. p. 223–230.
- 50 LU, Z. et al. Dynamic texture recognition by spatio-temporal multiresolution histograms. In: IEEE WORKSHOP ON APPLICATION OF COMPUTER VISION, 7, 2005, Breckenridge, USA. *Proceedings ...* Breckenridge, Colorado: IEEE Computer Society, 2005. v. 2, p. 241–246.
- 51 CHEN, J. et al. Automatic dynamic texture segmentation using local descriptors and optical flow. *IEEE Transactions on Image Processing*, v. 22, n. 1, p. 326–339, 2013.
- 52 DUBOIS, S.; PÉTERI, R.; MÉNARD, M. A comparison of wavelet based spatio-temporal decomposition methods for dynamic texture recognition. In: IBERIAN CONFERENCE ON PATTERN RECOGNITION AND IMAGE ANALYSIS, 4, 2009, Berlin. *Proceedings ...* Berlin, Heidelberg: Springer-Verlag, 2009. p. 314–321. ISBN 978-3-642-02171-8.
- 53 ZHONG, H.; SHI, J.; VISONTAI, M. Detecting unusual activity in video. In: IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, 2004, Pittsburgh. *Proceedings ...* Pittsburgh, PA, USA: IEEE Computer Society, 2004. p. 819–826.
- 54 DOLLAR, P. et al. Behavior recognition via sparse spatio-temporal features. In: INTERNATIONAL CONFERENCE ON COMPUTER COMMUNICATIONS AND NETWORKS, 14, 2005, Washington. *Proceedings ...* Washington, DC, USA: IEEE Computer Society, 2005. p. 65–72. ISBN 0-7803-9424-0.
- 55 GONÇALVES, W. N.; MACHADO, B. B.; BRUNO, O. M. Spatiotemporal gabor filters: a new method for dynamic texture recognition. In: WORKSHOP DE VISÃO COMPUTACIONAL, 2011, Curitiba. *Proceedings ...* Curitiba: WVC, 2011. p. 184–189.
- 56 DORETTO, G. et al. Dynamic textures. *International Journal of Computer Vision*, v. 51, n. 2, p. 91–109, 2003.
- 57 CHAN, A. B.; VASCONCELOS, N. Classifying video with kernel dynamic textures. In: IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, 2007, Los Alamitos. *Proceedings ...* Los Alamitos, CA, USA: IEEE Computer Society, 2007. p. 1–6.

58 SZUMMER, M.; PICARD, R. W. Temporal texture modeling. In: IEEE COMPUTER SOCIETY INTERNATIONAL CONFERENCE ON IMAGE PROCESSING, 1996, Cambridge. *Proceedings ...* Cambridge, MA, USA: IEEE Computer Society, 1996. v. 3, p. 823–826.

59 CHAN, A. B.; VASCONCELOS, N. Modeling, clustering, and segmenting video with mixtures of dynamic textures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 30, n. 5, p. 909–926, 2008.

60 RAVICHANDRAN, A.; CHAUDHRY, R.; VIDAL, R. Categorizing dynamic textures using a bag of dynamical systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 35, n. 2, p. 342–353, 2013.

61 OTSUKA, K. et al. Feature extraction of temporal texture based on spatiotemporal motion trajectory. In: INTERNATIONAL CONFERENCE ON PATTERN RECOGNITION, 14, 1998, Brisbane. *Proceedings ...* Brisbane, Qld: IEEE Computer Society, 1998. v. 2, p. 1047–1051. ISSN 1051-4651.

62 ZHONG, J.; SCARLAROFF, S. *Temporal texture recognition model using 3D features*. Cambridge: MIT Media Lab Perceptual Computing, 2002. Technical Report 346.

63 METZLER, R.; KLAFTER, J. The random walk's guide to anomalous diffusion: a fractional dynamics approach. *Physics Reports*, v. 339, n. 1, p. 1–77, 2000.

64 DERRIDA, B. From random walks to spin glasses. *Physica D: Nonlinear Phenomena*, v. 107, n. 2-4, p. 186–198, 1997. ISSN 0167-2789.

65 GALE, D. et al. Further travels with my ant. *Mathematical Intelligencer*, v. 17, p. 48–56, 1995.

66 BUNIMOVICH, L. A.; TROUBETZKOY, S. E. Recurrence properties of lorentz lattice gas cellular automata. *Journal of Statistical Physics*, v. 67, p. 289–302, 1992.

67 BUNIMOVICH, L. A. Deterministic walks in random environments. *Physica D: Nonlinear Phenomena*, v. 187, n. 1, p. 20–29, 2004.

68 CAMPITELI, M. G.; MARTINEZ, A. S.; BRUNO, O. M. An image analysis methodology based on deterministic tourist walks. In: ADVANCES IN ARTIFICIAL INTELLIGENCE, 2006, Ribeirão Preto. *Proceedings ...* Ribeirão Preto: Springer, 2006. p. 159–167.

69 SILVA, T. C.; ZHAO, L. *High level pattern classification via tourist walks in networks*. Disponível em: <<http://arxiv.org/pdf/1305.1679.pdf>>. Acesso em: 10 junho 2013.

70 CAMPITELI, M. G. et al. Deterministic walks as an algorithm of pattern recognition. *Physical Review E*, v. 74, n. 2, p. 026703, 2006.

- 71 KINOUCI, O. et al. Deterministic walks in random networks: an application to thesaurus graphs. *Physica A: Statistical Mechanics and its Applications*, v. 315, n. 3–4, p. 665 – 676, 2002. ISSN 0378-4371.
- 72 RISAU-GUSMAN, S.; MARTINEZ, A. S.; KINOUCI, O. Escaping from cycles through a glass transition. *Physica D: Nonlinear Phenomena*, v. 68, n. 1, p. 016104, 2003.
- 73 TERÇARIOL, C. A.; MARTINEZ, A. S. Analytical results for the statistical distribution related to a memoryless deterministic walk: dimensionality effect and mean-field models. *Physical Review E*, v. 72, n. 1, p. 021103, 2005.
- 74 LIMA, G. F.; MARTINEZ, A. S.; KINOUCI, O. Deterministic walks in random media. *Physical Review Letters*, v. 87, n. 1, p. 010603, 2001.
- 75 STANLEY, H. E.; BULDYREV, S. V. Statistical physics - the salesman and the tourist. *Nature*, v. 413, n. 6854, p. 373–374, 2001.
- 76 FALCONER, K. *Fractal geometry: mathematical foundations and applications*. 2nd. ed. Chinchester, UK: Wiley, 2003. ISBN 0470848626.
- 77 TRICOT, C. *Curves and fractal dimension*. New York, USA: Springer-Verlag, 1995. ISBN 9780387940953.
- 78 GONÇALVES, W. N.; BRUNO, O. M. Combining fractal and deterministic walkers to improve texture analysis and classification. *Pattern Recognition*, v. 46, n. 11, p. 2953 – 2968, 2013. ISSN 0031-3203.
- 79 BRUNO, O. M. et al. Fractal dimension applied to plant identification. *Information Sciences*, v. 178, n. 12, p. 2722–2733, 2008. ISSN 0020-0255.
- 80 DIAL, R. B. Algorithm 360: shortest-path forest with topological ordering. *Communications of the ACM*, v. 12, n. 11, p. 632–633, 1969. ISSN 0001-0782.
- 81 BACKES, A. R.; CASANOVA, D.; BRUNO, O. M. Color texture analysis based on fractal descriptors. *Pattern Recognition*, v. 45, n. 5, p. 1984–1992, 2012.
- 82 BRODATZ, P. *Textures: a photographic album for artists and designers*. New York: Dover Publications, 1966.
- 83 VISTEX: vision texture database. 1995. Maintained by the Vision and Modeling group at the MIT Media Lab. Disponível em: <<http://whitechapel.media.mit.edu/vismod/imagery/VisionTexture/vistex.html>>. Acesso em: 10 junho 2013.

- 84 HARALICK, R. M.; SHANMUGAM, K.; DINSTEN, I. Textural features for image classification. *IEEE Transactions on Systems, Man and Cybernetics*, v. 3, n. 6, p. 610–621, 1973.
- 85 AZENCOTT, R.; WANG, J.-P.; YOUNES, L. Texture classification using windowed Fourier filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 19, n. 2, p. 148–153, 1997. ISSN 0162-8828.
- 86 GONÇALVES, W. N.; BRUNO, O. M. Dynamic texture analysis and segmentation using deterministic partially self-avoiding walks. *Expert Systems with Applications*, v. 40, n. 11, p. 4283 – 4300, 2013. ISSN 0957-4174.
- 87 GONÇALVES, W. N.; SILVA, J. de A.; BRUNO, O. M. A rotation invariant face recognition method based on complex network. In: IBEROAMERICAN CONGRESS ON PROGRESS IN PATTERN RECOGNITION, IMAGE ANALYSIS, COMPUTER VISION, AND APPLICATIONS, 2010, São Paulo. *Proceedings ...* São Paulo: Springer Verlag, 2010, (Lecture Notes in Computer Science, v. 6419). p. 426–433. ISBN 978-3-642-16686-0.
- 88 COMIN, C. H. et al. *Diffusion in directed modular networks*. Disponível em: <<http://arxiv.org/pdf/1202.4047v2.pdf>>. Acesso em: 10 junho 2013.
- 89 COSTA, L. d. F. D.; CESAR JR., R. M. *Shape analysis and classification: theory and practice*. Boca Raton, FL, USA: CRC Press, Inc., 2000. ISBN 0849334934.
- 90 GABOR, D. Theory of communication. *Journal of Institute of Electronic Engineering*, London, v. 93, p. 429–457, 1946.
- 91 PETERI, R.; FAZEKAS, S.; HUISKES, M. Dyntex: a comprehensive database of dynamic textures. *Pattern Recognition Letters*, v. 31, n. 12, p. 1627–1632, 2010.
- 92 TORRE, F. de la; BLACK, M. J. Robust principal component analysis for computer vision. In: IEEE COMPUTER SOCIETY INTERNATIONAL CONFERENCE ON COMPUTER VISION, 2001, Vancouver. *Proceedings ...* Vancouver, BC: IEEE Computer Society, 2001. v. 1, p. 362–369.
- 93 ZHAO, G.; PIETIKAINEN, M. Dynamic texture recognition using local binary patterns with an application to facial expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 29, n. 6, p. 915–928, 2007. ISSN 0162-8828.
- 94 ZHAO, G. Y.; PIETIKAINEN, M. Local binary pattern descriptors for dynamic texture recognition. In: INTERNATIONAL CONFERENCE ON PATTERN RECOGNITION, 2006, Hong Kong. *Proceedings ...* Hong Kong: IEEE Computer Society, 2006. p. 211–214.
- 95 CHAN, A. B.; LIANG, Z. S. J.; VASCONCELOS, N. Privacy preserving crowd monitoring: counting people without people models or tracking. In: IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, 2008, Los Alamitos. *Proceedings ...* Los Alamitos, CA, USA: IEEE Computer Society, 2008. p. 1–7. ISBN 978-1-4244-2242-5.

- 96 VENDRAMIN, L.; CAMPELLO, R. J. G. B.; HRUSCHKA, E. R. Relative clustering validity criteria: a comparative overview. *Statistical Analysis and Data Mining*, v. 3, n. 4, p. 209–235, 2010. ISSN 1932-1864.
- 97 DEMSAR, J. Statistical comparisons of classifiers over multiple data sets. *Journal Machine Learning Research*, v. 7, p. 1–30, 2006. ISSN 1532-4435.
- 98 GHOREYSHI, A.; VIDAL, R. Segmenting dynamic textures with ising descriptors, arx models and level sets. In: EUROPEAN CONFERENCE ON COMPUTER VISION, 2006, Beijing. *Proceedings ...* Beijing, China: Springer-Verlag, 2006. p. 127–141.
- 99 VIDAL, R.; SINGARAJU, D. A closed form solution to direct motion segmentation. In: IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, 2005, San Diego. *Proceedings ...* San Diego, CA, USA: IEEE Computer Society, 2005. v. 2, p. 510–515. ISSN 1063-6919.
- 100 FAZEKAS, S. et al. Dynamic texture detection based on motion analysis. *International Journal of Computer Vision*, v. 82, p. 48–63, 2009. ISSN 0920-5691.
- 101 HOLMGREN, P. K.; HOLMGREN, N. H. *Plant specialists index: index to specialists in the systematics of plants and fungi, based on data from Index herbariorum (herbaria)*. 8th. ed. Königstein, Germany: Koeltz Scientific Books, 1992.
- 102 JUDD, W. S.; CAMPBELL, C. S.; KELLOGG, E. *Plant systematics: a phylogenetic approach with CDROM*. 2nd. ed. Sunderland, MA: Sinauer Associates, 2002.
- 103 CUCCHIARA, R.; PICCARDI, M.; MELLO, P. Image analysis and rule-based reasoning for a traffic monitoring system. *IEEE Transactions on Intelligent Transportation Systems*, v. 1, n. 2, p. 119–130, 2000. ISSN 1524-9050.
- 104 MAURIN, B.; MASOUD, O.; PAPANIKOLOPOULOS, N. Monitoring crowded traffic scenes. In: INTERNATIONAL CONFERENCE ON INTELLIGENT TRANSPORTATION SYSTEMS, 5, 2002, Singapore. *Proceedings ...* Singapore: IEEE Computer Society, 2002. p. 19–24.
- 105 JUNG, Y.-K.; LEE, K.-W.; HO, Y.-S. Content-based event retrieval using semantic scene interpretation for automated traffic surveillance. *IEEE Transactions on Intelligent Transportation Systems*, v. 2, n. 3, p. 151–163, 2001. ISSN 1524-9050.
- 106 CHAN, A.; VASCONCELOS, N. Classification and retrieval of traffic video using autoregressive stochastic processes. In: INTELLIGENT VEHICLES SYMPOSIUM, 2005, Las Vegas. *Proceedings ...* Las Vegas, USA: IEEE Computer Society, 2005. p. 771–776.
- 107 BURSTEIN, G. T.; SHREIR, L. L.; JARMAN, R. A. *Corrosion: corrosion control*. 3rd. ed. Oxford, UK: Butterworth-Heinemann, 1994. ISBN 9780750610773.

- 108 ARZOLA, S.; GENESCÁ, J. The effect of h₂s concentration on the corrosion behavior of api 5l x-70 steel. *Journal of Solid State Electrochemistry*, v. 9, p. 197–200, 2005. ISSN 1432-8488.
- 109 AZEVEDO, C. R. F. Failure analysis of a crude oil pipeline. *Engineering Failure Analysis*, v. 14, n. 6, p. 978–994, 2007. ISSN 1350-6307.
- 110 WU, X. et al. Top 10 algorithms in data mining. *Knowledge and Information Systems*, v. 14, p. 1–37, 2007. ISSN 0219-1377.
- 111 ASTRAHAN, M. M. Speech analysis by clustering or the hyperphome method. In: STANFORD ARTIFICIAL INTELLIGENCE PROJECT MEMORANDUM, 1970, Stanford. *Proceedings ...* Stanford, CA: Stanford University, 1970.
- 112 KATSAVOUNIDIS, I.; JAY, C.; ZHANG, Z. A new initialization technique for generalized lloyd iteration. *Signal Processing Letters*, v. 1, n. 10, p. 144–146, 1994.
- 113 SU, T.; DY, J. A deterministic method for initializing k-means clustering. In: INTERNATIONAL CONFERENCE ON TOOLS WITH ARTIFICIAL INTELLIGENCE, 16, 2004, Boca Raton. *Proceedings ...* Boca Raton, Florida: IEEE Computer Society, 2004. p. 784–786. ISBN 0-7695-2236-X.
- 114 CASANOVA, D. *Identificação de espécies vegetais por meio da análise de textura foliar*. 2008. 113p. Dissertação (Mestrado em Ciência da Computação) - Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2008.

Fundamentação Teórica

Neste apêndice, é apresentada a fundamentação teórica necessária para o entendimento deste trabalho. Para descrever os algoritmos abaixo, considere as seguintes definições da base de dados e distância entre dois vetores.

Definição A.1 A base de dados é descrita como um conjunto $X = \{x_j\}$ composto por N vetores $x_j \in \mathbb{R}^n$. A k -ésima dimensão do vetor é dada por x_{jk} .

Definição A.2 A distância Euclidiana $\|x_j, x_m\|$ entre dois vetores x_j e x_m é dada por:

$$\|x_j, x_m\| = \sqrt{\sum_{k=1}^n (x_{jk} - x_{mk})^2} \quad (\text{A.0.1})$$

A.1 Classificação de Padrões

A classificação de padrões é a tarefa de identificar a qual categoria uma instância pertence, com base em um conjunto de treinamento contendo instâncias cujas categorias são conhecidas. Cada instância é caracterizada por uma dupla (x_j, y_j) , onde x_j é o seu vetor de atributos e $y_j \in Y$ é um atributo especial que representa a categoria ou classe que a instância pertence. Dadas essas definições, uma das tarefas envolvidas na classificação é aprender uma função alvo f que mapeia cada vetor de atributos $x_j \in X$ em uma das classes predefinidas $Y = \{y_i\}$.

Cada algoritmo de classificação emprega um modelo de aprendizagem para identificar a função f que melhor se ajusta ao relacionamento entre os vetores de atributos e as classes.

O modelo deve ser capaz de se ajustar aos dados de treinamento, bem como, prever corretamente as classes de instâncias desconhecidas pelo algoritmo de aprendizagem. Portanto, um ponto chave dos algoritmos de aprendizagem é construir modelos com boa capacidade de generalização.

Basicamente, o processo de classificação é realizado em duas etapas: (i) um *conjunto de treinamento*, que consiste em instâncias com classes conhecidas, é utilizado para construir um modelo de classificação – (ii) o modelo de classificação é utilizado para prever as classes do *conjunto de teste*, um conjunto de instâncias com classes desconhecidas. A avaliação do desempenho de um modelo de classificação é baseada na contagem de instâncias corretamente previstas pelo modelo. Portanto, a taxa de classificação correta é dada pelo número de previsões corretas dividido pelo número total de previsões realizadas pelo modelo.

A.1.1 Validação-cruzada

Dado que o modelo de classificação foi construído, é possível aplicar o *conjunto de teste* para avaliar o classificador. A separação da base de dados X em *conjunto de treinamento* e *conjunto de teste* deve ser realizada com garantias estatísticas. Por exemplo, a escolha aleatória do *conjunto de teste* pode favorecer um modelo de classificação em relação a um outro modelo.

Para garantir que o modelo de classificação seja adequadamente avaliado, uma alternativa muito utilizada é a estratégia de validação cruzada. Nessa estratégia, cada instância é utilizada o mesmo número de vezes para treinamento e exatamente uma vez para teste. Ela separa os dados X em k partições $P_i \subset X \mid 1 \leq i \leq k$. Durante cada execução, uma partição é escolhida como *conjunto de teste* enquanto que as demais $k - 1$ partições são utilizadas como *conjunto de treinamento*. Este procedimento é repetido k vezes de modo que cada partição seja utilizada como *conjunto de teste* exatamente uma vez. Assim, a taxa de classificação correta é a média da taxa de classificação correta de cada uma das k execuções.

A.1.2 K-Vizinhos mais Próximos

O método *k-vizinhos mais próximos* faz parte de um conjunto de métodos de classificação conhecidos como aprendizagem baseada em instâncias. Esses métodos utilizam o conjunto de treinamento para fazer as previsões sem manter qualquer abstração derivada dos dados. Portanto, dada uma instância de teste $(x_m, y_m = ?)$, em que $?$ significa que a classe é desconhecida, e as instâncias de treinamento com classes conhecidas $T = \{(x_j, y_j)\}$, o método calcula as distâncias entre x_m e todas as instâncias de treinamento para determinar sua lista de k vizinhos mais próximos T_k . Uma vez que T_k foi obtida, a instância de teste é classificada com base na maioria das classes de seus vizinhos mais próximos. O Algoritmo 3 apresenta o pseudocódigo de classificação de uma instância de teste.

Para evitar casos de empates, é sugerido que o número de vizinhos mais próximos k seja sempre ímpar. Métodos baseados nessa estratégia não requerem que um modelo seja construído previamente com base no conjunto de treinamento. Devido a esse motivo, a fase de teste pode ser cara computacionalmente, pois é preciso calcular a distância entre a instância de teste e todas as demais instâncias. Por outro lado, nenhum tempo computacional é consumido na construção do modelo.

Algoritmo 3: Algoritmo de classificação k-vizinhos mais próximos.

Input : Conjunto de treinamento $T = \{x_j, y_j\}$, número de vizinhos mais próximos k , instância de teste x_m

Output: Classe predita y_m

```

1 for  $j = 1$  to  $|T|$  do
2    $d_j = \|x_j, x_m\|^2$ ;
3 end
4 Seleciona  $T_k \subset T$ , o conjunto dos  $k$  vizinhos mais próximos da instância de teste com base nas distâncias  $d_j$ ;
5  $y_m = \arg \max_v \sum_{(x_l, y_l) \in T_k} I(v, y_l)$ ; onde  $I(\cdot)$  é uma função que retorna 1 se seus argumentos forem iguais e 0 caso contrário;

```

A.1.3 Máquina de Vetores de Suporte

Um algoritmo que tem recebido uma atenção considerável é a máquina de vetores de suporte (*Support Vector Machine* - SVM). A SVM fornece bons resultados mesmo em dados com alta dimensão por evitar o mau da dimensionalidade. Outro aspecto importante da SVM é que ela representa a fronteira de decisão utilizando um subconjunto de vetores de treinamento, denominados vetores de suporte.

Dado o conjunto de treinamento, composto por duas classes, o objetivo é encontrar uma fronteira de decisão de modo que, todas as instâncias de uma classe fiquem de um lado do hiperplano e todas as instâncias da outra classe fiquem do lado oposto do hiperplano. Devido aos infinitos hiperplanos possíveis, a ideia é escolher o hiperplano que representa a fronteira de decisão com maior margem. Quanto maior a margem, maior a capacidade de generalização do modelo com relação a novos testes.

A.1.3.1 Fronteira de Decisão e Margem

A fronteira de decisão de um classificador linear pode ser descrita como $w \cdot x + b = 0$, onde w e b são parâmetros do modelo. Assim a classe de um vetor x_j pode ser predita de acordo com:

$$y_j = \begin{cases} 1, & \text{se } w \cdot x_j + b > 0 \\ -1, & \text{se } w \cdot x_j + b < 0 \end{cases} \quad (\text{A.1.1})$$

A margem de uma fronteira de decisão pode ser definida pelos vetores de suporte, aqueles vetores mais próximos da fronteira de decisão. Suponha que esses vetores são definidos como x_1 e x_2 para as duas classes respectivamente, assim sabe-se que:

$$\begin{aligned} w \cdot x_1 + b &= 1 \\ w \cdot x_2 + b &= -1 \end{aligned} \quad (\text{A.1.2})$$

Por meio da subtração das duas equações temos que $w(x_1 - x_2) = 2$. A distância entre os dois vetores $x_1 - x_2$ é justamente a margem que separa as duas classes. Portanto podemos reescrever como $\|w\|d = 2$ e assim a margem d é dada por:

$$d = \frac{2}{\|w\|} \quad (\text{A.1.3})$$

A.1.3.2 Estimando uma SVM Linear

A fase de treinamento da SVM envolve a estimação dos parâmetros w e b da fronteira de decisão do *conjunto de treinamento*. Esses parâmetros devem ser escolhidos de forma que as duas condições a seguir sejam satisfeitas:

$$\begin{aligned} w \cdot x_j + b &\geq 1 & \text{se } y_j = 1 \\ w \cdot x_j + b &\leq -1 & \text{se } y_j = -1 \end{aligned} \quad (\text{A.1.4})$$

Ambas as desigualdades podem ser sumarizadas na forma compacta:

$$y_j(w \cdot x_j + b) \geq 1, \quad j = 1 \dots N \quad (\text{A.1.5})$$

A ideia da SVM é que a margem da fronteira de decisão seja máxima. Maximizar a margem é equivalente a minimizar a seguinte função objetivo:

$$\min_w = \frac{\|w\|^2}{2} \quad (\text{A.1.6})$$

sujeito às restrições:

$$y_j(w \cdot x_j + b) \geq 1, \quad j = 1 \dots N \quad (\text{A.1.7})$$

Dado que a função objetivo é quadrática e as restrições são lineares nos parâmetros w e b , a minimização pode ser resolvida pelo método de multiplicadores de Lagrange e os parâmetros w e b são obtidos. Em problemas reais, usa-se uma transformação kernel nos vetores x_j para

transformá-los de um espaço não linear para um espaço linearmente separável.

A.2 Agrupamento de Dados

O agrupamento de dados consiste em dividir os dados em grupos que capturam a estrutura natural, sem a utilização de qualquer informação sobre a classe. Os grupos de objetos, que compartilham características em comum, desempenham um papel importante na forma como as pessoas analisam e descrevem o mundo. De fato, os seres humanos tendem a dividir os objetos em grupos, como por exemplo, os grupos de carros (diversas marcas, modelos), plantas (diferentes espécies), animais, entre outros.

Portanto, o objetivo do agrupamento de dados é que, os objetos do mesmo grupo sejam similares (ou relacionados) uns com os outros e diferentes (ou não relacionados com) os objetos em outros grupos. Quanto maior a similaridade dentro do grupo e maior a dissimilaridade entre os grupos, melhor ou mais distinto é o agrupamento dos dados. Na seção seguinte, é descrito o algoritmo k-means para agrupamento de dados.

A.2.1 K-means

O algoritmo k-means é um dos algoritmos mais utilizado para agrupamento devido a sua simplicidade de implementação, seu desempenho e seu baixo custo computacional. Recentemente, esse algoritmo foi eleito um dos 10 algoritmos mais utilizados em mineração de dados (110).

Para descrição do algoritmo k-means, considere um conjunto de vetores $X = \{x_j\}$ a ser

agrupado e que k centroides iniciais $\bar{x}_1, \bar{x}_2, \dots, \bar{x}_k$ foram definidos, onde k é o número de grupos desejado. Basicamente o algoritmo pode ser descrito em dois passos:

- Cada vetor $x_j \in X$ é atribuído ao centroide mais próximo por meio da distância Euclidiana. Um grupo $C_i \subset X, 1 \leq i \leq k$ para cada centroide é então definido.
- O centroide de cada grupo é atualizado com base nos vetores atribuídos a ele.

Os dois passos de atribuição e atualização são repetidos até que não haja mudança nos grupos. O algoritmo k-means é formalmente descrito no Algoritmo 4.

Algoritmo 4: Algoritmo k-means.

Input : $X = \{x_j\}$, número de grupos k

Output: Grupos $C_1, C_2, \dots, C_k \mid C_i \subset X, C_1 \cup \dots \cup C_k = X, C_1 \cap \dots \cap C_k = \emptyset$

- 1 Inicializa cada centroide \bar{x}_i ;
 - 2 **repeat**
 - 3 $d_j^i = \|x_j, \bar{x}_i\|^2 \mid 1 \leq j \leq N, 1 \leq i \leq k$;
 - 4 $n_j = \arg \min_{1 \leq i \leq k} d_j^i$;
 - 5 Atribuir o vetor x_j ao grupo C_{n_j} ;
 - 6 Atualizar os centroides $\bar{x}_i = \frac{1}{|C_i|} \sum_{x_m \in C_i} x_m$;
 - 7 **until** Nenhuma mudança nos grupos C_i ;
-

A.2.2 Métodos de Inicialização do K-means

Embora o k-means seja extensivamente utilizado pela comunidade de reconhecimento de padrões e visão computacional, a sua solução depende fortemente da escolha inicial dos centroides (linha 1 do Algoritmo 4). Devido à escolha inadequada dos centroides iniciais, a solução fornecida pelo k-means pode convergir para uma solução local cuja adequabilidade é significativamente inferior ao ótimo global. Diversas heurísticas têm sido propostas com o objetivo de inicializar o k-means de forma eficaz. As seções seguintes descrevem os seis principais métodos de inicialização do k-means que visam solucionar esse problema.

- Aleatório: apesar das muitas limitações, o método *aleatório* é um dos métodos mais utilizados devido a sua simplicidade e baixo custo computacional. Neste método, k

vetores da base de dados são aleatoriamente escolhidos como centroides, sendo que cada vetor possui a mesma probabilidade de ser escolhido. Esse método geralmente fornece uma solução local e a cada execução do algoritmo, uma solução diferente é obtida devido à escolha aleatória. O pseudocódigo desse método é apresentado no Algoritmo 5, onde $rand(.)$ aleatoriamente retorna um vetor da base de dados utilizando distribuição uniforme.

Algoritmo 5: Método aleatório.

Input : Base de dados $X = \{x_j\}$, número de grupos k

Output: $\bar{x}_1, \bar{x}_2, \dots, \bar{x}_k$

```

1 for  $i=1$  to  $k$  do
2   |  $\bar{x}_i = rand(X)$ ;
3 end

```

- Astrahan: Astrahan (111) propôs um método de inicialização baseado em densidade. O pseudocódigo do método é apresentado no Algoritmo 6. Inicialmente, um raio de vizinhança é definido como a média das distâncias entre todos os pares de vetores (linha 1). Em seguida, a densidade de cada vetor é calculada pela contagem do número de vetores que estão a uma distância menor ou igual a r . O vetor com maior densidade é escolhido como o primeiro centroide. Os demais $k - 1$ centroides são escolhidos em ordem decrescente de densidade, desde que eles estejam a uma distância r de todos os outros centroides já definidos.

Algoritmo 6: Método Astrahan.

Input : Base de dados $X = \{x_j\}$, número de grupos k

Output: $\bar{x}_1, \bar{x}_2, \dots, \bar{x}_k$

```

1  $r = \frac{1}{N(N-1)} \sum_{i=1}^{N-1} \sum_{j=i+1}^N \|x_j, x_i\|^2$ ;
2 for  $j=1$  to  $N$  do
3   |  $\delta_j = \sum_{i=1, i \neq j}^N \begin{cases} 1, & \text{se } \|x_j, x_i\|^2 \leq r \\ 0, & \text{caso contrário} \end{cases}$ ;
4 end
5  $\bar{x}_1 = \arg \max_{j=1}^N \delta_j$ ;
6 Os demais  $k - 1$  centroides são escolhidos em ordem decrescente de densidade desde que eles estejam a uma distância  $r$  dos centroides já escolhidos;

```

- KKZ: uma estratégia baseada em distâncias pareadas foi proposta por Katsavounidis (112). A ideia desse método é selecionar os vetores que estão mais distantes uns dos outros, seguindo a premissa de que esses vetores provavelmente pertencem a grupos diferentes. O pseudocódigo é apresentado no Algoritmo 7. O primeiro centroide é

definido como o vetor com a maior norma ($\bar{x}_1 = \arg \max |x_j|$, onde $|\cdot|$ é a norma do vetor). Em seguida, é calculada a menor distância d_j entre cada vetor x_j e os centroides já definidos. O vetor x_j com a maior distância d_j é escolhido como o próximo centroide. Os passos anteriores são repetidos até que o número de centroides já definidos seja igual a k .

Algoritmo 7: Método KKZ proposto por Katsavounidis.

Input : Base de dados $X = \{x_j\}$, número de grupos k

Output: $\bar{x}_1, \bar{x}_2, \dots, \bar{x}_k$

```

1  $\bar{x}_1 = \arg \max |x_j|$ , onde  $|\cdot|$  é a norma do vetor;
2 for  $i=2$  to  $k$  do
3    $d_j = \min_{m=1}^{i-1} \|x_j, \bar{x}_m\|$ ;
4    $\bar{x}_i = \arg \max d_j$ ;
5 end

```

- Múltiplas Execuções do K-means: um método comumente empregado e que apresenta resultados interessantes é inicializar o k-means M vezes de forma aleatória e selecionar os centroides que fornecem a melhor partição baseada em uma medida interna. A silhueta foi escolhida como medida interna, pois fornece uma alta correlação com a partição ótima dos dados (96). O pseudocódigo do método de múltiplas execuções do k-means é apresentado no Algoritmo 8 e a silhueta é apresentada na próxima seção.

Algoritmo 8: Múltiplas execuções do k-means.

Input : Base de dados $X = \{x_j\}$, número de grupos k

Output: $\bar{x}_1, \bar{x}_2, \dots, \bar{x}_k$

```

1 for  $j=1$  to  $M$  do
2   for  $i=1$  to  $k$  do
3      $\bar{x}_i^j = \text{rand}(X)$ ;
4   end
5    $s^j = \text{silhouette}(\text{kmeans}(\bar{x}_i^j, X))$ 
6 end
7  $m = \arg \max_j s^j$ ;
8  $\bar{x}_i = \bar{x}_i^m$ 

```

- PCA: o método PCA (113) (*Principal Component Analysis*) é baseado na aplicação iterativa da análise de componentes principais para dividir a base de dados em grupos. Inicialmente, todos os vetores pertencem ao mesmo grupo. Em seguida, é escolhido o grupo C_i (no início apenas um grupo está disponível) com a maior variância intra-grupo para realizar a divisão. Na divisão do grupo, os vetores $x_j \in C_i$ são projetados em direção da primeira componente λ obtida pela análise de componentes principais. Os

vetores projetados são denominados y_j . Então, C_i é dividido em dois subgrupos $C_i^{(1)}$ e $C_i^{(2)}$ de acordo com a regra: se $y_j \leq \bar{y}$, então $x_j \in C_i^{(1)}$; caso contrário, $x_j \in C_i^{(2)}$. Os passos acima são repetidos até que o número de grupos seja igual à k . O pseudocódigo é apresentado no Algoritmo 9.

Algoritmo 9: Método de inicialização baseado em PCA.

Input : Base de dados $X = \{x_j\}$, número de grupos k

Output: $\bar{x}_1, \bar{x}_2, \dots, \bar{x}_k$

```

1  $C_1 = X$ ;
2 for  $l=1$  to  $k - 1$  do
3    $C_i = \arg \max Var(C_m) \mid 1 \leq m < i$ ;
4    $\lambda = PCA(C_i)$ ;
5    $y_j = x_j \lambda^T \mid x_j \in C_i$ ;
6    $\bar{y} = \frac{1}{|C_i|} \sum_{y_j \in C_i} y_j$ ;
7   for  $y_j \in C_i$  do
8     if  $y_j \leq \bar{y}$  then  $x_j \in C_i^{(1)}$ ;
9     else  $x_j \in C_i^{(2)}$ ;
10  end
11   $C_i = C_i^{(1)}$ ;
12   $C_{l+1} = C_i^{(2)}$ ;
13 end
14  $\bar{x}_i = \frac{1}{|C_i|} \sum_{x_j \in C_i} x_j$ ;

```

- Inicialização Manual: em diversas aplicações, incluindo a segmentação de imagens, é comum que uma divisão semi-supervisionada seja executada. Assim, o usuário manualmente divide as regiões e o algoritmo utiliza essa divisão inicial para refinar a solução. Neste trabalho, as bases de dados fornecem uma segmentação inicial grosseira que é utilizada para inicializar o k-means nesse método. O pseudocódigo é apresentado no Algoritmo 10.

Algoritmo 10: Método de inicialização manual.

Input : Base de dados $X = \{x_j\}$, número de grupos k , regiões inicializadas manualmente R_1, \dots, R_k

Output: $\bar{x}_1, \bar{x}_2, \dots, \bar{x}_k$

```

1 for  $i=1$  to  $k$  do
2    $\bar{x}_i = \frac{1}{|R_i|} \sum_{x_j \in R_i} x_j$ ;
3 end

```

A.2.3 Validação do Agrupamento

Em classificação supervisionada, a validação do resultado dos classificadores é parte do processo de desenvolvimento de um modelo e existem medidas de avaliação amplamente aceitas, como a taxa de classificação correta. Embora seja menos desenvolvida, a validação de agrupamentos é uma etapa importante do processo de agrupamento. Nesta seção, são descritas duas medidas de avaliação comumente empregadas pela comunidade.

A.2.3.1 Rand index

O *rand index* é uma medida de similaridade entre dois agrupamentos. Portanto, esta medida compara o resultado do algoritmo de agrupamento com um agrupamento externo previamente conhecido e tido como o agrupamento ideal. Dada a base de dados X e dois agrupamentos a serem comparados, $S = \{C_{s_1}, \dots, C_{s_{k_s}} \mid C_{s_i} \subset X\}$ e o agrupamento ideal $T = \{C_{t_1}, \dots, C_{t_{k_t}} \mid C_{t_i} \subset X\}$, considere as seguintes definições:

- a , o número de pares de elementos em X que estão no mesmo grupo em S e no mesmo grupo em T .

$$a = |Y|, Y = \{(x_i, x_j) \mid x_i, x_j \in C_{s_1}, x_i, x_j \in C_{t_1}\} \quad (\text{A.2.1})$$

- b , o número de pares de elementos em X que estão em grupos diferentes em S e em grupos diferentes em T .

$$b = |Y|, Y = \{(x_i, x_j) \mid x_i \in C_{s_1}, x_j \in C_{s_2}, x_i \in C_{t_1}, x_j \in C_{t_2}\} \quad (\text{A.2.2})$$

- c , o número de pares de elementos em X que estão no mesmo grupo em S e em grupos diferentes em T .

$$c = |Y|, Y = \{(x_i, x_j) \mid x_i, x_j \in C_{s_1}, x_i \in C_{t_1}, x_j \in C_{t_2}\} \quad (\text{A.2.3})$$

- d , o número de pares de elementos em X que estão em grupos diferentes em S e no

mesmo grupo em T .

$$d = |Y|, Y = \{(x_i, x_j) \mid x_i \in C_{s_1}, x_j \in C_{s_2}, x_i, x_j \in C_{t_1}\} \quad (\text{A.2.4})$$

onde $1 \leq i, j \leq N, i \neq j, 1 \leq s_1, s_2 \leq k_s, s_1 \neq s_2, 1 \leq t_1, t_2 \leq k_t, t_1 \neq t_2$.

Dadas as definições acima, o *rand index* R é calculado como:

$$R = \frac{a + b}{a + b + c + d} \quad (\text{A.2.5})$$

Assim, o *rand index* retorna valores entre 0 e 1, com 0 indicando que os dois agrupamento estão em total desacordo e 1 indicando que os agrupamento são exatamente os mesmos.

A.2.3.2 Coeficiente de Silhueta

O coeficiente de silhueta combina a coesão interna de cada grupo C_i e a separação entre os grupos. Essa medida é adequada para avaliar a solução de agrupamento sem o conhecimento prévio do agrupamento ideal. Os três passos seguintes descrevem como calcular o coeficiente de silhueta.

1. Para cada vetor x_j , calcule a sua distância Euclidiana média para todos os outros vetores do seu grupo C_i . Denominamos essa distância como $a_{x_j} = \frac{1}{|C_i|} \sum_{x_m \in C_i} \|x_j, x_m\|^2$
2. Para cada vetor x_j , calcule a sua distância Euclidiana média para os objetos dos outros grupos $C_l \mid C_l \neq C_i$. Encontre o valor mínimo com respeito a todos os grupos C_l . Denominamos essa distância como $b_{x_j} = \min_{C_l, C_l \neq C_i} d_{C_l}^{x_j} \mid d_{C_l}^{x_j} = \frac{1}{|C_l|} \sum_{x_m \in C_l} \|x_j, x_m\|^2$
3. Para cada vetor x_j , seu valor de silhueta é dado por $s_{x_j} = (b_{x_j} - a_{x_j}) / \max(b_{x_j}, a_{x_j})$

O valor de silhueta para cada vetor pode variar de -1 até 1 . Os valores negativos são indesejáveis, pois correspondem ao caso em que a_{x_j} , a distância média de vetores dentro do grupo, é maior que b_{x_j} , a distância média para vetores do grupo mais próximo. Portanto, valores positivos são desejáveis para silhueta, casos em que $a_{x_j} < b_{x_j}$. O valor final de silhueta de um agrupamento é dado pela média da silhueta de cada vetor x_j :

$$s = \frac{1}{N} \sum_{x_j \in X} s_{x_j} \quad (\text{A.2.6})$$