



ANTÔNIO JOSÉ DE LIMA BATISTA

**A CAMINHADA DETERMINÍSTICA DO
TURISTA PARA A EXTRAÇÃO DE
CARACTERÍSTICAS EM IMAGENS DE
CANA DE AÇÚCAR**

LAVRAS – MG

2013

ANTÔNIO JOSÉ DE LIMA BATISTA

**A CAMINHADA DETERMINÍSTICA DO TURISTA PARA A EXTRAÇÃO
DE CARACTERÍSTICAS EM IMAGENS DE CANA DE AÇÚCAR**

Monografia de graduação apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras como parte das exigências do curso de Ciência da Computação para obtenção do título de Bacharel em Ciência da Computação.

Orientador

Dr. Cristiano Leite de Castro

LAVRAS – MG

2013

ANTÔNIO JOSÉ DE LIMA BATISTA

**A CAMINHADA DETERMINÍSTICA DO TURISTA PARA A EXTRAÇÃO
DE CARACTERÍSTICAS EM IMAGENS DE CANA DE AÇÚCAR**

Monografia de graduação apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras como parte das exigências do curso de Ciência da Computação para obtenção do título de Bacharel em Ciência da Computação.

APROVADA em 21 de Agosto de 2013.


Dr. Wilian Soares Lacerda

UFLA


Me. Gabriel Chaves Afonso Coutinho


Dr. Cristiano Leite de Castro

Orientador

LAVRAS – MG

2013

*A minha mãe, Deodata e ao meu pai, Antônio,
pela vida, pelos ensinamentos, apoio e amor.*

*Aos meus irmãos, Antônio Carlos, Ana Gabriela e Ana Paula,
pelo apoio, amizade e amor.*

A minha pequena sobrinha Cecília.

*Aos amigos Antônio Gustavo e Fabrício Torres,
que também fazem parte desta família.*

DEDICO

AGRADECIMENTOS

Agradeço a minha família, por me apoiar durante essa caminhada, por acreditarem em mim, pelo carinho, amizade e amor.

Ao professor Cristiano Leite por todos os anos em que trabalhamos juntos e que me fez despertar o prazer pela ciência e pesquisa. Também agradeço por suas orientações que me ajudaram na construção deste trabalho.

Aos amigos, pelos momentos felizes, pelos conselhos e apoio durante os difíceis.

Aos professores da UFLA, especialmente aos professores do Departamento de Ciência da Computação, que contribuíram para a construção do meu conhecimento.

RESUMO

A Caminhada Determinística do Turista (CDT) é um método capaz de explorar um conjunto de dados. A quantidade de informações, provenientes da exploração desse conjunto, depende de aspectos pré-determinados, tais como a memória do turista, a função determinística de caminhada e a disposição dos dados no ambiente modelado. Neste trabalho, objetivou-se modelar a CDT para análise de texturas de imagens de cortes, manuais e mecânicos, de canas de açúcar. Foram exploradas estratégias para a extração das características das informações obtidas pela CDT, com o objetivo de testar sua discriminação diante de classificadores baseados em Máquinas de Vetores de Suporte. Os resultados obtidos permitiram concluir que a CDT é um método promissor para análise de texturas.

Palavras-Chave: Caminhada Determinística do Turista, Máquinas de Vetores de Suporte, Análise de Textura.

ABSTRACT

Deterministic Tourist Walk (DTW) is a method to explore a set of data. The amount of information extracted from a studied dataset depends on predetermined aspects such as memory tourist, walk deterministic function and arrangement of data in the modeled environment. In this work the main goal was to model the DTW for analyzing textures pictures of cuts, manual and mechanical, of the sugar cane. We explored strategies for getting features from DTW information in order to test the differentiation based on classifiers Support Vector Machines. The results showed that the DTW is a promising method in texture analysis.

Keywords: Deterministic Tourist Walk, Support Vector Machines, Texture Analysis

SUMÁRIO

1	Introdução	18
1.1	Contextualização	18
1.2	Objetivos Gerais	20
1.3	Objetivos Específicos	21
1.4	Organização do Texto	21
2	Referencial Teórico	23
2.1	Imagens Digitais	23
2.2	Análise de Texturas	24
2.2.1	Abordagem Descritiva	26
2.2.1.1	Métodos Estatísticos	27
2.2.1.2	Métodos Espectrais	27
2.2.2	Abordagem Genérica	28
2.2.2.1	Métodos Sintéticos	28
2.2.2.2	Métodos Probabilísticos	28
2.3	Máquinas de Vetores de Suporte	29
2.4	Caminhadas Determinísticas	31
2.5	A Caminhada Determinística do Turista	32
2.5.1	A Memória do Turista	35
2.5.2	Condições de Contorno	36

2.5.3	Algoritmo para a Caminhada do Turista	37
2.5.4	Vetores de Características	39
2.5.4.1	Região	40
2.5.4.2	Histograma	41
2.5.4.3	Propriedades Estatísticas	42
2.5.4.4	Regiões de Diferentes Memórias	44
2.5.5	Estado da Arte	44
3	Metodologia	46
3.1	Aspectos e Parâmetros da Caminhada do Turista	46
3.1.1	Relação de Vizinhança	47
3.1.2	Função Determinística de Caminhada	48
3.1.3	Valores utilizados para a Memória	49
3.1.4	Condições de contorno	50
3.1.5	Critério de Desempate	51
3.1.6	Estratégia de Detecção de Atratores	52
3.1.7	Estratégias utilizadas para a extração de características	54
3.2	Bases de Dados	56
3.2.1	Outex	56
3.2.2	BTexCA	57
3.3	Modelos de Classificação	59
3.4	Configurações de Hardware e Softwares utilizados	60

4	Resultados e Discussões	61
4.1	Base Outex	61
4.1.1	Estratégia por Região	61
4.1.2	Estratégia por Histograma	63
4.1.3	Estratégia por Regiões de Diferentes Memórias	64
4.1.4	Estratégia utilizando Propriedades Estatísticas	66
4.2	Base BTexCA	70
4.2.1	Estratégia por Região	70
4.2.2	Estratégia por Histograma	74
4.2.3	Estratégia por Regiões de Diferentes Memórias	78
4.2.4	Estratégia utilizando Propriedades Estatísticas	81
5	Conclusão	85
A	Apêndice	92
A.1	Estratégia por Região	92
A.2	Estratégia por Histograma	93
A.3	Estratégia por Regiões de Diferentes Memórias	94
A.4	Estratégia utilizando as Propriedades Estatísticas	95

LISTA DE FIGURAS

1.1	Tipos de cortes de canas de açúcar.	19
2.1	Relação de vizinhança entre os <i>pixels</i>	24
2.2	Texturas obtidas a partir de cenas naturais.	25
2.3	Análise de textura com suas abordagens e métodos.	27
2.4	Ilustração da superfície ótima para padrões linearmente separáveis. . .	30
2.5	Ilustração do espaço de entrada e do espaço de característica.	30
2.6	Um exemplo de trajetória e do comportamento da memória do turista.	33
2.7	Um exemplo de trajetória do turista. Os quadrados em amarelo representam o tempo transiente, os quadrados azuis o período atrator. . .	34
2.8	Condições abertas e periódicas de contorno.	37
2.9	Distribuição conjunta (b) resultante da aplicação da CDT sobre a imagem em (a) retirada de http://sipi.usc.edu/database . Acessado em 20 de Julho de 2013).	39
2.10	Estratégia de extração de características por região. Em cinza uma parte da distribuição conjunta escolhida para compor o vetor de características. Cada bloco em cinza representa uma característica em ψ	41
2.11	Estratégia de extração de características por Histograma. Cada conjunto de cor representa uma característica do vetor ψ	42
2.12	Estratégia de extração de características por regiões de diferentes memórias. Em cinza uma parte da distribuição conjunta correspondente à uma memória μ	44

3.1	Considerando uma memória $\mu = 3$, ambas as figuras, (a) e (b), apresentam os melhores casos para a formação de um atrator.	47
3.2	A caminhada do turista foi aplicada a imagem da esquerda. As trajetórias (tempo transiente + período atrator) para diferentes valores de memórias foram sobrepostas e são mostradas na imagem à direita. Figura retirada de Campiteli (2007)	49
3.3	Uma imagem com os níveis de cinza associados a cada <i>pixel</i> , representando um caso em que a memória é maior que o número de vizinhos do <i>pixel</i> atual, fazendo com que o turista fique totalmente preso. A trajetória em vermelho indica o caminho percorrido pela turista.	50
3.4	O critério de desempate buscando a primeira cidade no sentido horário é variante a rotação.	52
3.5	Listas auxiliares para a detecção de atratores. A lista a esquerda armazena cada <i>pixel</i> e o instante em que ele foi visitado. A lista a direita mantém todos os <i>pixels</i> na ordem em que foram visitados. Um atrator é formado, sendo composto pelas cidades 13, 11 e 12. .	53
3.6	Uma amostra de cada uma das classes retiradas da base <i>Outex</i>	57
3.7	Etapas realizadas para a construção da base <i>BTexCA</i>	58
3.8	Uma amostra de cada uma das classes retiradas da base <i>BTexCA</i>	59
4.1	Porcentagem de acertos para cada uma das propriedades estatísticas sobre a distribuição conjunta obtida para a memória do turista igual a 3, sobre a base de dados <i>Outex</i> . Em (a) a acurácia dos modelos SVMs de <i>kernel linear</i> . Em (b) a acurácia dos modelos SVMs de <i>kernel RBF</i>	67

4.2	Parte da distribuição conjunta de cada uma das classes obtida para a memória do turista igual a 3 ($t_0 = 0, p_0 = 0, t_f = 20$ e $p_f = 20$).	68
4.3	Representação espacial entre as classes utilizadas pela bases de dados <i>Outex</i> . A imagem mostra as superfícies ótimas de separações para um modelo SVM com <i>kernel linear</i> , com e sem o uso da validação cruzada. X_1 e X_2 representam, respectivamente, δ_1 e δ_4 .	69
4.4	Representação Espacial entre as classes utilizadas pela bases de dados <i>Outex</i> . A imagem mostra as superfícies ótimas de separações para um modelo SVM com <i>kernel RBF</i> , com e sem o uso da validação cruzada. X_1 e X_2 representam, respectivamente, δ_1 e δ_4 .	70
4.5	Em (a) a porcentagem de acertos obtidos pelos modelos com validação cruzada para cada um dos intervalos. Em (b) o ganho médio em porcentagem obtido pelo uso da validação cruzada.	72
4.6	Em (a) a porcentagem de acertos obtidos pelos modelos com validação cruzada para cada um dos intervalos. Em (b) o ganho médio em porcentagem obtido pelo uso da validação cruzada.	73
4.7	Em (a) a porcentagem de acertos obtidos pelos modelos com validação cruzada para cada tamanho de caminhada ($t + p$). Em (b) o ganho médio em porcentagem obtido pelo uso da validação cruzada.	75
4.8	Em (a) a porcentagem de acertos obtidos pelos modelos com validação cruzada para cada tamanho de caminhada ($t + p$). Em (b) o ganho médio em porcentagem obtido pelo uso da validação cruzada.	76

4.9	Distribuição espacial das duas classes, corte manual e corte mecânico. A construção dessa figura foi possível por meio da estratégia por histograma considerando caminhadas de tamanho $\mu + 2$, onde apenas duas características para cada amostra dos dados são consideradas.	77
4.10	Em (a) a porcentagem de acertos obtidos pelos modelos com validação cruzada para cada tamanho de caminhada ($t + p$). Em (b) o ganho médio em porcentagem obtido pelo uso da validação cruzada. . . .	79
4.11	Em (a) a porcentagem de acertos obtidos pelos modelos com validação cruzada para cada tamanho de caminhada ($t + p$). Em (b) o ganho médio em porcentagem obtido pelo uso da validação cruzada. . . .	80
4.13	Porcentagem de acertos para cada uma das propriedades estatísticas sobre a distribuição conjunta para a memória do turista igual a 2. Em (a) a porcentagem de acertos obtidos pelos modelos SVMs de <i>kernel linear</i> . Em (b) a porcentagem de acertos obtidos pelos modelos SVMs de <i>kernel RBF</i>	81
4.14	Representação espacial entre as classes utilizadas pela bases de dados <i>BTexCA</i> . A imagem mostra as superfícies ótimas de separações para um modelo SVM com <i>kernel linear</i> , com e sem o uso da validação cruzada. X_1 e X_2 representam, respectivamente, δ_1 e δ_4	83
4.15	Representação Espacial entre as classes utilizadas pela bases de dados <i>BTexCA</i> . A imagem mostra as superfícies ótimas de separações para um modelo SVM com <i>kernel RBF</i> , com e sem o uso da validação cruzada. X_1 e X_2 representam, respectivamente, δ_1 e δ_4	83
4.12	Parte da distribuição conjunta de cada uma das classes obtida para a memória do turista igual a 2 ($t_0 = 0, p_0 = 0, t_f = 20$ e $p_f = 20$). . .	84

A.1	Porcentagem de acertos para cada uma das propriedades estatísticas sobre a distribuição conjunta obtida para a memória do turista igual a 3, sobre a base de dados <i>Outex</i>	95
-----	---	----

LISTA DE TABELAS

4.1	Porcentagem de acerto dos modelos SVM com <i>kernel linear</i> sobre a base de dados <i>Outex</i> , utilizando a estratégia de extração de características por Região.	62
4.2	Porcentagem de acerto dos modelos SVM com <i>kernel RBF</i> sobre a base de dados <i>Outex</i> , utilizando a estratégia de extração de características por Região.	62
4.3	Porcentagem de acerto do modelo SVM com <i>kernel linear</i> sobre a base de dados <i>Outex</i> , utilizando a estratégia de extração de características por Histograma.	63
4.4	Porcentagem de acerto do modelo SVM com <i>kernel RBF</i> sobre a base de dados <i>Outex</i> , utilizando a estratégia de extração de características por Histograma.	64
4.5	Porcentagem de acerto do modelo SVM com <i>kernel linear</i> sobre a base de dados <i>Outex</i> , utilizando a estratégia de extração de características por Região de Diferentes Memórias.	65
4.6	Porcentagem de acerto do modelo SVM com <i>kernel RBF</i> sobre a base de dados <i>Outex</i> , utilizando a estratégia de extração de características por Região de Diferentes Memórias.	65
4.7	Porcentagem de acerto dos modelos SVM com <i>kernel linear</i> sobre a base de dados <i>BTexCA</i> , utilizando a estratégia de extração de características por Região.	71
4.8	Porcentagem de acerto dos modelos SVMs com <i>kernel RBF</i> sobre a base de dados <i>BTexCA</i> , utilizando a estratégia de extração de características por Região.	73

4.9	Porcentagem de acerto do modelo SVM de <i>kernel linear</i> sobre a base de dados <i>BTexCA</i> , utilizando a estratégia de extração de características por Histograma.	74
4.10	Porcentagem de acerto do modelo SVM de <i>kernel RBF</i> sobre a base de dados <i>BTexCA</i> , utilizando a estratégia de extração de características por Histograma.	75
4.11	Porcentagem de acerto do modelo SVM com <i>kernel linear</i> sobre a base de dados <i>BTexCA</i> , utilizando a estratégia de extração de características por Região.	79
4.12	Porcentagem de acerto do modelo SVM com <i>kernel RBF</i> sobre a base de dados <i>BTexCA</i> , utilizando a estratégia de extração de características por Região.	80
4.13	Tabela com os melhores resultados para cada uma das estratégias. . .	82
A.1	Porcentagem de acerto dos modelos SVM com <i>kernel linear</i> sobre a base de dados <i>Outex</i> , utilizando a estratégia de extração de características por Região.	92
A.2	Porcentagem de acerto dos modelos SVM com <i>kernel RBF</i> sobre a base de dados <i>Outex</i> , utilizando a estratégia de extração de características por Região.	93
A.3	Porcentagem de acerto do modelo SVM com <i>kernel linear</i> sobre a base de dados <i>Outex</i> , utilizando a estratégia de extração de características por Histograma.	93
A.4	Porcentagem de acerto do modelo SVM com <i>kernel RBF</i> sobre a base de dados <i>Outex</i> , utilizando a estratégia de extração de características por Histograma.	94

A.5	Porcentagem de acerto do modelo SVM com <i>kernel linear</i> sobre a base de dados <i>Outex</i> , utilizando a estratégia de extração de características por Região de Diferentes Memórias.	94
A.6	Porcentagem de acerto do modelo SVM com <i>kernel RBF</i> sobre a base de dados <i>Outex</i> , utilizando a estratégia de extração de características por Região de Diferentes Memórias.	95

1 INTRODUÇÃO

Neste capítulo, será feita uma breve contextualização do problema, dos objetivos gerais e específicos, tratados neste trabalho.

1.1 Contextualização

A automatização de processos mecânicos abrange diversos setores industriais, dentre eles o setor canavieiro. Neste, a automatização favorece desde as etapas de plantio, colheita, separação, até as etapas de produção de derivados e comercialização. Atualmente o Brasil é o maior produtor de cana de açúcar do mundo¹, e se destaca pela produção do açúcar e do etanol, sendo a automatização um dos principais fatores que contribuiu para este destaque do setor canavieiro brasileiro.

Dentre as etapas envolvidas na produção de derivados da cana de açúcar, o processo de separação de cana, realizado após a colheita, é feito manualmente, ou por meio de algum processo automatizado. Nesta etapa, é frequente a separação das canas que são cortadas pelo homem, daquelas que são cortadas pelas máquinas. A Figura 1.1 mostra os dois tipos de cortes realizados na colheita de canas de açúcar.

A distinção e separação entre os dois tipos de cortes é importante, pois ambos devem receber tratamentos diferentes antes de serem enviados para as etapas seguintes. Esses tratamentos dependem de como o corte é realizado, e muitas vezes referem-se à ação de como os caules serão limpos. No corte manual, por exemplo, a integridade da cana é preservada, fazendo a produtividade aumentar.

¹Disponível em: <http://www.agricultura.gov.br/vegetal/culturas/cana-de-acucar>. Acessado em 20 Julho de 2013.



(a) Corte Manual

(b) Corte Mecânico

Figura 1.1: Tipos de cortes de canas de açúcar.

Por outro lado, em cortes realizados pelas máquinas, as canas são cortadas junto com suas folhas ou às vezes retiradas do solo, carregando além das folhas, terras e raízes. Além disso, os caules podem ser prejudicados durante o corte, caso a máquina não esteja corretamente regulada. Diante disso, percebe-se que os tratamentos a serem feitos com os cortes mecânicos tendem a ser mais rigorosos.

No processo manual da separação, a tarefa é realizada por um humano com base na sua experiência de trabalho. Já no processo automatizado, as canas cortadas são separadas através de um modelo mecânico provido de técnicas que permitem a distinção dos dois tipos de cortes.

Uma maneira de automatizar o processo de separação, é utilizar técnicas que permitem identificação dos dois tipos de cortes. Neste sentido, podemos considerar um braço mecânico associado à uma câmera, que registra, periodicamente, imagens, de cortes mecânicos e manuais, que passam sobre uma esteira. A câmera, ao registrar as imagens, envia-a para uma unidade de processamento. Tal unidade é provida de alguma técnica para a análise de imagens, capaz de identificar padrões

que permitam caracterizar os diferentes tipos de cortes. Caracterizados, a unidade aciona um braço mecânico para que este possa separar corretamente.

Um processo automatizado, tal como o descrito anteriormente, isenta o homem do contato com as canas cortadas, oferece mais segurança (tanto material quanto humana), otimiza o custo, o tempo, aumenta a capacidade de produção e qualidade, entre outros benefícios.

Neste contexto, a etapa de reconhecer os diferentes tipos de cortes diante de imagens deve ser desenvolvida através de modelos computacionais eficazes, capazes de extrair as principais características permitindo identificar um determinado tipo de imagem, em tempo real. O reconhecimento de imagens é uma tarefa comum em sistemas de visão computacional. É nessa tarefa que se encontram técnicas empregadas para a identificação e extração de padrões em imagens.

A análise de textura é uma etapa fundamental no processo de extração de características de imagens, e consiste em diversas abordagens, dentre elas a utilizada nessa pesquisa: Abordagem Descritiva. Dentro desta abordagem, pode-se citar o modelo da Caminhada Determinística do Turista (CDT) que tem mostrado resultados satisfatórios em diferentes aplicações (CAMPITELI, 2007; BACKES *et al.*, 2006; LIMA, 2002; BACKES *et al.*, 2010; GONÇALVES, 2010).

1.2 Objetivos Gerais

O objetivo geral deste trabalho é: modelar a CDT como uma ferramenta para análise de texturas de imagens de cortes, manuais e mecânicos, de canas de açúcar. Explorar estratégias para a extração de características das informações obtidas pela CDT, avaliando a discriminação das características extraídas diante de classificadores baseados em Máquinas de Vetores de Suporte (*Support Vectors Machines*).

1.3 Objetivos Específicos

- Modelar os aspectos da CDT através das propriedades que o problema nos oferece, tornando possível sua implementação e sua aplicação nas imagens.
- Adotar estratégias para melhorar a eficiência do algoritmo na detecção de atratores.
- Aplicar a CDT sobre o *benchmark Outex*.
- Explorar estratégias que permitam a extração de características das informações obtidas com o algoritmo da CDT.
- Avaliar o uso de modelos de classificação baseados em *Support Vectors Machines* (SVMs) diante das estratégias de extração de características adotadas.
- Verificar se CDT pode ser usada como uma ferramenta eficiente para a extração de características em imagens de canas de açúcar.

1.4 Organização do Texto

Este trabalho está dividido da seguinte forma.

No Capítulo 2 será introduzido o conceito de imagem digital, seguido da definição de Análise de Texturas e suas respectivas abordagens. Também será tratado o conceito por trás das SVMs, os aspectos e definição da CDT, bem como as estratégias para a extração de características das informações resultantes da aplicação do algoritmo, e trabalhos correlatos.

No Capítulo 3 são discutidos aspectos e estratégias considerados para a implementação e aplicação da CDT para a extração de características em imagens

e as bases de dados sobre o qual foram feitos os testes, bem como os modelos de classificação baseados em SVMs.

No Capítulo 4, são apresentados e discutidos os resultados obtidos pelos modelos SVMs.

Por último, o Capítulo 5, apresenta as conclusões obtidas a partir deste trabalho.

2 REFERENCIAL TEÓRICO

2.1 Imagens Digitais

A imagem digital de um objeto do mundo real é obtida através de aparelhos capazes de capturar e transformar os fenômenos físicos para o formato digital (GONZALEZ; WOODS, 2010). Ela pode ser representada matematicamente por uma função bidimensional $f(x,y)$, onde x e y são coordenadas no plano e o valor de f é o brilho (intensidade) da imagem sobre o ponto (x,y) (MASCARENHAS; VELASCO, 1984).

A função $f(x,y)$ pode ser representada por duas formas eficientes, são elas: a matriz de intensidade visual e a matriz numérica. A primeira, é a mais comum e trata-se de como uma imagem seria apresentada no monitor de um computador, por exemplo. A segunda, é útil em elaboração de algoritmos e utilizada no Processamento de Imagens (GONZALEZ; WOODS, 2010). A representação de $f(x,y)$ por uma matriz numérica $M \times N$ é dada pela Equação 2.1 a seguir.

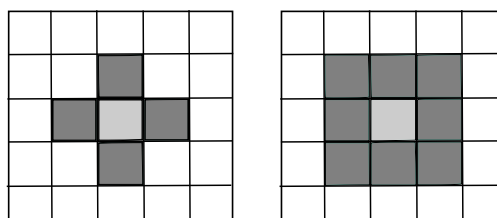
$$f(x,y) = \begin{pmatrix} f(0,0) & f(0,1) & \dots & f(0,N-1) \\ f(1,0) & f(1,1) & \dots & f(1,N-1) \\ \vdots & \dots\dots\dots & & \vdots \\ f(M-1,0) & f(M-1,1) & \dots & f(M-1,N-1) \end{pmatrix}_{M \times N} \quad (2.1)$$

onde cada elemento da matriz é um *pixel* da imagem.

Pela equação definida anteriormente, podemos ver que para cada *pixel* têm-se um determinado valor de intensidade. Em uma imagem em níveis de cinza, por exemplo, os valores de intensidade variam de 0 a 255, onde o 0 significa a cor preta e 255 a cor branca. Este intervalo é a quantidade de valores que um *byte* (2^8)

pode assumir. Neste sentido, podemos interpretar o *pixel* como a menor unidade presente em uma imagem digital.

Os *pixels* e seus relacionamentos desempenham papéis fundamentais em aplicações de Processamento de Imagens, tais como, restauração e segmentação de imagens (GONZALEZ; WOODS, 2010). Um relacionamento importante entre eles é a de vizinhança, mostrada pela Figura 2.1 (a) e (b). Em (a), a relação de um *pixels* com os quatro vizinhos, verticais e horizontais, é denominada de *vizinhança-4*. Esses *pixels*, junto com os *pixels* que estão na diagonal, mostrado em (b), compõe o que se denomina de *vizinhança-8* (MASCARENHAS; VELASCO, 1984).

(a) *Vizinhança-4*(b) *Vizinhança-8***Figura 2.1:** Relação de vizinhança entre os *pixels*.

2.2 Análise de Texturas

O fato de podermos distinguir diferentes formas de objetos, cores e perspectivas é fruto de um processo de aprendizagem, pelo qual, sem que percebamos, estamos submetidos a medida que crescemos, sendo a visão humana elemento ímpar nesse processo de aprendizagem. A concepção que temos de um objeto, quando não o vemos ou não o observamos por completo, é uma habilidade construída durante este processo.

Quando temos uma representação do objeto, por meio de imagens por exemplo, tal habilidade de reconhecimento e concepção do objeto se mantém, mesmo que as imagens estejam em níveis de cinza. A Figura 2.2 apresenta quatro imagens em níveis de cinza que estão relacionadas à um objeto do mundo real. A imagem (a) representa um pedaço de madeira e podemos perceber suas características, dentre elas o seu polimento; em (b) podemos perceber a imagem de um tipo de gramado; na imagem (c), deduzimos que provavelmente seja um muro feito de tijolos, e por último, em (d) a porção de uma imagem que representa algo líquido.

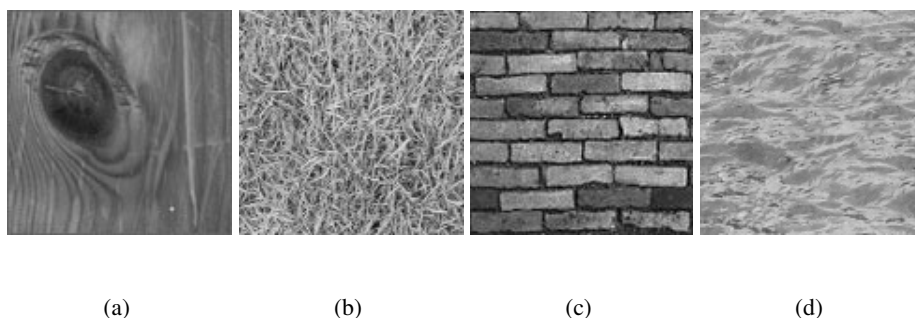


Figura 2.2: Texturas obtidas a partir de cenas naturais.

Diferente de nós, seres humanos, os computadores são desprovidos de qualquer benefício/habilidade que a visão nos oferece. Entretanto podem ser capazes de trabalhar com imagens geradas por fontes que estão além das que os seres humanos são aptos a perceber por meio da visão.

Sem qualquer tipo de técnica ou ferramenta, seria impossível para os computadores associar os objetos apresentados acima, com suas respectivas cenas naturais. Sendo assim, diversas técnicas e ferramentas tem sido criadas, a fim de conceder aos computadores o poder de enxergar, representar e caracterizar objetos (GONZALEZ; WOODS, 2010).

Uma das técnicas para caracterizar objetos parte do princípio de explorar suas superfícies e retirar padrões que possibilite distingui-lo de outros. A textura é uma terminologia usada para caracterizar superfícies de um objeto e é um dos principais atributos usados no Processamento de Imagens para a classificação, segmentação, síntese e geração de imagens (WECHSLER, 1980; CROSS; JAIN, 1983; BLINN; NEWELL, 1976; ZHOU, 2006). Foi a partir dela que nós, ao observarmos as imagens da Figura 2.2, conseguimos associar os diferentes objetos apresentados às suas respectivas cenas naturais. Em Haralick, Shanmugam e Dinstein (1973) uma definição bastante elementar sobre a textura é dada:

A textura é uma propriedade natural presente em todas as superfícies, seja ela um pedaço de madeira, a ondulação de um tecido, o padrão de uma cultura no campo, entre outras. A textura contém informações importantes sobre o arranjo estrutural das superfícies e do meio em que fazem partes. (...) Uma vez que as propriedades das texturas de imagens consideram informações úteis para fins de discriminação, torna-se necessário a construção de técnicas para a extração de suas características. (p. 611)

O processo de extração e agrupamento de características extraídas da textura de uma imagem, é conhecido como Análise de Textura (ZHOU, 2006). A Análise de Textura é dividida em duas abordagens: *descritivas* e *genéricas*. Cada abordagem, por sua vez, é dividida, de acordo com as propriedades de seus métodos, tal como é apresentado pela Figura 2.3.

A seguir serão discutidas as principais propriedades de cada abordagem e de seus respectivos métodos.

2.2.1 Abordagem Descritiva

As abordagens descritivas envolvem métodos que extraem e criam conjuntos de características, capazes de representar determinadas texturas (ZHOU, 2006). Esta abordagem é dividida em métodos *estatísticos* e *espectrais*.

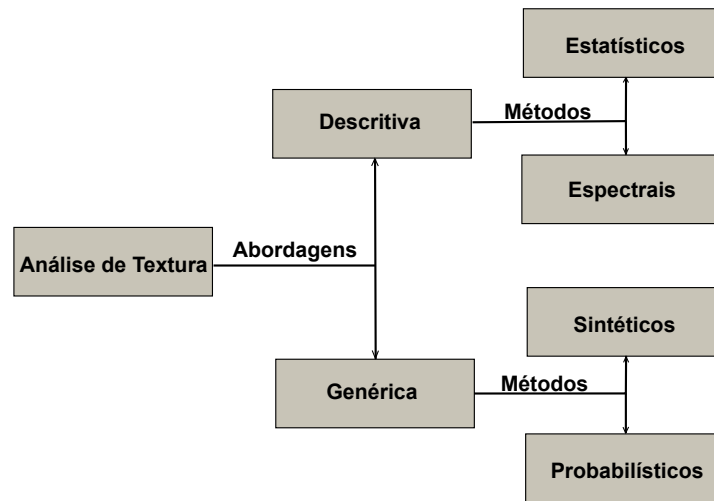


Figura 2.3: Análise de textura com suas abordagens e métodos.

2.2.1.1 Métodos Estatísticos

Métodos estatísticos se caracterizam pela coleta de características que estão sobre um domínio espacial. Técnicas de histogramas (GONZALEZ; WOODS, 2010) e matrizes de co-ocorrência em níveis de cinza (CLAUSI; ZHAO, 2003) são exemplos desses tipos de domínios.

2.2.1.2 Métodos Espectrais

Métodos espectrais são caracterizados pelo uso do domínio da frequência como descritores gerais de características. Métodos dessa abordagem geralmente usam filtros que transformam o domínio espacial em um domínio de frequência, capaz de destacar determinadas características, enquanto outras são eliminadas. Filtros de Gabor e Descritores de Fourier são exemplos de métodos espectrais (JAIN; FARROKHANIA, 1991; MATSUYAMA; MIURA; NAGAO, 1983).

2.2.2 Abordagem Genérica

A abordagem genérica faz o uso de técnicas geométricas e probabilísticas para descrever uma determinada textura (ZHOU, 2006). Os principais métodos dessa abordagem são os *sintéticos* e *probabilísticos*.

2.2.2.1 Métodos Sintéticos

Envolvem explorar a geometria das texturas por meio de técnicas espaciais analíticas. O uso de fractais e abordagens estruturais são os principais meios para a criação de métodos sintéticos de texturas (ZHOU, 2006).

A maioria das texturas apresentam padrões bem similares, às vezes, dispostos em diferentes escalas. Fractais são padrões que se repetem em diferentes escalas, e por isso podem servir como ferramentas para descrever determinadas texturas (MANDELBROT, 1983). Nos estudos de Pentland (1984), por exemplo, eles são usados para modelar cenas naturais.

As abordagens estruturais partem da ideia de que a textura é formada por elementos mais simples capazes de descrever a textura por completo (GONZALEZ; WOODS, 2010). Bordas de objetos nas imagens e polígonos de Voronoi são exemplos de elementos de texturas (TUCERYAN; JAIN, 1990).

2.2.2.2 Métodos Probabilísticos

A ideia por trás desses métodos é construir modelos paramétricos, criados a partir de características retiradas de uma determinada textura, para a criação de outras novas texturas semelhantes (ZHOU, 2006). Um exemplo promissor de métodos probabilísticos são os campos aleatórios de Markov-Gibbs (CROSS; JAIN, 1983).

2.3 Máquinas de Vetores de Suporte

As máquinas de vetores de suporte, do inglês *Support Vectors Machines* (*SVMs*), são técnicas de aprendizagem de máquina estabelecidas a partir da Teoria de Aprendizagem Estatística (VAPNIK, 1999) e vêm sendo aplicadas em diversos problemas de classificação de padrões (BYUN; LEE, 2002).

Dado um conjunto de treinamento (\mathbf{x}_i, y_i) , $i = 1, \dots, l$, onde $\mathbf{x}_i \in \mathfrak{R}$, $y_i \in \{1, -1\}^l$, as SVMs consistem no seguinte problema de otimização (CAMPITELLI, 2007):

$$\begin{aligned} & \text{minimizar } \frac{1}{2} \mathbf{w}^t \mathbf{w} + C \sum_{i=1}^l \xi_i \\ & \text{sujeito a } y_i (\mathbf{w}^t \phi(\mathbf{x}_i) + b) \geq 1 - \xi_i, \xi_i \leq 0 \end{aligned} \quad (2.2)$$

A finalidade de um modelo baseado em SVM, é construir uma superfície de separação, tal que a margem de separação entre duas classes sejam máximas. A construção desta superfície de separação é feita com o auxílio de vetores de suporte, que conceitualmente são as amostras de dados mais próximas da superfície de separação, sendo portanto, as mais difíceis de serem classificados, e por isso desempenham um papel fundamental na localização ótima da superfície de decisão (HAYKIN, 2001). A Figura 2.4 ilustra o conceito por trás das SVMs. Na figura, a superfície δ , também denominada de hiperplano ótimo, separa as classes σ_1 e σ_2 , tal que a separação entre elas é máxima.

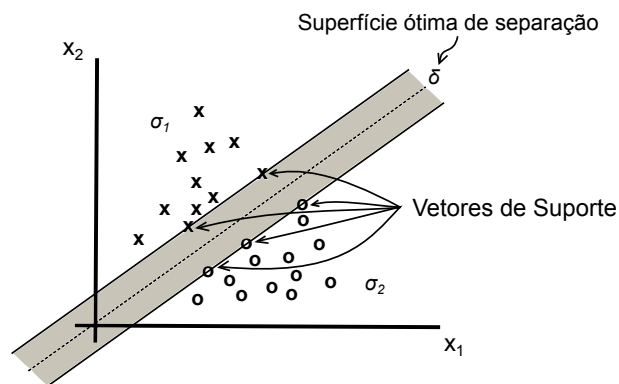


Figura 2.4: Ilustração da superfície ótima para padrões linearmente separáveis.

Normalmente, problemas de classificação envolvem dados que não são linearmente separáveis. Para tratar isso as SVMs fazem o uso de funções *kernel* (Φ), cuja finalidade é levar os vetores de suporte para um espaço de alta dimensão (espaço de características) tornando-os linearmente separáveis (HAYKIN, 2001). O *kernel* Gaussiano (RBF - *Radial Basis Functions*) dado por $e^{-\gamma\|x-y\|^2}$ é um exemplo desse tipo de função (CAMPITELI, 2007). A Figura 2.5 ilustra o conceito do uso de funções *kernel*.

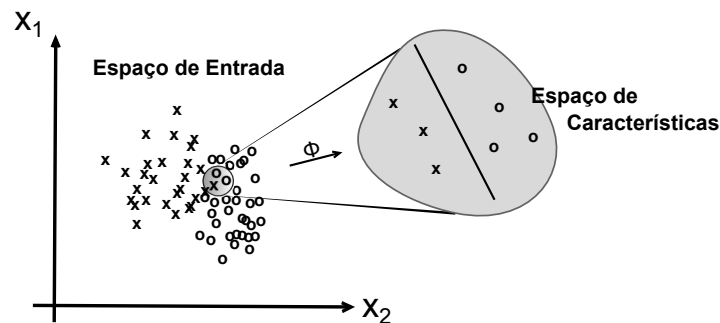


Figura 2.5: Ilustração do espaço de entrada e do espaço de característica.

As SVMs apresentam algumas características que as tornam atrativas para seu uso neste trabalho. Dentre elas (LORENA; CARVALHO, 2007): boa capacidade de generalização, robustas diante de objetos com grandes dimensões, tais

como imagens, e possuem uma teoria bem estabelecida dentro da matemática e estatística.

Um problema no uso das SVMs é a escolha dos parâmetros que serão utilizados no treinamento. As vezes a escolha cabe somente ao parâmetro C , que controla o número de dados não separáveis, como é o caso de modelos SVMs de *kernel linear*. Já para modelos SVMs de *kernel RBF*, a decisão cabe na melhor combinação de C e γ , este por sua vez consiste na largura da função RBF. A escolha desses parâmetros envolve o uso de estratégias, tal como, a validação cruzada (HSU; CHANG; LIN, 2010; HAYKIN, 2001).

2.4 Caminhadas Determinísticas

Caminhadas realizadas em conjuntos de dados podem ser divididas em duas categorias: caminhadas aleatórias e caminhadas determinísticas. A primeira categoria é representada por modelos onde o movimento é realizado por resultados totalmente aleatórios, enquanto que na segunda, a movimentação é realizada por regras determinísticas previamente definidas. Entretanto, ambas são aplicadas com o objetivo de explorar e caracterizar o meio modelado pelos dados (CAMPITELI, 2007).

Diferente das caminhadas determinísticas, as caminhadas aleatórias são mais estudadas e possuem uma base teórica bem estabelecida (LIMA, 2002). Porém, nos últimos anos, modelos de caminhadas determinísticas têm despertado o interesse de pesquisadores e o desenvolvimento de aplicações em diversas áreas de conhecimento (CAMPITELI, 2007; GONÇALVES, 2010).

No contexto de caminhadas determinísticas, o modelo de caminhada do turista tem apresentado bons resultados na identificação de padrões (BACKES *et al.*, 2006; BACKES *et al.*, 2010; CAMPITELI, 2007). Neste trabalho, o modelo

de caminhada do turista foi usado como uma ferramenta para a análise de texturas. As seções a seguir, apresentam o conceito e as propriedades deste modelo de caminhada.

2.5 A Caminhada Determinística do Turista

A Caminhada Determinística do Turista (CDT) foi proposta por Lima (2002). Nela podemos imaginar um turista que caminha por um número de cidades N sobre um mapa de dimensão d . A dinâmica de movimentação do turista é dada por um regra bem estabelecida, que pode ser entendida como uma função que leva o turista de uma cidade x_i para uma cidade próxima x_j , tal que x_i e $x_j \in N$. Em Campiteli (2007) três regras de caminhadas são definidas, em uma delas, por exemplo, o caminhante escolhe dentro de sua vizinhança a cidade com o menor gradiente de energia com relação a sua cidade atual.

A movimentação do turista também está associada à uma restrição: o turista não pode passar pelas cidades visitadas nos últimos μ passos de tempos. Esta restrição é denominada de memória do turista (LIMA, 2002).

A memória de tamanho μ , pode ser entendida como o tempo necessário para que uma cidade possa ser visitada novamente pelo turista. Este conceito da memória do turista é exemplificado pela Figura 2.6, em vermelho temos um exemplo de trajetória do turista.

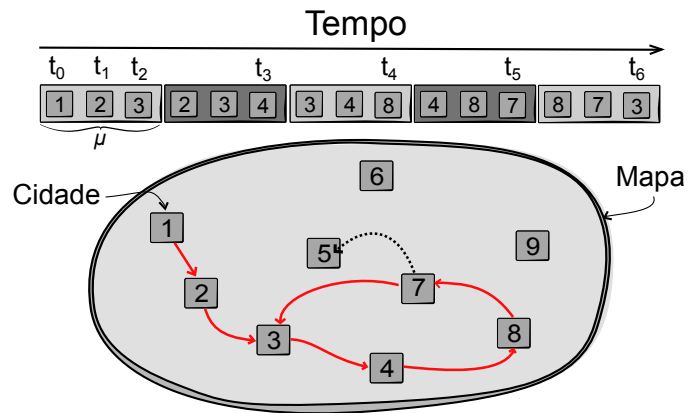


Figura 2.6: Um exemplo de trajetória e do comportamento da memória do turista.

Considerando uma memória de tamanho $\mu = 3$, uma regra determinística que faça o turista caminhar pelas cidades mais próximas e que em cada instante de tempo t_i o turista visita uma determinada cidade, temos: do instante t_0 até t_2 o turista visita e armazena, respectivamente, na sua memória as cidades 1, 2 e 3, fazendo com que a mesma fique completamente ocupada. No instante seguinte, t_3 , a cidade 1 excede o tempo de permanência na memória e portanto deve ser “descartada” pelo turista. Esta ação faz com que a memória tenha um “espaço livre” para armazenar a cidade 4 imediatamente visitada. Um fato interessante ocorre nos tempos t_5 e t_6 ; em t_5 o turista esquece que visitou 3 e armazena 7 na sua memória, no instante t_6 , 3 é novamente visitada e colocada na memória. Isso é possível, uma vez que o turista já esqueceu que 3 foi visitada. Caso a memória fosse uma unidade maior, a cidade 3 ainda seria “lembrada”, forçando o turista a procurar por outra cidade próxima a ser visitada, por exemplo a cidade 5, onde a direção pontilhada indica este caminho alternativo. Do instante t_6 em diante, as futuras ações do turista o levam a passar sempre pelas mesmas cidades: 3, 4, 8 e 7. Este ciclo no qual o turista está sujeito é uma característica da caminhada e também um ponto de parada para o algoritmo.

A regra dinâmica imposta pela memória produz uma trajetória de passos iniciais, denominada de tempo *transiente* e uma trajetória periódica final, denominada de *atrator* (CAMPITELI, 2007). O período atrator p e o tempo transiente t , representam respectivamente, o número de cidades na qual o turista está aprisionado e o número de cidades que o turista visitou desde o seu ponto de partida até a primeira cidade antes de cair em um ciclo. A Figura 2.7 ilustra esses dois conceitos.

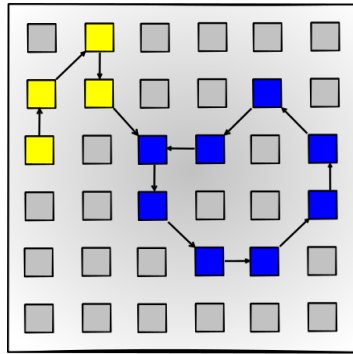


Figura 2.7: Um exemplo de trajetória do turista. Os quadrados em amarelo representam o tempo transiente, os quadrados azuis o período atrator.

O comportamento do turista não está estritamente ligado somente ao conceito da memória, mas também a configuração das cidades no mapa, a cidade de partida e a função determinística de caminhada. Neste sentido, um mapa de dimensão d com N cidades organizadas, terão N trajetórias diferentes, e para cada trajetória haverá um determinado valor para t e p . Esses valores podem ser combinados em uma distribuição conjunta $S_{\mu,d}^{(N)}$ definida pela equação 2.3 (GONÇALVES, 2010).

$$S_{\mu,d}^{(N)}(t,p) = \frac{1}{N} \sum_{i=1}^N \begin{cases} 1, \text{ se } t_i = t, p_i = p \\ 0, \text{ caso contrário} \end{cases} \quad (2.3)$$

Uma representação da distribuição conjunta resultante pode ser dada pela matriz 2.4, onde t_{max} e p_{max} representam, respectivamente, o tempo transiente e período atrator máximos obtido após a aplicação da CDT sobre um determinado mapa.

$$\begin{pmatrix} S(0, \mu + 1) & S(0, \mu + 2) & \dots & S(0, p_{max}) \\ S(1, \mu + 1) & S(1, \mu + 2) & \dots & S(1, p_{max}) \\ \vdots & \dots & \dots & \vdots \\ S(t_{max}, \mu + 1) & S(t_{max}, \mu + 2) & \dots & S(t_{max}, p_{max}) \end{pmatrix}_{t_{max} \times p_{max}} \quad (2.4)$$

2.5.1 A Memória do Turista

A memória do turista impõe uma condição de restrição para o algoritmo, fazendo com que o caminhante seja parcialmente impedido de passar pelo mesmo ponto (*self-avoiding*). Por este motivo, a caminhada determinística do turista também é conhecida como *partial self-avoiding deterministic walk - PSADW* (CAMPITELI, 2007).

Um turista com memória igual a zero ($\mu = 0$), não consegue se mover, e a resposta para suas trajetórias são triviais, sendo o tempo transiente igual a zero, pois o turista não se move, e o período atrator igual a um, que é a própria cidade inicial de uma trajetória.

Para uma memória igual a um ($\mu = 1$), o turista tem informações somente de seus vizinhos. Neste caso, o turista caminha até encontrar um par de cidades em que ficará preso. Sendo portanto, o período atrator igual a dois e o período transiente dependente da disposição das cidades no mapa e da função determinística de caminhada. Apesar de poder conseguir caminhar por várias cidades, esta regra não produz uma caracterização do ambiente (BACKES *et al.*, 2010).

Casos em que a memória assume valores no intervalo $1 < \mu < N - 1$ são diferentes dos anteriores. Neste caso, nada pode-se concluir a respeito dos valores dos tempos transientes e dos períodos atratores, sendo $p \geq \mu + 1$ a única restrição válida (LIMA, 2002). Esta restrição é devido ao número mínimo de caminhadas que o turista deve visitar para que uma cidade esteja acessível novamente. Isto explica porque os elementos da primeira coluna da matriz 2.4 são valores cuja a distribuição conjunta tem valores $p = \mu + 1$ e os elementos das colunas seguintes valores maiores que $\mu + 1$.

2.5.2 Condições de Contorno

Em um meio limitado, as cidades que se encontram na fronteira do mapa sofrem um determinado isolamento e acabam relacionando com seus vizinhos de maneira diferente das cidades que estão localizadas na região central. Um ambiente com esta configuração é dito apresentar *condições abertas de contorno*. Por outro lado, quando todas as cidades são igualmente acessíveis, então o ambiente apresenta *condições periódicas de contorno* (CAMPITELI, 2007).

Conectar as extremidades de um ambiente com condições abertas de contorno é uma maneira de permitir que todas as cidades sejam igualmente acessíveis, em um mapa onde as disposições das cidades são regulares. Fazendo isso, o ambiente passará a ter as mesmas configurações de um meio com condições periódicas de contorno. A Figura 2.8 ilustra este processo para um ambiente de uma e duas dimensões.

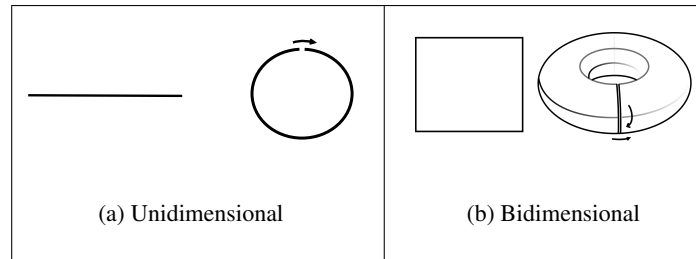


Figura 2.8: Condições abertas e periódicas de contorno.

2.5.3 Algoritmo para a Caminhada do Turista

Baseado no algoritmo proposto por Campitelli (2007), o algoritmo descrito a seguir, apresenta as variáveis básicas e os procedimentos auxiliares usados para a implementação da caminhada determinística do turista deste trabalho.

No Algoritmo 1, temos como entrada a memória do turista, que pode variar no intervalo $0 \leq \mu \leq N - 1$ e uma matriz de vizinhança (**MV**), que armazena os respectivos vizinhos de cada uma das cidades do mapa.

O algoritmo é aplicado para cada cidade do mapa, e cada trajetória vai sendo formada a medida que um atrator não é detectado. O procedimento responsável pela detecção dos atratores (**verifica_atrator**), recebe como parâmetro uma lista de cidades visitadas pelo turista (Σ) e se um atrator é formado, então a trajetória é encerrada trocando o valor da variável `parou` para verdadeiro e armazenando o tempo transiente e período atrator na matriz conjunta **MC**, através dos respectivos procedimentos: **armazena_atrator** e **armazena_transiente**. Caso um atrator não seja formado, é necessário prosseguir com a trajetória; para isso, uma nova cidade, dentro de sua vizinhança, é determinada e atribuída a variável `cidade` a partir de função (ϕ). Com esta função é possível através dos parâmetros (`cidade`, **MV** e μ) e de uma regra determinística de caminhada pré-estabelecida, continuar a trajetória do turista.

Algoritmo 1: Algoritmo Da Caminhada Determinística do Turista.

Entrada: Matriz de vizinhança: \mathbf{MV} e Memória do Turista: μ

Saída: Matriz Conjunta: \mathbf{MC}

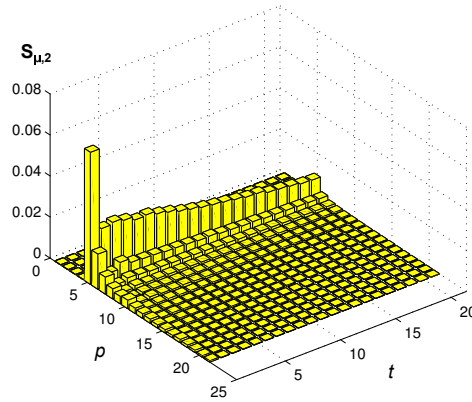
```

1 início:
2   para (cidade = 0 até N) faça:
3     parou = falso;
4      $\Sigma$  = [cidade]
5     enquanto ( $\neg$  parou) faça:
6       se (verifica_atrator( $\Sigma$ )) então:
7         armazena_atrator( $\mathbf{MC}$ );
8         armazena_transiente( $\mathbf{MC}$ );
9         parou = verdadeiro;
10      fim-se
11      se-não
12        cidade =  $\phi$ (cidade,  $\mathbf{MV}$ ,  $\mu$ );
13         $\Sigma$  = [ $\Sigma$ , cidade];
14      fim-se
15    fim-equanto
16  fim-para
17 fim.
```

Ao término do algoritmo têm-se a distribuição conjunta em forma de matriz com os tempos transientes e períodos atratores de todas as caminhadas efetuadas. A Figura 2.9 (b) mostra o resultado da distribuição conjunta do algoritmo desenvolvido neste trabalho (mais detalhes dos aspectos do algoritmo implementado serão tratados no Capítulo 3) e aplicado sobre uma imagem em (a).



(a)



(b)

Figura 2.9: Distribuição conjunta (b) resultante da aplicação da CDT sobre a imagem em (a) retirada de <http://sipi.usc.edu/database>. Acessado em 20 de Julho de 2013).

Um dos maiores problemas da CDT está no procedimento para a detecção de atratores. Este procedimento possui um alto custo computacional e ainda é um ponto em que se exige estudos (CAMPITELI, 2007). Mais detalhes sobre a detecção de atratores, serão apresentados no Capítulo 3, quando for discutido a estratégia adotada para a implementação deste procedimento.

2.5.4 Vetores de Características

A distribuição conjunta agrupa todas as informações resultante das trajetórias do turista. Para que a CDT possa ser usada de forma eficiente, as informações mais relevantes devem ser extraídas desta distribuição formando os vetores de características, posteriormente usados para treinar e testar modelos de classificação. Em Gonçalves (2010), são apresentadas três estratégias para a construção dos vetores de características. São elas: *Região*, *Histograma* e *Propriedades Estatísticas*.

Outra estratégia avaliada em Backes *et al.* (2006), permite a construção do vetor de característica por meio da junção de regiões da distribuição conjunta para diferentes valores de memórias. Neste trabalho, esta estratégia será denominada de *Regiões de Diferentes Memórias*. As seções a seguir descrevem cada uma dessas estratégias.

2.5.4.1 Região

A estratégia por região consiste em obter um vetor de características (ψ) através das informações contidas em uma região da distribuição conjunta (Figura 2.10). Esta região é dada por um intervalo de tempo transiente $[t_0, t_f]$ e por um intervalo de período atrator $[\mu + p_0, \mu + p_f]$, tal que $t_{min} \leq t_0 < t_f < t_{max}$ e $1 \leq p_0 < p_f < p_{max}$, onde t_{min} , t_{max} e p_{max} representam, respectivamente, o tempo transiente mínimo, tempo transiente máximo e período atrator máximo da distribuição conjunta. O vetor de características é dado pela Equação 2.5.

$$\psi_{\mu} = [S_{\mu,d}^{(N)}(t_0 : t_f, \mu + p_0), S_{\mu,d}^{(N)}(t_0 : t_f, \mu + p_0 + 1)m, \dots, S_{\mu,d}^{(N)}(t_0 : t_f, \mu + p_f)] \quad (2.5)$$

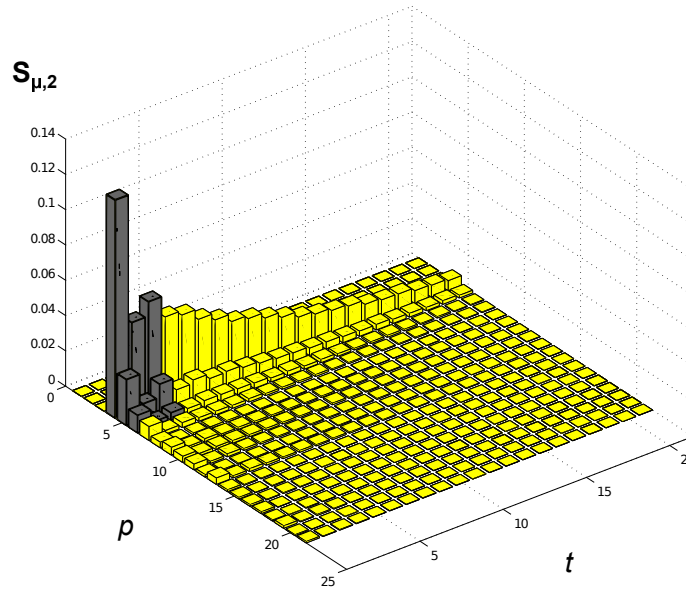


Figura 2.10: Estratégia de extração de características por região. Em cinza um parte da distribuição conjunta escolhida para compor o vetor de características. Cada bloco em cinza representa uma característica em ψ .

2.5.4.2 Histograma

Esta estratégia consiste em construir um histograma $h_{\mu}(p+t)$, com uma caminhada de tamanho $(p+t)$. Cada histograma h_{μ} corresponde a uma atributo do vetor de características ψ . A Equação 2.6 define o vetor de características para esta estratégia.

$$\psi_{\mu} = [h_{\mu}(\mu+1), \dots, h_{\mu}(\mu+n)] \mid h_{\mu}(\mu+n) = \sum_{i=0}^{n-1} S_{\mu,d}^{(N)}(i, \mu+n-i) \quad (2.6)$$

Esta estratégia consegue agrupar mais informações da distribuição conjunta em um número menor de características comparada com a estratégia anterior. A Figura 2.11 ilustra esta estratégia.

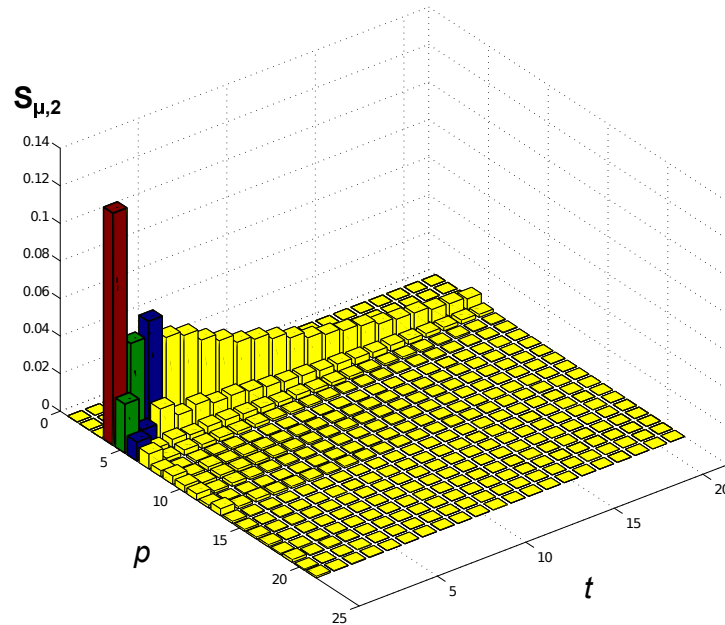


Figura 2.11: Estratégia de extração de características por Histograma. Cada conjunto de cor representa uma característica do vetor ψ .

2.5.4.3 Propriedades Estatísticas

As estratégias discutidas anteriormente não consideram todas as informações disponíveis pela distribuição conjunta. No trabalho de Haralick, Shanmugam e Dinstein (1973), são propostas 14 propriedades estatísticas capazes de considerar todas as informações contidas em uma determinada distribuição bidimensional, tal como pode ser o caso da distribuição conjunta. Entretanto, trabalhos tais como os de Weszka, Dyer e Rosenfeld (1976), Connors e Harlow (1980) e Clausi e Zhao

(2003) utilizaram apenas algumas dessas propriedades. Em conjunto, elas são descritas a seguir.

$$\text{Energia} = \delta_1 = \sum_i^{t_{\max}} \sum_j^{p_{\max}} S_{\mu,2}^{(N)}(i, j)^2 \quad (2.7)$$

$$\text{Inércia} = \delta_2 = \sum_i^{t_{\max}} \sum_j^{p_{\max}} (i - j)^2 S_{\mu,2}^{(N)}(i, j)^2 \quad (2.8)$$

$$\text{Entropia} = \delta_3 = - \sum_i^{t_{\max}} \sum_j^{p_{\max}} S_{\mu,2}^{(N)}(i, j) \log_2 S_{\mu,2}^{(N)}(i, j) \quad (2.9)$$

$$\text{Momento da Diferença Inversa} = \delta_4 = \sum_i^{t_{\max}} \sum_j^{p_{\max}} \frac{S_{\mu,2}^{(N)}(i, j)}{1 + (i - j)^2} \quad (2.10)$$

$$\text{Correlação de Haralik} = \delta_5 = \frac{\sum_i^{t_{\max}} \sum_j^{p_{\max}} (ij) S_{\mu,2}^{(N)}(i, j) - \gamma_x \gamma_y}{\sigma_x \sigma_y} \quad (2.11)$$

$$\text{Correlação} = \delta_6 = \frac{- \sum_i^{t_{\max}} \sum_j^{p_{\max}} (i - \gamma_x)(j - \gamma_y) S_{\mu,2}^{(N)}(i, j)}{\sigma_x \sigma_y} \quad (2.12)$$

onde $\gamma_x, \gamma_y, \sigma_x$ e σ_y , são dadas pelas seguintes equações:

$$\sigma_x = \sum_i^{t_{\max}} i \sum_j^{p_{\max}} S_{\mu,2}^{(N)}(i, j) \quad (2.13)$$

$$\sigma_y = \sum_j^{p_{\max}} j \sum_i^{t_{\max}} S_{\mu,2}^{(N)}(i, j) \quad (2.14)$$

$$\gamma_x^2 = \sum_i^{t_{max}} (i - \sigma_x)^2 \sum_j^{p_{max}} S_{\mu,2}^{(N)}(i, j) \quad (2.15)$$

$$\gamma_y^2 = \sum_j^{p_{max}} (j - \sigma_y)^2 \sum_i^{t_{max}} S_{\mu,2}^{(N)}(i, j) \quad (2.16)$$

2.5.4.4 Regiões de Diferentes Memórias

Para esta estratégia, primeiro obtêm-se os vetores de características das distribuições conjuntas obtidas por diferentes valores de memórias usando a estratégia por região, e por último concatena-se esses vetores, obtendo o vetor de característica final ψ , tal como ilustrado pela Figura 2.12.

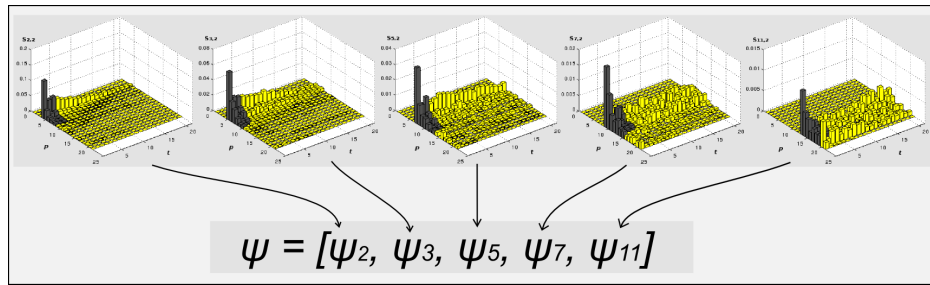


Figura 2.12: Estratégia de extração de características por regiões de diferentes memórias. Em cinza uma parte da distribuição conjunta correspondente à uma memória μ .

2.5.5 Estado da Arte

No contexto deste trabalho, pode-se encontrar trabalhos que fizeram o uso da Caminhada Determinística do Turista para a extração de características em imagens. Esses trabalhos são citados a seguir.

No trabalho de Backes *et al.* (2006) a CDT foi aplicada sobre imagens em níveis de cinza, sendo possível estabelecer uma regra determinística de movimentação que permitia o turista caminhar para os *pixels* próximos onde a diferença

de intensidade era mínima. Utilizando a extração de características por região e por regiões de diferentes memórias foi possível obter bons resultados. Entretanto, os resultados obtidos pela última estratégia foram os melhores, diante de regiões onde $t_0 = 0$, $t_1 = 1$, $p_0 = 1$ e $p_f = 2$.

Campitelli (2007) utilizou a caminhada determinística do turista para a exploração de ambientes aleatórios e estruturados. Em ambientes estruturados a caminhada foi aplicada como um método de análise de imagens obtendo bons resultados. Neste mesmo trabalho, Campitelli (2007) propõe um algoritmo baseado na CDT para reconhecimento de padrões e os resultados obtidos mostram um método eficiente e não supervisionado de agrupamento de dados.

O trabalho desenvolvido por Backes *et al.* (2010) avalia a CDT sobre diversos aspectos e também como metodologia para a análise de imagens. Neste trabalho, uma imagem foi modelada como um grafo (redes complexas), sendo cada grafo constituído de um conjunto de vértices representados pelos *pixels* da imagem e um conjunto de arestas com pesos representados pelas diferenças entre os níveis de cinza de seus vizinhos. Em seguida, grafos reduzidos foram obtidos a partir de uma transformação preservando arestas de interesses. Por último, a caminhada do turista foi aplicada sobre cada grafo, obtendo vários vetores de características, que foram concatenados resultando em um vetor final com diferentes informações da imagem. Os resultados obtidos superaram resultados obtidos por métodos tradicionais encontrados na literatura, tais como: descritores de Fourier, Matrizes de Co-ocorrência e Filtros de Gabor.

Em Gonçalves (2010) a CDT também é aplicada sobre uma imagem modelada como um rede complexa. O uso das estratégias de extração de características por região, histograma e propriedades estatísticas foram avaliadas, sendo a estratégia por histograma a que obteve os melhores resultados.

3 METODOLOGIA

Neste Capítulo serão tratados os aspectos e estratégias considerados para a implementação do algoritmo da caminhada determinística do turista, bem como os parâmetros utilizados para sua aplicação em imagens. Também são apresentadas as bases de dados de imagens, sobre o qual foram realizados os testes. Por último, são discutidos os modelos de classificação baseados em Máquinas de Vetores de Suporte e as estratégias usadas para melhorar suas capacidade de generalização.

3.1 Aspectos e Parâmetros da Caminhada do Turista

A Seção 2.5 abordou a caminhada determinística do turista de tal forma que não se especifica as aplicações dessa técnica, mostrando apenas, os principais aspectos e objetivos quando de sua utilização. O interesse deste trabalho é aplicar a CDT em dois conjuntos de classes de imagens para que possamos extrair informações significativas capazes de representar cada uma delas.

A primeira tarefa a ser feita, antes de ajustar o algoritmo da CDT para uma determinada aplicação, é relacionar suas propriedades com as propriedades do problema em estudo. A fim de tornar sua aplicação possível para este trabalho, partiu-se da seguinte concepção: “Dada uma imagem representada por uma matriz $M \times N$, podemos relacionar esta imagem ao mapa de cidades que o turista estará sujeito a caminhar, neste caso um mapa bidimensional. Cada elemento (i, j) da matriz é um *pixel* da imagem e portanto uma cidade com coordenada (x, y) no mapa”. Com esta analogia pode-se adaptar corretamente a CDT e aplicá-lo sobre qualquer imagem representada por uma matriz $M \times N$. As seções a seguir abordam todas as considerações feitas para a implementação e utilização do algoritmo em imagens.

3.1.1 Relação de Vizinhança

Dado que a regra de movimentação leva o turista para uma cidade próxima, é necessário, antes de definirmos uma regra determinística de caminhada, estabelecer uma relação de vizinhança para cada um dos *pixels* da imagem. Esta relação permitiu a implementação da função para a construção da matriz de vizinhança, que é uma das entradas do algoritmo definido na Seção 2.5.3.

A seção 2.1 estabeleceu dois tipos de relacionamentos básicos entre os *pixels*: *vizinhança-4* e *vizinhança-8*. Para facilitar o uso desses termos, será usada as notações Γ_4 e Γ_8 para representar, respectivamente, cada uma das relações.

A utilização de qualquer uma das relações torna possível a identificação dos vizinhos de cada um dos *pixels*. No entanto, Γ_4 limita pela metade as chances que o turista tem de efetuar o próximo passo, além disso, se observarmos a Figura 3.1 podemos notar que os períodos atratores a serem formados são maiores do que a relação Γ_8 . Isto aumenta a dificuldade de encontrar um atrator e consequentemente o custo computacional. Por esses motivos e com base nos trabalhos de Backes *et al.* (2006) e Campiteli (2007), foi escolhida a relação Γ_8 .

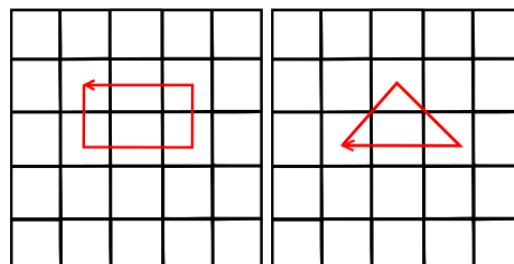
(a) Γ_4 (b) Γ_8

Figura 3.1: Considerando uma memória $\mu = 3$, ambas as figuras, (a) e (b), apresentam os melhores casos para a formação de um atrator.

3.1.2 Função Determinística de Caminhada

A princípio, a CDT proposta em Lima (2002) tem por finalidade fazer com que o turista caminhe seguindo uma regra determinística de menor custo, ou seja, caminhar sempre para as cidades mais próximas, tal que o custo de se deslocar de uma para outra seja mínimo. Em Campitelli (2007) e Backes *et al.* (2006) onde a CDT foi aplicada sobre imagens, o menor custo é avaliado pela diferença entre os níveis de cinza dos *pixels*.

A fim de se adotar a mesma regra determinística de caminhada, as imagens utilizadas neste trabalho passaram por um pré-processamento (mais detalhes sobre as imagens serão discutidas na Seção 3.2), possibilitando transformá-las para uma faixa com 256 tons de cinza. A partir dessa transformação e respeitando a disponibilidade de *pixels* dada pela janela de memória, a regra determinística de menor custo proposta nos trabalhos citados acima é definida da seguinte forma: **“vá para o *pixel* vizinho que tenha o nível de cinza mais próximo do nível de cinza do *pixel* atual”**. Matematicamente temos:

$$p_{t+1} = \min_{p_i} (|p(t) - p_i(t)|) \mid p_i \in \Gamma_8. \quad (3.1)$$

onde, $p_i(t)$ é o nível de cinza associado ao *pixel* vizinho de p no instante t da caminhada.

Nesta regra determinística de caminhada, o caminhante tende a permanecer em regiões da imagem onde as tonalidades de cinzas são mais homogêneas. Além disso, caminhadas iniciadas em regiões onde as diferenças de tonalidades são altas não se cruzam, possibilitando dessa forma, a identificação de bordas em imagens (CAMPITELLI, 2007) (Figura 3.2). Esta característica é interessante, pelo fato das bordas serem importantes atributos visuais para a caracterização de objetos (COSTA; CESAR JR., 2000). Por esses motivos e também com o propósito de

se manter a originalidade do algoritmo, esta foi a regra escolhida para a movimentação do turista.

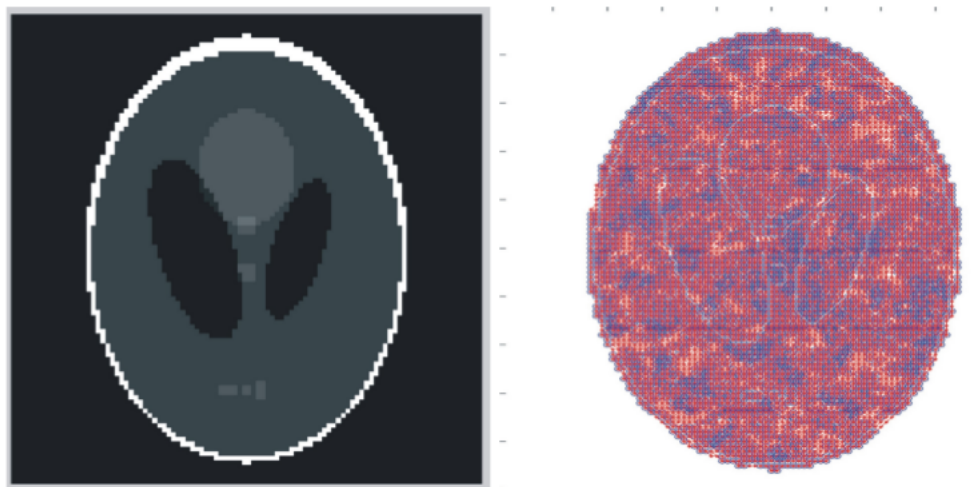


Figura 3.2: A caminhada do turista foi aplicada a imagem da esquerda. As trajetórias (tempo transiente + período atrator) para diferentes valores de memórias foram sobrepostas e são mostradas na imagem à direita. Figura retirada de Campitelli (2007)

3.1.3 Valores utilizados para a Memória

Outro aspecto interessante e discutido foi a memória do turista. No contexto deste trabalho, queremos que as caminhadas possam ser capazes de extrair informações a partir da disposições dos *pixels* de uma determinada imagem.

Foi visto que memórias com valores pequenos iguais a 0 e 1 são incapazes de caracterizar um determinado ambiente, e portanto, são dispensáveis. Por outro lado, para valores grandes, as caminhadas do turista se tornam longas e complexas, aumentando o custo computacional, devido a complexidade em detectar atratores. Além disso, casos raros em que o turista não encontra um vizinho para poder continuar sua trajetória podem se tornar cada vez mais frequentes (Figura 3.3).

acessíveis, a condição de contorno adotada durante a implementação do algoritmo foi a condição periódica de contorno.

3.1.5 Critério de Desempate

Até então, foi apresentado o problema em que o turista fica encurralado e sem chances de continuar com sua caminhada. Este problema ocorre principalmente devido aos valores altos de memória. Entretanto, a relação de vizinhança e a disposições dos *pixels* na imagem, também são fatores que influenciam neste problema.

Um outro problema surge quando dois ou mais *pixels* satisfazem a Equação 3.1. Este problema, denominado de empate, ocorre principalmente devido a regra determinística de caminhada (CAMPITELI, 2007). Uma solução adotada por Campiteli (2007) e Backes *et al.* (2006) se resume em: caso haja um empate a próxima cidade é a primeira cidade no sentido horário. A Figura 3.4 (a), ilustra um momento de empate do turista e a solução propriamente dita. Na imagem, cada *pixel* está associado ao seu valor de cinza e o turista se encontra no *pixel* mais interno, cujo o valor de cinza é 220, além disso, seus vizinhos estão enumerados de 1 a 8 no sentido horário. Devido a regra de movimentação definida na seção anterior, as cidades rotuladas com 1 e 5 sofrem um empate e pela solução proposta, ordena que o *pixel* de rótulo 1 seja a próxima cidade da caminhada.

Esta solução de desempate é interessante pois preserva a natureza determinística do algoritmo (GONÇALVES, 2010). No entanto, diferentes caminhadas podem ser obtidas para uma mesma imagem, caso esta venha sofrer uma transformação de rotação (BACKES *et al.*, 2006). A Figura 3.4 (b) é a mesma em (a), porém rotacionada 90° à esquerda, nota-se agora que o *pixel* a ser escolhido é o *pixel* que havia sido rotulado como 5.

0	0	20	80	52
0	⁸ 45	¹ 230	² 200	52
0	⁷ 80	↑ 220	³ 185	65
0	⁶ 100	⁵ 230	⁴ 58	65
0	112	197	231	36

52	52	65	65	36
80	² 200	³ 185	⁴ 58	231
20	¹ 230	220	↓ ⁵ 230	197
0	⁸ 45	⁷ 80	⁶ 100	112
0	0	0	0	0

(a) (b)

Figura 3.4: O critério de desempate buscando a primeira cidade no sentido horário é variante a rotação.

3.1.6 Estratégia de Detecção de Atratores

A detecção de atratores também é um problema da caminhada determinística do turista, sendo o ponto do algoritmo com o maior custo computacional. Este custo se deve à memória do turista que permite a repetições de cidades sem que isso forme um atrator (BACKES, 2010; GONÇALVES, 2010).

A primeira versão do algoritmo implementado neste trabalho, realizou a detecção de atratores por meio de força bruta, onde duas sequências iguais em uma lista de cidades visitadas era procurada. Entretanto viu-se que, o tempo gasto na realização das caminhadas em imagens era muito alto, isso fez com que novas estratégias fossem estudadas. Por fim, uma estratégia apresentada por Backes *et al.* (2010) foi adotada e permitiu uma boa redução no custo computacional. Ela é descrita a seguir.

Para facilitar a detecção de atratores duas listas são mantidas, uma que armazena todas as *pixels* visitados à medida que o turista caminha pela imagem, e uma segunda lista que mantém as informações dos instantes em que cada um dos *pixels* foi visitado. Na Figura 3.5 podemos ter a ideia de como essas duas listas foram tratadas durante uma caminhada.

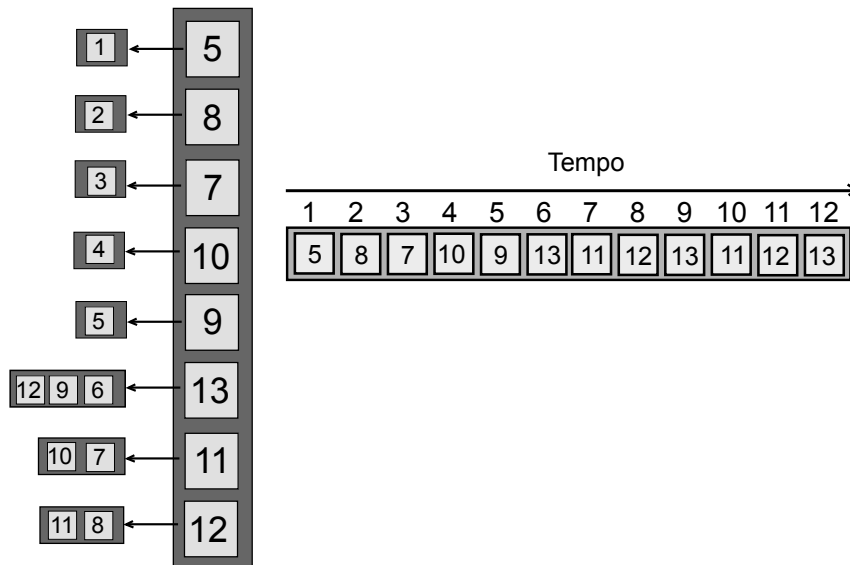


Figura 3.5: Listas auxiliares para a detecção de atratores. A lista a esquerda armazena cada *pixel* e o instante em que ele foi visitado. A lista a direita mantém todos os *pixels* na ordem em que foram visitados. Um atrator é formado, sendo composto pelas cidades 13, 11 e 12.

A Figura 3.5 também ilustra o momento em que um atrator foi formado. Se observarmos as seqüências que vão dos instantes 6 a 8 e de 9 a 11 podemos ver que elas são iguais e que a partir do instante 12 em diante, o turista irá repetir essas mesmas cidades. Portanto, pode-se concluir que é necessário, no mínimo, três visitas para a identificação de um atrator.

Manter uma lista auxiliar que armazena os instantes em que os *pixels* foram visitados é uma estratégia interessante, pois pode-se consultar diretamente esta lista e verificar se o *pixel* atual do turista já foi visitado no mínimo três vezes.

Se sim, os dois trechos, limitados pelos instantes em que este *pixel* foi visitado da lista auxiliar, são comparados. Caso eles sejam iguais então um atrator foi formado. No exemplo descrito pela Figura 3.5, consta-se que o *pixel* atual 13 já foi visitado pela terceira vez, fazendo que os trechos de 6 a 8 e de 9 a 11 sejam comparados na lista a direita, identificando o atrator.

Pode-se observar que esta estratégia para a detecção de atratores evita a busca por força bruta melhorando o desempenho do algoritmo.

3.1.7 Estratégias utilizadas para a extração de características

A Seção 2.5.4 apresentou quatro estratégias para a extração de características da distribuição conjunta resultante da caminhada do turista. Foram elas: Região, Histograma, Propriedades Estatísticas e Regiões de Diferentes Memórias. Neste trabalho, todas foram utilizadas.

A estratégia por Região, tal como foi discutido, extrai as características a partir de uma região da distribuição conjunta. Apesar de não haver uma regra para a escolha da região, em Backes *et al.* (2006) valores para $t_0 = 0, t_f = 4, p_0 = 1, p_f = 5$ apresentaram resultados interessantes. Neste trabalho, foram consideradas regiões do início da distribuição conjunta assim como valores distantes. Em Gonçalves (2010) esta estratégia foi usada e permitiu concluir que as informações mais relevantes estão no início da distribuição conjunta.

Na estratégia por Histograma, o tamanho das caminhadas utilizadas para compor o vetor de características foram: $2+\mu, 3+\mu, 5+\mu, 10+\mu, 20+\mu$ e $35+\mu$. Com esses valores é possível agrupar informações que estão somente no início e também agregar as que estão no início e no fim da distribuição conjunta, para a criação do vetor de característica.

Para a estratégia utilizando regiões de diferentes memórias, adotou-se os mesmos intervalos utilizados para a estratégia por região, variando a junção das distribuições conjuntas obtidas por diferentes memórias. A variação da junção de diferentes distribuições fizeram o uso dos seguintes vetores de características.

$$\psi = [\psi_2, \psi_3] \quad (3.2)$$

$$\psi = [\psi_2, \psi_3, \psi_5] \quad (3.3)$$

$$\psi = [\psi_2, \psi_3, \psi_5, \psi_7] \quad (3.4)$$

$$\psi = [\psi_2, \psi_3, \psi_5, \psi_7, \psi_{11}] \quad (3.5)$$

onde ψ_i é o vetor de característica obtido pela estratégia por Região da distribuição conjunta resultante da memória i .

Por último, a estratégia utilizando propriedades estatísticas foi aplicada sobre a distribuição conjunta que melhor descreveu as texturas. Em Gonçalves (2010), avaliou-se cada uma das propriedades. Aquelas que resultaram os quatro melhores desempenhos foram agrupadas e reavaliadas. Neste sentido, além da aplicação das propriedades estatísticas sobre a distribuição conjunta que melhor descreveu a textura, também serão adotadas as estratégias descritas pelo trabalho citado. Entretanto, serão agrupadas somente as duas propriedades de melhor desempenho, para que possa ser possível visualizar a representação espacial das classes de imagens.

3.2 Bases de Dados

A caminhada do turista considerando os aspectos discutidos nas seções anteriores foi aplicada sobre duas bases de dados de imagens. São elas: *Outex* (OJALA *et al.*, 2002) e a que denominaremos neste trabalho de *BTexCA - Base de Texturas de Cana de Açúcar*. Elas são discutidas a seguir.

3.2.1 Outex

A base de dados *Outex* criada por Ojala *et al.* (2002), contém uma variedade de texturas com o propósito de aplicações de algoritmos de classificação (supervisionados e não supervisionados) e segmentação. Atualmente ela contém 320 tipos de texturas provenientes de superfícies naturais apresentando variações de iluminação, rotação e escala em suas estruturas. Além disso, a base é de acesso livre¹ e tem a contribuição de diversos grupos de pesquisas.

Neste trabalho, as classes de texturas utilizadas foram retiradas da coleção *Outex_TC_00012*², que se trata de um conjunto de 24 classes de texturas de dimensões 128x128 em níveis de cinza, para serem empregadas em problemas de classificação. As classes selecionadas foram: *canvas001* e *canvas002*, cada uma com 20 amostras de treinamento e 180 amostras de testes, estas com rotações de 45°, 90°, 135°, 180° e 225°. A Figura 3.6 apresenta uma amostra para cada uma das classes.

As proporções em que os dados nesta base estão separados seguem de 10% para treinamento e 90% para testes. Normalmente, o que se faz é separar mais amostras para o conjunto de treinamento do que para o conjunto de testes. Em

¹Disponível em: <http://www.outex.oulu.fi>. Acessado em 20 de Julho de 2013.

²Disponível em: <http://www.outex.oulu.fi/index.php?page=classification>. Acessado em 20 de Julho de 2013.

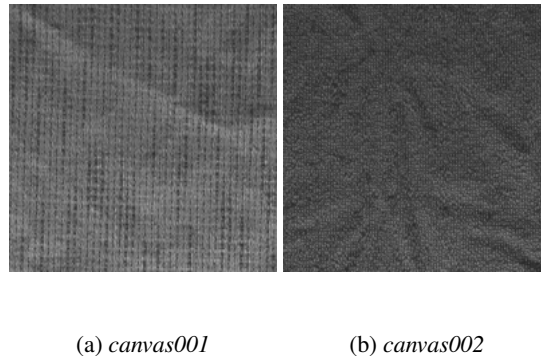


Figura 3.6: Uma amostra de cada uma das classes retiradas da base *Outex*.

Haykin (2001) é sugerido 50% das amostras para treinamento, 25% para validação cruzada e 25% para testes.

Por esses motivos e também para tornar o processo de separação da base de dados *Outex* similar ao que foi feito com a base de dados das imagens de canas, 70% das 200 amostras foram separadas para compor o conjunto de treinamento, sendo as restantes utilizadas para validar os modelos SVMs. A escolha das 70% das amostras foi feita de maneira aleatória, para que as amostras sobre o efeito das rotações tenham as mesmas chances de serem selecionadas diante das outras.

Os resultados obtidos pela aplicação da CDT sobre a coleção *Outex_TC_00012* na sua forma original é apresentado pelo Apêndice A.

3.2.2 BTexCA

A *BTexCA*, *Base de Texturas de Canas de Açúcar*, é uma base de dados criada para atender os objetivos deste trabalho. Ela contém 800 amostras de texturas, no qual 400 são provenientes de cortes manuais e 400 de cortes mecânicos.

A base de dados foi construída seguindo as etapas ilustrada pela Figura 3.7 com o auxílio do *Software Matlab*³. Essas etapas são descritas a seguir.

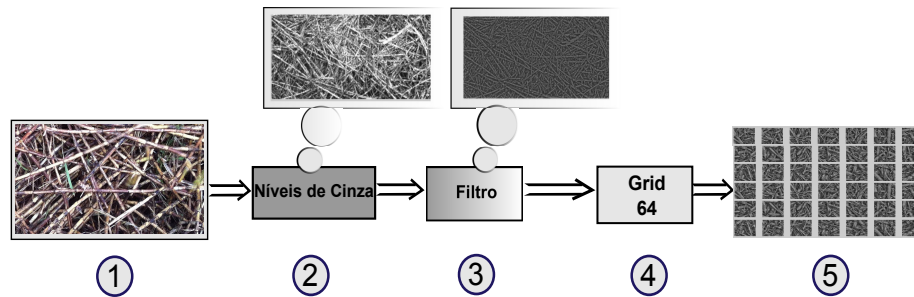
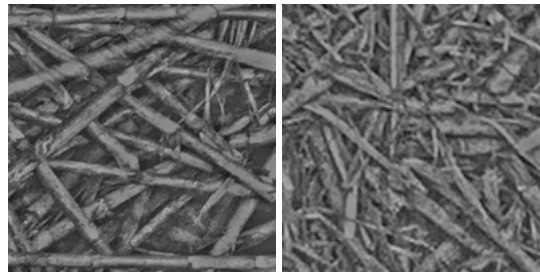


Figura 3.7: Etapas realizadas para a construção da base BTexCA

Na etapa 1 a região de uma imagem de alta dimensão foi selecionada, derivando outra imagem que exibia somente um dado tipo de corte (manual ou mecânico). Esta nova imagem, ainda relativamente grande, foi apresentada à função, etapa 2, cuja saída foi a imagem em níveis de cinza. Em seguida, a imagem proveniente foi apresentada para uma segunda função, etapa 3, um filtro que possibilitou retirar as diferenças de iluminação presentes na imagem. Na etapa 4, uma função de *grid* permitiu recortar a imagem em várias outras de dimensões 64x64, posteriormente salvas em uma base de dados, etapa 5. Este processo repetiu-se por 5 imagens de alta dimensão, gentilmente fornecidas por uma empresa de consultoria do setor de Automação Industrial, até que fossem obtidas 400 imagens de cortes mecânicos e 400 de cortes manuais, criando a base de dados final, denominada *BTexCA*. As proporções de amostras usadas no treinamento e teste foram respectivamente, 70% e 30%. A Figura 3.8 mostra uma amostra de dados para cada uma das classes retirada desta base de dados.

³ <http://www.mathworks.com/products/matlab/>. Acessado em 20 de Julho de 2013.



(a) Corte Manual

(b) Corte Mecânico

Figura 3.8: Uma amostra de cada uma das classes retiradas da base *BTexCA*.

3.3 Modelos de Classificação

A Seção 2.3 apresentou os conceitos por trás das Máquinas de Vetores de Suporte e suas principais características, dentre elas a capacidade de generalização e a robustez em grandes dimensões (LORENA; CARVALHO, 2007), tornando-as ferramentas atrativas para este trabalho.

Neste trabalho investigamos dois tipos de modelos SVMs: SVM com *kernel linear* e SVM com *kernel RBF*. Para a implementação dos modelos SVMs contamos com o auxílio da biblioteca *LIBSVM*⁴(CHANG; LIN, 2011), que provê uma interface simples de programação, modelos SVMs já implementados e extensões para o *Matlab*.

A estratégia usada para a escolha do parâmetro C para *kernel linear* e a melhor combinação (C, γ) para o *kernel RBF* foi a estratégia de busca em *grid* com *5-fold crossvalidation*, onde $2^{-5} \leq C \leq 2^{15}$ e $2^{-15} \leq \gamma \leq 2^3$. Também avaliamos modelos sem o uso da validação cruzada, para isso os valores de C e γ utilizados

⁴Disponível em: <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>. Acessado em 20 de Julho de 2013

foram padrões da biblioteca *LIBSVM*, respectivamente iguais a 1 e $1/(\text{número de características do conjunto de treinamento})$.

3.4 Configurações de Hardware e Softwares utilizados

O Hardware e o Sistema Operacional, no qual foram utilizados os testes, têm as seguintes configurações:

- Sistema Operacional Ubuntu 11.04, 64 bits.
- Core i3 de 8 GB de memória.

4 RESULTADOS E DISCUSSÕES

Neste Capítulo apresentaremos os resultados obtidos. Primeiramente, iremos mostrar os resultados obtidos sobre a base de dados *Outex*, seguidos da base de dados *BTexCA*.

A obtenção dos resultados partiu da aplicação da CDT sobre as imagens. Para cada valor de memória utilizado, foi obtida uma determinada distribuição conjunta, sobre a qual foram utilizadas as estratégias de extração de características para a geração dos vetores de características. Estes, foram usados para treinar e avaliar modelos SVMs.

4.1 Base Outex

Esta Seção apresenta os resultados obtidos sobre a base de dados *Outex*. Para esta base foi considerado os valores de memórias iguais a 2, 3, 5 e 7.

4.1.1 Estratégia por Região

Os resultados obtidos pelos modelos SVMs, *Linear* e *RBF*, utilizando a estratégia por Região são mostrados, respectivamente, pelas Tabelas 4.1 e 4.2. Nas tabelas, a primeira coluna refere-se ao intervalo considerado para extrair as características da distribuição conjunta. Os valores apresentados indicam o percentual de amostras do conjunto de teste corretamente classificadas pelos modelos SVMs, portanto quanto maior o valor melhor é o desempenho do modelo SVM. Também é indicado se os modelos fizeram ou não o uso da validação cruzada. Se sim, então os resultados são apresentados pelas colunas CV (com validação cruzada), caso contrário são apresentados pelas colunas SV (sem validação cruzada). As tabe-

las também indicam o valor da memória utilizada pelo turista para a obtenção da distribuição conjunta.

Tabela 4.1: Porcentagem de acerto dos modelos SVM com *kernel linear* sobre a base de dados *Outex*, utilizando a estratégia de extração de características por Região.

Intervalo $[t_0, t_f]$ e $[p_0, p_f]$	Memórias							
	2		3		5		7	
	SV	CV	SV	CV	SV	CV	SV	CV
[0, 1] e [1, 2]	55.83%	72.50%	88.33%	94.16%	58.33%	74.16%	75.00%	78.33%
[0, 4] e [1, 5]	67.50%	87.50%	92.50%	96.66%	79.16%	83.33%	91.66%	93.33%
[3, 5] e [1, 3]	71.66%	75.00%	63.33%	64.16%	67.50%	75.83%	80.00%	80.83%
[6, 8] e [4, 7]	55.83%	55.83%	65.83%	65.83%	64.16%	65.00%	67.50%	70.00%
[9, 11] e [5, 8]	55.00%	55.00%	52.50%	52.50%	56.66%	60.00%	59.16%	59.16%

Tabela 4.2: Porcentagem de acerto dos modelos SVM com *kernel RBF* sobre a base de dados *Outex*, utilizando a estratégia de extração de características por Região.

Intervalo $[t_0, t_f]$ e $[p_0, p_f]$	Memórias							
	2		3		5		7	
	SV	CV	SV	CV	SV	CV	SV	CV
[0, 1] e [1, 2]	55.83%	75.00%	88.33%	94.16%	58.33%	74.16%	75.00%	81.66%
[0, 4] e [1, 5]	67.50%	86.66%	92.50%	92.50%	78.33%	81.66%	91.66%	92.50%
[3, 5] e [1, 3]	71.66%	71.66%	63.33%	61.66%	67.50%	75.00%	80.00%	80.83%
[6, 8] e [4, 7]	55.83%	54.16%	65.83%	65.00%	64.16%	63.33%	67.50%	70.00%
[9, 11] e [5, 8]	53.33%	58.33%	54.16%	52.50%	56.66%	58.33%	59.16%	55.83%

Podemos observar que o melhor resultado obtido foi de 96.66% pelo modelo SVM de *kernel linear*, com o uso da validação cruzada. As características extraídas considerou a região $t_0 = 0$, $t_f = 4$, $p_0 = 1$ e $p_f = 5$ da distribuição conjunta obtida para memória do turista igual a 3. Em Backes *et al.* (2006) este mesmo intervalo conseguiu resultados interessantes.

Também é observado que o desempenho de ambos os modelos é melhor quando consideramos regiões do início da distribuição conjunta. Note que a porcentagem de acertos tende a diminuir para intervalos distantes (três últimas linhas de cada tabela). Este aspecto foi avaliado em Gonçalves (2010), no qual é concluído que os valores mais representativos para a composição das características estão no início da distribuição conjunta.

4.1.2 Estratégia por Histograma

Os resultados utilizando a estratégia baseada em Histograma são mostrados nas Tabelas 4.3 para os modelos SVMs com *kernel linear* e 4.4 com *kernel RBF*. Aqui, a primeira coluna das tabelas indicam o tamanho das caminhadas consideradas para a construção do vetor de característica.

Tabela 4.3: Porcentagem de acerto do modelo SVM com *kernel linear* sobre a base de dados *Outex*, utilizando a estratégia de extração de características por Histograma.

$t + p$	Memórias							
	2		3		5		7	
	SV	CV	SV	CV	SV	CV	SV	CV
$2+\mu$	61.66%	62.50%	85.83%	88.33%	63.33%	65.83%	74.16%	74.16%
$3+\mu$	58.33%	55.83%	85.00%	87.50%	67.50%	73.33%	82.50%	85.00%
$5+\mu$	60.00%	60.00%	89.16%	90.83%	80.00%	84.16%	94.16%	95.00%
$10+\mu$	65.00%	66.66%	91.66%	91.66%	90.00%	90.00%	90.83%	93.33%
$20+\mu$	68.33%	66.66%	91.66%	92.50%	86.66%	88.33%	90.00%	92.50%
$35+\mu$	64.16%	68.33%	90.83%	91.66%	86.66%	90.00%	89.16%	92.50%

Para esta estratégia, a melhor acurácia foi do modelo SVM com *kernel linear*, cujo valor foi de 95%. A extração de características ocorreu a partir da distribuição conjunta obtida com o uso da memória igual a 7, com caminhadas de tamanho $5 + \mu$. O tamanho da caminhada considerada é coerente com o intervalo

Tabela 4.4: Porcentagem de acerto do modelo SVM com *kernel RBF* sobre a base de dados *Outex*, utilizando a estratégia de extração de características por Histograma.

$t + p$	Memórias							
	2		3		5		7	
	SV	CV	SV	CV	SV	CV	SV	CV
$2 + \mu$	61.66%	60.83%	85.83%	88.33%	63.33%	66.66%	74.16%	75.83%
$3 + \mu$	58.33%	55.83%	85.00%	87.50%	67.50%	68.33%	82.50%	85.00%
$5 + \mu$	60.00%	58.33%	89.16%	92.50%	80.00%	83.33%	94.16%	94.16%
$10 + \mu$	65.00%	65.00%	91.66%	91.66%	90.00%	90.00%	90.83%	93.33%
$20 + \mu$	68.33%	64.16%	91.66%	91.66%	86.66%	87.63%	90.00%	91.66%
$35 + \mu$	64.16%	68.33%	90.83%	90.83%	86.66%	86.82%	89.16%	90.11%

do melhor resultado obtido pela estratégia anterior, uma vez que as caminhadas deste tamanho são limitadas pelos intervalos $t_0 = 0$, $t_f = 4$, $p_0 = 1$ e $p_f = 5$.

Diferente do que foi obtido em Gonçalves (2010), os modelos não tiveram seus resultados prejudicados à medida que o intervalo foi aumentado e, de certa forma, permaneceram ligeiramente próximos a partir de $5 + \mu$. Isso é aceitável, uma vez que os valores de interesse, estão no início da distribuição conjunta e diferente da estratégia por região, onde podemos considerar valores particulares da distribuição conjunta para compor o vetor de características final, nesta estratégia, mesmo agrupando os valores que estão distantes na distribuição, os valores no início sempre serão considerados.

4.1.3 Estratégia por Regiões de Diferentes Memórias

A realização dos testes usando a estratégia de extração de características por regiões de diferentes memórias, adotou os mesmos intervalos utilizados na estratégia por região. Os resultados são apresentados pelas Tabelas 4.5 e 4.6.

Tabela 4.5: Porcentagem de acerto do modelo SVM com *kernel linear* sobre a base de dados *Outex*, utilizando a estratégia de extração de características por Região de Diferentes Memórias.

Intervalo $[t_0, t_f]$ e $[p_0, p_f]$	Memórias					
	[2, 3]		[2, 3, 5]		[2, 3, 5, 7]	
	SV	CV	SV	CV	SV	CV
[0, 1] e [1, 2]	80.83%	95.00%	79.16%	96.66%	83.33%	97.50%
[0, 4] e [1, 5]	90.00%	95.00%	92.50%	95.83%	95.83%	96.66%
[3, 5] e [1, 3]	70.00%	75.00%	76.66%	77.50%	81.66%	78.33%
[6, 8] e [4, 7]	55.83%	55.83%	66.66%	66.66%	66.66%	74.16%
[9, 11] e [5, 8]	52.50%	54.16%	58.33%	58.33%	56.66%	63.33%

Tabela 4.6: Porcentagem de acerto do modelo SVM com *kernel RBF* sobre a base de dados *Outex*, utilizando a estratégia de extração de características por Região de Diferentes Memórias.

Intervalo $[t_0, t_f]$ e $[p_0, p_f]$	Memórias					
	[2, 3]		[2, 3, 5]		[2, 3, 5, 7]	
	SV	CV	SV	CV	SV	CV
[0, 1] e [1, 2]	80.83%	90.00%	79.16%	92.50%	83.33%	94.16%
[0, 4] e [1, 5]	90.00%	90.00%	92.50%	92.63%	95.83%	96.66%
[3, 5] e [1, 3]	70.00%	75.83%	76.66%	75.83%	81.66%	78.33%
[6, 8] e [4, 7]	66.66%	61.66%	66.66%	67.50%	66.66%	72.50%
[9, 11] e [5, 8]	53.33%	56.66%	58.33%	57.50%	56.66%	52.50%

Como é observado, esta foi a estratégia de maior percentual de acertos, com 97.50% considerando regiões onde $t_0 = 0$, $t_f = 1$, $p_0 = 1$ e $p_f = 2$ de distribuições conjuntas obtidas de memórias iguais a 2, 3, 5 e 7.

As mesmas observações feitas pela utilização da estratégia por região também podem ser analisadas para esta estratégia, segue: melhor acurácia para regiões no início da distribuição conjunta e a contribuição da validação cruzada para a melhoria dos resultados.

Melhores valores de acurácia utilizando esta estratégia podem ser explicados com base na Figura 3.2, mostrada na Seção 3.1.2. Na figura, a junção de diferentes trajetórias obtidas por diferentes memórias permitiu a reconstrução de uma determinada imagem. Neste sentido, a utilização da estratégia por regiões de diferentes memórias possibilitou uma descrição mais geral das texturas das imagens, permitindo obter mais características que distinguem melhor as duas classes e conseqüentemente melhorando a acurácia dos modelos.

Foi observado até então que as distribuições conjuntas obtidas a partir das memórias 3 e 7, dentre as utilizadas, são as que melhor descrevem as texturas desta base de dados. Por este motivo, também foi feito o uso da estratégia de regiões de diferentes memórias, considerando o intervalo $t_0 = 0$, $t_f = 1$, $p_0 = 1$ e $p_f = 2$, para essas distribuições, já que, somente as duas não foram consideradas. A taxa de acerto obtida foi de 98.33%, para um modelo SVM com *kernel linear* e com o uso de validação cruzada. Este resultado mostra que a junção de informações da distribuição conjunta, cujo resultados não são satisfatórios, podem acabar prejudicando o desempenho dos modelos.

4.1.4 Estratégia utilizando Propriedades Estatísticas

Apesar da distribuição conjunta obtida para a memória igual a 7 ter apresentado bons resultados, no geral, a distribuição conjunta obtida pela memória de valor 3 obteve o melhor desempenho. Para verificarmos se todas as informações desta distribuição conjunta podem ser usadas para descreverem as classes de texturas, aplicamos as propriedades estatísticas sobre esta distribuição. Os resultados são apresentados pela Figura 4.1. A Figura 4.2 apresenta parte da distribuição conjunta obtida pela memória igual a 3 para cada uma das classes.

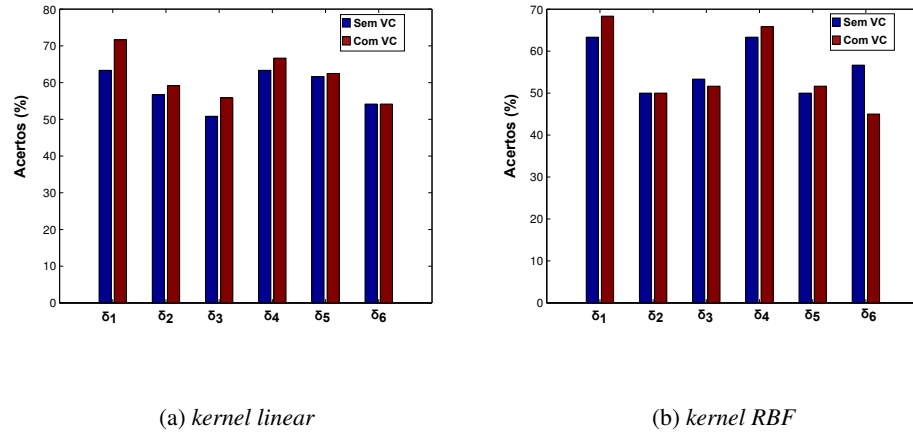
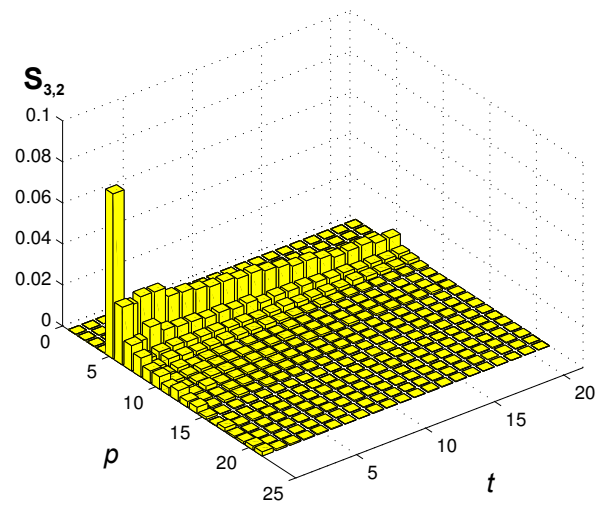


Figura 4.1: Porcentagem de acertos para cada uma das propriedades estatísticas sobre a distribuição conjunta obtida para a memória do turista igual a 3, sobre a base de dados *Outex*. Em (a) a acurácia dos modelos SVMs de *kernel linear*. Em (b) a acurácia dos modelos SVMs de *kernel RBF*.

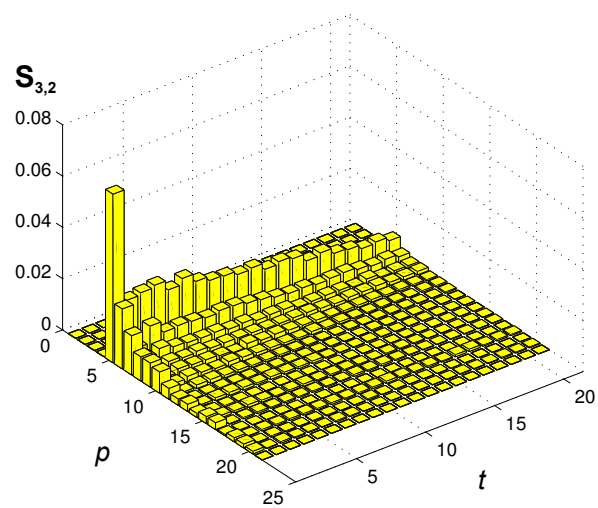
Na figura, também é apresentado os resultados obtidos com o uso da validação cruzada (em azul) e sem o uso da mesma (em vermelho). Pelos resultados, as propriedades com os melhores resultados foram δ_1 e δ_4 , respectivamente definidos pelas Equações 2.7 e 2.10.

A junção das duas propriedades para a formação de um novo vetor de características apresentou uma taxa de 66.66% de acerto tanto para modelos SVMs de *kernel linear*, quanto para os de *kernel RBF*, com o uso da validação cruzada.

Também foi avaliado a discriminação dos modelos para todas as propriedades em conjunto. Os resultados obtidos foram de 50% para o modelo de *kernel RBF* e 59.16% com *kernel linear*, ambos com o uso da validação cruzada.



(a) Canvas001



(b) Canvas002

Figura 4.2: Parte da distribuição conjunta de cada uma das classes obtida para a memória do turista igual a 3 ($t_0 = 0, p_0 = 0, t_f = 20$ e $p_f = 20$).

Para esta base de dados as propriedades estatísticas não indicaram bons resultados, significando que considerar todas as informações da distribuição conjunta não é interessante. Em Gonçalves (2010) a energia (δ_1) também foi a propriedade que obteve o melhor resultado, sendo que o uso das 4 melhores estatísticas resultou em uma acurácia de 91.98%.

A justificativa para esses resultados obtidos com o uso das propriedades estatísticas, também pode ser explicado com base nas Figuras 4.3 e 4.4. Na figura, pode-se notar que as classes estão praticamente sobrepostas resultando em uma separação ruim por parte do classificador.

Foi possível realizar esta apresentação espacial entre as classes devido a junção das duas melhores propriedades estatísticas (δ_1 e δ_4) obtidas.

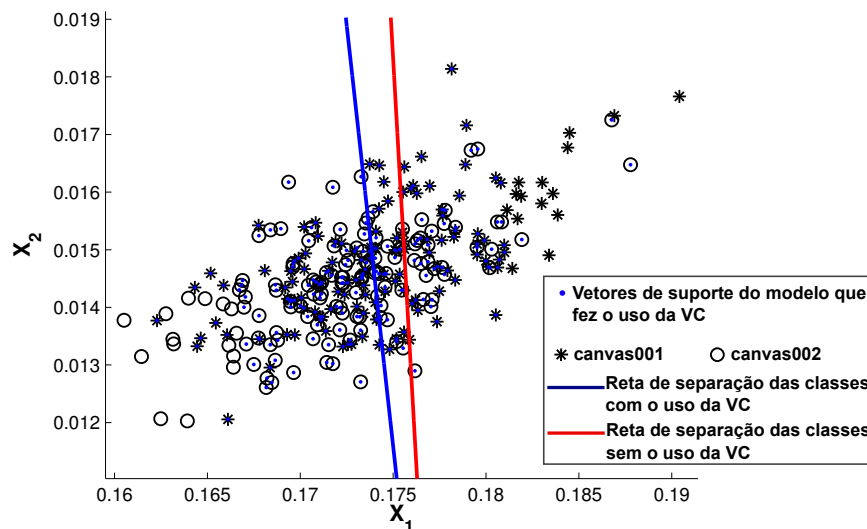


Figura 4.3: Representação espacial entre as classes utilizadas pela bases de dados *Outex*. A imagem mostra as superfícies ótimas de separações para um modelo SVM com *kernel linear*, com e sem o uso da validação cruzada. X_1 e X_2 representam, respectivamente, δ_1 e δ_4

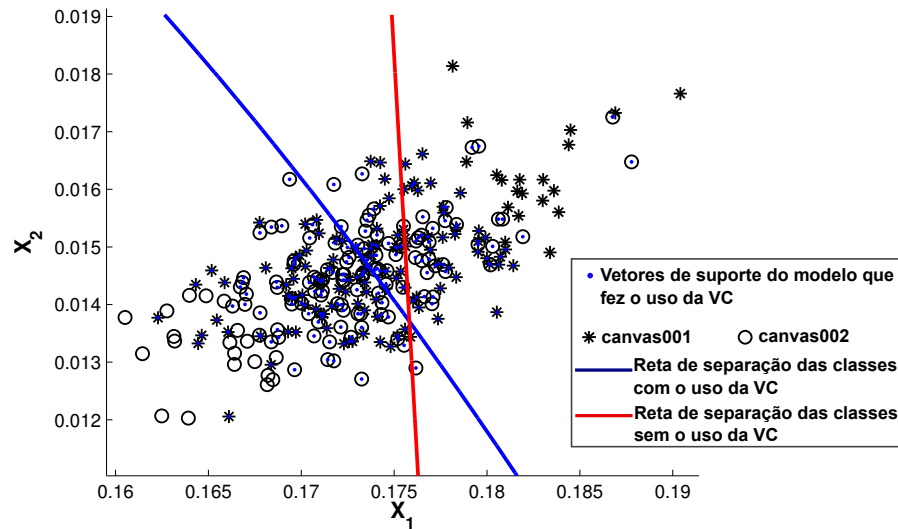


Figura 4.4: Representação Espacial entre as classes utilizadas pela bases de dados *Outex*. A imagem mostra as superfícies ótimas de separações para um modelo SVM com *kernel RBF*, com e sem o uso da validação cruzada. X_1 e X_2 representam, respectivamente, δ_1 e δ_4

4.2 Base BTexCA

A seguir são apresentados e discutidos os resultados obtidos para a base de dados *BTexCA*.

4.2.1 Estratégia por Região

Os resultados obtidos pelos modelos SVMs sobre a base de dados *Outex*, nos permitiu concluir que os valores mais interessantes para a representação das texturas estão no início da distribuição conjunta. Devido a isso, resolveu variar um pouco mais os intervalos do início das distribuições conjuntas e desconsiderar intervalos muito distantes. Além disso, as distribuições conjuntas obtidas para essa base de dados resultaram de valores de memórias iguais a 2, 3, 5, 7 e 11.

Na utilização do modelo SVM com *kernel linear* obtemos os resultados mostrados pela Tabela 4.7. A melhor porcentagem de classificação obtida foi de 95%, onde a região da distribuição conjunta considerada foi $t_0 = 0$, $t_f = 2$, $p_0 = 1$ e $p_f = 3$ obtida da memória do turista igual a 2.

Tabela 4.7: Porcentagem de acerto dos modelos SVM com *kernel linear* sobre a base de dados *BTexCA*, utilizando a estratégia de extração de características por Região.

Intervalo $[t_0, t_f]$ e $[p_0, p_f]$	Memórias									
	2		3		5		7		11	
	SV	CV	SV	CV	SV	CV	SV	CV	SV	CV
[0, 1] e [1, 2]	88.33%	93.33%	75.83%	85.00%	55.41%	68.75%	60.00%	67.50%	54.58%	65.83%
[0, 2] e [1, 3]	89.58%	95.00%	76.25%	88.75%	53.75%	69.58%	59.58%	68.75%	57.50%	64.16%
[0, 3] e [1, 4]	90.41%	93.33%	77.08%	87.91%	55.41%	69.16%	62.91%	70.41%	54.58%	68.75%
[0, 4] e [1, 5]	90.83%	92.50%	77.91%	88.33%	56.66%	65.00%	61.66%	71.66%	53.33%	70.41%
[0, 5] e [1, 6]	90.41%	92.50%	76.66%	87.08%	56.25%	69.58%	60.83%	74.58%	56.25%	70.83%
[0, 7] e [1, 8]	90.00%	92.91%	75.83%	88.75%	55.41%	71.66%	59.16%	72.50%	57.91%	72.08%
[3, 4] e [1, 2]	72.91%	81.66%	75.83%	76.25%	56.25%	58.75%	48.33%	49.16%	50.41%	46.25%
[3, 5] e [1, 3]	73.33%	82.08%	76.66%	75.00%	57.50%	62.50%	52.08%	59.58%	48.75%	49.16%
[3, 4] e [2, 3]	67.91%	68.75%	56.66%	57.91%	60.83%	61.66%	47.08%	49.16%	50.41%	53.33%
[3, 5] e [2, 4]	65.00%	69.58%	55.00%	58.33%	57.91%	65.41%	50.83%	50.41%	52.08%	52.91%

Uma melhor visualização dos resultados da Tabela 4.7 é mostrado pela Figura 4.5. Em (a) é apresenta a porcentagem de acertos obtida pelos modelos que fizeram o uso da validação cruzada. Em (b) é apresentado a porcentagem média do aumento da taxa de acertos pelos modelos que adotaram o uso da validação cruzada.

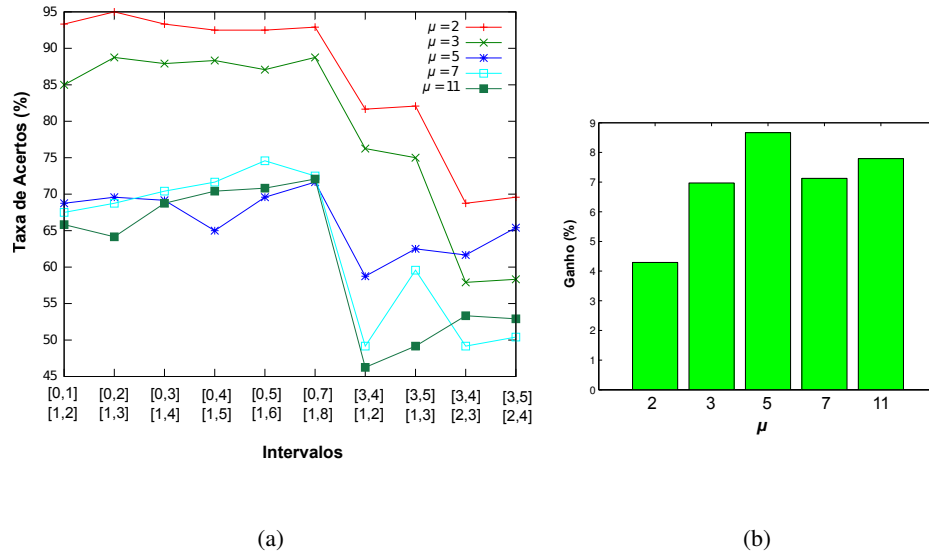


Figura 4.5: Em (a) a porcentagem de acertos obtidos pelos modelos com validação cruzada para cada um dos intervalos. Em (b) o ganho médio em porcentagem obtido pelo uso da validação cruzada.

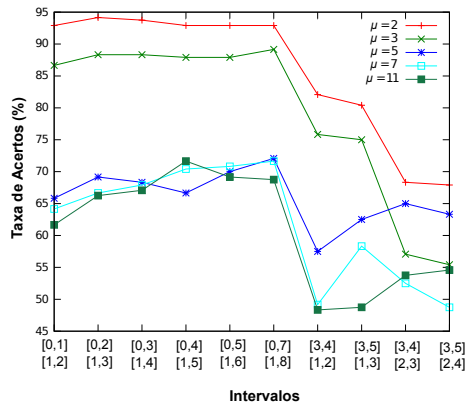
Para o modelo com *kernel RBF* os resultados são mostrados na Tabela 4.8. Neste modelo o melhor resultado obtido foi 94.16% de acertos. A Figura 4.6 ilustra a porcentagem de acertos para os modelos que fizeram o uso da validação cruzada e a contribuição da mesma para o aumento da porcentagem de acertos.

Assim como os resultados obtidos sobre a base de dados *Outex*, nós observamos mais uma vez que valores do início da distribuição mostraram ser mais interessantes para o desempenho dos modelos SVMs e o quanto a validação cruzada é importante para a melhoria da acurácia dos mesmos.

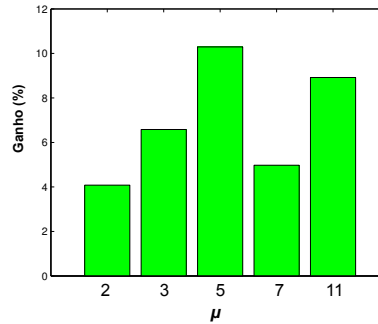
Outro aspecto interessante foram as distribuições conjuntas obtidas. Na base *Outex* as distribuições que melhor descreviam as texturas foram obtidas a partir da memória igual a 3 e 7. Para a base *BTexCA*, memórias altas não mostraram resultados satisfatórios, sendo as distribuições obtidas de memórias iguais a

Tabela 4.8: Porcentagem de acerto dos modelos SVMs com *kernel RBF* sobre a base de dados *BTexCA*, utilizando a estratégia de extração de características por Região.

Intervalo $[t_0, t_f]$ e $[p_0, p_f]$	Memórias									
	2		3		5		7		11	
	SV	CV	SV	CV	SV	CV	SV	CV	SV	CV
[0, 1] e [1, 2]	88.75%	92.91%	75.41%	86.66%	54.58%	65.83%	56.66%	64.16%	53.75%	61.66%
[0, 2] e [1, 3]	89.58%	94.16%	76.25%	88.33%	53.33%	69.16%	62.91%	66.66%	54.16%	66.25%
[0, 3] e [1, 4]	90.41%	93.75%	77.08%	88.33%	53.75%	68.33%	58.33%	67.91%	56.25%	67.08%
[0, 4] e [1, 5]	90.83%	92.91%	77.91%	87.91%	56.25%	66.66%	58.75%	70.41%	55.00%	71.66%
[0, 5] e [1, 6]	90.41%	92.91%	76.66%	87.91%	56.25%	70.00%	60.00%	70.83%	54.16%	69.16%
[0, 7] e [1, 8]	90.00%	92.91%	77.50%	89.16%	55.00%	72.08%	58.75%	71.66%	57.08%	68.75%
[3, 4] e [1, 2]	72.08%	82.08%	76.25%	75.83%	51.25%	57.50%	48.75%	59.16%	48.33%	48.33%
[3, 5] e [1, 3]	73.33%	80.41%	77.08%	75.00%	58.33%	62.50%	51.25%	58.33%	49.16%	48.75%
[3, 4] e [2, 3]	67.08%	68.33%	56.66%	57.08%	60.83%	65.00%	50.41%	52.50%	50.83%	53.75%
[3, 5] e [2, 4]	65.00%	67.91%	55.00%	55.41%	57.91%	63.33%	50.00%	48.75%	52.08%	54.58%



(a)



(b)

Figura 4.6: Em (a) a porcentagem de acertos obtidos pelos modelos com validação cruzada para cada um dos intervalos. Em (b) o ganho médio em porcentagem obtido pelo uso da validação cruzada.

2 e 3 mais adequadas. Isso mostra que não há necessidade do turista realizar caminhadas tão distantes para obter informações suficientes para a representação da

textura, uma vez que as informações mais relevantes estão no início da distribuição conjunta, bem como para valores pequenos de memórias.

4.2.2 Estratégia por Histograma

Os resultados usando a estratégia por Histograma são apresentados pela Tabela 4.9 para modelos SVM de *kernel linear* e pela Tabela 4.10 para modelos de *kernel RBF*. A ilustração dos resultados são mostrados pelas Figuras 4.7 e 4.8.

Tabela 4.9: Porcentagem de acerto do modelo SVM de *kernel linear* sobre a base de dados *BTexCA*, utilizando a estratégia de extração de características por Histograma.

$t + p$	Memórias									
	2		3		5		7		11	
	SV	CV	SV	CV	SV	CV	SV	CV	SV	CV
2+ μ	92.08%	92.50%	75.00%	75.83%	53.33%	52.50%	46.66%	49.58%	48.33%	48.33%
3+ μ	92.50%	92.91%	73.33%	78.75%	55.41%	60.41%	50.41%	48.75%	48.75%	49.16%
5+ μ	92.91%	92.91%	76.66%	79.16%	54.58%	54.58%	51.25%	52.91%	47.08%	47.08%
10+ μ	91.66%	91.66%	78.33%	80.41%	60.00%	59.58%	53.33%	50.41%	48.33%	48.33%
20+ μ	91.25%	91.25%	77.91%	80.41%	61.66%	65.83%	55.00%	55.41%	46.66%	43.75%
35+ μ	92.08%	92.08%	77.08%	80.83%	62.08%	65.83%	56.25%	52.50%	51.66%	45.83%

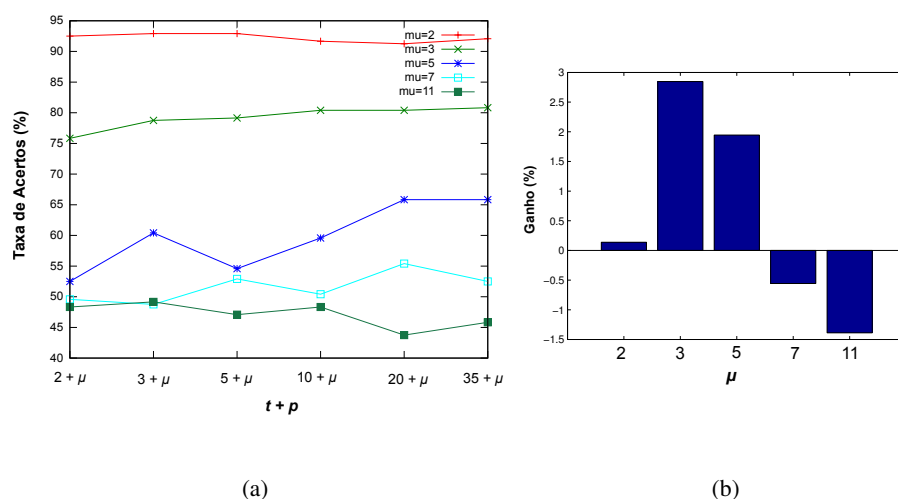


Figura 4.7: Em (a) a porcentagem de acertos obtidos pelos modelos com validação cruzada para cada tamanho de caminhada ($t + p$). Em (b) o ganho médio em porcentagem obtido pelo uso da validação cruzada.

Tabela 4.10: Porcentagem de acerto do modelo SVM de *kernel RBF* sobre a base de dados *BTexCA*, utilizando a estratégia de extração de características por Histograma.

$t+p$	Memórias									
	2		3		5		7		11	
	SV	CV	SV	CV	SV	CV	SV	CV	SV	CV
$2+\mu$	92.08%	92.50%	75.00%	75.41%	53.33%	53.75%	46.66%	49.58%	48.33%	46.66%
$3+\mu$	92.50%	92.91%	73.33%	78.75%	55.41%	52.91%	50.41%	50.41%	48.75%	50.41%
$5+\mu$	92.91%	92.91%	76.66%	79.16%	54.58%	54.58%	51.25%	52.50%	47.08%	47.08%
$10+\mu$	91.66%	91.66%	78.33%	80.41%	60.00%	60.00%	53.75%	55.41%	48.33%	48.33%
$20+\mu$	91.25%	91.25%	77.91%	80.41%	61.66%	65.83%	55.00%	55.00%	46.60%	50.00%
$35+\mu$	92.08%	92.08%	77.08%	79.16%	62.08%	65.83%	56.25%	52.91%	51.66%	46.66%

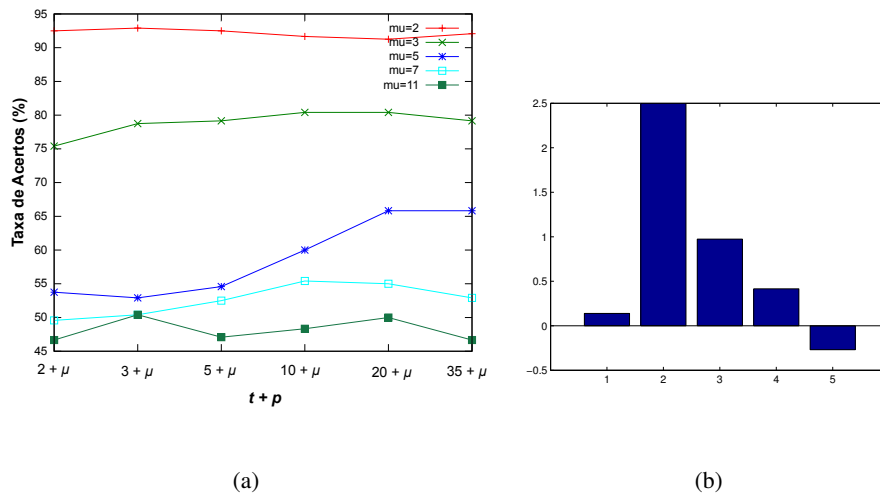


Figura 4.8: Em (a) a porcentagem de acertos obtidos pelos modelos com validação cruzada para cada tamanho de caminhada ($t + p$). Em (b) o ganho médio em porcentagem obtido pelo uso da validação cruzada.

Esta estratégia reforça ainda mais a proposta de que os valores distantes não são representativos. Pode-se notar que para as distribuições conjuntas obtidas de memórias iguais a 2 e 3 a taxa de acerto foi quase que constante, isto também foi observado na base *Outex* a partir de caminhadas $5 + \mu$.

Um caso a ser discutido é o que pode ser visto pelas Figuras 4.7 e 4.8 (b), onde a validação cruzada não contribuiu no aumento da acurácia para a maioria dos modelos. Este fato também foi observado em alguns resultados obtidos na utilização da base *Outex* e pode melhor ser explicado através da Figura 4.9.

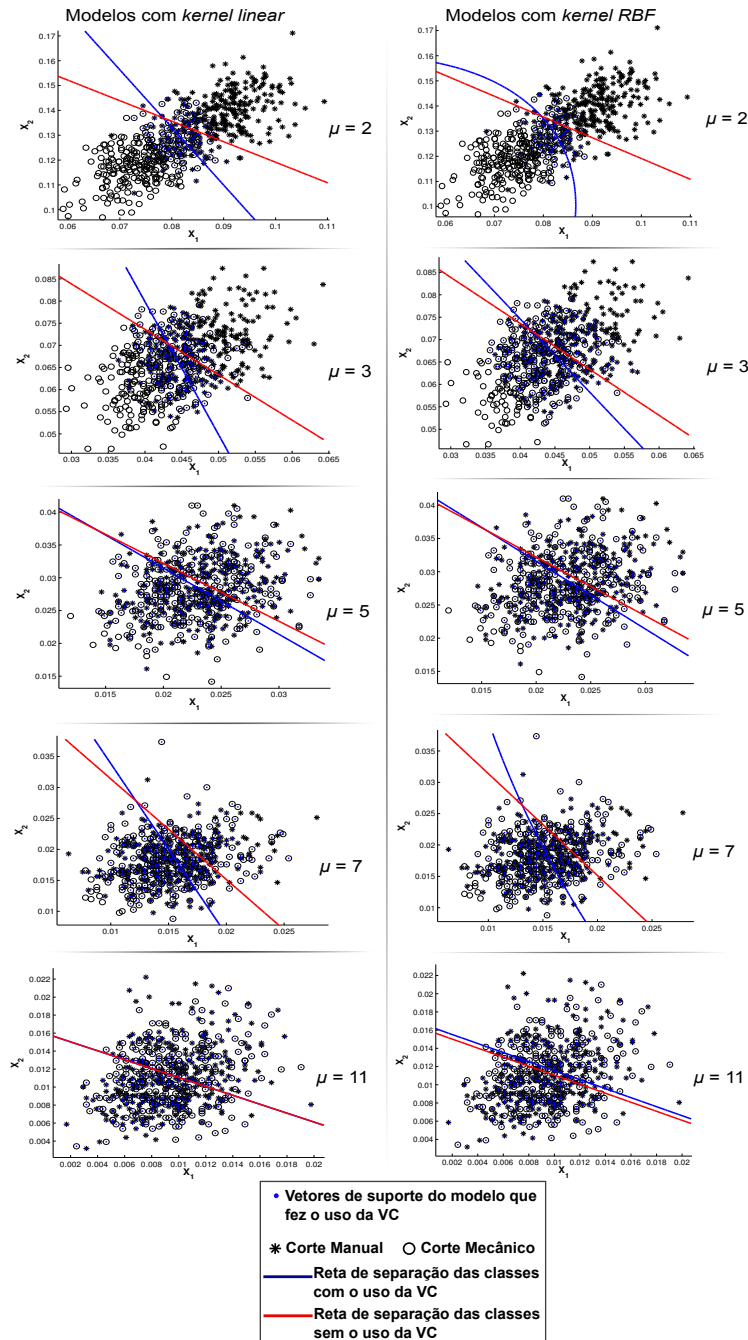


Figura 4.9: Distribuição espacial das duas classes, corte manual e corte mecânico. A construção dessa figura foi possível por meio da estratégia por histograma considerando caminhadas de tamanho $\mu + 2$, onde apenas duas características para cada amostra dos dados são consideradas.

Tal como é observado na Figura 4.9, a partir da memória igual a 5, as classes, corte manual e corte mecânico, estão bastante sobrepostas, tornando praticamente inviável a tarefa de separá-las. Dessa forma, o classificador acaba decidindo de maneira aleatória qual é a classe para uma amostra desconhecida (amostras de testes).

Este problema não ocorre somente na classificação das amostras do conjunto de teste, mas também no procedimento da validação cruzada, dado que o mesmo também consiste em classificar as amostras de dados dos subconjuntos de validação, obtidos a partir da divisão do conjunto de treinamento em *5-folds*.

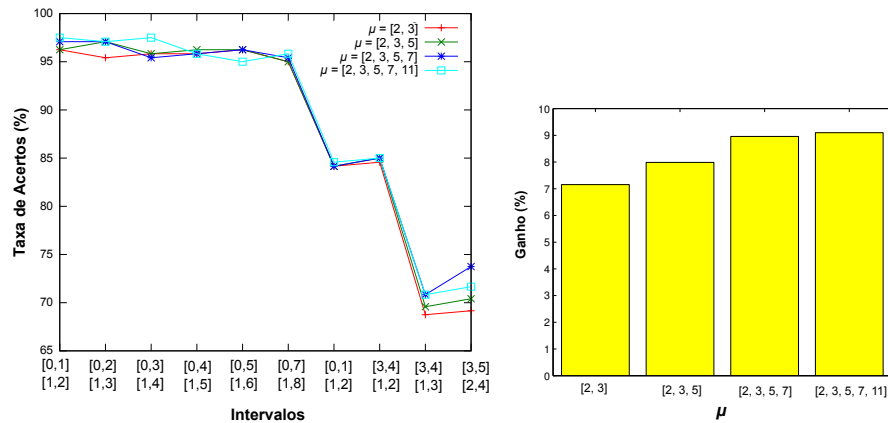
Por outro lado, quando é observado a representação espacial das classes para características retiradas da distribuição conjunta obtidas de memórias iguais a 2 e 3, pode-se ver uma distinção mais clara entre as classes e que a superfície de separação com validação cruzada (em azul) tende para a separação de margem máxima entre as mesmas.

4.2.3 Estratégia por Regiões de Diferentes Memórias

Para a utilização da estratégia por regiões de diferentes memórias, os resultados para modelos SVMs com *kernel linear* e *kernel RBF* são mostrados pelas Tabelas 4.11 e 4.12, e Figuras 4.10 e 4.11, respectivamente.

Tabela 4.11: Porcentagem de acerto do modelo SVM com *kernel linear* sobre a base de dados *BTexCA*, utilizando a estratégia de extração de características por Região.

Intervalo $[t_0, t_f]$ e $[p_0, p_f]$	Memórias							
	[2, 3]		[2, 3, 5]		[2, 3, 5, 7]		[2, 3, 5, 7, 11]	
	SV	CV	SV	CV	SV	CV	SV	CV
[0, 1] e [1, 2]	90.41%	96.25%	90.00%	96.25%	90.00%	97.08%	90.00%	97.50%
[0, 2] e [1, 3]	91.25%	95.41%	91.25%	97.08%	91.66%	97.08%	91.66%	97.08%
[0, 3] e [1, 4]	91.66%	95.83%	91.66%	95.83%	91.66%	95.41%	91.66%	97.50%
[0, 4] e [1, 5]	91.66%	95.83%	91.66%	96.25%	91.66%	95.83%	91.66%	95.83%
[0, 5] e [1, 6]	91.66%	96.25%	91.66%	96.25%	91.25%	96.25%	91.25%	95.00%
[0, 7] e [1, 8]	91.66%	95.00%	91.66%	95.00%	91.66%	95.41%	91.66%	95.83%
[3, 4] e [1, 2]	79.58%	84.16%	78.33%	84.16%	77.91%	84.16%	77.91%	84.58%
[3, 5] e [1, 3]	79.58%	84.58%	79.58%	85.00%	79.58%	85.00%	79.58%	85.00%
[3, 4] e [2, 3]	64.58%	68.75%	65.83%	69.58%	65.41%	70.83%	65.00%	70.83%
[3, 5] e [2, 4]	66.25%	69.16%	66.25%	70.41%	66.25%	73.75%	65.83%	71.66%



(a)

(b)

Figura 4.10: Em (a) a porcentagem de acertos obtidos pelos modelos com validação cruzada para cada tamanho de caminhada ($t + p$). Em (b) o ganho médio em porcentagem obtido pelo uso da validação cruzada.

Tabela 4.12: Porcentagem de acerto do modelo SVM com *kernel RBF* sobre a base de dados *BTexCA*, utilizando a estratégia de extração de características por Região.

Intervalo $[t_0, t_f]$ e $[p_0, p_f]$	Memórias							
	[2, 3]		[2, 3, 5]		[2, 3, 5, 7]		[2, 3, 5, 7, 11]	
	SV	CV	SV	CV	SV	CV	SV	CV
[0, 1] e [1, 2]	90.41%	96.25%	90.00%	97.08%	90.00%	97.08%	90.00%	97.50%
[0, 2] e [1, 3]	91.25%	95.41%	91.25%	97.08%	91.66%	97.08%	96.66%	97.50%
[0, 3] e [1, 4]	91.66%	96.66%	91.66%	95.83%	91.66%	95.41%	91.66%	97.50%
[0, 4] e [1, 5]	91.66%	95.83%	91.66%	96.25%	91.66%	95.00%	91.66%	95.83%
[0, 5] e [1, 6]	91.66%	96.25%	91.66%	96.25%	91.25%	93.75%	91.25%	95.00%
[0, 7] e [1, 8]	91.66%	95.00%	91.66%	95.00%	91.66%	95.00%	91.66%	95.83%
[3, 4] e [1, 2]	79.58%	84.16%	78.33%	84.16%	77.91%	84.16%	77.91%	84.58%
[3, 5] e [1, 3]	79.58%	84.16%	79.58%	85.00%	79.58%	85.00%	79.58%	85.00%
[3, 4] e [2, 3]	64.58%	68.75%	65.83%	69.58%	65.41%	69.16%	65.00%	70.41%
[3, 5] e [2, 4]	66.25%	68.75%	66.25%	70.41%	66.25%	73.75%	65.83%	68.75%

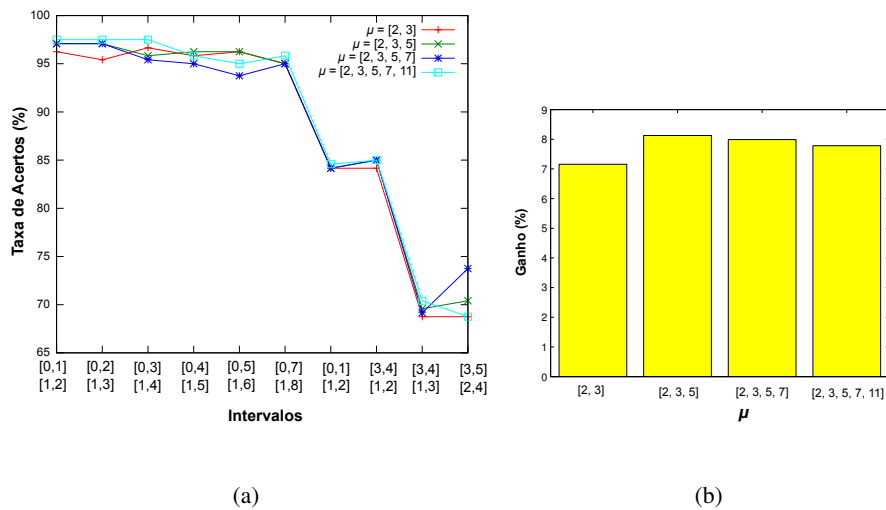


Figura 4.11: Em (a) a porcentagem de acertos obtidos pelos modelos com validação cruzada para cada tamanho de caminhada ($t + p$). Em (b) o ganho médio em porcentagem obtido pelo uso da validação cruzada.

Como era esperado, os melhores resultados foram obtidos utilizando esta estratégia. A acurácia foi de 97.50% tanto para os modelos SVMs de *kernel linear* quanto para os modelos de *kernel RBF*. Além disso, a junção das características

extraídas das regiões das duas melhores distribuições conjuntas obtidas, tiveram acurácia de 96.66% para o modelo de *kernel RBF*.

4.2.4 Estratégia utilizando Propriedades Estatísticas

Os resultados obtidos pelas estratégias acima, permitem dizer que a distribuição conjunta que melhor representa as informações das texturas é a que foi obtida para memória do turista igual a 2. A Figura 4.12 mostra parte da distribuição conjunta obtida pela memória igual a 2 para cada uma das classes.

Com o objetivo de testar se todas as informações desta distribuição podem ser consideradas para descrever as texturas, os resultados da Figura 4.13 apresentam as porcentagens de acertos obtidas utilizando a extração de características por cada uma das propriedades estatísticas, apresentadas na Seção 2.5.4.3.

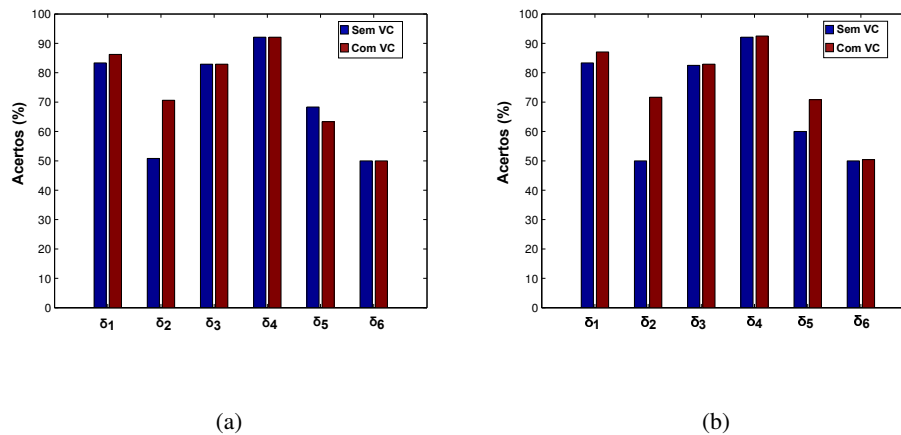


Figura 4.13: Porcentagem de acertos para cada uma das propriedades estatísticas sobre a distribuição conjunta para a memória do turista igual a 2. Em (a) a porcentagem de acertos obtidos pelos modelos SVMs de *kernel linear*. Em (b) a porcentagem de acertos obtidos pelos modelos SVMs de *kernel RBF*.

Tal como foi obtido pela base de dados *Outex*, as propriedades com os melhores resultados também foram δ_1 e δ_4 . A partir dessas duas propriedades um novo vetor de característica foi feito e utilizado para treinar os modelos SVMs. A acurácia para ambos foi de 92.91% com o uso da validação cruzada. A utilização de todas as propriedades estatísticas resultou em uma acurácia de 67.08% para o modelo com *kernel linear* com o uso da validação cruzada.

Diferentemente dos resultados obtido pela base *Outex*, nesta base, a utilização das propriedades estatísticas como estratégia para a extração de características tiveram resultados satisfatórios, comparáveis com os que foram obtidos em Gonçalves (2010). Entretanto, os resultados obtidos pelas estratégias anteriores foram melhores quando comparados aos primeiros.

Assim como na base *Outex*, as Figuras 4.14 e 4.15 mostram a representação espacial das classes de cortes manual e mecânico. Dessa vez, é possível perceber uma distinção melhor entre as duas classes tornando possível uma melhor separação entre elas. Uma Tabela final com os melhores resultados é apresentada pela Tabela 4.13.

Tabela 4.13: Tabela com os melhores resultados para cada uma das estratégias.

Estratégia	Resultados	
	SV	CV
Região	90.83%	95.00%
Histograma	92.91%	92.91%
Regiões de Diferentes Memórias	91.66%	97.50%
Propriedades Estatísticas	-	92.91%

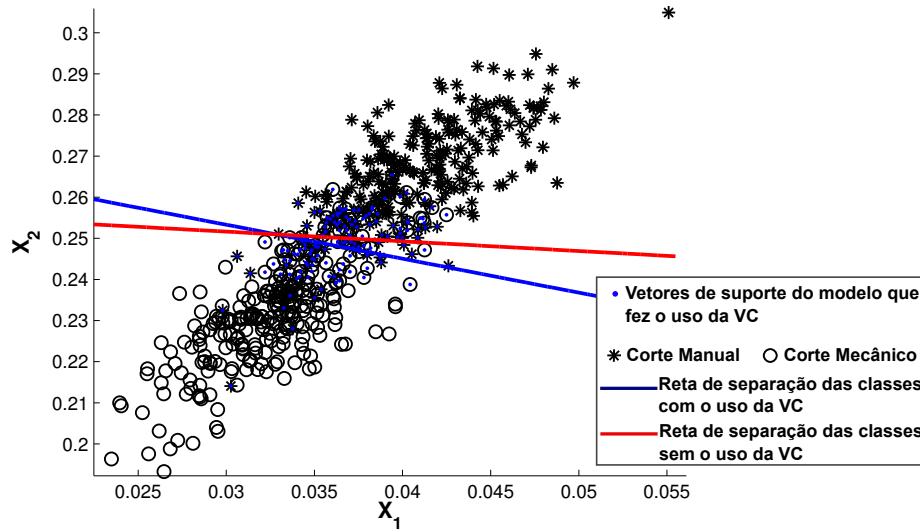


Figura 4.14: Representação espacial entre as classes utilizadas pela bases de dados *BTexCA*. A imagem mostra as superfícies ótimas de separações para um modelo SVM com *kernel linear*, com e sem o uso da validação cruzada. X_1 e X_2 representam, respectivamente, δ_1 e δ_4

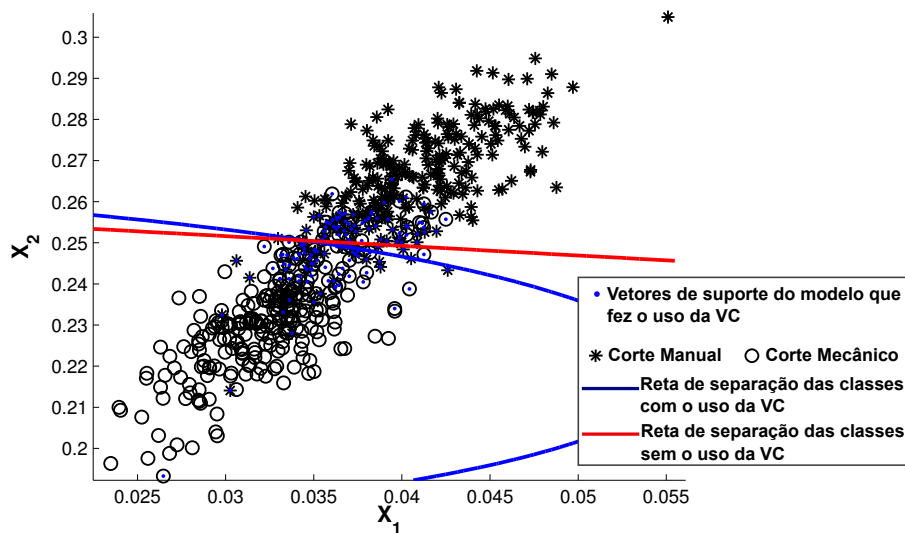
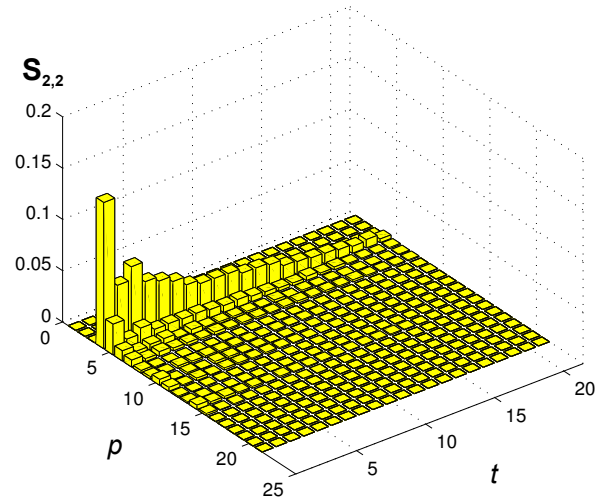
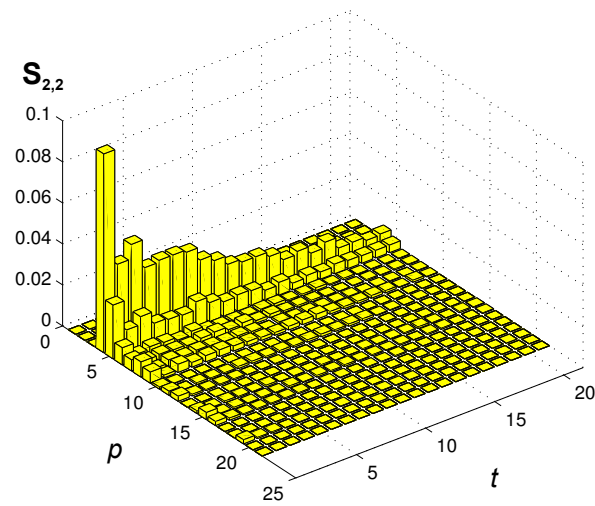


Figura 4.15: Representação Espacial entre as classes utilizadas pela bases de dados *BTexCA*. A imagem mostra as superfícies ótimas de separações para um modelo SVM com *kernel RBF*, com e sem o uso da validação cruzada. X_1 e X_2 representam, respectivamente, δ_1 e δ_4



(a) Manual



(b) Mecânico

Figura 4.12: Parte da distribuição conjunta de cada uma das classes obtida para a memória do turista igual a 2 ($t_0 = 0, p_0 = 0, t_f = 20$ e $p_f = 20$).

5 CONCLUSÃO

Durante o desenvolvimento deste trabalho discutimos diversos aspectos da Caminhada Determinística do Turista. O primeiro deles foi a sua adaptação para o problema em estudo. Antes do desenvolvimento do algoritmo para sua aplicação em imagens, procurou-se associar as características do problema com as propriedades que o algoritmo da CDT nos oferece. Diante disto, foi possível definir a função determinística de caminhada, estabelecer uma relação de vizinhança entre os *pixels* e utilizar uma estratégia de condições de contorno, baseando-se em trabalhos que também aplicaram a CDT para a análise de imagens.

Todos esses aspectos pré-determinados influenciaram os resultados obtidos, que mostraram ser bastante promissores. Não se sabe se eles foram os mais adequados, pois há outras maneiras de se definir as regras de caminhadas e as relações de vizinhança. Se a regra determinística de caminhada fosse diferente, mas os outros aspectos determinados fossem mantidos, certamente outros resultados seriam obtidos, podendo estes ser melhores ou piores do que os obtidos neste trabalho.

Portanto não há uma regra para a determinação de tais aspectos. O que deve ser feito, é uma análise do problema a fim de procurar a melhor forma para adaptar a CDT. Neste trabalho foi discutido o porque da utilização de cada um desses aspectos, com base nos trabalhos encontrados na literatura.

Discutiu-se também a utilização de uma estratégia para uma detecção de atratores. Esta estratégia permitiu uma melhora na eficiência do algoritmo, comparado com a estratégia de força bruta inicialmente utilizada. A detecção de atratores ainda é um problema da caminhada determinística do turista, sendo a principal desvantagem deste método.

A partir dos resultados, foi observado que as principais informações estão no início da distribuição conjunta, e que para valores altos de memórias os resultados mostraram não ser tão interessantes. Neste sentido, pode-se concluir que não há necessidade do turista realizar uma caminhada muito longa para encontrar padrões representativos nas texturas. Dessa forma, pode-se adicionar passos para o algoritmo desconsiderar caminhadas muito longas, facilitando o processo para encontrar atratores e conseqüentemente melhorando o desempenho.

Outro fator que contribui para mostrar que as informações mais representativas estão no início da distribuição conjunta foram os resultados obtidos pelas estratégias de histograma e propriedades estatísticas. Nos resultados obtidos pela estratégia de histograma, vimos que não há necessidade de agregar informação distantes na distribuição, já que esses valores não influenciaram em quase nada no aumento da acurácia dos modelos, sendo que para algumas vezes, a mesma acabou diminuindo. Além disso, quanto maior a caminhada para o histograma, maior o número de características consideradas, o que pode acabar afetando o desempenho dos treinamentos dos modelos SVMs.

Diante dos resultados obtidos pelas propriedades estatísticas, nota-se que elas não foram eficientes para o caso da base *Outex*. Além disso o treinamento dos modelos usando essa estratégia foi mais demorado. Os resultados diante dessa estratégia, apenas reforçam que não há necessidade de considerar todas as informações da distribuição conjunta.

Era esperado que a estratégia utilizando regiões de diferentes memórias obtivessem os melhores resultados, uma vez que os trabalhos que a utilizaram obtiveram os melhores resultados. Os resultados pela junção de diferentes regiões de distribuições obtidas por diferentes valores de memórias mostraram caracterizar melhor a textura. Por outro lado, os resultados obtidos com a junção das duas melhores distribuições obtidas na base *Outex* foi melhor do que os resulta-

dos obtidos com a junção de todas as distribuições, significando que distribuições que não contribuem tanto para a caracterização da texturas acabam prejudicando o desempenho dos modelos SVMs.

Um ponto positivo para o algoritmo da CDT é que não foi preciso fazer o uso de técnicas para a normalização dos dados, uma vez que as informações da distribuição conjunta estão entre 0 e 1.

O uso da validação cruzada foi promissor. Os resultados e as representações espaciais permitem concluir que esta estratégia é importante para o aumento da acurácia dos modelos SVMs. Houve casos em que ela não apresentou resultados bons, entretanto explicamos porque esses casos foram mais comuns quando as classes estavam sobrepostas, tornando a atividade de separá-las onerosa.

No geral, os resultados obtidos permitem concluir que a CDT pode ser usada como uma ferramenta para a análise de texturas em imagens de cortes, manuais e mecânicos, de canas de açúcar. Entretanto, o processo para obter as distribuições conjuntas, mesmo para uma memória e imagens pequenas, ainda é o ponto do algoritmo com o maior custo computacional, e para uma aplicação que exige resultados em tempo real, o uso da CDT é inviável.

Trabalhos futuros podem analisar o desempenho da CDT com o uso de ferramentas que possibilite programações paralelas e distribuídas. Seria vantajoso atribuir para cada processo independente a atividade de efetuar uma determinada trajetória.

A julgar que uma das características de modelos de caminhadas determinísticas é a exploração dos dados, torna-se possível trabalhos envolvendo clusteração de dados e conseqüentemente maneiras não supervisionadas de classificação, tal como foi feito em Campitelli (2007).

REFERÊNCIAS BIBLIOGRÁFICAS

BACKES, A. R. *Estudos de métodos de análise de complexidade em imagens*. Tese (Doutorado em Ciências de Computação e Matemática Computacional) — Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2010.

BACKES, A. R.; BRUNO, O. M.; CAMPITELI, M. G.; MARTINEZ, A. S. Deterministic tourist walks as an image analysis methodology based. In: *Progress in Pattern Recognition, Image Analysis and Applications*. [S.l.]: Springer Berlin Heidelberg, 2006. v. 4225, p. 784–793.

BACKES, A. R.; GONÇALVES, W. N.; MARTINEZ, A. S.; BRUNO, O. M. Texture analysis and classification using deterministic tourist walk. *Pattern Recognition*, v. 43, p. 685 – 694, 2010.

BLINN, J. F.; NEWELL, M. E. Texture and reflection in computer generated images. *Commun. ACM*, v. 19, n. 10, p. 542–547, 1976.

BYUN, H.; LEE, S.-W. Applications of support vector machines for pattern recognition: A survey. In: *Pattern Recognition with Support Vector Machines*. [S.l.]: Springer Berlin Heidelberg, 2002. v. 2388, p. 213–236.

CAMPITELI, M. G. *A caminhada do turista como ferramenta na identificação de padrões*. Tese (Doutorado em Física Aplicada à Medicina e Biologia) — Faculdade de Filosofia, Ciências e Letras de Ribeirão Preto, Universidade de São Paulo, Ribeirão Preto, 2007.

CHANG, C.-C.; LIN, C.-J. Libsvm: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, v. 2, 2011.

CLAUSI, D. A.; ZHAO, Y. Grey level co-occurrence integrated algorithm (glcia): a superior computational method to rapidly determine co-occurrence probability texture features. *Computers & Geosciences*, v. 29, n. 7, p. 837 – 850, 2003.

CONNERS, R. W.; HARLOW, C. A. A theoretical comparison of texture algorithms. *Pattern Analysis and Machine Intelligence, IEEE Transactions on, IEEE*, n. 3, p. 204–222, 1980.

COSTA, L. d. F. D.; CESAR JR., R. M. *Shape Analysis and Classification: Theory and Practice*. 1st. ed. [S.l.]: CRC Press, Inc., 2000.

CROSS, G. R.; JAIN, A. K. Markov random field texture models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on, IEEE*, n. 1, p. 25–39, 1983.

GONZALEZ, R.; WOODS, R. *Processamento Digital De Imagens*. [S.l.]: Addison Wesley Bra, 2010. 624 p.

GONÇALVES, W. N. *Caminhadas determinísticas em redes complexas aplicadas em visão computacional*. Dissertação (Mestrado em Ciências de Computação e Matemática Computacional) — Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2010.

HARALICK, R. M.; SHANMUGAM, K.; DINSTEN, I. H. Textural features for image classification. *Systems, Man and Cybernetics, IEEE Transactions on, IEEE*, n. 6, p. 610–621, 1973.

HAYKIN, S. *Redes Neurais: Princípios e Práticas*. [S.l.]: Bookman, 2001. 700 p.

HSU, C. wei; CHANG, C. chung; LIN, C. jen. A practical guide to support vector classification. 2010.

JAIN, A. K.; FARROKHANIA, F. Unsupervised texture segmentation using gabor filters. *Pattern recognition*, Elsevier, v. 24, p. 1167–1186, 1991.

LIMA, G. F. de. *Caminhadas determinísticas em meios desordenados: problema da caminhada do turista*. Tese (Doutorado em Física Aplicada à Medicina e Biologia) — Faculdade de Filosofia, Ciências e Letras de Ribeirão Preto, Universidade de São Paulo, Ribeirão Preto, 2002.

LORENA, A. C.; CARVALHO, A. C. P. L. F. de. Uma introdução às support vector machines. *Revista de Informática Teórica e Aplicada - RITA*, v. 14, n. 2, p. 43–67, 2007.

MANDELBROT, B. *The Fractal Geometry of Nature*. [S.l.]: Henry Holt and Company, 1983.

MASCARENHAS, N.; VELASCO, F. *Processamento digital de imagens*. [S.l.]: INPE, 1984. 315 p.

MATSUYAMA, T.; MIURA, S.-I.; NAGAO, M. Structural analysis of natural textures by fourier transformation. *Computer vision, graphics, and image processing*, Elsevier, v. 24, n. 3, p. 347–362, 1983.

OJALA, T.; MÄENPÄÄ, T.; PIETIKÄINEN, M.; VIERTOLA, J.; KYLLÖNEN, J.; HUOVINEN, S. Outex - new framework for empirical evaluation of texture analysis algorithms. In: . [S.l.: s.n.], 2002. Proc. 16th International Conference on Pattern Recognition, Quebec, Canada, 1:701 - 706.

PENTLAND, A. P. Fractal-based description of natural scenes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, IEEE, n. 6, p. 661–674, 1984.

TUCERYAN, M.; JAIN, A. K. Texture segmentation using voronoi polygons. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, IEEE, v. 12, n. 2, p. 211–216, 1990.

VAPNIK, V. *The nature of statistical learning theory*. [S.l.]: Springer, 1999. 314 p.

WECHSLER, H. Texture analysis — a survey. *Signal Processing*, v. 2, n. 3, p. 271 – 282, 1980.

WESZKA, J. S.; DYER, C. R.; ROSENFELD, A. A comparative study of texture measures for terrain classification. *Systems, Man and Cybernetics, IEEE Transactions on*, IEEE, n. 4, p. 269–285, 1976.

ZHOU, D. *Texture analysis and synthesis using a generic Markov-Gibbs image model*. Tese (Doutorado) — ResearchSpace@ Auckland, 2006.

A APÊNDICE

Resultados obtidos por modelos SVMs, onde a CDT foi aplicada sobre as imagens da base de dados *Outex* na sua forma original, ou seja, 20 amostras para treinamento e 180 amostras de teste.

A.1 Estratégia por Região

Tabela A.1: Porcentagem de acerto dos modelos SVM com *kernel linear* sobre a base de dados *Outex*, utilizando a estratégia de extração de características por Região.

Intervalo $[t_0, t_f]$ e $[p_0, p_f]$	Memórias							
	2		3		5		7	
	SV	CV	SV	CV	SV	CV	SV	CV
[0, 1] e [1, 2]	47.77%	52.22%	77.77%	90.27%	54.44%	59.72%	69.72%	77.50%
[0, 4] e [1, 5]	49.16%	49.72%	77.22%	87.77%	50.00%	50.27%	50.83%	53.05%
[3, 5] e [1, 3]	64.16%	62.50%	66.94%	61.66%	67.22%	69.16%	78.61%	76.94%
[6, 8] e [4, 7]	49.44%	49.44%	60.00%	60.00%	60.83%	60.83%	60.83%	61.38%
[9, 11] e [5, 8]	55.55%	55.55%	53.05%	53.05%	54.16%	54.16%	59.72%	59.72%

Tabela A.2: Porcentagem de acerto dos modelos SVM com *kernel RBF* sobre a base de dados *Outex*, utilizando a estratégia de extração de características por Região.

Intervalo [t_0, t_f] e [p_0, p_f]	Memórias							
	2		3		5		7	
	SV	CV	SV	CV	SV	CV	SV	CV
[0, 1] e [1, 2]	47.77%	65.83%	77.77%	88.05%	54.16%	71.94%	69.72%	79.16%
[0, 4] e [1, 5]	49.16%	50.00%	77.22%	90.83%	50.00%	50.55%	50.55%	72.50%
[3, 5] e [1, 3]	64.16%	56.94%	66.94%	60.27%	79.16%	71.11%	79.16%	71.11%
[6, 8] e [4, 7]	48.05%	48.61%	60.00%	58.05%	61.66%	59.72%	60.55%	57.50%
[9, 11] e [5, 8]	56.66%	55.83%	53.05%	51.11%	54.16%	55.55%	59.44%	51.11%

A.2 Estratégia por Histograma

Tabela A.3: Porcentagem de acerto do modelo SVM com *kernel linear* sobre a base de dados *Outex*, utilizando a estratégia de extração de características por Histograma.

$t + p$	Memórias							
	2		3		5		7	
	SV	CV	SV	CV	SV	CV	SV	CV
$2+\mu$	48.33%	49.16%	77.22%	83.33%	58.33%	56.38%	70.55%	77.50%
$3+\mu$	48.33%	50.00%	76.66%	87.50%	67.22%	65.83%	79.44%	83.05%
$5+\mu$	48.88%	49.72%	78.61%	88.61%	76.66%	79.16%	87.77%	89.16%
$10+\mu$	51.94%	55.55%	90.55%	89.72%	86.11%	82.77%	92.22%	90.83%
$20+\mu$	53.33%	56.11%	91.94%	89.44%	85.27%	84.44%	91.11%	91.11%
$35+\mu$	53.05%	56.66%	91.94%	89.72%	85.83%	82.22%	91.11%	92.50%

Tabela A.4: Porcentagem de acerto do modelo SVM com *kernel RBF* sobre a base de dados *Outex*, utilizando a estratégia de extração de características por Histograma.

$t + p$	Memórias							
	2		3		5		7	
	SV	CV	SV	CV	SV	CV	SV	CV
$2+\mu$	48.33%	49.16%	77.22%	84.16%	58.33%	56.66%	70.55%	76.38%
$3+\mu$	48.33%	56.11%	76.66%	85.00%	67.22%	67.77%	79.44%	81.38%
$5+\mu$	48.88%	57.50%	78.61%	85.83%	76.66%	76.38%	87.77%	89.72%
$10+\mu$	51.94%	53.61%	90.55%	86.94%	86.11%	77.77%	92.22%	89.16%
$20+\mu$	53.33%	56.94%	91.94%	89.72%	85.27%	77.50%	91.11%	85.55%
$35+\mu$	53.05%	54.72%	91.94%	89.72%	86.38%	79.72%	91.11%	87.50%

A.3 Estratégia por Regiões de Diferentes Memórias

Tabela A.5: Porcentagem de acerto do modelo SVM com *kernel linear* sobre a base de dados *Outex*, utilizando a estratégia de extração de características por Região de Diferentes Memórias.

Intervalo $[t_0, t_f]$ e $[p_0, p_f]$	Memórias					
	[2, 3]		[2, 3, 5]		[2, 3, 5, 7]	
	SV	CV	SV	CV	SV	CV
[0, 1] e [1, 2]	64.16%	86.66%	64.44%	86.94%	65.27%	86.66%
[0, 4] e [1, 5]	64.72%	91.66%	65.83%	91.66%	69.16%	88.33%
[3, 5] e [1, 3]	68.05%	62.77%	71.66%	67.22%	78.05%	71.11%
[6, 8] e [4, 7]	58.61%	58.61%	65.27%	63.88%	66.11%	66.38%
[9, 11] e [5, 8]	57.50%	57.50%	58.88%	58.88%	58.05%	58.05%

Tabela A.6: Porcentagem de acerto do modelo SVM com *kernel RBF* sobre a base de dados *Outex*, utilizando a estratégia de extração de características por Região de Diferentes Memórias.

Intervalo $[t_0, t_f]$ e $[p_0, p_f]$	Memórias					
	[2, 3]		[2, 3, 5]		[2, 3, 5, 7]	
	SV	CV	SV	CV	SV	CV
[0, 1] e [1, 2]	64.16%	91.66%	64.44%	87.77%	65.27%	87.50%
[0, 4] e [1, 5]	64.72%	91.94%	65.83%	91.38%	69.16%	92.50%
[3, 5] e [1, 3]	68.05%	57.50%	71.66%	61.66%	78.05%	68.33%
[6, 8] e [4, 7]	57.22%	57.77%	65.00%	60.27%	65.55%	65.83%
[9, 11] e [5, 8]	57.77%	56.94%	58.05%	56.38%	58.61%	54.16%

A.4 Estratégia utilizando as Propriedades Estatísticas

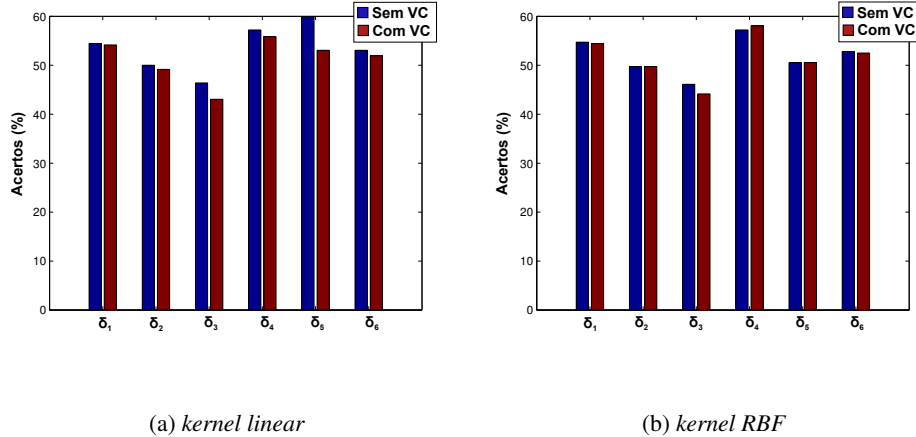


Figura A.1: Porcentagem de acertos para cada uma das propriedades estatísticas sobre a distribuição conjunta obtida para a memória do turista igual a 3, sobre a base de dados *Outex*.