

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE INFORMÁTICA
CURSO DE ENGENHARIA DE COMPUTAÇÃO**

ANDRÉ LUIZ MARASCA

**ANÁLISE DE TEXTURA ATRAVÉS DE DESCRITORES FRACTAIS POR MEIO DA
UTILIZAÇÃO DE RÓTULOS NA TRANSFORMADA DE DISTÂNCIA 3D**

TRABALHO DE CONCLUSÃO DE CURSO

PATO BRANCO

2016

ANDRÉ LUIZ MARASCA

**ANÁLISE DE TEXTURA ATRAVÉS DE DESCRITORES FRACTAIS POR MEIO DA
UTILIZAÇÃO DE RÓTULOS NA TRANSFORMADA DE DISTÂNCIA 3D**

Trabalho de Conclusão de Curso como requisito parcial à obtenção do título de Bacharel em Engenharia de Computação, do Departamento Acadêmico de Informática da Universidade Tecnológica Federal do Paraná.

Orientador: Prof. Dr. Dalcimar Casanova

PATO BRANCO

2016



TERMO DE APROVAÇÃO

Às 8 horas e 30 minutos do dia 28 de junho de 2016, na sala V006, da Universidade Tecnológica Federal do Paraná, Câmpus Pato Branco, reuniu-se a banca examinadora composta pelos professores Dalcimar Casanova (orientador), Pablo Gauterio Cavalcanti e Marco Antonio de Castro Barbosa para avaliar o trabalho de conclusão de curso com o título **Análise de textura através de descritores fractais por meio da utilização de rótulos na transformada de distância 3D**, do aluno **Andre Luiz Marasca**, matrícula 01377841, do curso de Engenharia de Computação. Após a apresentação o candidato foi arguido pela banca examinadora. Em seguida foi realizada a deliberação pela banca examinadora que considerou o trabalho aprovado.

Dalcimar Casanova
Orientador (UTFPR)

Pablo Gauterio Cavalcanti
(UTFPR)

Marco Antonio de Castro Barbosa
(UTFPR)

Beatriz Terezinha Borsoi
(UTFPR)

Pablo Gauterio Cavalcanti
Coordenador do Curso de
Engenharia de Computação

RESUMO

MARASCA, André Luiz. Análise de textura através de descritores fractais por meio da utilização de rótulos na transformada de distância 3D. 2016. 72 f. Trabalho de Conclusão de Curso de bacharelado em Engenharia de Computação - Universidade Tecnológica Federal do Paraná. Pato Branco, 2016.

Este trabalho propõe um método para análise de texturas coloridas baseado na geometria fractal. A proposta consiste em converter uma imagem bidimensional em um cubo, criar rótulos para determinados grupos de *pixels* e, então, dilatar os componentes de figuras presentes neste cubo. Os descritores são obtidos a partir da relação entre os volumes dos componentes dilatados, seus rótulos e os raios de dilatação. Desta forma, os descritores podem medir tanto a complexidade dos modelos de intensidade dos *pixels*, assim como os padrões de interferência entre os componentes rotulados. Esta combinação proporciona um descritor de textura robusto e preciso. A eficiência do novo método é avaliada através de análises estatísticas sobre o reconhecimento de padrões supervisionados.

Palavras-chave: Reconhecimento de padrões. Dimensão Fractal. Descritores Fractais. Análise de textura colorida.

ABSTRACT

MARASCA, André Luiz. Texture analysis using fractal descriptors by labels in the 3D distance transform. 2016. 72 f. Trabalho de Conclusão de Curso de bacharelado em Engenharia de Computação - Universidade Tecnológica Federal do Paraná. Pato Branco, 2016.

This work proposes a method for analysis of color textures based on fractal geometry. The proposal is based in conversion of a two-dimensional image in a 3 dimensional cloud points where each pixel corresponds to a point. After that we set labels for certain groups of points and then dilate this cloud structure. The descriptors are obtained by the volume ratio of components, their labels and dilatation radius. In this way the descriptors can measure both the complexity of the intensity of the pixels models, as well the interference patterns between the labeled components. This combination provides a descriptor robust and precise for texture analysis. The new method is evaluated through statistical analyzes over supervised machine learning problems.

Keywords: Pattern Recognition. Fractal Dimension. Descriptors Fractals. Analysis colorful texture.

LISTA DE FIGURAS

| | |
|---|----|
| Figura 1 – Dilatação de uma curva com um disco..... | 19 |
| Figura 2 – Regressão linear para o método de Minkowski..... | 19 |
| Figura 3 – Distância $d(P, S)$ entre o ponto P e a forma S | 20 |
| Figura 4 – As possíveis distâncias em uma grade ortogonal 3×3 | 21 |
| Figura 5 – Forma S qualquer..... | 22 |
| Figura 6 – Transformada de distância euclidiana exata para forma apresentada na Figura 5 em níveis de cinza..... | 22 |
| Figura 7 – Transformada de distância euclidiana exata para forma apresentada na Figura 5, representada como superfície tridimensional | 22 |
| Figura 8 – A SEDR para as quatro primeiras distâncias exatas | 24 |
| Figura 9 – A SEDR representada em formato matriz | 24 |
| Figura 10 – Pseudocódigo para Transformada de distância euclidiana exata | 24 |
| Figura 11 – Exemplo da transformação 1 para o algoritmo básico de Saito e Toriwaki (1994) | 27 |
| Figura 12 – Exemplo da transformação 2 para o algoritmo básico de Saito e Toriwaki (1994) | 27 |
| Figura 13 – Pseudocódigo para os passos <i>forward scan</i> e <i>backward scan</i> do algoritmo mais eficiente para EEDT de Saito e Toriwaki (1994) | 28 |
| Figura 14 – Um conjunto de pontos isolados (a) e sua respectiva transformada de distância (b) e Diagrama de Voronoi (c). Cada região de influência foi representada por um nível de cinza distinto..... | 29 |
| Figura 15– Um conjunto de pontos isolados (a) e seu respectivo Diagrama de Voronoi (b) e Diagrama de Voronoi modificado (c). Cada região de influência foi representada por um nível de cinza distinto | 30 |
| Figura 16 – Gráficos de fdps normais | 34 |
| Figura 17 – Amostras de classes de textura contidos na base de conjunto de testes da Outex com ID Outex_TC_00013 | 44 |
| Figura 18 – Fluxograma ilustrando o procedimento necessário para o método proposto neste trabalho .. | 45 |
| Figura 19 – Imagem de um inseto preso em uma teia de aranha..... | 46 |
| Figura 20 – Um conjunto de pontos isolados, conversão da Figura 19 em um cubo | 47 |
| Figura 21 – Esboço de assinatura para uma instância de cada classe de textura | 48 |
| Figura 22 – Três canais de cores de uma imagem de exemplo | 49 |
| Figura 23– EEDT e segmentação do espaço para o exemplo de imagem da Figura 22. Cortes no eixo z ... | 50 |
| Figura 24 – Transformada de distância bidimensional representada em níveis de cinza (b) para uma forma binária bidimensional (a)..... | 53 |

| | |
|---|----|
| Figura 25 – Comparação (d) entre os Diagramas de Voronoi através do algoritmo força bruta (b) e através do algoritmo de Saito e Toriwaki (1994) (c), para as formas (a) representadas em preto e rótulos representados em níveis de cinza..... | 54 |
| Figura 26 – Comparação os possíveis resultados Diagramas de Voronoi com tratamento de empate (a), sem tratamento de empate, com prioridade para o vermelho (b) e com prioridade para o verde (c)..... | 55 |
| Figura 27 – Comparação entre os Diagramas de Voronoi tridimensionais através do algoritmo para o método força bruta (b), para o método de Casanova et al. (2016) (c) e para o método aqui proposto (d), ambos para imagem de entrada (a). Corte no Eixo Z | 56 |
| Figura 28 – Exemplos da relação dos canais de cores no processo de dilatação. (a) Textura original; (b) r = 0; (c) r = 1; (d) r = 3; (e) r = 4..... | 57 |
| Figura 29 – Exemplos da relação dos canais de cores no processo de dilatação. (a) Textura original; (b) Canal vermelho; (c) Canal verde; (d) Canal Azul; (e) Os três canais..... | 58 |
| Figura 30 – Curva log-log para diferentes classes de textura presentes na base de conjunto de testes com ID Outex_TC_00013..... | 59 |
| Figura 31 – Curva log-log para uma imagem de cada classe de textura presentes na base de conjunto de testes com ID Outex_TC_00013 | 60 |
| Figura 32 – Curva log-log para todas as imagens pertencentes a todas as classes de textura presentes na base de conjunto de testes com ID Outex_TC_00013 | 60 |
| Figura 33 – Curva ln raio x ln volume para cada canal de cor para uma imagem de textura. | 61 |
| Figura 34 – Mapa de correlação entre os atributos utilizados para caracterização de texturas..... | 61 |
| Figura 35 – Matriz de confusão para todas as classes de textura presentes na base de conjunto de testes com ID Outex_TC_00013 utilizando o vetor de características obtidos através do método proposto. Classificador 1-NN | 62 |
| Figura 36 – Matriz de confusão para todas as classes de textura presentes na base de conjunto de testes com ID Outex_TC_00013 utilizando o vetor de características obtidos através do método proposto. Classificador LDA..... | 63 |
| Figura 37 – Curva ln raio x ln volume para cada rótulo para uma imagem de textura. | 66 |
| Figura 38 – Matriz de confusão para todas as classes de textura presentes na base de conjunto de testes com ID Outex_TC_00013 utilizando o vetor de características obtidos através do novo método proposto. Classificador LDA | 67 |

LISTA DE SIGLAS

| | |
|------|---|
| DT | <i>Distance Transform</i> ou transformada de distância |
| EDT | <i>Euclidean Distance Transform</i> ou transformada de distância euclidiana |
| EEDT | <i>Exact Euclidean Distance Transform</i> ou transformada de distância euclidiana exata |
| GLCM | <i>Gray-level Co-occurrence Matrix</i> ou Matriz de Co-ocorrência |
| GLDM | <i>Gray-level Difference Method</i> ou Método de Diferenças de Tons de Cinza |
| k-NN | <i>k-Nearest Neighbor</i> ou k-vizinhos mais próximos |
| LBP | <i>Local Binary Patterns</i> ou Padrões Locais Binários |
| LDA | <i>Linear discriminant analysis</i> ou Análise de Discriminante Linear |
| SEDR | <i>Sorted Euclidean Distance Representation</i> ou representação ordenada da distância euclidiana |

SUMÁRIO

| | | |
|----------|--|-----------|
| 1 | INTRODUÇÃO | 10 |
| 1.1 | PROBLEMA | 11 |
| 1.2 | OBJETIVOS | 12 |
| 1.2.1 | Objetivo Geral | 12 |
| 1.2.2 | Objetivos Específicos | 12 |
| 1.3 | JUSTIFICATIVA | 12 |
| 1.4 | ESTRUTURA DO TRABALHO | 13 |
| 2 | REFERENCIAL TEÓRICO | 14 |
| 2.1 | ANÁLISE DE TEXTURA | 14 |
| 2.1.1 | LBP..... | 14 |
| 2.1.2 | GLDM | 15 |
| 2.1.3 | GLCM..... | 15 |
| 2.1.4 | Descritores de Fourier | 15 |
| 2.1.5 | Gabor-wavelets | 16 |
| 2.2 | DIMENSÃO FRACTAL..... | 16 |
| 2.2.1 | Dimensão fractal em processamento de imagens | 17 |
| 2.3 | DIMENSÃO FRACTAL DE BOULIGAND-MINKOWSKI..... | 18 |
| 2.4 | TRANSFORMADA DE DISTÂNCIA | 20 |
| 2.4.1 | Algoritmo força bruta para transformada de distância | 23 |
| 2.4.2 | Algoritmo otimizado para transformada de distância..... | 25 |
| 2.4.3 | Algoritmo de Saito e Toriwaki (1994) para EEDT | 25 |
| 2.5 | DIAGRAMAS DE VORONOI | 28 |
| 2.5.1 | Rotulação | 29 |
| 2.6 | APRENDIZAGEM DE MÁQUINA | 30 |
| 2.6.1 | Aprendizado baseado em instâncias | 31 |
| 2.6.2 | Aprendizado <i>k-Nearest Neighbor</i> | 32 |
| 2.6.3 | <i>Discriminant Analysis</i> | 33 |
| 2.7 | VALIDAÇÃO ESTATÍSTICA | 33 |
| 2.7.1 | Distribuição Normal | 34 |
| 2.7.2 | Distribuição Normal Padrão | 35 |
| 2.7.3 | Validação cruzada por <i>k-fold</i> | 36 |
| 2.7.4 | Matriz de confusão | 36 |

| | | |
|--------------|---|-----------|
| 2.8 | ANÁLISE DE TEXTURAS BASEADA EM DESCRITORES FRACTAIS..... | 37 |
| 2.8.1 | Método de Backes, Casanova e Bruno (2009) | 38 |
| 2.8.2 | Método de Backes, Casanova e Bruno (2012) | 38 |
| 2.8.3 | Método de Casanova et al. (2016)..... | 40 |
| 3 | MATERIAIS E MÉTODOS | 42 |
| 3.1 | MATERIAIS | 42 |
| 3.2 | METODOLOGIA..... | 44 |
| 3.2.1 | Definição da região de trabalho..... | 45 |
| 3.2.2 | Obtenção dos descritores fractais | 47 |
| 3.2.3 | Validação do método | 51 |
| 4 | RESULTADOS | 52 |
| 4.1 | COMPARAÇÃO DAS IMPLEMENTAÇÕES | 52 |
| 4.1.1 | EEDT através dos algoritmos força bruta e Saito e Toriwaki (1994) | 52 |
| 4.1.2 | Diagrama de Voronoi 2D através dos algoritmos força bruta e Saito e Toriwaki (1994) sem tratamento de empate | 53 |
| 4.1.3 | Diagrama de Voronoi 3D Modificado através dos algoritmos de força bruta e Saito e Toriwaki (1994) com tratamento de empate..... | 54 |
| 4.2 | DESCRITORES FRACTAIS DE BOULIGAND-MINKOWSKI..... | 59 |
| 4.3 | RESULTADOS ESTATÍSTICOS..... | 62 |
| 4.4 | PROPOSTA DE METODOLOGIA DE ROTULAÇÃO | 64 |
| 4.4.1 | Resultados para nova proposta de metodologia de rotulação..... | 65 |
| 5 | CONCLUSÃO..... | 68 |
| | REFERÊNCIAS | 69 |

1 INTRODUÇÃO

A análise de textura é uma área de grande importância. Dentre suas aplicações está o controle de qualidade aplicado à automação da inspeção industrial, como detecção de pragas em frutas, classificação de materiais e objetos, inspeção de placas de circuito impresso e detecção de discontinuidades significativas em superfícies de materiais. A medicina também se beneficia da análise de textura como, por exemplo, no diagnóstico e acompanhamento da doença de Alzheimer, anormalidades intestinais, exames de mamografia, e classificação de lesões melanocíticas de pele Cavalcanti; Scharcanski (2013 e Khademi; Krishnan (2008 e Patients et al. (1998).

Apesar de o conceito de textura ser algo aparentemente intuitivo, ainda não se tem uma definição única aceita na literatura. Segundo Backes (2006) a textura pode ser caracterizada como a repetição de padrões nas variações de cores e tons de iluminação, ou quaisquer outros modelos presentes em uma determinada área, podendo haver pequenas diferenças na repetição dos mesmos, como cor ou forma. Para Tamra et al (1978), textura é caracterizada por padrões repetitivos no posicionamento de primitivas ou elementos em uma imagem. Já para Sklansky (1978) uma imagem deve possuir estatísticas locais, ou outras propriedades locais constantes, com variações suaves, para que possua uma textura constante.

Mesmo não havendo definição exata para textura, a análise de textura é uma área da ciência bastante complexa. Há na literatura vários métodos destinados à caracterização de texturas, dentre eles estão *Local Binary Patterns* (LBP) ou Padrões Locais Binários, *Gray-level Difference Method* (GLDM) ou Método de Diferenças de Tons de Cinza, *Gray-level Co-occurrence Matrix* (GLCM) ou Matriz de Co-ocorrência, Descritores de Fourier, Gabor-wavelets, Laws, Histograma de Cores e Momentos Cromáticos. Uma breve descrição destes métodos pode ser encontrada na tese doutorado de Florindo (2013). Uma outra forma recente de abordar o problema é a partir de descritores de fractais.

A geometria fractal é capaz de descrever situações que não podem ser explicadas facilmente através da geometria euclidiana, como fenômenos caóticos. Um objeto fractal perfeito só pode ser encontrado em teorias matemáticas, porém, alguns objetos na natureza têm semelhanças com fractais Backes (2006).

Mesmo objetos ou imagens artificiais, que não são fractais perfeitos, podem ter suas dimensões fractais estimadas. O processo de estimação dessa dimensão pode ser realizado de diversas formas e por diversos métodos, como pode ser visto em Florindo e Bruno (2012),

Peleg et al. (1984) e Pentland (1984). Alguns desses métodos podem ser adaptados para, além de estimarem a dimensão fractal da imagem/objeto, também caracterizar textura. As informações obtidas por esses métodos convencionou-se chamar de descritores fractais, que são utilizados com sucesso em Backes, Casanova e Bruno (2012).

É nesse escopo que se insere esse trabalho, mais especificamente, o foco está na investigação e proposta de melhoria dos descritores fractais do método de Bouligand-Minkowski.

1.1 PROBLEMA

A análise de texturas a partir de descritores fractais por Boulingard-Minkowski já existe e os resultados obtidos indicam elevada eficiência, como mostram Backes, Casanova e Bruno (2009), Backes, Casanova e Bruno (2012) e Casanova et al. (2016). Embora Casanova et al. (2016) apresente bons resultados, o custo computacional desta metodologia é relativamente alta, e está diretamente relacionada com o número de rótulos analisados. Outra desvantagem é a incapacidade de generalizar a maneira como é feita a escolha dos rótulos em análise (o trabalho é restrito apenas aos canais R, G e B como sendo rótulos), dificultando a continuação de seu trabalho.

Diferentes metodologias podem ser empregadas para realizar a escolha de rótulos, por exemplo, a utilização de mais de um rótulo para descrever cada *pixel*, inserção de informações de vizinhança, ou ainda, informações provenientes da análise do domínio da frequência.

Nesse contexto, propõe-se desenvolver uma metodologia capaz de suportar a generalização no uso de rótulos em grupos de *pixels* de maneira regrada utilizando a transformada de distância euclidiana exata (*Exact Euclidean Distance Transform*, EEDT), de forma que tal transformada permaneça com tempo polinomial de baixa ordem. Como consequência, pode-se realizar escolhas diversas de rótulos que podem melhor caracterizar determinados nichos de textura, ou seja, a maneira que melhor atende a determinadas bases de dados, além da redução no custo computacional.

1.2 OBJETIVOS

1.2.1 Objetivo Geral

Melhorar a análise de textura através de descritores fractais por meio da utilização de rótulos na transformada de distância 3D.

1.2.2 Objetivos Específicos

- Implementar a transformada de distância 3D com rótulos através do algoritmo de Saito e Toriwaki (1994).
- Utilizar a transformada com rótulos na obtenção de descritores fractais através do método Bouligand-Minkowski.
- Classificar texturas utilizando as características obtidas através dos descritores fractais como base para um método de aprendizagem de máquina supervisionado.
- Validar os resultados obtidos utilizando métodos de análise estatística.

1.3 JUSTIFICATIVA

A análise de textura como já discutido anteriormente, é uma área de grande importância, e melhorar os métodos que realizam a extração de características à partir de imagens de textura é uma das etapas mais importantes desta área.

Como já citado anteriormente, diversos métodos que realizam a extração destas características vem sendo propostos na literatura recentemente. Contudo os métodos apresentados em Backes, Casanova e Bruno (2009), Backes, Casanova e Bruno (2012) e Casanova et al. (2016) apresentam ótimos resultados, já foram melhorados neste trabalho e ainda existe a possibilidade de melhora.

O foco deste trabalho foi justamente melhorar o método proposto Casanova et al. (2016), tanto no sentido de agilidade temporal, como também possibilitar um aperfeiçoamento no processo de extração de características de imagens de textura.

Para isso realizou-se adaptações no algoritmo proposto por Saito e Toriwaki (1994), porém, não utilizando somente a transformada de distância, mas também realizando a segmentação e rotulação do espaço tridimensional (i.e. Diagramas de Voronoi). Tal rotulação pode ser feita via força bruta, no entanto, a alta complexidade impede sua utilização em aplicações computacionais. De forma alternativa, o algoritmo de Saito e Toriwaki (1994) além de consumir pouca memória, tem seu tempo de computação razoavelmente pequeno se comparado ao força bruta. Essas características tornam o algoritmo mais rápido, muito robusto, sendo candidato ideal para o uso em rotulação de um dado espaço tridimensional.

Portanto, a ideia foi utilizar o algoritmo proposto por Saito e Toriwaki (1994) de forma que, além da transformada de distância, também seja possível obter os rótulos correspondentes em tempo polinomial de baixa ordem.

A razão da busca de tal flexibilidade na rotulação, é a possibilidade de estimar a dimensão fractal de superfícies específicas geradas através da textura analisada. Visando a escolha de tais rótulos de maneira a aumentar a riqueza dos detalhes extraídos de uma dada imagem de textura, com a expectativa de obtenção de assinaturas únicas para cada uma das dadas classes de textura analisadas.

1.4 ESTRUTURA DO TRABALHO

Este trabalho está estruturado da seguinte forma: O Capítulo 2 apresenta o referencial teórico, onde tem-se uma breve descrição sobre textura e análise de textura bem como alguns métodos para este fim (Seção 2.1). Na sequência, são apresentados os conceitos sobre dimensão fractal, bem como a forma de sua aplicação em processamento de imagens (Seção 2.2), e a técnica de dimensão fractal utilizada neste trabalho (Dimensão de Minkowski) (Seção 2.3). Em seguida, são apresentados os conceitos sobre transformada de distância (Seção 2.4) e Diagramas de Voronoi (Seção 2.5). Então, são apresentados conceitos básicos sobre aprendizagem de máquina e uma descrição sobre os métodos para este fim utilizados neste trabalho (Seção 2.6). Por fim, são apresentados os métodos de validação utilizados na técnica aqui proposta (Seção 2.8). No Capítulo 3 são apresentados os materiais (Seção 3.1) e a metodologia empregada neste trabalho (Seção 3.2). Finalizando, no Capítulo 4 são apresentados os resultados e comparações com outros trabalhos, e o Capítulo 5 mostra a conclusão deste trabalho.

2 REFERENCIAL TEÓRICO

2.1 ANÁLISE DE TEXTURA

A análise de textura consiste basicamente em extrair características marcantes de uma imagem (i.e. características que possivelmente são únicas para grupos de textura), em seguida, utilizar esta informação para caracterizar e reconhecer imagens, através de algum método de aprendizagem.

Este tópico irá apresentar brevemente alguns métodos de análise de textura presentes na literatura, sendo eles o *Local Binary Patterns* (LBP) ou Padrões Locais Binários, *Gray-level Difference Method* (GLDM) ou Método de Diferenças de Tons de Cinza e *Gray-level Co-occurrence Matrix* (GLCM) ou Matriz de Co-ocorrência, Descritores de Fourier e Gabor-wavelets.

Uma outra forma recente de abordar o problema de análise de texturas, e que vem apresentando bons resultados é através de descritores de fractais Backes et al. (2009, 2012 e Casanova et al. (2016), sendo esta, a abordagem utilizada neste trabalho.

2.1.1 LBP

O método *Local Binary Patterns* (LBP) ou Padrões Locais Binários originalmente proposto em Pietikäinen et al. (2011), é um operador de textura simples, porém muito eficiente, sendo atualmente considerado um dos métodos mais precisos e robustos na classificação e segmentação de texturas.

A abordagem consiste em analisar janelas quadradas de tamanho N . Por exemplo, se selecionada uma janela com $N = 3$, o *pixel* central terá 8 vizinhos. A análise é feita através do ato de percorrer os vizinhos de mesma distância ao centro (círculo centrado no *pixel* centroide da janela) em um sentido horário ou anti-horário. Se o *pixel* central for maior que o *pixel* analisado o mesmo receberá valor 1, senão 0. Em sequência, é gerado um código binário de 8 bits, que é então convertido em byte. E, finalmente, o LBP é o vetor de características obtido pela contagem da frequência de cada um dos bytes (histograma sobre a análise anterior).

2.1.2 GLDM

O Método de Diferenças de Tons de Cinza ou *Gray-level Difference Method* (GLDM) ou é uma técnica antiga e simples de análise de textura, porém ainda produz bons resultados Weszka et al. (1976).

Segundo Kim e Park (1999), O GLDM baseia-se na ocorrência de dois pixels que possuem uma determinada diferença absoluta no nível de cinza e que são separadas por um deslocamento específico δ . Assumindo que e os valores que representam a intensidade luminosa dos *pixels* de uma dada imagem são expressos por $S(x, y)$. Para qualquer deslocamento, o vetor δ é dado por $\delta = (\Delta x, \Delta y)$, a função de deslocamento é $S_\delta(x, y) = |S(x, y) - S(x + \Delta x, y + \Delta y)|$ e $D(i|\delta)$ é a função densidade de probabilidade, definida por $D(i|\delta) = \text{Prob}[S(x, y) = i]$.

São realizadas operações matemáticas sobre a função densidade de probabilidade, e por fim quatro descritores de textura são obtidos para a classificação de padrões: contraste, segundo momento angular, entropia, média Weszka et al. (1976).

2.1.3 GLCM

Gray-level Co-occurrence Matrix (GLCM) ou Matriz de Co-ocorrência consiste em um método estatístico de examinar textura, o qual considera que a relação espacial dos *pixels* é o nível de cinza da matriz de co-ocorrência.

Esta técnica caracteriza texturas calculando o número de vezes que pares de *pixels* com valores específicos em um determinado relacionamento espacial ocorrem em uma imagem, criando uma GLCM. Por fim o vetor de características é obtido assim como o GLDM, através de operações matemáticas sobre a GLCM, obtendo-se assim algumas medidas estatísticas Haralick et al. (1973).

2.1.4 Descritores de Fourier

É uma metodologia que utiliza o espectro de Fourier da imagem de textura para computar os descritores Gonzalez; Woods (2009). Os descritores são obtidos através da média dos valores do espectro de Fourier em uma dada circunferência de distância r do centro, para diversos valores de r . Outra forma para obtenção dos Descritores de Fourier consiste em realizar

a média dos valores do espectro para diferentes ângulos. Na prática, são mescladas ambas as técnicas para obter os Descritores de Fourier Florindo (2013).

2.1.5 Gabor-wavelets

Assim como os descritores de Fourier, os descritores de Gabor-wavelets Manjunath (1996) também são baseados na análise espectral da imagem. A técnica consiste em aplicar filtros Gabor-wavelets bidimensionais sobre a imagem e calcular as energias dos resultados obtidos. O vetor de características é obtido da concatenação dos valores de energia de cada filtro.

2.2 DIMENSÃO FRACTAL

Até meados do século 19 a geometria euclidiana fora considerada a que melhor descrevia o mundo, sendo representado por objetos e formas extraídos desta geometria Backes (2006). Entre 1875 e 1925 os matemáticos reconheceram que uma compreensão adequada da irregularidade ou fragmentação não podiam ser satisfeitas com a ideia de dimensão com determinado número de coordenadas (MANDELBROT, 1983, p. 14). A ideia de dimensão não inteira não é intuitiva, porém a multiplicidade de dimensões é essencial para definir de forma eficiente muitos objetos e formas analisados, sendo estes os fractais.

Com a introdução do conceito de fractais a representação do mundo começou a mudar, pois verificou-se que algumas formas eram impossíveis de serem representadas utilizando a geometria euclidiana, por serem constituídas de formas geradas a partir de autorreplicas.

Segundo Mandelbrot (1983, p. 15), um fractal é, por definição, um conjunto para o qual a dimensão Hausdorff Besicovitch excede estritamente a dimensão topológica. A dimensão de Besicovitch é deduzida do conceito que para uma reta de comprimento L , esta pode ser dividida em n partes de comprimento $H = L/n$. Já um quadrado com aresta de comprimento L , dividindo suas arestas em n partes iguais, serão obtidos n^2 quadrados de área $H = (L/n)^2$. Um cubo de arestas de mesmo comprimento seguindo o mesmo procedimento serão obtidos n^3 partes de volume $H = (L/n)^3$. Generalizando, para um hipercubo de dimensão d tem-

se n^d hipercubos iguais com ocupação espacial $H = (L/n)^d$. Logo a dimensão d de Hausdorff Besicovitch é calculada pela Equação (1).

$$d = \frac{\log H}{\log \frac{L}{n}} \quad (1)$$

Onde L é o comprimento da aresta, n é o número de vezes que a aresta pode ser dividida para determinado fractal em uma iteração i , e H é o comprimento do segmento na iteração i , onde $i \in \mathbb{N}$ (MANDELBROT, 1983, p. 16).

A dimensão fractal é uma medida da fragmentação de um objeto fractal Mandelbrot (1983).

2.2.1 Dimensão fractal em processamento de imagens

Segundo Backes (2006), as características básicas que um objeto necessita para ser considerado um fractal é a dimensão fractal, a auto semelhança em escala, e a complexidade infinita.

Porém objetos do mundo real estão limitados as leis da física, dessa forma não apresentando complexidade infinita, nem auto semelhança em escala Carlin (2000). Mesmo assim estes objetos podem ter suas dimensões fractais estimadas. O processo de estimação da dimensão fractal de modo geral pode ser realizado de diversas formas e por diversos métodos. Segundo Florindo e Bruno (2012), os métodos utilizados para o cálculo da dimensão fractal, podem ser essencialmente divididos em duas categorias: os métodos espaciais, como o box-counting Costa; Cesar Jr. (2009), Bouligand-Minkowski Tricot (1995) e *mass-radius* Costa; Cesar Jr. (2009) e os métodos espectrais, como Fourier Russ (2013) e wavelets Florindo et al. (2012).

Basicamente a ideia é adaptar a técnica de estimativa da dimensão fractal de Hausdorff Besicovitch com o intuito de estimar a dimensão fractal de objetos não-fractais. Dessa forma segundo Backes (2006), faz-se necessário encontrar um valor para que a Equação (1) convirja, que pode ser estimado pela Equação (2).

$$d = \lim_{n \rightarrow \infty} \frac{\log H}{\log \frac{L}{n}} \quad (2)$$

No domínio discreto (como no caso de imagens), a Equação (2) deve ser adaptada. Assim o que pode ser feito é traçar a relação $\ln(H) \times \ln(L/n)$. O resultado é aproximadamente

uma reta, então calcula-se o coeficiente angular, α , desta reta, obtendo-se assim a dimensão fractal $D = \alpha$ Backes (2006).

Dentre as aplicações da dimensão fractal em imagens, algumas destas destacam-se e serão apresentadas brevemente a seguir:

Segmentação de imagens: A estimativa da dimensão fractal pode ser vista como uma medida de ocupação espacial, podendo esta ser usada como ferramenta para separar e rotular partes de uma imagem Costa; Cesar Jr. (2009).

Comprimento de Curvas: Segundo Backes (2006), é possível estimar comprimento de curvas irregulares e contornos através do uso das técnicas de estimativa da dimensão fractal, e não propriamente a dimensão fractal. Mais detalhes são encontrados em Falconer (2003).

Análise e reconhecimento de texturas: A extração de características de imagens através dos dados obtidos por métodos de estimativa da dimensão fractal possibilita a análise e reconhecimento de texturas. O método de estimação da dimensão fractal de Bouligand-Minkowski foi usado por Backes, Casanova e Bruno (2012) como ferramenta para esta tarefa.

2.3 DIMENSÃO FRACTAL DE BOULIGAND-MINKOWSKI

Segundo Backes (2006), o método de Bouligand-Minkowski ou dimensão de Minkowski é a mais sensível a mudanças na estrutura da forma analisada. Desta forma, trata-se de um dos mais consistentes e apurados para o cálculo da dimensão fractal Tricot (1995).

Dada uma forma qualquer $f \in \mathbb{R}^2$, considere o conjunto de pontos $Pf = \{pf_i\}$, tal que todo pf_i está contido em f . Este método se baseia na análise da área de influência, $A_{influência}$, de todo pf_i sob um disco de raio r (Figura 1). Consequentemente, variações na forma geram alterações na área de influência analisada Tricot (1995). Define-se formalmente a área de influência para o espaço bidimensional \mathbb{R}^2 , a partir da Equação (3).

$$A_{influência} = \{ p_j \in \mathbb{R}^2 \text{ e } pf_i \in f \mid \|p_j - pf_i\| \leq r \} \quad (3)$$

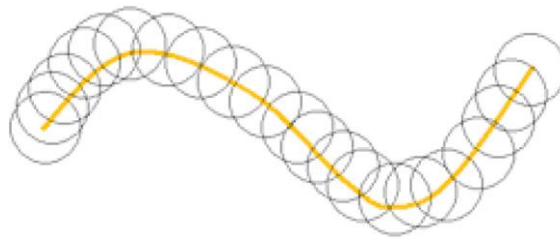


Figura 1 – Dilatação de uma curva com um disco
Fonte: Backes (2006, p. 22).

Em uma imagem digital, para obter a Dimensão de Minkowski, cada um dos *pixels* da forma de interesse tem sua região de influência analisada a partir de um disco de raio r centrado nele. A área dilatada $A(r)$ da imagem é calculada somando os *pixels* sob a região de influência Serra (1982). Segundo Backes (2006) a área dilatada $A(r)$ está relacionada ao raio r conforme a Equação (4), onde μ é um coeficiente qualquer.

$$A(r) = \mu r^{2-D} \quad (4)$$

A dimensão fractal, D , obtida pelo método de Minkowski pode ser calculada através da Equação (5).

$$D = 2 - \lim_{r \rightarrow 0} \frac{\ln(A(r))}{\ln(r)} \quad (5)$$

A Dimensão de Minkowski, D , também pode ser aproximada calculando-se a regressão de uma reta sobre o gráfico log-log da relação $\ln(A(r)) \times \ln(r)$, como demonstrado na Figura 2, onde o coeficiente angular, α , desta reta é utilizado fazendo $D = 2 - \alpha$.

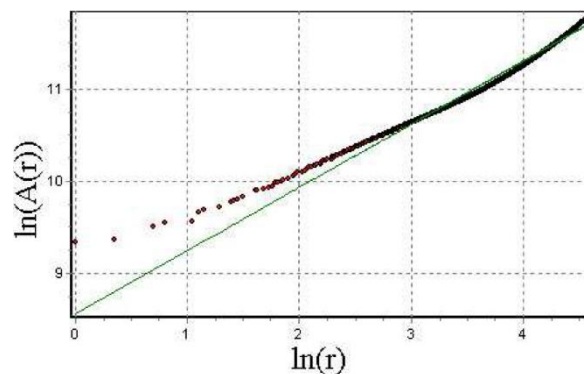


Figura 2 – Regressão linear para o método de Minkowski
Fonte: Backes (2006, p. 23).

Para otimizar o processo de estimativa da Dimensão de Minkowski em uma imagem, pode-se utilizar a EEDT, de modo que não seja necessário computar cálculos massivos para realizar a análise da região de influência, essa transformada é apresentada abaixo.

2.4 TRANSFORMADA DE DISTÂNCIA

A transformada de distância (*Distance Transform*, DT) é um operador geométrico fundamental com grande aplicabilidade em visão computacional, dentre elas estão a análise de formas, a esqueletonização, o *anti-aliasing*, o reconhecimento de padrões, a detecção e rastreamento de pista e a geometria computacional Costa (2000 e Fabbri et al. (2008 e Gustavson; Strand (2011 e Ruyi et al. (2011). Porém, o cálculo de uma transformada de distância euclidiana (*Euclidean Distance Transform*, EDT) é relativamente custoso, pois além do tamanho da imagem, o seu conteúdo também influencia no desempenho deste algoritmo.

Dada uma forma binária bidimensional S (conjunto de pontos conectados), e ainda, dado um ponto qualquer, P , não contido em S , a *distância* $d(P, S)$ entre o ponto P e a forma S é definida como a menor distância entre P e um ponto qualquer contido em S Costa; Cesar Jr. (2009). Várias métricas de distância podem ser consideradas, como *chessboard*, *city-block* ou distância euclidiana. A Figura 3 ilustra a distância $d(P, S)$ utilizando a métrica euclidiana.

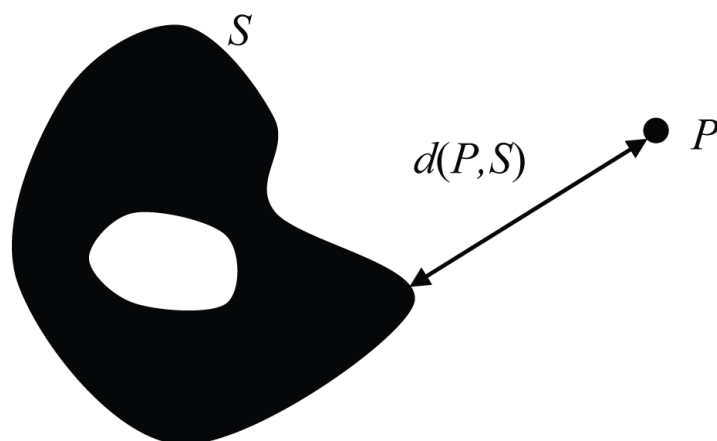


Figura 3 – Distância $d(P, S)$ entre o ponto P e a forma S
Fonte: Costa e Cesar JR. (2009, p. 388).

No domínio contínuo existem infinitas distâncias e infinitos pontos dentro de uma área de interesse para se atribuir suas respectivas distâncias. Diferentemente, no domínio discretizado (como para imagens) existe limitado número de pontos e distâncias. Por exemplo,

uma imagem de 3 x 3 permite apenas seis distâncias diferentes (contando com a distância 0), como ilustrado na Figura 4 (COSTA; CESAR JR., 2009, p. 338).

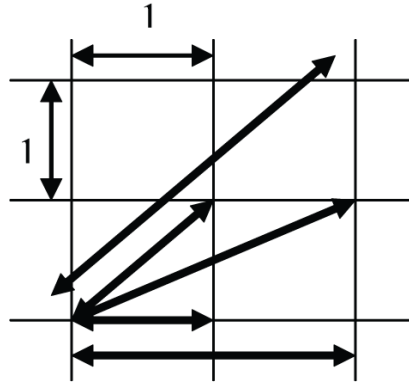


Figura 4 – As possíveis distâncias em uma grade ortogonal 3 x 3
Fonte: Costa e Cesar JR. (2009, p. 389).

Segundo Costa e Cesar JR. (2009), o cálculo da transformada de distância em uma malha ortogonal através de algoritmos força bruta apresentam complexidade polinomial de ordem elevada, implicando em uma série de dificuldades práticas que levaram a muitos algoritmos simplificados (que utilizam muitas vezes de mecanismos gulosos) para o cálculo da transformada de distância euclidiana aproximada. Porém, os avanços recentes têm permitido cálculo da transformada de distância euclidiana exata. Mais detalhes sobre estas técnicas podem ser encontrados em Fabbri et al. (2008).

Para Fabbri et al. (2008) a razão por tanto esforço para cálculo da transformada de distância exata com métrica euclidiana são suas propriedades altamente aplicáveis. Sendo radialmente simétrica, ou seja, gera representações e propriedades praticamente invariantes a rotações, tornando o reconhecimento de forma robusto. Além disso, métricas não euclidianas além de não possuírem tais propriedades, podem não corresponder ao resultado esperado Paglieroni (1992).

Para melhor entendimento, a transformada de distância para a forma da Figura 5 é ilustrada na Figura 6 representada em níveis de cinza. Já na Figura 7 é representada como uma superfície tridimensional, onde regiões mais elevadas representam maiores distâncias.



Figura 5 – Forma S qualquer
Fonte: Costa e Cesar JR. (2009, p. 387).

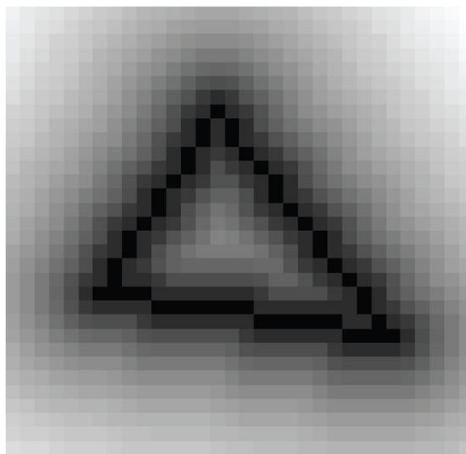


Figura 6 – Transformada de distância euclidiana exata para forma apresentada na Figura 5 em níveis de cinza
Fonte: Costa e Cesar JR. (2009, p. 389).

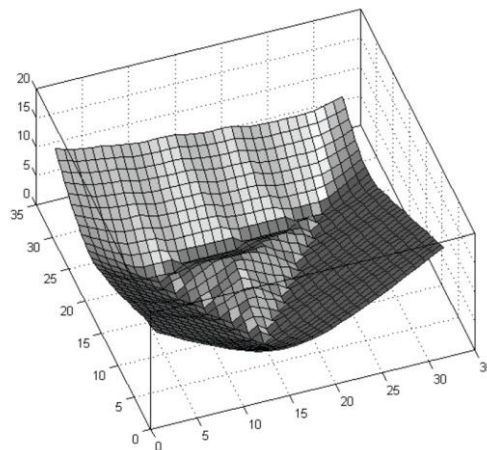


Figura 7 – Transformada de distância euclidiana exata para forma apresentada na Figura 5, representada como superfície tridimensional
Fonte: Costa e Cesar JR. (2009, p. 389).

2.4.1 Algoritmo força bruta para transformada de distância

O algoritmo força bruta para EEDT tem complexidade alta. Uma vez que no pior caso, levando em consideração a forma da imagem, estima-se uma complexidade $O(n^4)$ no pior caso, $\Omega(n^2)$ no melhor caso, mas ficando no caso médio em $\Theta(n^3)$, dada uma imagem com dimensões $C \times R$ composta por N pixels, sendo $n = \sqrt{N}$ Fabbri et al. (2008).

A base para o funcionamento do algoritmo força bruta proposto por Costa e Cesar JR. (2009) é uma estrutura de dados auxiliar denominada *Sorted Euclidean Distance Representation* (SEDR). Essa estrutura contém dados como as possíveis distâncias (ou quadrado das distâncias) até um dado ponto, p , contido em uma forma S , a quantidade de pontos com determinada distância e a posição de seus respectivos pontos. A Figura 8 ilustra a organização geral para uma SEDR, considerando as quatro primeiras distâncias exatas em uma malha ortogonal. A lista disposta verticalmente contém um apontador para o próximo elemento desta lista, uma informação de distância e um ponteiro para uma lista de pontos que apresentam respectiva distância para um dado ponto p .

Segundo Costa e Cesar JR. (2009, p. 385), uma forma simples de implementar a SEDR é como uma matriz, onde o elemento SEDR $(k, 1)$ irá conter o valor da distância d , o elemento SEDR $(k, 2)$ indica o número $M(k)$ de pontos que apresentam uma distância d , e SEDR $(k, 3)$ até SEDR $(k, 3 + M(k))$ irá conter as coordenadas x e y dos respectivos pontos, sequencialmente, armazenados em pares. A Figura 9 ilustra a organização geral para uma SEDR representada em uma matriz para os pontos apresentados na Figura 8.

A estrutura SEDR deve ser construída para cada ponto, P , contido na forma analisada S , tornando-se altamente custoso em termos de memória e processamento. Uma vez construída, a SEDR contendo Nm distâncias (não incluindo o zero), pode ser utilizada para o processo da transformada de distância sem alterações adicionais Costa; Cesar Jr. (2009).

Dessa forma, inicialmente é representado a forma binária S em duas listas L_x e L_y , as quais contém as coordenadas x e y , respectivamente de todos os N pontos de S . O processo consiste basicamente em dilatar a forma S para cada um de seus raios sequencialmente sobre a matriz destino, a dilatação ocorre apenas quando o determinado ponto na matriz de destino apresenta valor -1 (valor de inicialização desta matriz), evitando assim que pontos com maior distância sobrescrevam lugares já calculados com distância inferior Costa; Cesar Jr. (2009). O pseudocódigo para a transformada de distância é ilustrado na Figura 10.

2.4.2 Algoritmo otimizado para transformada de distância

Algoritmos eficientes não-euclidianos (i.e. não utilizam métrica euclidiana para o cálculo da distância) da transformada de distância começaram a surgir a partir de 1966, enquanto apenas a partir da década de 1990 começaram a surgir algoritmos rápidos para EEDT Fabbri et al. (2008). Alguns destes algoritmos são apresentados nos trabalhos de Cuisenaire e Macq (1999), Lotufo e Zampiroli (2001) e Saito e Toriwaki (1994), estes e outros podem ser vistos também na *survey* de Fabbri et al. (2008).

Como há interesse na EDT tridimensional exata, devido a sua característica de invariância a rotação, algo essencial para um método de análise de textura robusto, optou-se pela utilização do algoritmo de Saito e Toriwaki (1994).

2.4.3 Algoritmo de Saito e Toriwaki (1994) para EEDT

Saito e Toriwaki (1994) desenvolveram um algoritmo para realizar a EEDT para uma imagem k -dimensional, utilizando k transformações unidimensionais (i.e. uma transformação para cada eixo geométrico). Este método garante as seguintes características citadas pelo autor:

- EEDT para o espaço n -dimensional sem grandes alterações no algoritmo.
- Aplicável com leve modificações para imagem digitalizadas utilizando amostragem com intervalos diferentes em cada eixo de coordenada. A exemplo disso, imagens de raio X, onde os intervalos entre as camadas apresentam intervalos de aquisição diferentes ao dos *pixels*.
- Requisitos mínimos de memória. Apenas uma matriz n -dimensional para armazenar a imagem de entrada e um único vetor para trabalho local são necessários para executar a EEDT n -dimensional.
- Algoritmos iterativos e transformações locais sempre unidimensionais.
- Tempo computacional relativamente pequeno, uma vez que, grosso modo, o tempo do algoritmo é proporcional tanto ao número de *voxels* da imagem de entrada quanto o raio médio de uma figura.
- Apresenta maior eficiência para computadores de uso geral, porém não necessariamente para computadores de hardware específico (ex: *Field Programmable Gate Array* - FPGA), pois não possui mascaras para execução simultânea.

O trabalho de Saito e Toriwaki (1994) apresenta quatro algoritmos diferentes, para realização da EEDT, desde o mais básico até o mais eficiente. Ambos apresentam a mesma ideia, porém o algoritmo mais eficiente foi projetado com uma otimização significativa relacionada a redução da região de busca para minimização de distâncias. Para fins didáticos, aqui é brevemente explicado apenas sua versão mais básica.

O algoritmo de Saito e Toriwaki (1994) para realização da EEDT como já dito acima, realiza uma transformação para cada eixo de coordenadas. Porém, apenas a transformada para o primeiro eixo é um caso particular. A partir do segundo eixo em diante o algoritmo pode ser generalizado, com mudanças apenas no sentido do eixo analisado.

Dada uma imagem constituída por M linhas e N colunas, a inicialização é feita marcando o plano de fundo (*background*) com 1 e pontos pertencentes a figura analisada (*foreground*) com 0.

Em sequência, a transformação 1 (Figura 11) consiste em minimizar a distância quadrática euclidiana entre *background* e *foreground*. Isso pode ser feito através de duas varreduras, uma varredura para frente (*forward scan*), que simplesmente consiste em marcar a distância quadrática euclidiana entre *background* e o último ponto encontrado do *foreground*, e então uma varredura para trás (*backward scan*), onde de fato é minimizada a distância entre *background* e o último ponto encontrado do *foreground*. Em outras palavras, considerando que a saída da transformação 1 é um conjunto $G = \{g_{ij}\}$ e imagem de entrada para esta é um conjunto de pixels $F = \{f_{ij}\}$, então g_{ij} é dado pela Equação 19.

$$g_{ij} = \min\{(i - x)^2; f_{ij} = 0 \text{ e } 1 \leq x \leq N\} \quad (19)$$

A segunda transformação (e as demais a partir deste ponto), utilizam da saída da transformação anterior como região de trabalho. Consiste basicamente em minimizar a soma dos valores de distâncias presentes atualmente em cada posição da vizinhança com a distância euclidiana quadrática entre o ponto a ser computado e seu vizinho analisado. Em outras palavras, considerando que a saída da transformação 2 é um conjunto $H = \{h_{ij}\}$ para uma imagem bidimensional, e ainda, a saída da transformação 1 e também entrada para transformação 2 é um conjunto $G = \{g_{ij}\}$. Considere a análise sobre o eixo y , portanto o valor h_{ij} é dado pela Equação 20. A Figura 12 ilustra este processo.

$$h_{ij} = \min\{g_{iy} + (j - y)^2; 1 \leq y \leq M\} \quad (20)$$

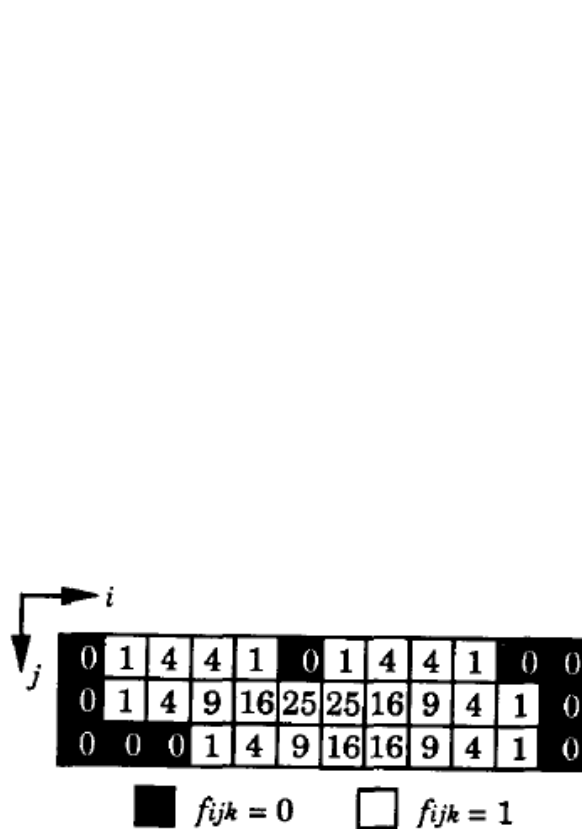


Figura 11 – Exemplo da transformação 1 para o algoritmo básico de Saito e Toriwaki (1994).
 Fonte: Saito e Toriwaki (1994).

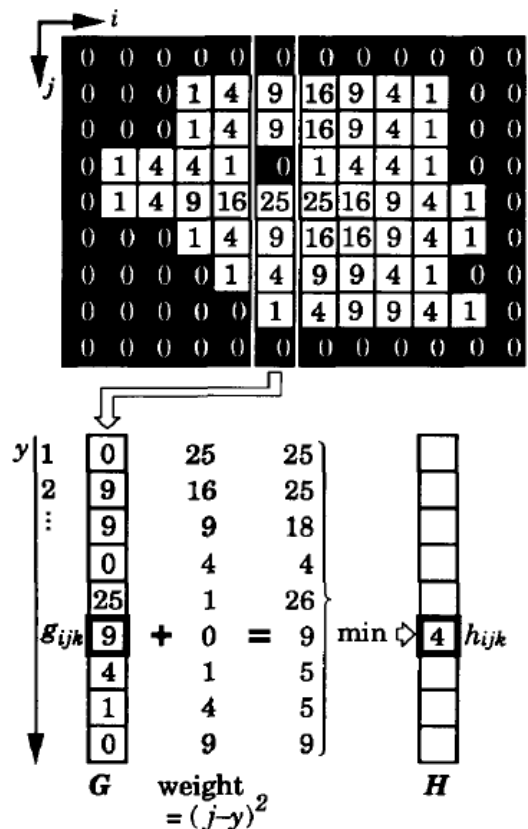


Figura 12 – Exemplo da transformação 2 para o algoritmo básico de Saito e Toriwaki (1994).
 Fonte: Saito e Toriwaki (1994).

Apesar de não parecer intuitivo o funcionamento do algoritmo proposto por Saito e Toriwaki (1994), ele resulta sempre resposta correta. Estes autores apresentam provas matemáticas quanto eficácia do método proposto, mostrando que de fato este algoritmo minimiza o quadrado da distância euclidiana. Sendo assim, se necessário a obtenção do valor da distância, deve-se apenas calcular a raiz quadrada para cada elemento contido no conjunto de saída.

A análise de complexidade apenas pode ser realizada com exatidão para o pior caso, pois como já discutido anteriormente, o tempo do algoritmo é proporcional tanto ao número de *voxels* da imagem de entrada quanto o raio médio da saída da transformação anterior.

Considerando o algoritmo mais eficiente proposto por Saito e Toriwaki (1994), onde o pseudocódigo para a segunda transformação é ilustrado na Figura 13, o pior caso acontece quando ambas as condições “ $(y + b) > (YYY - 1)$ ” e “ $(y - b) < 0$ ” forem satisfeitas para todos os *voxels* da imagem analisada.

Utilizando a mesma abordagem de análise de complexidade para o algoritmo força bruta, considerando uma EEDT bidimensional, dada uma imagem com dimensões C x R

composta por N pixels, considerando $n = \sqrt[3]{N}$, a complexidade no pior caso é $O(n^3)$, e grosso modo, a complexidade no caso médio poderia ser estimada por $O(R_m \cdot n^2)$, onde R_m é o raio médio da saída da transformação 1 e 2.

```

//FORWARD
a = 0;
for (y = 1; y < YYY; y++) {
  if (a > 0) a--;
  if (buff[y] > buff[y - 1] + 1) {
    b = (buff[y] - buff[y - 1] - 1) / 2;
    if (y + b > (YYY - 1)) b = (YYY - 1) - y;
    for (n = a; n <= b; n++) {
      m = buff[y - 1] + (n + 1) * (n + 1);
      if (buff[y + n] <= m) break;
      if (m < A[x][y + n][z]) {
        A[x][y + n][z] = m;
      }
    }
    a = b;
  }
  else {
    a = 0;
  }
}

//BACKWARD
a = 0;
for (y = YYY - 2; y >= 0; y--) {
  if (a > 0) a--;
  if (buff[y] > buff[y + 1]) {
    b = (buff[y] - buff[y + 1] - 1) / 2;
    if (y - b < 0) b = y;
    for (n = a; n <= b; n++) {
      m = buff[y + 1] + (n + 1) * (n + 1);
      if (buff[y - n] <= m) break;
      if (m < A[x][y - n][z]) {
        A[x][y - n][z] = m;
      }
    }
    a = b;
  }
  else {
    a = 0;
  }
}

```

Figura 13 – Pseudocódigo para os passos *forward scan* e *backward scan* do algoritmo mais eficiente para EEDT de Saito e Toriwaki (1994)
Fonte: Saito e Toriwaki (1994).

2.5 DIAGRAMAS DE VORONOI

Um Diagrama de Voronoi é uma forma de decomposição de um dado espaço, segundo Costa e Cesar JR. (2009, p. 390), qualquer conjunto de pontos isolados P_i , $i = 1, 2, \dots, N$ sobre um espaço R^2 definem um Diagrama de Voronoi do R^2 . Um Diagrama de Voronoi consiste em particionar o espaço R^2 em N regiões, denominadas R_i associadas a cada ponto P_i , de forma que qualquer ponto dentro da região R_i é mais próximo do ponto P_i Costa; Cesar Jr. (2009). A Figura 14-c ilustra o Diagrama de Voronoi obtidos para o conjunto de pontos isolados em Figura 14-a.

Segundo Costa e Cesar JR. (2009, p. 390), os Diagramas de Voronoi fornecem uma boa maneira de particionar o espaço bidimensional, pois cada região R_i pode ser entendida como a região de influência. Os Diagramas de Voronoi podem ser generalizados para um conjunto de formas isoladas, de modo que ao invés de considerar um ponto P_i para rotulação, considera-se uma forma (conjunto de pontos conectados).

Uma das formas de obtenção dos Diagramas de Voronoi é através de um algoritmo de propagação de etiquetas, o qual pode ser obtido por meio de uma ligeira adaptação do algoritmo da EEDT força bruta (COSTA; CESAR JR., 2009, p. 391). Neste trabalho a divisão de um

espaço tridimensional é realizada através de adaptações no algoritmo de Saito e Toriwaki (1994), utilizando outras métricas e regras para geração dos Diagramas de Voronoi.

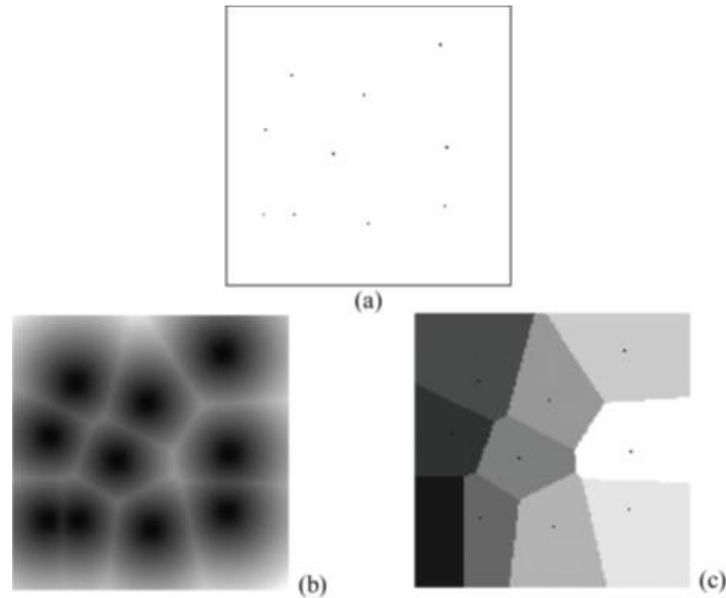


Figura 14 – Um conjunto de pontos isolados (a) e sua respectiva transformada de distância (b) e Diagrama de Voronoi (c). Cada região de influência foi representada por um nível de cinza distinto
 Fonte: Costa e Cesar JR. (2009, p. 391).

2.5.1 Rotulação

Por definição um Diagrama de Voronoi particiona o espaço considerando pontos ou formas (conjunto de pontos conectados) isoladas, de maneira que cada um destes receba uma região de Voronoi no processo de decomposição deste espaço Costa; Cesar Jr. (2009). Diferentemente, neste trabalho a segmentação do espaço é realizada através de rótulos, portanto, conjuntos de pontos desconectados podem ser considerados pertencentes a uma mesma “forma”.

A Figura 15-b ilustra o Diagrama de Voronoi obtido de um conjunto de pontos (Figura 15-a), já a Figura 15-c ilustra a segmentação do espaço através de apenas dois rótulos (representados através de níveis de cinza). Repare que as regiões de interesse para um mesmo rótulo podem ser desconexas.

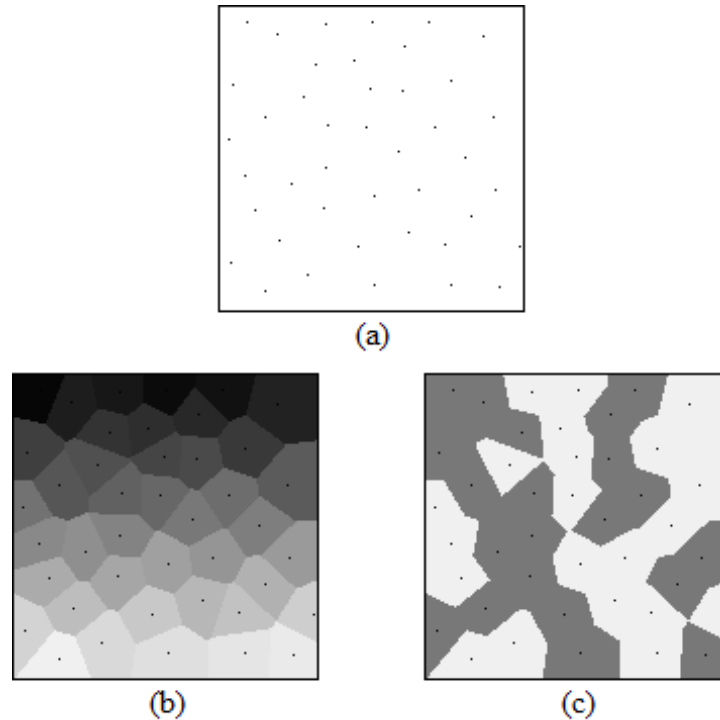


Figura 15– Um conjunto de pontos isolados (a) e seu respectivo Diagrama de Voronoi (b) e Diagrama de Voronoi modificado (c). Cada região de influência foi representada por um nível de cinza distinto

2.6 APRENDIZAGEM DE MÁQUINA

Dados os descritores de Minkowski anteriormente apresentados, e tendo como premissa deste trabalho realizar análise de textura utilizando os mesmos, é necessária a utilização dos métodos de aprendizado de máquina com objetivo de classificar as texturas dadas.

O problema de modelar funções para classificação geral de amostras a partir de exemplos específicos é fundamental para a aprendizagem. Grande parte da aprendizagem envolve a aquisição de conceitos gerais de exemplos de treinamento específicos Mitchell (1997). Por exemplo, pessoas aprendem continuamente os conceitos gerais de “flor”, “arvore”, “Situações em que eu deveria comer menos para não passar mal”, etc. Esta forma de generalização está vinculada ao raciocínio indutivo, o qual parte de exemplos específicos para chegar a uma conclusão generalizada.

Pode-se classificar subconjuntos contidos em conjuntos maiores, por exemplo, o subconjunto de flores está contido no conjunto de plantas. Dessa forma, pode-se distinguir um

subconjunto definindo uma variável booleana, sendo “verdadeiro” se um dado elemento está contido neste subconjunto, “falso” caso contrário Mitchell (1997).

Dessa forma, pode-se entender aprendizagem de máquina como sendo a capacidade de inferir automaticamente a definição geral de um dado conceito, apenas tendo alguns exemplos rotulados como pertencentes ou não deste conceito Mitchell (1997).

Existem várias formas de aprendizado de máquina, porém estes se dividem em três grupos principais, aprendizagem supervisionada, não-supervisionada e por reforço Russell; Norvig (2003). A aprendizagem supervisionada consiste em realizar o aprendizado a partir de amostras rotuladas previamente. De forma contrária, a aprendizagem não-supervisionada consiste em inserir amostras não rotuladas no algoritmo de aprendizagem, deixando-o sozinho para encontrar uma estrutura para suas entradas. Já o aprendizado por reforço é baseado num ambiente totalmente observável, onde ele deve descobrir a utilidade de suas ações.

Vários métodos de aprendizagem supervisionada são propostos na literatura, neste trabalho, optou-se utilizar o aprendizado *k-Nearest Neighbor* Mitchell (1997) e o classificador *Discriminant Analysis* Fisher (1936).

2.6.1 Aprendizado baseado em instâncias

Diferentemente dos métodos que constroem uma descrição geral e explícita para função de classificação dos exemplos fornecidos, métodos de aprendizado baseado em instâncias simplesmente armazenam os exemplos fornecidos. Esta classe de métodos é por vezes referidos como aprendizagem "preguiçosa", uma vez que, o processamento é adiado até que uma amostra necessite ser classificada.

A vantagem nesta abordagem é, que em vez de estimar a função de destino para todo espaço, esses métodos podem estimá-lo localmente e de forma diferente para cada nova instância a ser classificada. Isto tem vantagens significativas quando a função alvo é muito complexa, mas ainda pode ser descrita por um conjunto de aproximações locais menos complexas Mitchell (1997).

Uma desvantagem deste método está relacionada ao custo de classificação de novas instâncias, o qual pode ser significativamente elevado, pois quase todo processo de computação ocorre em tempo de classificação ao invés do momento em que a amostra de treinamento é inserida.

Outra desvantagem é a consideração de todos os atributos dos casos no processo de recuperação de exemplos de treinamento similares. Se o conceito alvo depende apenas de alguns dos muitos atributos disponíveis, amostras similares podem ser distanciadas dele Mitchell (1997).

Dentre as abordagens vinculadas ao processo de aprendizado baseado em instâncias está o algoritmo de aprendizagem *k-Nearest Neighbor*, que é discutido no tópico a seguir.

2.6.2 Aprendizado *k-Nearest Neighbor*

Segundo Mitchell (1997), o método mais básico de aprendizado baseado em instâncias é o algoritmo de aprendizagem *k-Nearest Neighbor* (k-NN). Este assume que todas as instâncias correspondem a pontos no espaço n -dimensional R^n . Os vizinhos mais próximos de uma dada instância são definidos como, aqueles que possuem menor distância euclidiana Mitchell (1997).

Dado uma instância arbitrária x , a qual pode ser descrita pelo vetor $\{a_1(x), a_2(x), \dots, a_n(x)\}$, onde a_r denota o r -ésimo atributo da instância x . Assim, a distância entre duas instâncias x_i e x_j é definida como $d(x_i, x_j)$, onde:

$$d(x_i, x_j) = \sqrt{\sum_{r=1}^n (a_r(x_i) - a_r(x_j))^2} \quad (6)$$

Dada uma função $f: R_n \rightarrow V$, onde V é um conjunto finito $\{v_1, \dots, v_s\}$. O algoritmo de aprendizagem *k-Nearest Neighbor*, consiste basicamente em retornar o valor, $f(x_q)$, mais frequente (moda) entre os k elementos vizinhos mais próximos de x_q . Se escolhermos $k = 1$, então o algoritmo *1-Nearest Neighbor* irá retornar o valor do elemento mais próximo da amostra em análise. Uma boa técnica para otimizar o processo de estimação quando $k = 1$ é decompor o espaço em regiões de Voronoi Mitchell (1997).

2.6.3 Discriminant Analysis

Segundo Cai, He e Han, (2008), LDA é um método popular de extração de características que preserva a separação entre classes, ele é uma generalização do método *Fisher's linear discriminant* Fisher (1936). Porém também pode ser aplicado como classificador.

De acordo com Mathworks (2016), o modelo para o classificador *Discriminant Analysis*, é o seguinte: Cada classe (Y) gera dados (X) utilizando uma distribuição normal multivariada. Isto é, o modelo assume que X tem uma distribuição de mistura de Gauss McLachlan; Peel (2005).

Para a análise através do discriminante linear, o modelo utiliza médias variáveis e a mesma matriz de covariância para cada classe. Já para a análise discriminante quadrática as médias e matrizes de covariância variam.

Uma das características importantes deste método é a compatibilidade com conjuntos de dados de treinamento que apresentam correlação. Dessa forma, trazendo um modelo muito mais robustos para o método proposto neste trabalho.

2.7 VALIDAÇÃO ESTATÍSTICA

Dado as texturas, dadas suas características obtidas via descritores de Minkowski, e ainda, dado o método de aprendizagem de máquina (k-NN) a utilizar, é necessário avaliar o resultado com base em fundamentos estatísticos.

Dentro da estatística existem várias formas de se avaliar um experimento, mais especificamente para um experimento de classificação supervisionado, a medida mais utilizada é a taxa de erro (ou sua contraparte, a taxa de acerto). Para obtenção de tal métrica é necessário porém seguir alguns conceitos estatísticos, no sentido de que os resultados obtidos se tornem válidos.

Nesse contexto, é necessária uma breve explicação acerca da normalização dos dados (z-score), da validação cruzada e da obtenção das métricas de análise (taxas de acerto e matriz de confusão).

2.7.1 Distribuição Normal

Segundo Devore (1982), em probabilidade e estatística, a distribuição normal é a mais importante de todas. Diversas populações numéricas podem ser ajustadas para uma distribuição normal adequada. Por exemplo, medidas de acerto em tiro ao alvo, pesos, alturas, erros de medições científicas e pontuações em testes variados. Mesmo que as próprias variáveis medidas individualmente não obedeçam a distribuição normal, geralmente a soma e as médias das variáveis conseguem seguir esta distribuição Devore (1982).

De acordo com Devore (1982), diz-se que uma variável contínua X possui uma distribuição normal com parâmetros μ (média) e σ (desvio padrão), onde $-\infty < \mu < \infty$ e $0 < \sigma$ se a função densidade de probabilidade (fdp) de X for:

$$f(x; \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(x-\mu)^2/(2\sigma^2)} \quad (7)$$

Onde e denota a base do sistema de logaritmos naturais sendo aproximadamente igual a 2,71828 e π representa uma constante matemática familiar com valor aproximado de 3,14159.

A Figura 16 apresenta gráficos de $f(x; \mu, \sigma)$ para diversos pares de (μ, σ) . Cada gráfico em formato de sino (gaussiana) é simétrico em torno de μ , ou seja o centro do gráfico é tanto média quanto mediana Devore (1982).

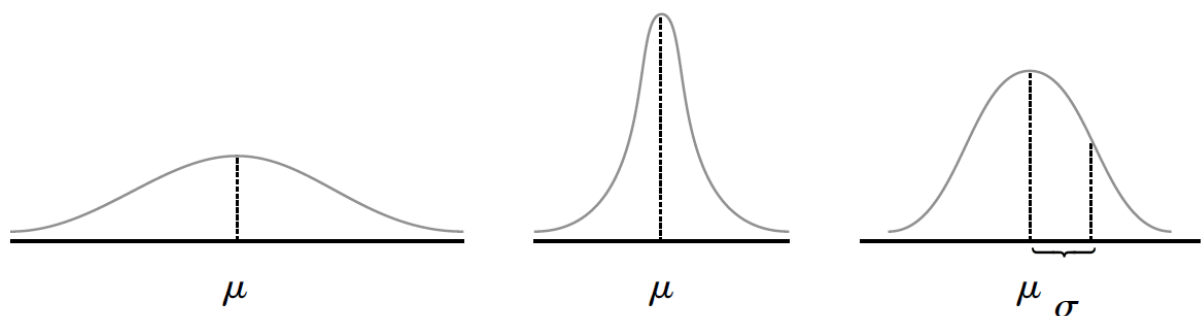


Figura 16 – Gráficos de fdps normais
Fonte: Devore (1982).

2.7.2 Distribuição Normal Padrão

O cálculo da probabilidade de uma variável normal, X , com parâmetros μ e σ estando entre dois pontos a e b , $P(a \leq X \leq b)$, é dado resolvendo a Equação (8), porém nenhuma técnica de integração analítica pode resolver este problema. A solução para isso foi a padronização dos valores de μ e σ e criação de uma tabela através de integração numérica Devore (1982).

$$\int_a^b \frac{1}{\sqrt{2\pi}\sigma} e^{-(x-\mu)^2/(2\sigma^2)} dx \quad (8)$$

De acordo com Devore (1982), a distribuição normal com parâmetros $\mu = 0$ e $\sigma = 1$ é denominada **distribuição normal padrão**. Uma variável aleatória com distribuição normal padrão é denotada por Z e a função densidade de probabilidade de Z está expressa na Equação (9). Sendo está, uma distribuição de referência, a partir da qual, podem ser obtidas informações sobre outras distribuições normais.

$$f(z; 0, 1) = \frac{1}{\sqrt{2\pi}} e^{-z^2/2} \quad -\infty \leq z < \infty \quad (9)$$

A normalização z-score, é uma ferramenta frequentemente utilizada para comparar uma amostra a um desvio normal padrão.

$$z = \frac{x - \mu}{\sigma} \quad (10)$$

O valor absoluto de z representa a distância entre o escore bruto e a média da população em unidades de desvio padrão, conforme a Equação (10).

Neste trabalho, a normalização z-score é utilizada para normalização do vetor de características. Normalizando cada índice do vetor de características com toda a população de instâncias presentes, com objetivo de igualar a influência para todos os elementos do vetor de características.

2.7.3 Validação cruzada por *k-fold*

Na área de aprendizagem de máquina, o desempenho de um classificador é geralmente medida em termos de erro de predição. Na maioria dos problemas do mundo real, o erro não pode ser calculado com exatidão, dessa forma deve ser estimado Rodríguez et al. (2010).

Geralmente, classificadores são induzidos a classificação através de instancias de treinamento, usando um algoritmo de aprendizagem. Todo classificador tem um erro de predição associado. No entanto, geralmente este erro é desconhecido, e deve ser calculado a partir de dados. Um estimador de erro de um dado classificador é uma variável aleatória, e sua qualidade é medida geralmente por meio de seu *bias* e variância Rodríguez et al. (2010).

Segundo Rodríguez et al. (2010), uma destas técnicas de estimativa de erro de classificação, e provavelmente a mais popular, é a validação cruzada por *k-fold*. Resumidamente, este método consiste em dividir o conjunto total de dados de treinamento em k subconjuntos mutuamente exclusivos de igual tamanho, então o algoritmo de classificação é ensinado utilizando $k-1$ subconjuntos, e o subconjunto restante é utilizado como teste. Este processo é repetido k vezes, de forma circular, ou seja, sempre escolhendo um subconjunto inédito dos k subconjuntos criados inicialmente.

A estimativa de erro é o valor médio dos erros cometidos em cada etapa. Assim o estimador de erro validação cruzada por *k-fold*, depende de dois fatores: o conjunto de treinamento e a forma com que são criados os k subconjuntos Rodríguez et al. (2010). Frequentemente a metodologia de validação empregada é por *10-fold*.

2.7.4 Matriz de confusão

A matriz de confusão oferece uma medida efetiva de um dado modelo de classificação, pois apresenta o número de classificações corretas para dadas amostras, além do número de classificações incorretas, mostrando quando e onde algoritmo de classificação “se confundiu”.

Uma matriz de confusão de tamanho $n \times n$ associada com um classificador mostra a classificação prevista e real, onde n é o número de classes diferentes Visa et al. (2011). A Tabela 1 mostra uma matriz de confusão para $n = 2$, onde os elementos têm os seguintes significados:

- a é o número de previsões negativas corretas;
- b é o número de previsões positivas incorretas;
- c é o número de previsões negativas incorretas;
- d é o número de previsões positivas corretas;

Tabela 1 - Matrix de confusão para um problema de classificação de duas classes.

| | Previsto Negativo | Previsto Positivo |
|---------------|----------------------|----------------------|
| Real Negativo | a | b |
| Real Positivo | c | d |

Fonte: Visa et al. (2011, p. 2).

Segundo Visa et al. (2011), a precisão e o erro de classificação pode ser obtido a partir dessa matriz como se segue:

$$Precisão = \frac{a + d}{a + b + c + d} \quad (11)$$

$$Erro = \frac{b + c}{a + b + c + d} \quad (12)$$

2.8 ANÁLISE DE TEXTURAS BASEADA EM DESCRITORES FRACTAIS

O método proposto neste trabalho segue a linha de pesquisa de outros trabalhos em análise de texturas através de descritores fractais Backes et al. (2009, 2012 e Casanova et al. (2016).

Inicialmente foi proposto o método Bouligand-Minkowski em tons de cinza em Backes, Casanova e Bruno (2009). Após isso, veio o método proposto por Backes, Casanova e Bruno (2012), que consiste numa análise através dos canais de cores porém sem distinção dos mesmos. Por fim, veio o trabalho de Casanova et al. (2016), que também realiza a análise através dos canais de cores porém agora com distinção entre eles, gerando assinaturas mais ricas.

Objetivando maior esclarecimento do método proposto neste trabalho, além de deixar claro o que já foi pesquisado e o que ainda pode ser pesquisado na área, esta seção apresenta resumidamente os métodos destes trabalhos na mesma linha de pesquisa.

2.8.1 Método de Backes, Casanova e Bruno (2009)

A complexidade de objetos pode ser medida através da estimativa da dimensão fractal de tais objetos Backes et al. (2009). Segundo Tricot (1995), o método de Bouligand-Minkowski ou Dimensão de Minkowski é um dos métodos que produz os resultados mais acurados e consistentes para a Dimensão Fractal, Backes (2006) confirma isso a partir de resultados experimentais.

O trabalho de Backes, Casanova e Bruno (2009) apresenta uma abordagem com base na Dimensão de Minkowski para identificar plantas com base na textura/complexidade de suas folhas. Tal abordagem também poderia ser utilizado em outras aplicações na área de análise de textura. A metodologia proposta por eles, tem como base a dimensão fractal volumétrica, o qual extrai as características a partir das imagens das folhas que fornecem a informação para identificá-las.

Este método proposto por Backes, Casanova e Bruno (2009) gera uma superfície em um espaço tridimensional como representação de uma imagem de textura bidimensional (é importante ressaltar que a representação desta imagem de textura é feita em escala de cinza). Em sequência a análise de complexidade desta superfície tridimensional é feita aplicando-se o método de estimativa da dimensão fractal de Bouligand-Minkowski. Como principal interesse conseguir uma assinatura (vetor de características) a partir do volume calculado na região de influência da superfície dilatada.

2.8.2 Método de Backes, Casanova e Bruno (2012)

Segundo Backes, Casanova e Bruno (2012) a informação de cor é muito importante em texturas de cenários naturais, tais como superfícies de folhas. A proposta do trabalho “*Color texture analysis based on fractal descriptors*” é uma abordagem baseada na dimensão fractal de Bouligand-Minkowski, com objetivo de realizar a análise de texturas coloridas.

Um estudo com a dimensão fractal em imagens coloridas foi proposta recentemente em Ivanovici e Richard (2011), que sugeriu utilizar a dimensão fractal como métrica para caracterizar textura. No entanto, segundo Backes, Casanova e Bruno (2012), a utilização da dimensão fractal sozinha (valor único), como característica única para descrever texturas, não apresenta bons resultados para problemas de classificação com muitas classes.

A abordagem proposta por Backes, Casanova e Bruno (2012), realiza a análise de cada canal de cor de forma independente (como no trabalho de She e Huang (1994)), assim estuda a relação entre canais de cor de modo geral em uma espécie de abordagem multiespectral.

Estes autores fazem uso do método de Bouligand-Minkowski para o cálculo da dimensão fractal. O processo de estimação da dimensão fractal necessita do estudo da região de influência, o qual é feito através da transformada de distância, uma vez que é uma forma rápida de realizar o processo necessário Backes et al. (2012).

Após o processo de dilatação realizado pela transformada de distância (necessário para calcular o volume de influência), é computado o volume, $V(r_i)$, de cada superfície ou forma tridimensional presente no espaço em análise sob um raio r_i Backes et al. (2012).

O processo clássico para análise de textura a partir da dimensão fractal do método de Bouligand-Minkowski é realizada através da inclinação, α , do plot (contínuo) da relação $\log(V(r))$ versus $\log(r)$. Com a justificativa que este processo não consegue descrever a riqueza de detalhes que a curva log-log computada pelo método de Bouligand-Minkowski proporciona, o trabalho de Backes, Casanova e Bruno (2012) apresenta uma abordagem de caracterizar textura utilizando todo o vetor $V(r)$ que os autores convencionaram chamar de descritores fractais Backes et al. (2012).

O vetor de características, $\Psi(n) = [e_1, e_2, \dots, e_n]$, é obtido através das diferenças, e_i , entre a reta de regressão linear, $y = a * \log(r_i) + b$, e o log do volume dos raios, $\log(V(r_i))$, como segue na Equação (11) Backes et al. (2012).

$$e_i = a * \log(r_i) + b - \log(V(r_i)) \quad (11)$$

De acordo com Mäenpää; Pietikäinen (2004), abordagens atuais para análise de texturas coloridas podem ser divididas em duas categorias: métodos que processam cor e textura informações separadamente, e aqueles que consideram cor e textura um fenômeno comum. Segundo Backes, Casanova e Bruno (2012), o trabalho deles pode se encaixar nas duas categorias.

Para a primeira categoria, eles utilizam os canais de cores separadamente, obtendo vetores característicos para cada canal de cor, conseguindo assim um vetor de características com a união (concatenação) destes. Ou seja, para uma imagem RGB (*Red* - Vermelho, *Green* - Verde, *Blue* - Azul), é calculado um vetor $\Phi(n) = [\Psi_R(n), \Psi_G(n), \Psi_B(n)]$ Backes et al. (2012).

Já para segunda categoria, o trabalho proposto por estes autores, consiste em analisar as cores em uma única transformação. Considerando cada canal de cor RGB como uma superfície, obtém-se o vetor de características, $\Psi(n)$, colocando ambas as superfícies em um único espaço tridimensional. Esta assinatura nos permite estudar todos os canais em combinação, levando em consideração as correlações entre eles Backes et al. (2012).

Esta segunda técnica leva em consideração a proximidade dos canais de cor, uma vez que a proximidade influencia no volume final de cada cor. Consequentemente, esta abordagem considera o conceito de auto similaridade e auto afinidade entre os canais de cor Backes et al. (2012).

A diferença para a abordagem proposta neste trabalho e o trabalho de Backes, Casanova e Bruno (2012) é a utilização de rótulos em conjunto com a transformada de distância. Assim é possível, além de analisar o conceito de auto similaridade a auto afinidade dos *pixels*, computar de forma individual qualquer canal de cor e/ou objeto de interesse na imagem, gerando assim descritores mais ricos.

2.8.3 Método de Casanova et al. (2016)

O trabalho aqui apresentado segue a mesma linha de raciocínio utilizada por Casanova et al. (2016), porém com modificações que generalizam o problema, não limitando a análise em rotulações fixas, dando liberdade para a escolha da forma como realizar a rotulação, sem alterar a complexidade temporal do algoritmo. Como a linha de raciocínio é a mesma, a descrição nesta seção do trabalho de Casanova et al. (2016) é resumida.

O trabalho antecessor de Casanova et al. (2016), sendo ele Backes, Casanova e Bruno (2012) apresentado anteriormente, realiza a análise de textura utilizando multicanais de cores, além disso, também é proposto um vetor de característica alternativa, em que todos os três canais são analisados simultaneamente (isto é, num volume tridimensional único).

Segundo Casanova et al. (2016), embora a ideia principal desta alternativa seja muito boa e interessante, ela não apresenta melhores resultados do que a versão multicanal. Assim Casanova et al. (2016) propõe uma modificação no método a fim de explorar a relação entre os canais de cores em padrões de textura em vez das características de um único canal de cor.

Este método não mescla os canais de cores como sendo apenas um objeto no espaço tridimensional. Porém também não separa os canais de cores em espaços distintos, eles estão

contidos no mesmo espaço “competindo pelo mesmo recurso”, ou seja, interferem entre si no processo de dilatação, sendo assim, é possível analisar relação entre os canais de cores.

O método inicialmente mapeia cada canal de cor para o espaço tridimensional, gerando assim uma “nuvem de pontos”, em seguida, dilata cada ponto da nuvem de pontos utilizando uma esfera com um raio variável. Para cada canal de cor e para cada raio dilatado, obtêm-se os volumes da estrutura dilatada. Estes valores expressam a complexidade da distribuição de intensidade/cor Casanova et al. (2016).

As medidas de volume (os descritores fractais) são então concatenadas para compor o vetor de características. Os descritores são ordenados pelo raio de influência, e o raio máximo utilizado é definido como aquele que melhor descreve as imagens de textura no geral. Ou seja, são realizados testes, e ao fim, é escolhido quantos descritores usar como parte do vetor de características.

A diferença principal entre o método de Casanova et al. (2016) e o proposto neste trabalho está na forma como a segmentação do espaço é realizada. Aqui utiliza-se a segmentação através dos Diagramas de Voronoi, permitindo assim, a escolha de formas alternativas ou complementares de rotulação, pois Casanova et al. (2016) tem uma implementação limitada a apenas rótulos RGB, e a apenas 3 rótulos.

3 MATERIAIS E MÉTODOS

3.1 MATERIAIS

O método de análise de textura proposto neste trabalho utiliza a transformada de distância como ferramenta para obtenção do vetor de características. Porém, mesmo utilizando um algoritmo eficiente para esta tarefa, o processo necessita de uma linguagem de baixo nível para obtenção de um melhor desempenho temporal. Requisito este, essencial para maior flexibilidade em momento de testes. Portanto, optou-se pelo uso da linguagem C como instrumento para o cálculo da EEDT.

Objetivando continuação deste trabalho e, para melhor manuseio dos dados obtidos através do método de análise de textura aqui proposto, decidiu-se a utilização do software Matlab como ferramenta para leitura das imagens em análise e pós-processamento dos dados.

Para isso, utilizou-se o compilador MEX disponibilizado pelo próprio Matlab, o qual permite utilizar a linguagem C como ferramenta para implementação de funções. Em outras palavras, foi implementado um código em linguagem C que recebe do software Matlab uma nuvem de pontos (coordenadas e rótulos), em sequência, realiza a transformada de distância sobre esta nuvem de pontos e a segmentação do espaço por Diagramas de Voronoi, e por fim extrai os descritores fractais, que são retornadas para o software Matlab.

Para uma análise precisa e com maior confiança sobre a eficiência do método proposto, optou-se pela utilização do Software Weka, pois este, é capaz de realizar normalizações, validação cruzada, além de gerar matriz de confusão e avaliar a eficiência dos dados. Sendo assim, após a obtenção dos atributos de todas as imagens pertencentes a base de dados utilizada, a matriz de atributos é convertida no formato Arff, o qual é aceito pelo software Weka, e os dados são então validados. Também foi realizado a etapa de validação através do software Matlab utilizando o classificador LDA, para realizar a comparação dos resultados com o trabalho de Casanova et al. (2016).

A base de textura utilizada foi a Outex Ojala; Maenpaa (2002), que consiste em um *benchmark* para avaliação empírica dos algoritmos de classificação de textura. Essa base contém uma grande coleção de texturas de superfície capturadas utilizando os mesmos critérios de angulação e iluminação. Porém apresentam variados tipos de texturas de superfície e cenas

naturais e artificiais, facilitando assim, a construção de uma ampla gama de problemas de análise de textura.

Também existem coleções de texturas de superfície que exibem variações para uma determinada referência bem definida em termos de iluminação, rotação e resolução espacial Ojala; Maenpaa (2002).

A Outex contém várias bases de dados para diferentes tipos de aplicações, porém optou-se pela utilização da base de conjunto de testes com ID Outex_TC_00013 (a Figura 17 ilustra uma imagem por classe de textura, para algumas classes desta base), por ser projetada especificamente para análise de textura colorida, sendo um dos objetivos deste trabalho. Este conjunto de testes apresenta as seguintes características:

| | |
|---------------------|-------------|
| Texturas | 68 texturas |
| O tamanho da janela | 128x128 |
| Imagens por textura | 20 imagens |
| Iluminação | Inca |
| Rotação | 0 |
| Resolução | 100dpi |

Quadro 1 - Características do conjunto de testes Outex_TC_00013
Fonte: Ojala e Maenpaa (2002).

As imagens da base de conjunto de testes com ID Outex_TC_00013 são disponibilizadas em um único diretório nomeadas numericamente em ordem crescente, iniciando em zero, de forma que conjuntos de 20 imagens sequencialmente pertencem a uma mesma classe de textura.



Figura 17 – Amostras de classes de textura contidos na base de conjunto de testes da Outex com ID Outex_TC_00013

3.2 METODOLOGIA

Seguindo o raciocínio de Casanova et al. (2016), inicialmente foi necessário a implementação do processo de conversão de imagens de textura para objetos segmentados por rótulos, onde estes objetos devem estar presentes em um ambiente propício para realização do processo de estimativa da dimensão fractal do mesmo.

Em seguida foi realizada a implementação da transformada de distância juntamente com os Diagramas de Voronoi, os quais são imprescindíveis para eficiência do processo da estimativa da dimensão fractal.

Na sequência, tendo em mãos a EEDT segmentada por rótulos, é possível realizar a computação do volume dilatado para cada raio analisado (i.e. obtenção dos “descritores fractais”).

Á partir deste ponto, estes descritores podem ser vistos como atributos presente em um vetor de características para descrição de imagens de textura, que por fim, são utilizados na validação do método proposto.

Para facilitar o entendimento, o procedimento descrito nesta sessão é ilustrado através de um fluxograma como apresentado na Figura 18, todo este procedimento é melhor detalhado nas subsecções a seguir.

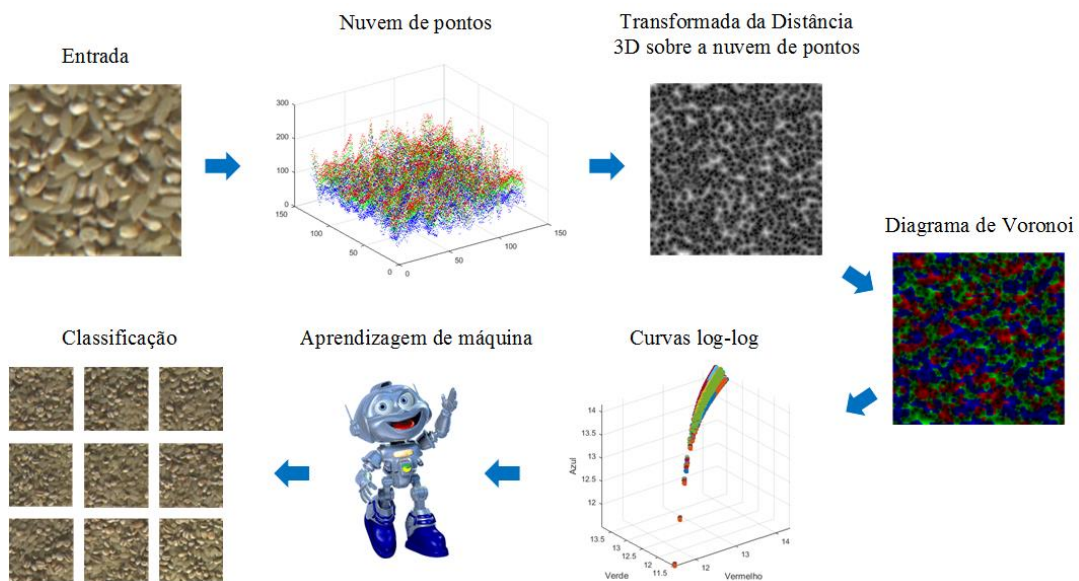


Figura 18 – Fluxograma ilustrando o procedimento necessário para o método proposto neste trabalho

3.2.1 Definição da região de trabalho

A discretização de uma imagem contínua (processo de amostragem), resulta em uma imagem digital, onde o menor elemento deste processo é denominado *pixel*. Dessa forma, uma imagem digital é composta por $M \times N$ *pixels*, e o valor presente em um *pixel* corresponde a intensidade luminosa de uma cor. A maioria das imagens digitais tem três canais de cores, vermelho, verde e azul (quando adotado o sistema *RGB*) e a intensidade luminosa está entre 0 a 255.

O método de análise de textura proposto torna necessária a conversão de uma imagem bidimensional em um espaço tridimensional. Portanto, a definição da região de trabalho para EEDT é a conversão de uma imagem bidimensional (ver Figura 19) em um cubo de 256 camadas (ver Figura 20).

O processo consiste em transformar a intensidade, k , de um dado *pixel* na posição, i, j , de uma imagem bidimensional em uma coordenada, i, j, k , para o cubo. Desta forma, é marcado com 0 cada um dos *voxels* correspondentes a um *pixel* (i.e., dada a intensidade k de um *pixel* na posição i, j de uma imagem bidimensional, definir o *voxel* de coordenada i, j, k no cubo de 256 camadas como 0), e os que não corresponde a *pixels* (plano de fundo) são marcados com 1.

Assim como para a EEDT, o processo de segmentação do espaço via Diagramas de Voronoi também necessita de uma matriz cubica para armazenar informações. Neste trabalho, foi pensada em uma forma de segmentar o espaço considerando regiões de empate de rótulos (i.e. regiões equidistantes entre pontos com rotulações diferentes), os quais devem receber parcelas de espaço iguais quando isso ocorre. Dessa forma, cada posição da matriz de rotulação armazena um número inteiro de 32 bits, e ainda, cada um dos bits deste número representa a presença ou não de um rótulo nesta dada posição espacial.

O processo de inicialização da matriz de rotulação ocorre de forma semelhante a definição da região de trabalho para EEDT. Ou seja, para um *voxel* correspondente a um *pixel* da imagem bidimensional é analisado seu rótulo e então o bit correspondente a este é marcado.

A exemplo, considerando que cada canal de cor em uma imagem de textura RGB é um rótulo, os bits 0, 1 e 2 correspondem aos canais R, G, e B respectivamente. Desta forma, um *pixel* na posição, i, j , da imagem bidimensional com cor branca (255 255 255) apresenta três pontos coincidindo no mesmo *voxel* (posição, i, j, k , onde k é 255 para ambos), porém com rotulação diferente, assim, o valor em binário deste *voxel* presente na matriz de rotulação é 0...0111.

Para permitir livre expansão dos objetos fractais torna-se necessário um aumento no tamanho total do cubo. Ou seja, todas as fronteiras devem ser expandidas de forma que elas não limitem a EEDT até um raio preestabelecido. Este procedimento é melhor detalhado na subseção seguinte à partir de um exemplo numérico.



Figura 19 – Imagem de um inseto preso em uma teia de aranha

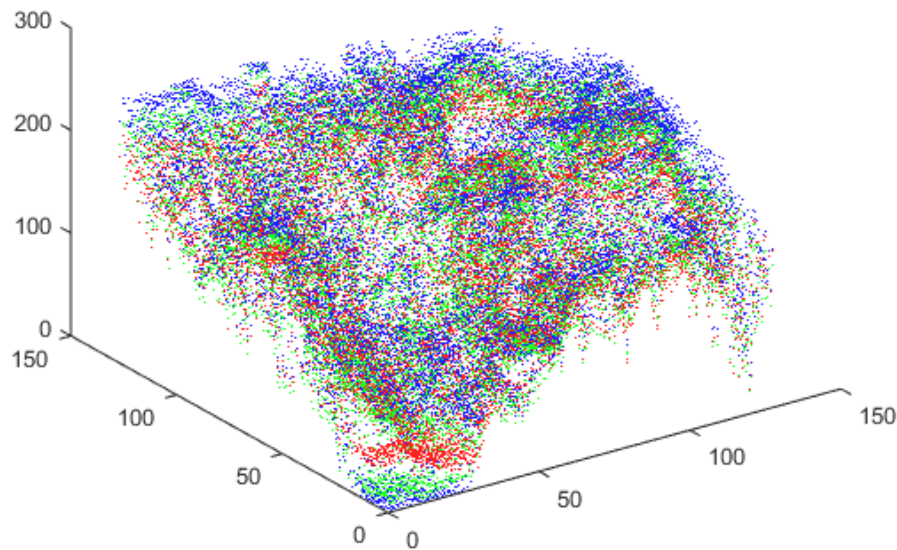


Figura 20 – Um conjunto de pontos isolados, conversão da Figura 19 em uma nuvem de pontos

3.2.2 Obtenção dos descritores fractais

O método de estimativa da dimensão fractal de Bouligand-Minkowski (Seção 2.4) pode ser auxiliado através de alguma técnica para EEDT (Seção 2.5). De forma que, dado um conjunto de pontos no espaço tridimensional (ver Figura 20), a realização da transformada de distância exata com métrica euclidiana utilizando rótulos sobre este conjunto de pontos gera um Diagrama de Voronoi modificado (ver Figura 15-c).

O processo de estimativa da dimensão fractal de Bouligand-Minkowski calcula o volume sob a região de expansão dos pontos no espaço tridimensional, para cada raio r em uma malha ortogonal contidos na faixa $1 \leq r \leq x$, onde x é um valor preestabelecido. A relação entre um dado raio e a contagem do volume de todos os pontos expandidos para este respectivo raio, r , para $1 \leq r \leq x$, dará os denominados “descritores fractais”, que podem ser utilizados para estimar a dimensão fractal (Figura 2), ou ainda, utiliza-los individualmente como partes do vetor de características.

A diferença entre o método apresentado neste trabalho e os já existentes Backes et al. (2009, 2012 e Casanova et al. (2016) está no processo de estimativa da dimensão fractal através do método de Bouligand-Minkowski, o qual é computado para cada rótulo. Pois a forma como os rótulos são escolhidos é livre, dependendo exclusivamente do objetivo a ser alcançado. Pode-se, por exemplo, utilizar cada *pixel* como sendo um rótulo, um determinado objeto ou ainda um canal de cor.

Para facilitar o entendimento do método, considera-se a priori 3 rótulos, um para cada canal de cor (R, G e B), assim como apresentado em Casanova et al., (2016).

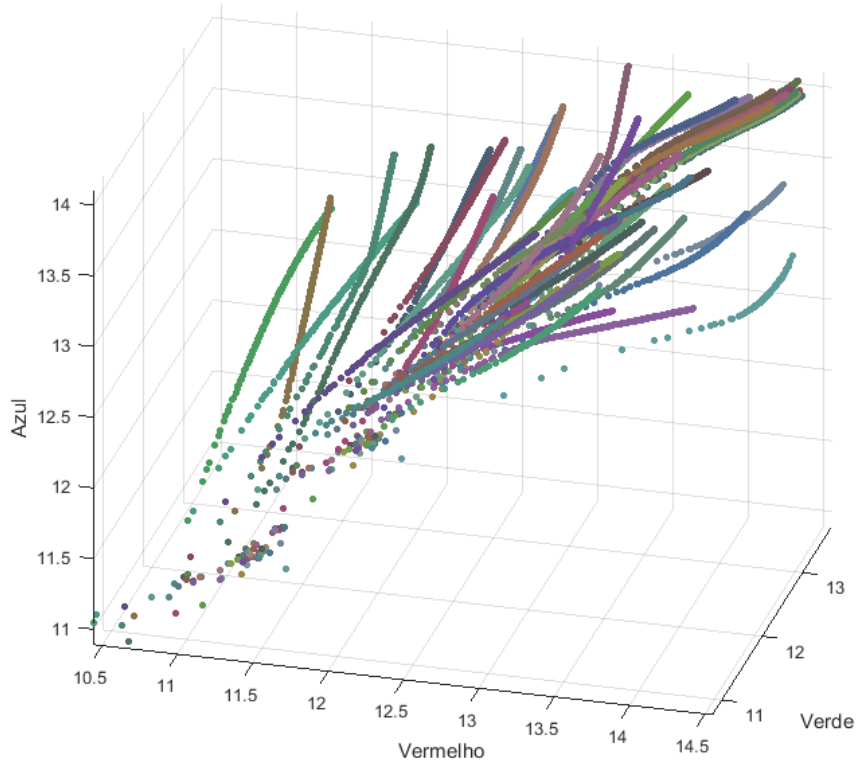


Figura 21 – Esboço de assinatura para uma instância de cada classe de textura

Nesse caso, onde cada rótulo representa um canal de cor, percebe-se que uma determinada cor influencia no volume de outra por estarem no mesmo espaço durante o processo EEDT (Figura 23). Observa-se que a região de expansão de um *pixel* é reprimida devido à expansão de seus *pixels* vizinhos, limitando não somente o raio de expansão de canais de cores diferentes, mas também os raios de expansão dos *pixels* do próprio canal. Diferentes classes de textura apresentam assinaturas (vetor de características) diferentes, isso pode ser observado na Figura 21, onde foi representado o vetor de características de uma instância de cada classe de uma dada base de imagens, cada uma das curvas representa uma instância.

A seguir é apresentado um exemplo numérico do processo de EEDT segmentada por rótulos, utilizando um rótulo para cada canal de cor, este exemplo pode ser encontrado também em Casanova et al. (2016). A imagem de textura utilizada neste exemplo tem valores de intensidade luminosa entre 0 e 2 (Figura 22), e o raio r analisado está contido na faixa $1 \leq r \leq 2$. Portanto, o cubo terá 3 camadas devido à faixa de intensidade luminosa, e mais 4 camadas para permitir livre expansão na nuvem de pontos (2 camadas a mais em cima e em baixo, para permitir expansão até $r = 2$ para os *pixels* contidos nas bordas da imagem).

Ressaltando, para calcular o resultado obtido na figura Figura 23 foi utilizada a transformada de distância exata com métrica euclidiana, onde as “sementes” para a EEDT são os *pixels* apresentados na Figura 22, que geram as distâncias zero na nuvem de pontos, Figura 23. Em sequência, calcula-se as distâncias euclidianas dos pontos do cubo externos a nuvem de pontos, como sendo a distância euclidiana quadrática $((x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2)$, onde x_0 , y_0 e z_0 expressam a coordenada do ponto, contido na nuvem de pontos, mais próximo a um ponto externo com coordenadas x , y e z .

Após a computação do método, obtém-se a segmentação do espaço por Diagramas de Voronoi juntamente com a EEDT, como ilustrado na Figura 23, que utiliza como imagem de entrada o exemplo da Figura 22. Durante o processo da transformada de distância ocorreram vários empates, todos foram representados com as cores resultantes da combinação (no sistema RGB) dos canais empatados. Ao término da computação da EEDT pode-se realizar a extração dos descritores fractais (Tabela 3).

| | | | | | |
|----------|---|-------|---|------|---|
| 0 | 2 | 2 | 0 | 1 | 2 |
| 1 | 1 | 0 | 1 | 0 | 1 |
| Vermelho | | Verde | | Azul | |

Figura 22 – Três canais de cores de uma imagem de exemplo

Verificou-se que o tratamento de empates feitos de maneira ponderada apresenta melhores resultados. Portanto, seguindo a mesma metodologia de Casanova et al. (2016), foi adotado neste trabalho a ponderação da região de empate como sendo a divisão do volume presente num *voxel* para ambos os rótulos de maneira igual, como exemplificado na Tabela 2.

A computação do volume é efetuada inicialmente calculando o “histograma” dos raios dilatados, em seguida este histograma é acumulado de maneira sequencial, dando lugar a um vetor contendo o volume dilatado para cada raio analisado. Como anteriormente definiu-se que o raio r analisado está contido na faixa $1 \leq r \leq 2$, então apenas estes valores serão utilizados para extração dos descritores fractais.

Após a obtenção dos descritores, deve-se calcular o log para cada um deles e colocá-los em sequência, formando assim o vetor de características. Recomenda-se utilizar todos os raios possíveis no momento da extração dos descritores fractais, para que em momento de testes tenha-se a possibilidade de escolher até quantos atributos serão utilizados. Ou seja, deve-se realizar testes para saber até qual raio torna-se vantajoso utilização como parte do vetor de

características. Uma vez que, excesso de atributos pode ser altamente prejudicial para métodos de aprendizagem de máquina, ou em contrapartida, falta de atributos pode não descrever com eficiência o objeto em análise.

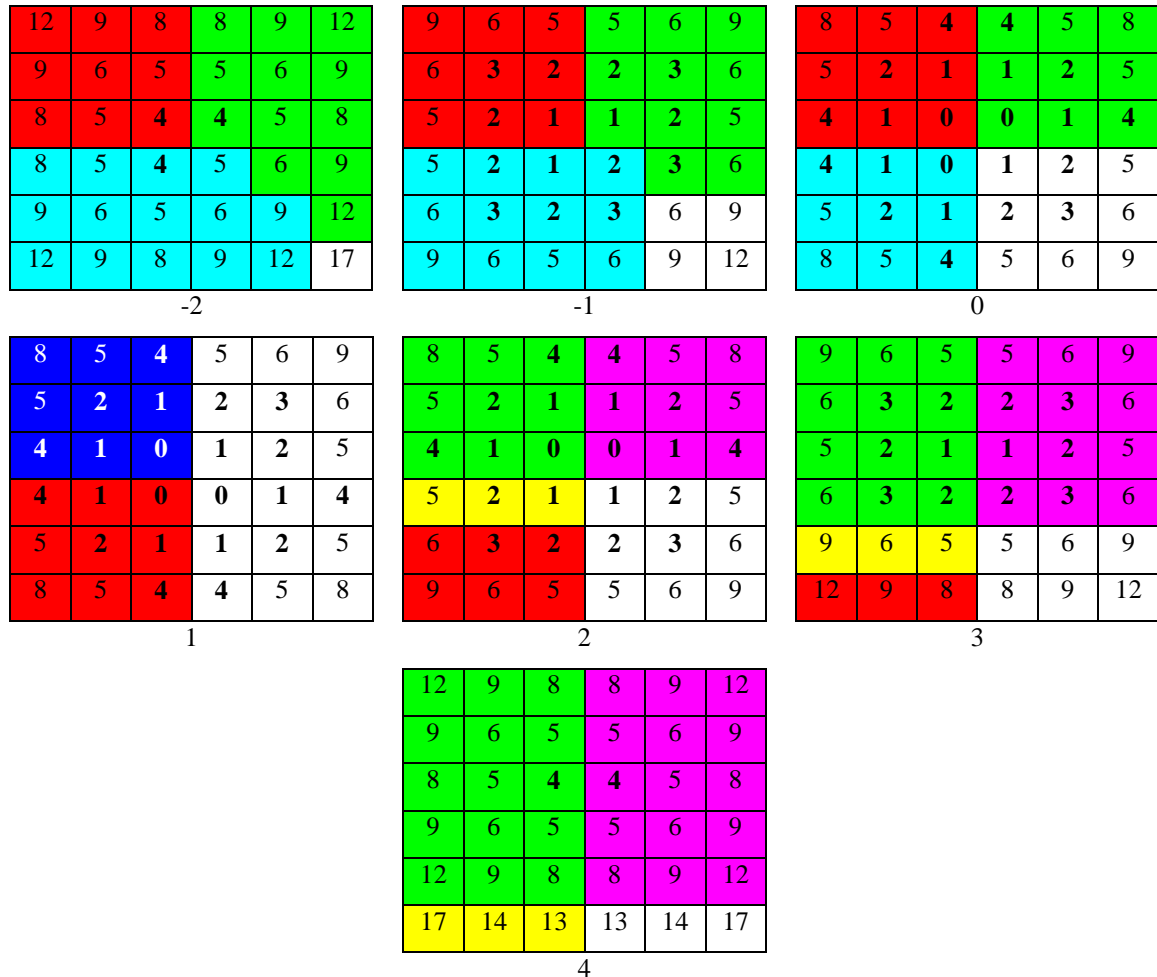


Figura 23– EEDT e segmentação do espaço para o exemplo de imagem da Figura 22. Cortes no eixo z

Tabela 2 - Pontuação para cada combinação de rótulos ao ocorrer empate. As cores amarelo, ciano e magenta são resultado da junção de duas cores, e a cor branca é resultado da junção das três cores.

| Cor | Pontuação de Volume |
|------------|---------------------|
| R | 1 |
| G | 1 |
| B | 1 |
| RG | 1/2 |
| RB | 1/2 |
| GB | 1/2 |
| RGB | 1/3 |

Os descritores fractais podem ser claramente observados na Tabela 3, que mostra a relação, raio ao quadrado versus volume dilatado para cada canal de cor. É importante lembrar que o conjunto de pontos pertencentes a um determinado rótulo é visto como sendo um objeto fractal, sendo assim, deve ter sua dimensão fractal estimada. Porém como já discutido anteriormente, a riqueza dos detalhes nos descritores fractais torna vantajosa sua utilização em lugar de apenas um número representando a dimensão fractal do dado conjunto de pontos.

Tabela 3 - Volumes obtidos para os três canais de cor para cada raio ($r_{max} = 2$). Volumes obtidos utilizando a Figura 22 e as regras da Tabela 2.

| Raio | 1 | 2 | 3 | 4 |
|-------------|----------|----------|----------|----------|
| R | 11,50 | 21,33 | 25,33 | 32,50 |
| G | 12,50 | 24,33 | 30,33 | 38,50 |
| B | 9,00 | 16,33 | 19,33 | 25,00 |

3.2.3 Validação do método

Para verificar a eficácia do método, e a riqueza dos detalhes contidos no vetor de características obtido, deve-se fazer uso de algoritmos de aprendizagem de máquina. Muitos métodos de aprendizagem estão dispostos na literatura, porém, geralmente deve-se utilizar, a priori, o método mais simples. Dessa forma, optou-se pela utilização do método k-NN como processo de aprendizagem supervisionado.

Objetivando maior confiança na análise, ou seja, evitar inserção de possíveis erros desnecessários, optou-se a utilização da implementação do classificador k-NN contida no software Weka.

A matriz de vetores de características para cada uma das imagens de textura da base de dados utilizada deve ser convertida em formato Arff, para ser utilizada como entrada para o software Weka.

Após receber os dados, pode-se realizar os testes. O software Weka realiza a normalização dos dados e a validação cruzada. Portanto, deve-se configurar este software para realizar normalização através da técnica z-score, e ainda, a validação cruzada por *k-fold* utilizando $k = 10$, resultando assim em testes robustos.

Os resultados de saída após treinamento e classificação para cada uma das *10-folds* serão análises estatísticas, como percentual de acerto, matriz de confusão, dentre outros.

4 RESULTADOS

Nessa seção serão mostrados os resultados obtidos tanto da transformada de distância sem rótulos, bem como sua versão rotulada. Ambas analisadas sobre o aspecto de 3 implementações diferentes, que são força bruta, o método proposto por Casanova et al. (2016) e o método proposto neste trabalho.

Ao fim é apresentada a taxa de eficiência do método proposto utilizando o classificador K-NN, em seguida, uma comparação entre o método proposto e de Casanova et al. (2016) utilizando classificador LDA, por fim, é apresentada uma nova abordagem de rotulação, e verificada sua taxa de sucesso.

4.1 COMPARAÇÃO DAS IMPLEMENTAÇÕES

Nesta seção, o processo de realização da EEDT e Segmentação do espaço através de Diagramas de Voronoi foi implementado e comparado para os 3 métodos: força bruta, o método proposto por Casanova et al. (2016) e o método proposto neste trabalho.

A aplicação do algoritmo força bruta é apenas possível para imagens de dimensões em torno de 15 x 15 *pixels* para um notebook simples, pois ele apresenta altas complexidades de tempo e espaço, limitando assim a comparação entre as outras implementações utilizando as imagens de textura que serão usadas para validação do método proposto.

No entanto, é realizar comparações com Casanova et al. (2016), pois apresenta complexidade polinomial de baixa ordem.

4.1.1 EEDT através dos algoritmos força bruta e Saito e Toriwaki (1994)

O algoritmo de transformada de distância, como já comentado anteriormente, apresenta complexidade polinomial elevada ($\Theta(n^3)$ para o caso médio) para realização dos cálculos necessários para a EEDT, no sentido temporal, assim como o consumo de memória. Contudo, este procedimento apresenta garantia de resposta correta. Dessa forma, se fez necessário sua implementação a fim de comparação com o método de Saito e Toriwaki (1994).

A comparação de resultados entre o algoritmo força bruta e o algoritmo de Saito para EDT foi feita utilizando raciocínio indutivo, para os casos bidimensional e tridimensional. O

mesmo conjunto de imagens de entrada foi executado para todos os algoritmos em teste, e suas saídas foram analisadas através de um algoritmo, simplesmente comparando *pixel a pixel* se as saídas estavam iguais. Para o conjunto testado, todas as saídas apresentaram equivalência, induzindo assim que as implementações apresentam resultados corretos no caso genérico, como o esperado. A Figura 24-b apresenta a transformada de distância em níveis de cinza para a forma binária bidimensional apresentada na Figura 24-a.

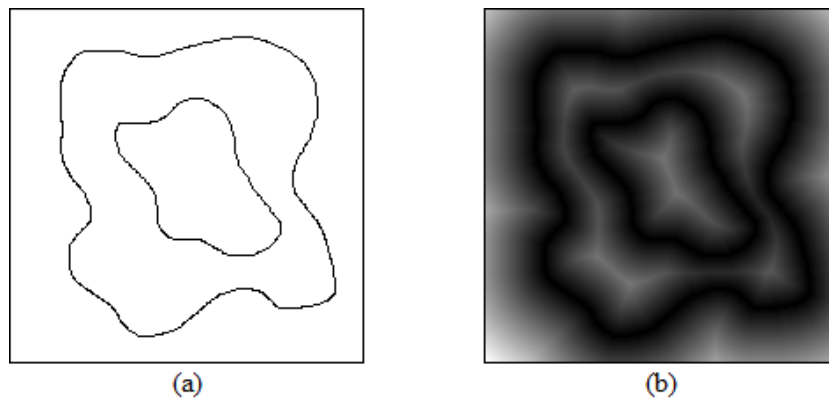


Figura 24 – Transformada de distância bidimensional representada em níveis de cinza (b) para uma forma binária bidimensional (a)

4.1.2 Diagrama de Voronoi 2D através dos algoritmos força bruta e Saito e Toriwaki (1994) sem tratamento de empate

A etapa seguinte necessária para o método proposto neste trabalho é a modificação nos algoritmos de transformada de distância para realização da segmentação do espaço via Diagramas de Voronoi. Esta segmentação do espaço através do algoritmo de EEDT força bruta sempre apresentará a resposta correta, uma vez que o mesmo realiza todos os testes possíveis. Porém a modificação do algoritmo mais eficiente proposto por Saito e Toriwaki (1994) precisa ser testada e comparada com o algoritmo força bruta.

A Figura 25 apresenta a comparação entre as implementações para a segmentação do espaço bidimensional via Diagramas de Voronoi, obtidos à partir dos algoritmos força bruta e Saito e Toriwaki (1994) para EEDT. Nota-se que existem diferenças entre os diagramas de Voronoi apresentados na Figura 25-b e Figura 25-c, estas diferenças estão representadas por pontos vermelhos na Figura 25-d. Contudo, esta diferença era esperada, uma vez que a forma tradicional de segmentar o espaço via Diagramas de Voronoi não prevê o tratamento de regiões

equidistantes entre dois pontos. Ou seja, a região de empate de distância entre dois pontos recebe sua rotulação de maneira pré-determinada em alguns casos, ou aleatória em outros, podendo variar dependendo da rotação da figura, ou seja, tornando o método menos robusto.

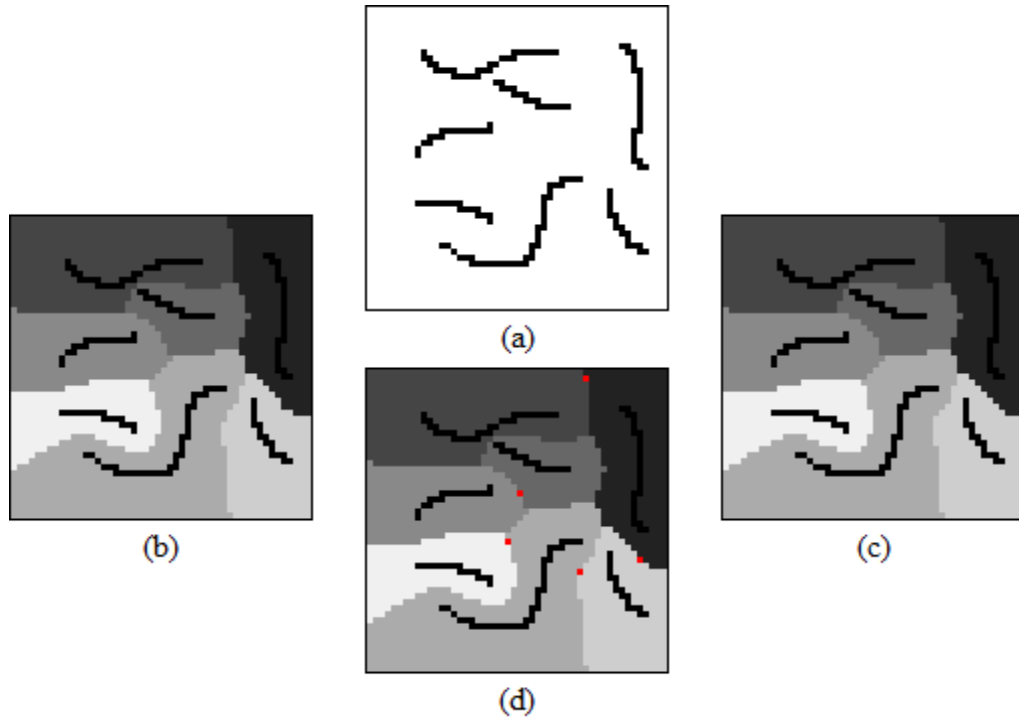


Figura 25 – Comparação (d) entre os Diagramas de Voronoi através do algoritmo força bruta (b) e através do algoritmo de Saito e Toriwaki (1994) (c), para as formas (a) representadas em preto e rótulos representados em níveis de cinza

4.1.3 Diagrama de Voronoi 3D Modificado através dos algoritmos de força bruta e Saito e Toriwaki (1994) com tratamento de empate

Assim como discutido anteriormente, o erro gerado ao ignorar o tratamento de empates em regiões equidistantes durante a realização da segmentação do espaço via Diagramas de Voronoi tem consequências na robustez do método, pois pode não apresentar o mesmo resultado dependendo da forma de implementação e rotação da imagem de textura.

A Figura 26 mostra o não determinismo ao ignorar o tratamento de empates ao gerar Diagramas de Voronoi. Neste caso, foram mostradas imagens dando-se prioridades para determinadas cores, observa-se que podem haver diversos resultados diferentes, dependendo do eixo analisado por primeiro no processo de EEDT, ou ainda a rotação da imagem, sendo estes, problemas indesejáveis em métodos de análise de textura. Objetivando contornar este

problema, foi implementado neste trabalho uma metodologia de tratamento de empate, que foi apresentada na Seção anterior.

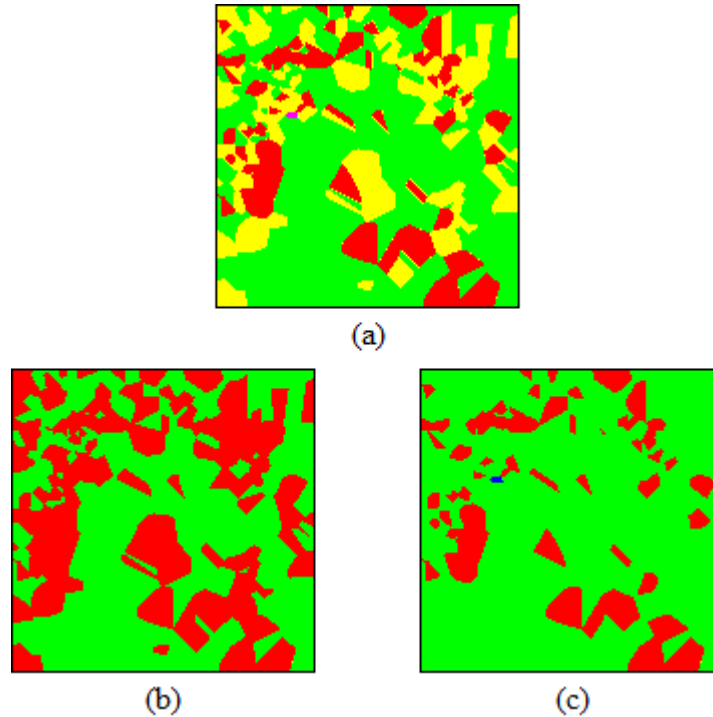


Figura 26 – Comparação os possíveis resultados Diagramas de Voronoi com tratamento de empate (a), sem tratamento de empate, com prioridade para o vermelho (b) e com prioridade para o verde (c)

A questão de tratamentos de empate é algo natural no algoritmo força bruta, e também já está previsto na metodologia proposta por Casanova et al. (2016). Para fins de comparação, foram implementados neste trabalho estes três algoritmos (incluindo o método proposto neste trabalho) para realização da EEDT, com objetivo da segmentação do espaço via Diagramas de Voronoi com tratamento de empate.

Como já discutido anteriormente, o método aqui proposto suporta até 32 rótulos diferentes, se utilizado uma matriz de inteiros como região de trabalho. Porém optou-se utilizar apenas 3 rótulos, que correspondem diretamente ao canal de cor de cada *pixel* analisado, para tornar possível realizar a comparação com o trabalho de Casanova et al. (2016). A razão disto é que a inserção de novos rótulos na metodologia utilizada por Casanova et al. (2016) torna necessária uma modificação drástica em seu algoritmo, ao contrário do método aqui proposto, onde a inserção de novos rótulos, juntamente com o tratamento de empate destes, acontece de maneira natural.

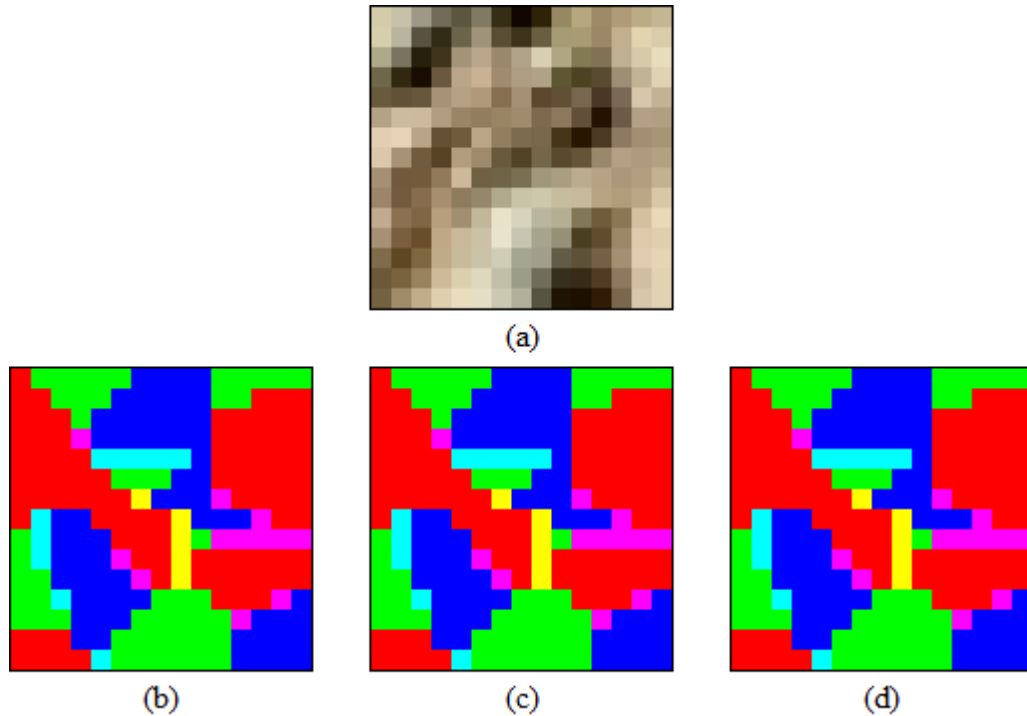


Figura 27 – Comparação entre os Diagramas de Voronoi tridimensionais através do algoritmo para o método força bruta (b), para o método de Casanova et al. (2016) (c) e para o método aqui proposto (d), ambos para imagem de entrada (a). Corte no Eixo Z

A Figura 27 apresenta a comparação entre os Diagramas de Voronoi através dos três métodos implementados. Como pode ser observado nos cortes no Eixo Z, os três algoritmos geram o mesmo resultado. Para comprovar foram feitos testes para diferentes imagens, e realizada a comparação automatizada para cada *Voxel* no cubo resultante para os Diagramas de Voronoi.

A fim de melhor visualização do processo de segmentação do espaço via Diagramas de Voronoi serão apresentadas a seguir imagens (Figura 28) de etapas deste procedimento (i.e. imagens de diferentes raios de dilatação), onde foi selecionado $r_{max} = 4$ para dilatação. Nestas figuras, da mesma forma como apresentado anteriormente, cada canal de cor é utilizado como sendo um rótulo, e apenas para distinção de regiões de empate, estas são representadas com cores diferentes, onde estas são a combinação das cores dos rótulos empatados. As imagens foram quantizadas de 255 cores para apenas 32 de uma imagem 15 x 15.

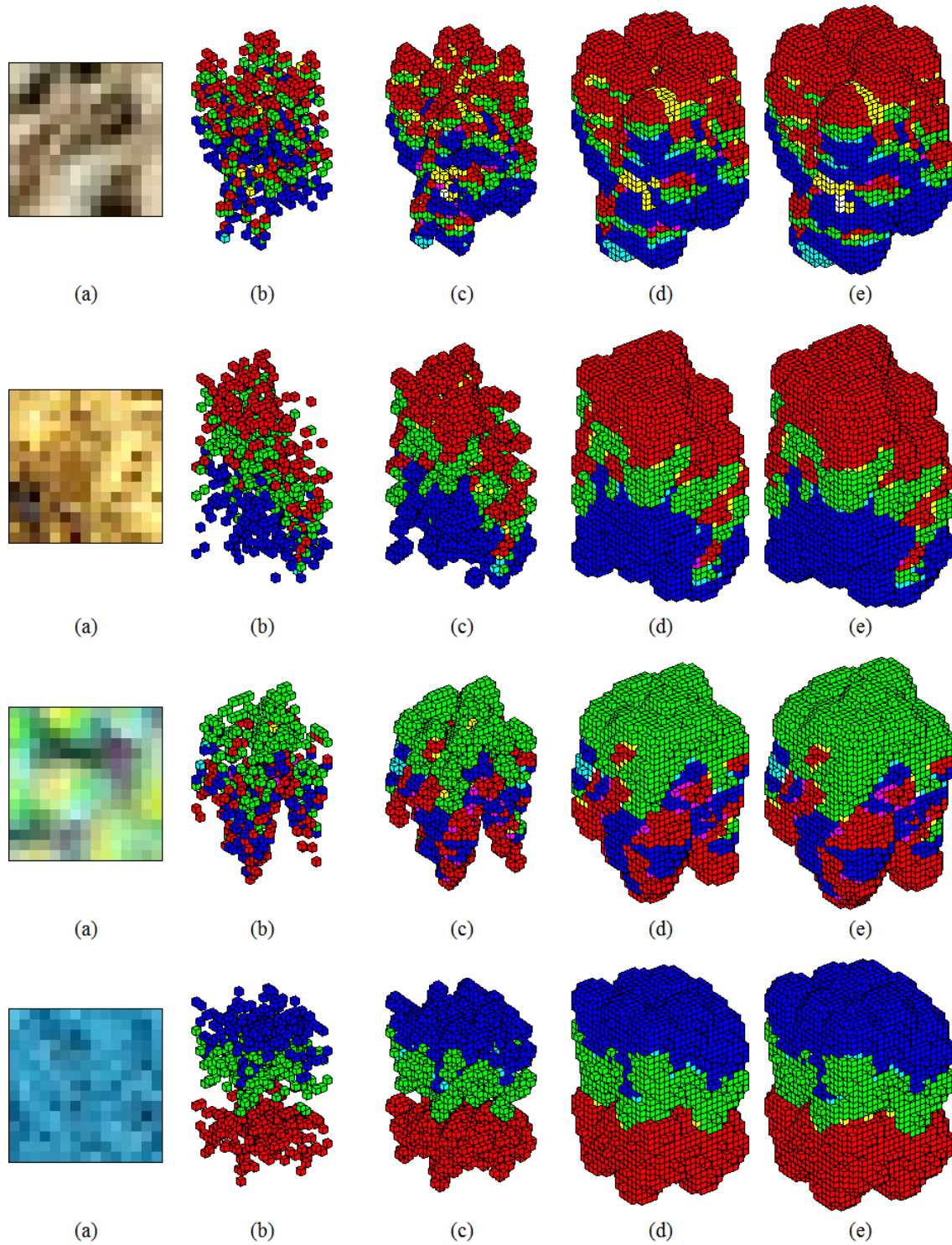


Figura 28 – Exemplos da relação dos canais de cores no processo de dilatação. (a) Textura original; (b) $r = 0$; (c) $r = 1$; (d) $r = 3$; (e) $r = 4$

A Figura 29 ilustra a separação dos rótulos ao fim do processo de dilatação. Observa-se a influência gerada por ambos os objetos fractais (considerando que a nuvem de pontos para cada rótulo é um objeto fractal) estarem no mesmo ambiente. As regiões de empate são mostradas juntamente com os canais os quais possuem uma parcela de volume dilatado.

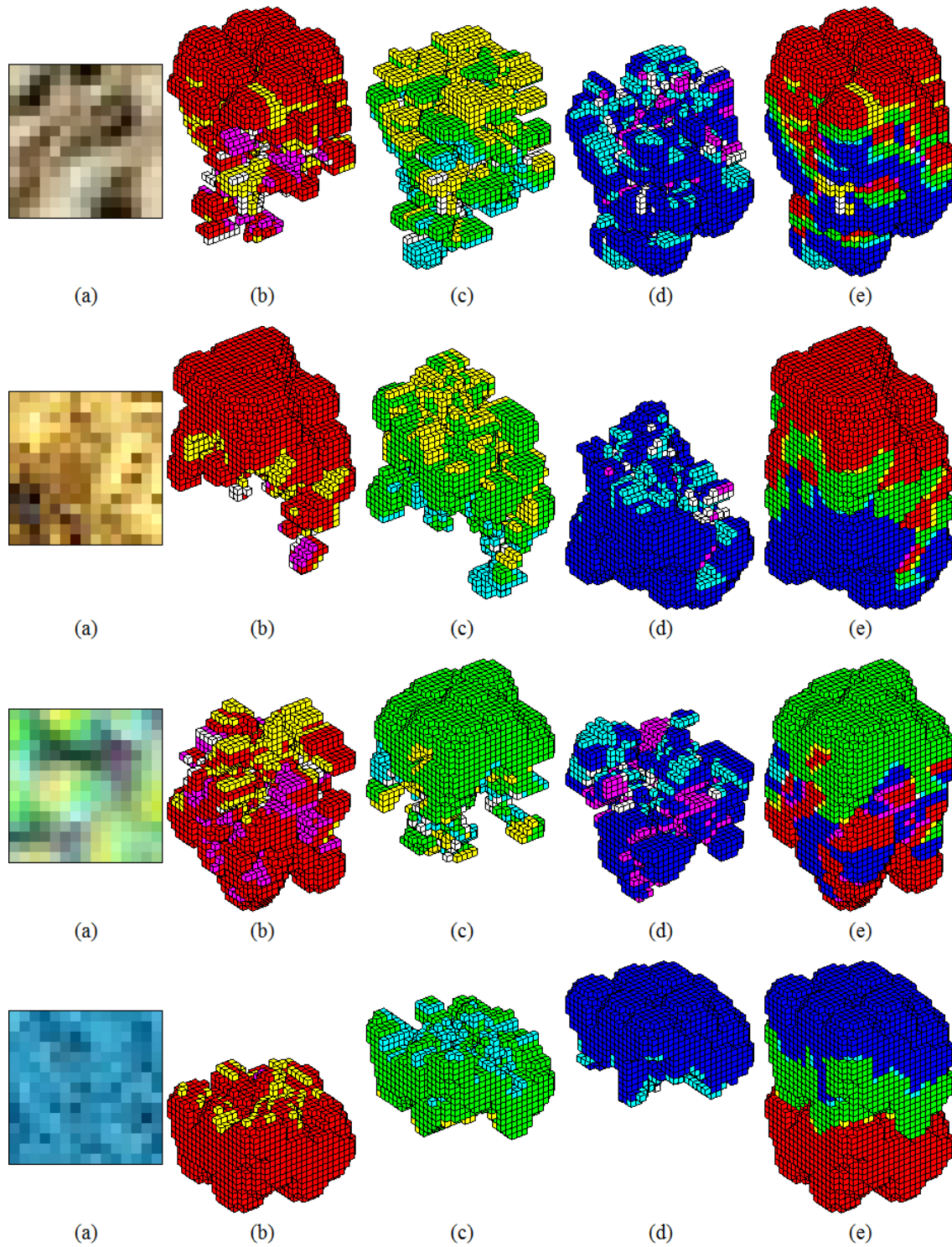


Figura 29 – Exemplos da relação dos canais de cores no processo de dilatação. (a) Textura original; (b) Canal vermelho; (c) Canal verde; (d) Canal Azul; (e) Os três canais

4.2 DESCRITORES FRACTAIS DE BOULIGAND-MINKOWSKI

Uma vez efetuada a transformada de distância e os Diagramas de Voronoi, se torna possível procedimento para obtenção dos descritores fractais (Figura 30), que como descrito anteriormente, consiste em realizar a contagem do volume proveniente de cada raio de dilatação, para cada rótulo existente.

A Figura 30 foi gerada baseada na seguinte ideia: cada rótulo representa um canal de cor, vermelho, verde e azul. Sendo assim, para cada raio r entre $1 \leq r \leq 20$ são geradas três contagens diferentes, cada uma destas contagens representa uma coordenada no espaço tridimensional, resultando em um ponto. Para cada raio r é gerado então um ponto, de forma que cada conjunto de pontos de mesma cor expresso abaixo representa uma imagem. Estas pertencem à base de conjunto de testes com ID Outex_TC_00013, onde foram utilizadas quatro classes de textura diferentes com duas imagens de cada classe, diferenciadas por tons da mesma cor.

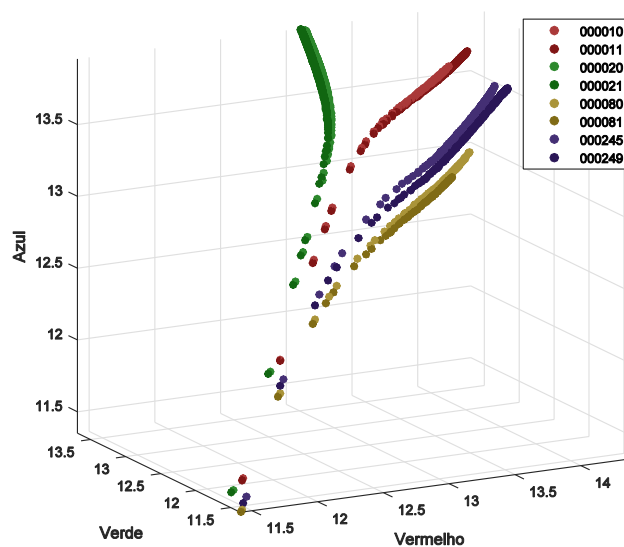


Figura 30 – Curva log-log para diferentes classes de textura presentes na base de conjunto de testes com ID Outex_TC_00013

Cada uma das curvas presentes no gráfico ilustrado na Figura 32 representa o vetor de características para cada uma das imagens pertencem à base de conjunto de testes com ID Outex_TC_00013, totalizando assim 1360 curvas. Cada uma das classes dessa base de imagens é apresentada por uma cor, totalizando 68 cores distintas (algumas muito próximas as outras, podendo gerar engano na visualização).

Nota-se que os descritores fractais para imagens de mesma classe apresentam alta correlação, pois estão muito próximas umas das outras. Além disso, como observado na Figura 31, classes de texturas diferentes tendem a distanciar-se entre si. A conclusão empírica ao observar visualmente esta imagem é que as imagens pertencentes a determinadas classes de textura podem eficientemente categorizadas.

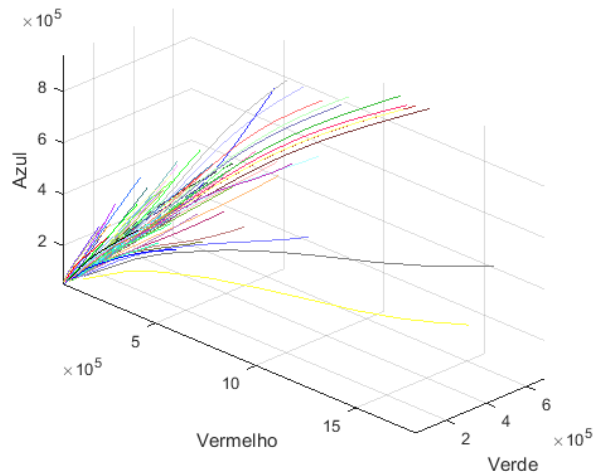


Figura 31 – Curva log-log para uma imagem de cada classe de textura presentes na base de conjunto de testes com ID Outex_TC_00013

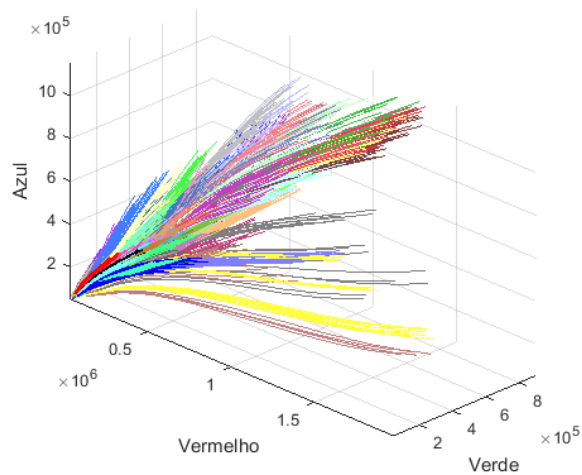


Figura 32 – Curva log-log para todas as imagens pertencentes a todas as classes de textura presentes na base de conjunto de testes com ID Outex_TC_00013

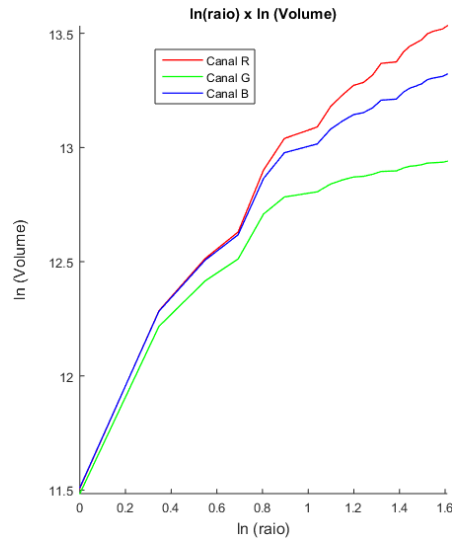


Figura 33 – Curva ln raio x ln volume para cada canal de cor para uma imagem de textura.

Para exemplificar a obtenção do vetor de características, considere cada canal de cor como sendo um rótulo e o raio de dilatação máximo $r_{max} = 5$ (não é necessário utilizar um número inteiro para r_{max} , desde que este esteja contido numa malha ortogonal). Após computar todas as possíveis distâncias em uma malha ortogonal e, eliminar previamente o raio zero para evitar possíveis complicações, obtém-se 22 características para cada um dos três rótulos utilizados. Portanto, ao fim, a junção de todas as características resulta em um vetor de características com 66 atributos. A Figura 33 ilustra os descritores fractais de cada um dos canais de cores, cada um com 22 pontos (relação raio vs volume dilatado).

A Figura 34 ilustra um mapa de correlação para cada um dos atributos utilizados (sem normalização). Observa-se forte correlação, principalmente entre os atributos pertencentes ao mesmo rótulo.

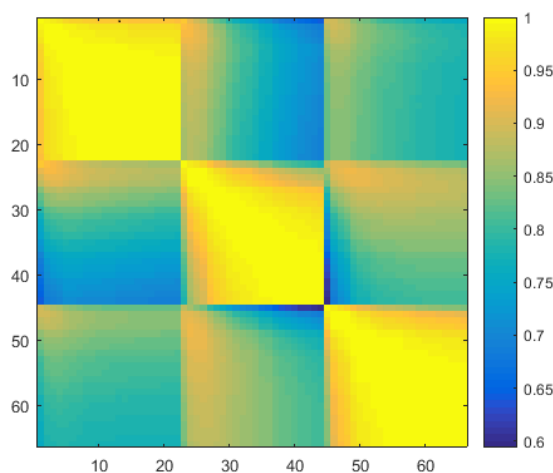


Figura 34 – Mapa de correlação entre os atributos utilizados para caracterização de texturas

4.3 RESULTADOS ESTATÍSTICOS

Tendo em mãos a matriz contendo todos os atributos para todas as classes de textura da base de dados utilizada, pode-se realizar os testes estatísticos sobre estes dados, a fim de verificar a eficácia do método.

Neste contexto, após a utilização do software Weka para validação dos dados obteve-se os resultados expressos na Tabela 4. Além disso, obteve-se a matriz de confusão (Figura 35) para todas as classes de textura presentes na base de conjunto de testes com ID Outex_TC_00013.

Devido ao mal da dimensionalidade Bellman (1957), ao aumentar o número de atributos aumenta-se o “volume” do espaço de decisão, dessa forma, se não houver um aumento no número de instancias, haverá um baixo preenchimento do espaço de decisão, reduzindo o desempenho de algoritmos de aprendizagem de máquina. Dessa forma, fez-se uma avaliação da taxa de sucesso em função ao número de atributos utilizados, aumentando R_{max} , dessa forma aumentando o número de atributos, e comparando com a taxa de sucesso, para o classificador LDA, e obteve-se 72 atributos. Porém utilizou-se também um número de 66 atributos para fins de comparação com Casanova et al. (2016).

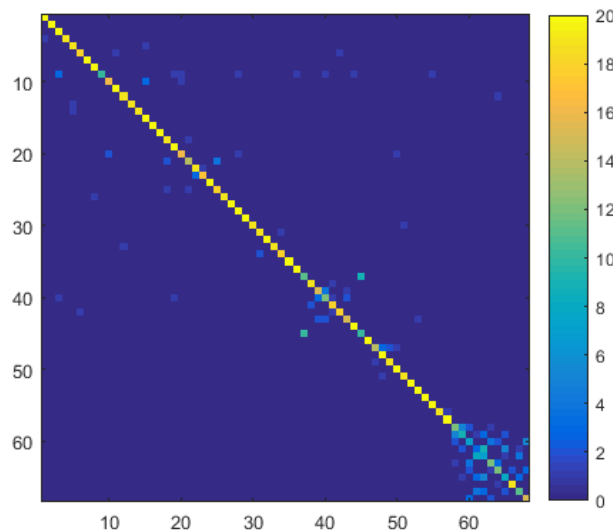


Figura 35 – Matriz de confusão para todas as classes de textura presentes na base de conjunto de testes com ID Outex_TC_00013 utilizando o vetor de características obtidos através do método proposto. Classificador 1-NN

A matriz de confusão expressa na Figura 35 apresenta baixa eficiência na região das últimas classes analisadas, presume-se que isto ocorra pois as classes de textura contidas nesta

região são visualmente muito semelhantes, como pode ser observado na Figura 17. Dessa forma, figuras que teoricamente são distintas, apresentam muita semelhança em relação aos seus vetores de características pelo método proposto.

Tabela 4 – Resultado obtido

| Base de dados | Classificador | k-fold | Taxa de sucesso | Número de atributos |
|---------------|---------------|--------|-----------------|---------------------|
| Outex | 1-NN | 10 | 85,2941 % | 66 |

Observa-se que ao utilizar o classificador k-NN para classificação dos dados obtidos, este não consegue generalizar de maneira tão satisfatória quanto poderia, porque, como já discutido anteriormente, os atributos utilizados apresentam alta correlação. Portanto, torna-se possível melhorar o percentual de acerto utilizando um classificador que considere esta alta correlação entre os dados. Além disso, utilizando o classificador LDA, torna possível a comparação dos resultados obtidos com o trabalho de Casanova et al. (2016).

A Tabela 5 mostra a comparação dos resultados obtidos com o trabalho de Casanova et al. (2016), utilizando classificador LDA, validação cruzada por 2-fold e a base de conjunto de testes com ID Outex_TC_00013. A Figura 36 ilustra a matriz de confusão para todas as classes de textura presentes na base de conjunto de testes com ID Outex_TC_00013 utilizando o vetor de características obtidos através do método proposto e classificador LDA.

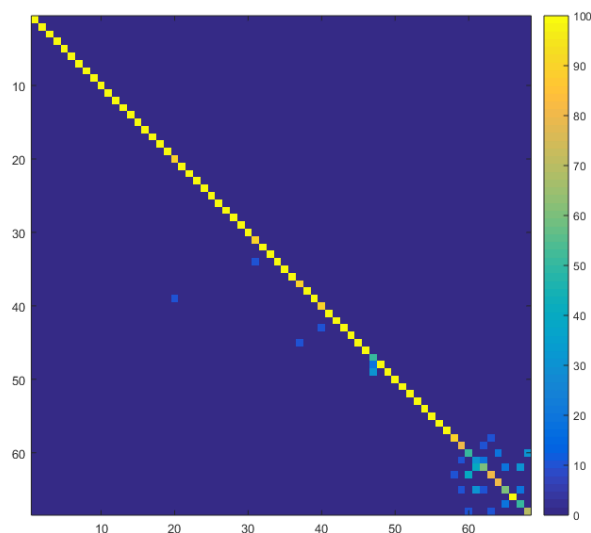


Figura 36 – Matriz de confusão para todas as classes de textura presentes na base de conjunto de testes com ID Outex_TC_00013 utilizando o vetor de características obtidos através do método proposto. Classificador LDA

Tabela 5 – Comparação entre os resultados obtidos neste trabalho e os resultados obtidos de Casanova et al. (2016), ambos utilizando classificador LDA. Taxa de acerto e número de atributos utilizados.

| Metodologia | Base | Classificador | k-fold | Taxa de sucesso | Número de atributos |
|-------------------------------|-------|---------------|--------|-----------------|---------------------|
| Método proposto | Outex | LDA | 2 | 92,94 % | 72 |
| Método proposto | Outex | LDA | 2 | 92,50 % | 66 |
| Casanova et al. (2016) | Outex | LDA | 2 | 92,79 % | 66 |

O número de atributos é escolhido de forma que maximize a taxa de sucesso. Nota-se que além do número de atributos, houve uma pequena diferença nos resultados de taxa de sucesso apresentados na Tabela 5. Esta variação pode ser consequência das diferenças presentes nas metodologias empregadas por ambos os trabalhos. Porém esta diferença não é o suficiente para invalidar a equivalência entre as duas metodologias.

4.4 PROPOSTA DE METODOLOGIA DE ROTULAÇÃO

O objetivo principal deste trabalho é melhorar a análise de textura através de descritores fractais por meio da utilização de rótulos na transformada de distância 3D. Uma das formas de fazer isso, e já alcançada, é simplesmente a otimização no quesito complexidade temporal, de forma que o desempenho dependa apenas da EEDT sobre a imagem de entrada, livre da metodologia de rotulação empregada. Outra forma, pode ser vista como sendo uma melhora diretamente sobre a taxa de sucesso em relação a outras metodologias na área como Casanova et al. (2016).

A alteração na forma como é realizada a rotulação do espaço pode trazer melhorias significativas na taxa de acerto. Porém podem existir inúmeras metodologias para isso, mas todas elas devem ser projetadas visando extração de características únicas para cada classe de textura analisada. Nesse contexto, com a intenção de testar o potencial do método apresentado neste trabalho, propõe-se uma alternativa a metodologia de Casanova et al. (2016) para rotulação do espaço.

As informações de cores são muito importantes no processo de análise de textura. Portanto, pretende-se manter estas informações sobre rotulação. Porém, a metodologia proposta neste trabalho possibilita que um *voxel* tenha mais de um rótulo simultaneamente, permitindo assim adição de informações extras, podendo beneficiar a análise.

Nota-se que a análise da dimensão fractal apenas através dos canais de cores não traz informações no quesito intensidade luminosa. A exemplo disso, dada uma imagem de textura com histograma de cores aglomerado entre uma faixa restrita, se todos os pixels dessa imagem tiverem sua intensidade subtraída ou incrementada sem ocorrer saturação, então o resultado da análise será o mesmo. Isso pode ser útil para imagens amostradas utilizando iluminação não controlada, porém quando ocorre o oposto, torna-se vantajosa as informações de intensidade luminosa.

Nesse contexto, propõe-se a utilização de dois rótulos como informações de intensidade luminosa. Pixels com intensidade luminosa acima de um limiar recebem a rotulação “cima”, caso contrário recebe o rótulo “baixo”. Sendo assim, tem-se 5 rótulos, canal R, canal G, canal B, cima e baixo, que são representados pelos bits de 0 à 4 respectivamente.

O tratamento de empates é realizado apenas entre os canais de cores e, separadamente, entre os rótulos de intensidade luminosa. Pois as informações intensidade luminosa não devem afetar a relação entre os canais de cores.

4.4.1 Resultados para nova proposta de metodologia de rotulação

Para avaliar os resultados para nova proposta de metodologia de rotulação é utilizado apenas o classificador LDA, pois como já observado anteriormente, os dados provenientes dos descritores fractais apresentam alta correlação.

Definiu-se limiar como sendo a metade da escala de quantização das imagens de entrada, no caso, as imagens tem 256 níveis de intensidade luminosa, então define-se o limiar como sendo 127. Outra metodologia que poderia ser aplicada é encontrando um limiar que divide a metade dos *pixels* em relação ao eixo Z, não tornando o método variante a escala de cinza, porém sensível a intensidade luminosa dos *pixels*.

A alta correlação dos dados pode ser observada na Figura 37, onde apresenta a comparação entre as curvas de descrição fractal para cada um dos rótulos utilizado. As curvas são traçadas obedecendo aos critérios da obtenção da dimensão fractal de Minkowski, $\log \text{raio} \times \log \text{volume}$.

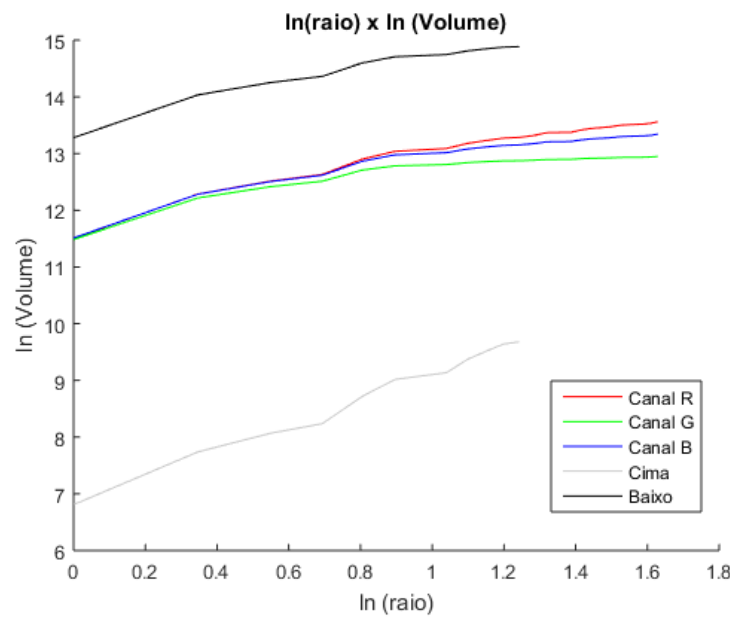


Figura 37 – Curva ln raio x ln volume para cada rótulo para uma imagem de textura.

Como pode ser observado na Figura 37, os comprimentos das curvas dos rótulos dos canais de cores apresentam maior comprimento que as curvas que trazem a informação de limiar. Isso se deve ao fato de se escolher diferentes números de descritores entre as duas categorias de rótulos, visando a maximização da taxa de sucesso. A busca pela quantidade de atributos para cada categoria é feita através de força bruta.

A Tabela 6 apresenta os resultados obtidos utilizando esta metodologia de rotulação, utilizando classificador LDA, validação cruzada por 2-fold e a base de conjunto de testes com ID Outex_TC_00013. E a Figura 38 ilustra a matriz de confusão para essa nova metodologia

Tabela 6 – Resultados obtidos neste trabalho utilizando uma metodologia de rotulação RGB e de intensidade luminosa.

| Base | Classificador | k-fold | Taxa de sucesso | Atributos por cor | Atributos por limiar | Total atributos |
|-------|---------------|--------|-----------------|-------------------|----------------------|-----------------|
| Outex | LDA | 2 | 93,53 % | 23 | 11 | 91 |

Tabela 7 – Comparação entre os resultados obtidos neste trabalho e os resultados obtidos de Casanova et al. (2016), ambos utilizando classificador LDA. Taxa de acerto e número de atributos utilizados.

| Metodologia | Base | Classificador | k-fold | Taxa de sucesso | Número de atributos |
|------------------------|-------|---------------|--------|-----------------|---------------------|
| Proposto RGB e Limiar | Outex | LDA | 2 | 93,53 % | 91 |
| Proposto RGB | Outex | LDA | 2 | 92,94 % | 72 |
| Proposto RGB | Outex | LDA | 2 | 92,50 % | 66 |
| Casanova et al. (2016) | Outex | LDA | 2 | 92,79 % | 66 |

Observa-se (Tabela 7) que a taxa de sucesso teve um aumento de 0,59 % com a adição de 22 atributos, os quais continham informações sobre a intensidade luminosa da figura. Aparentemente é um pequeno ganho, porém qualquer ganho quando a taxa de sucesso já é alta, se torna significativo. Além disso, este é apenas um pequeno exemplo de como a metodologia de rotulação pode influenciar na taxa de sucesso.

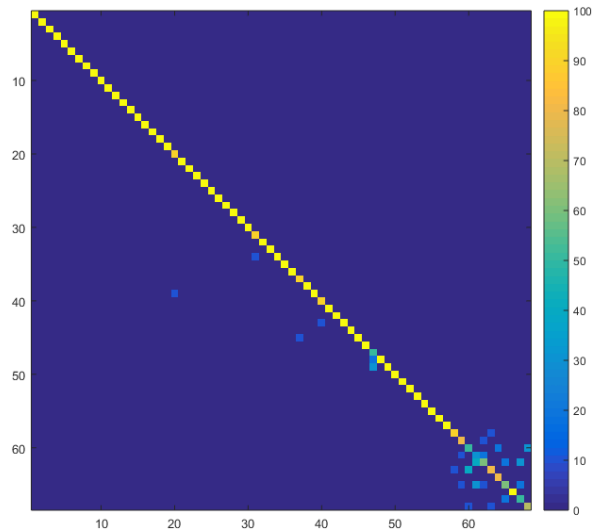


Figura 38 – Matriz de confusão para todas as classes de textura presentes na base de conjunto de testes com ID Outex_TC_00013 utilizando o vetor de características obtidos através do novo método proposto. Classificador LDA

5 CONCLUSÃO

Este trabalho propôs uma nova metodologia para análise de textura com base na dimensão fractal. Esta técnica utiliza descritores fractais como fonte de informações de ocupação espacial em um Diagrama de Voronoi para grupos de *pixels*. Consequentemente, este método é capaz analisar a complexidade destes grupos, bem como a interferência em função da região analisada entre grupos de *pixels* com rotulação diferente.

Os resultados obtidos através deste trabalho mostraram-se satisfatórios, uma vez que além de agilizar o processo de obtenção de descritores fractais, quando utilizado múltiplos rótulos, foi possível reproduzir e melhorar os resultados obtidos em Casanova et al. (2016).

O fato da metodologia aqui proposta utilizar a transformada de distância euclidiana exata com suporte a rotulação, tornou-a muito ágil, pois seu processamento não está mais ligado ao número de rótulos utilizados na transformação. Ou seja, a complexidade temporal do método aqui proposto está indexado de forma relevante apenas à EEDT.

A grande flexibilidade de rotulação obtida por esta metodologia fica limitada apenas aos requisitos de memória, pois cada rótulo utilizado está associado a um bit na matriz de rotulação, ou seja, para um número muito grande de rótulos, seria necessário combinar matrizes de rotulação. A metodologia proposta abre as portas para trabalhos futuros na mesma área, visando alternativas para definição de rótulos, com objetivo de descrever de forma única grupos de textura, distinguindo assim grupos aparentemente semelhantes.

O ponto chave aqui o melhoramento da análise de textura no sentido de extração de características que podem descrever texturas de alta complexidade, uma vez que este método é baseado na dimensão fractal, que é capaz de descrever objetos caóticos, como fractais.

REFERÊNCIAS

- BACKES, A. R. Implementação e comparação de métodos de estimativa da dimensão fractal e sua aplicação à análise e processamento de imagens. , 2006.
- BACKES, A. R.; CASANOVA, D.; BRUNO, O. M. Plant Leaf Identification Based on Volumetric Fractal Dimension. **International Journal of Pattern Recognition and Artificial Intelligence**, v. 23, n. 06, p. 1145–1160, 2009.
- BACKES, A. R.; CASANOVA, D.; BRUNO, O. M. Color texture analysis based on fractal descriptors. **Pattern Recognition**, v. 45, n. 5, p. 1984–1992, 2012.
- BELLMAN, R. Dynamic programming. , 1957.
- CAI, D. C. D.; HE, X. H. X.; HAN, J. H. J. Training Linear Discriminant Analysis in Linear Time. **2008 IEEE 24th International Conference on Data Engineering**, v. 00, p. 209–217, 2008.
- CARLIN, M. Measuring the complexity of non-fractal shapes by a fractal method. **Pattern Recognition Letters**, v. 21, n. 11, p. 1013–1017, 2000. Disponível em: <<http://linkinghub.elsevier.com/retrieve/pii/S0167865500000611>>. .
- CASANOVA, D.; FLORINDO, J. B.; FALVO, M.; BRUNO, O. M. Texture analysis using fractal descriptors estimated by the mutual interference of color channels. **Information Sciences**, v. 347, p. 58–72, 2016. Disponível em: <<http://linkinghub.elsevier.com/retrieve/pii/S0020025516300135>>. .
- CAVALCANTI, P. G.; SCHARCANSKI, J. A coarse-to-fine approach for segmenting melanocytic skin lesions in standard camera images. **Computer Methods and Programs in Biomedicine**, v. 112, n. 3, p. 684–693, 2013. Elsevier Ireland Ltd. Disponível em: <<http://dx.doi.org/10.1016/j.cmpb.2013.08.010>>. .
- COSTA, L.; CESAR JR., R. **Shape classification and analysis: theory and practice**. CRC Press, 2009.
- COSTA, L. D. F. Robust Skeletonization through Exact Euclidean Distance Transform and its Application to Neuromorphometry. **Real-Time Imaging**, v. 6, n. 6, p. 415–431, 2000. Disponível em: <<http://linkinghub.elsevier.com/retrieve/pii/S1077201499901778>>. .
- CUISENAIRE, O.; MACQ, B. Fast Euclidean Distance Transformation by Propagation Using Multiple Neighborhoods. **Computer Vision and Image Understanding**, v. 76, n. 2, p. 163–172, 1999.
- DEVORE, J. L. **Probability and Statistics for Engineering and the Sciences**. 1982.

FABBRI, R.; COSTA, L. D. F.; TORELLI, J. C.; BRUNO, O. M. 2D Euclidean distance transform algorithms. **ACM Computing Surveys**, v. 40, n. 1, p. 1–44, 2008.

FALCONER, K. **Fractal Geometry: Mathematical Foundations and Applications**. 2003.

FISHER, R. A. The use of multiple measurements in taxonomic problems. **Annals of eugenics**, v. 7, n. 2, p. 179–188, 1936. Wiley Online Library.

FLORINDO, J. B. **Descritores fractais aplicados à análise de texturas**, 2013. São Carlos: Universidade de São Paulo.

FLORINDO, J. B.; BACKES, A. R.; DE CASTRO, M.; BRUNO, O. M. A comparative study on multiscale fractal dimension descriptors. **Pattern Recognition Letters**, v. 33, n. 6, p. 798–806, 2012. Elsevier B.V. Disponível em: <<http://dx.doi.org/10.1016/j.patrec.2011.12.016>>. .

FLORINDO, J. B.; BRUNO, O. M. Fractal descriptors based on Fourier spectrum applied to texture analysis. **Physica A: Statistical Mechanics and its Applications**, v. 391, n. 20, p. 4909–4922, 2012. Elsevier B.V. Disponível em: <<http://dx.doi.org/10.1016/j.physa.2012.03.039>>. .

GONZALEZ, R. C.; WOODS, R. E. **Digital Image Processing**. 2009.

GUSTAVSON, S.; STRAND, R. Anti-aliased Euclidean distance transform. **Pattern Recognition Letters**, v. 32, n. 2, p. 252–257, 2011. Elsevier B.V. Disponível em: <<http://dx.doi.org/10.1016/j.patrec.2010.08.010>>. .

HARALICK, R.; SHANMUGAN, K.; DINSTEN, I. Textural features for image classification. **IEEE Transactions on Systems, Man and Cybernetics**, 1973. Disponível em: <[http://dceanalysis.bigr.nl/Haralick73-Textural features for image classification.pdf](http://dceanalysis.bigr.nl/Haralick73-Textural%20features%20for%20image%20classification.pdf)>. .

IVANOVICI, M.; RICHARD, N. Fractal dimension of color fractal images. **IEEE Transactions on Image Processing**, v. 20, n. 1, p. 227–235, 2011.

KHADEMI, A.; KRISHNAN, S. Medical image texture analysis: A case study with small bowel, retinal and mammogram images. **Electrical and Computer Engineering, 2008. CCECE 2008. Canadian Conference on**, p. 001949–001954, 2008. Disponível em: <http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4564884>. .

KIM, J. K.; PARK, H. W. Statistical textural features for detection of microcalcifications in digitized mammograms. **IEEE Transactions on Medical Imaging**, v. 18, n. 3, p. 231–238, 1999.

LOTUFO, R.; ZAMPIROLI, F. Fast Multidimensional Parallel Euclidean Distance Transform based on Mathematical Morphology. , p. 100–105, 2001.

MÄENPÄÄ, T.; PIETIKÄINEN, M. Classification with color and texture: Jointly or separately? **Pattern Recognition**, v. 37, n. 8, p. 1629–1640, 2004. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0031320303004321>>. .

MANDELBROT, B. B. **The Fractal Geometry of Nature**. ilustrada, ed. New York: Henry Holt and Company, 1983.

MANJUNATH, B. S. Texture features for browsing and retrieval of image data. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 18, n. 8, p. 837–842, 1996.

MATHWORKS, I. fitcdiscr. Disponível em: <<http://www.mathworks.com/help/stats/fitcdiscr.html>>. Acesso em: 5/6/2016.

MCLACHLAN, G.; PEEL, D. **Multivariate Normal Mixtures**. John Wiley & Sons, Inc., 2005.

MITCHELL, T. M. **Machine Learning**. 1997.

OJALA, T.; MAENPAA, T. Outex-new framework for empirical evaluation of texture analysis algorithms. **6th International Conference on Pattern Recognition, Quebec, Canada**, p. 701 – 706, 2002. Disponível em: <http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1044854>. .

PAGLIERONI, D. W. Distance transforms: Properties and machine vision applications. **CVGIP: Graphical Models and Image Processing**, v. 54, n. 1, p. 56–74, 1992. Disponível em: <<http://www.sciencedirect.com/science/article/pii/104996529290034U>>. .

PATIENTS, A; FREEBOROUGH, P. A; FOX, N. C. MR Image Texture Analysis Applied to the Diagnosis and Tracking of Alzheimer’s Disease. **IEEE Transactions on Medical Imaging**, v. 17, n. 3, p. 8–11, 1998.

PELEG, S.; NAOR, J.; HARTLEY, R.; AVNIR, D. Multiple resolution texture analysis and classification. **IEEE transactions on pattern analysis and machine intelligence**, v. 6, n. 4, p. 518–523, 1984.

PENTLAND, A. P. Fractal-Based Description of Natural Scenes. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. PAMI-6, n. 6, p. 661–674, 1984.

PIETIKÄINEN, M.; HADID, A.; ZHAO, G.; AHONEN, T. Local Binary Patterns for Still Images. **Computer Vision Using Local Binary Patterns**. v. 40, p.13–43, 2011. Disponível em: <http://link.springer.com/10.1007/978-0-85729-748-8\nhttp://books.google.com/books?hl=en&lr=&id=wBrZz9FiERsC&oi=fnd&pg=PR3&dq=Computer+Vision+Using+Local+Binary+Patterns&ots=XncQtF1fds&sig=veOPKjBE02EuobCln_qT5F4UAI>. .

RODRÍGUEZ, J. D.; PEREZ, A.; LOZANO, J. A.; PE, A.; LOZANO, J. A. Sensitivity analysis of k-fold cross validation in prediction error estimation. **Pattern Analysis and**

Machine Intelligence, IEEE Transactions on, v. 32, n. 3, p. 569–575, 2010. IEEE.

RUSS, J. C. **Fractal Surfaces**. 2013.

RUSSELL, S.; NORVIG, P. **Artificial Intelligence: A Modern Approach**. 2003.

RUYI, J.; REINHARD, K.; TOBI, V.; SHIGANG, W. Lane detection and tracking using a new lane model and distance transform. **Machine Vision and Applications**, v. 22, n. 4, p. 721–737, 2011.

SAITO, T.; TORIWAKI, J. New algorithms for Euclidean distance transformation of an n-dimensional digitised picture with applications. , v. 27, n. 11, 1994.

SERRA, J. **Image Analysis and Mathematical Morphology**. 1982.

SHE, A. C.; HUANG, T. S. Segmentation of road scenes using color and fractal-based texture classification. **Image Processing, 1994. Proceedings. ICIP-94., IEEE International Conference**, v. 3, p. 1026–1030 vol.3, 1994.

SKLANSKY, J. Image segmentation and feature extraction. **Systems, Man and Cybernetics, IEEE Transactions ...**, v. 75, n. 4, p. 237–247, 1978. Disponível em: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4309944. .

TAMURA, H.; MORI, S.; YAMAWAKI, T. Textural Features Corresponding to Visual Perception. **IEEE Transactions on Systems, Man, and Cybernetics**, v. 8, n. 6, p. 460–473, 1978.

TRICOT, C. **Curves and Fractal Dimension**. New York: Springer-Verlag, 1995.

VISA, S.; RAMSAY, B.; RALESCU, A.; VAN DER KNAAP, E. Confusion matrix-based feature selection. **CEUR Workshop Proceedings**, v. 710, p. 120–127, 2011.

WESZKA, J. S.; DYER, C. R.; ROSENFELD, A. Comparative Study of Texture Measures for Terrain Classification. **IEEE Transactions on Systems, Man and Cybernetics**, v. SMC-6, n. 4, p. 269–285, 1976.