

GUSTAVO PICHORIM BOIKO  
HERTON RONALDO KRZESINSKI

**ANÁLISE DE IMAGENS BASEADA EM DESCRITORES DE  
TEXTURA**

Trabalho de Graduação apresentado ao Curso  
de Bacharelado em Ciência da Computação, Se-  
tor de Ciências Exatas, Universidade Federal do  
Paraná.

Orientador: Prof. Dr. Hélio Pedrini

CURITIBA

2005

# Sumário

|   |            |
|---|------------|
| <b>Lista de Figuras</b>   | <b>iii</b> |
| <b>Lista de Abreviaturas e Siglas</b>                               | <b>iv</b>  |
| <b>Resumo</b>   | <b>v</b>   |
| <b>1 Introdução</b>   | <b>1</b>   |
| <b>2 Tópicos Relacionados ao Tema</b>                               | <b>3</b>   |
| 2.1 Textura . . . . .   | 3          |
| 2.2 Métodos . . . . .   | 5          |
| 2.2.1 GLCM - Grey Level Co-Occurrence Matrix . . . . .              | 5          |
| 2.2.2 HSVCM - Hue, Saturation, Value Co-Occurrence Matrix . . . . . | 8          |
| 2.3 Classificadores . . . . .                                       | 9          |
| 2.3.1 Redes Neurais . . . . .                                       | 9          |
| 2.3.2 Distância Mínima Euclidiana . . . . .                         | 10         |
| <b>3 Método Desenvolvido</b>  | <b>11</b>  |
| 3.1 Extração de amostras . . . . .                                  | 12         |
| 3.2 Treinamento do Classificador . . . . .                          | 13         |
| 3.3 Classificação da Imagem . . . . .                               | 13         |
| 3.4 Ferramenta Desenvolvida - KImageProcess . . . . .               | 14         |
| 3.4.1 Recursos Utilizados . . . . .                                 | 14         |
| 3.4.2 Organização da Ferramenta . . . . .                           | 15         |
| <b>4 Resultados</b>   | <b>20</b>  |
| 4.1 Imagem dos feijões (beans) . . . . .                            | 20         |
| 4.1.1 Resultado Caso 1 . . . . .                                    | 20         |
| 4.1.2 Resultado Caso 2 . . . . .                                    | 21         |
| 4.2 Mosaico de 5 Texturas . . . . .                                 | 23         |
| 4.2.1 Resultado Caso 1 . . . . .                                    | 23         |
| 4.2.2 Resultado Caso 2 . . . . .                                    | 24         |
| 4.3 Imagem letra “ka” . . . . .                                     | 25         |

|   |           |
|---|-----------|
| <i>SUMÁRIO</i>  | ii        |
| 4.4 Imagem Continente/Oceano (Sensoriamento Remoto) . . . . . | 26        |
| 4.5 Imagem Colorida - “Continente Verde” . . . . .            | 26        |
| <b>5 Conclusões e Trabalhos Futuros</b>                       | <b>31</b> |
| <b>Referências Bibliográficas</b>                             | <b>32</b> |

# Lista de Figuras

|      |  |    |
|------|--|----|
| 2.1  | Amostras . . . . .   | 4  |
| 2.2  | Imagem original a ser classificada . . . . .   | 4  |
| 2.3  | Direções para analisar transições de valores para gerar a matriz de co-ocorrência  | 6  |
| 2.4  | Pedaço de uma imagem com 2 níveis de cinza, Para o pixel central de valor 1 é<br>apresentado seus vizinhos de distância 1 ao redor . . . . . | 6  |
| 2.5  | Direções para analisar transições de valores para gerar a matriz de co-ocorrência<br>com os valores dos canais HSV . . . . .                 | 9  |
| 3.1  | Algoritmo Seqüencial de Iteração em Amostras . . . . .   | 12 |
| 3.2  | Algoritmo Cascata de Iteração em Amostras . . . . .  | 12 |
| 3.3  | Janela principal . . . . .   | 17 |
| 3.4  | Opções de Textura . . . . .  | 18 |
| 3.5  | Seleção de Características . . . . .   | 19 |
| 4.1  | Imagem dos feijões (beans) . . . . .   | 21 |
| 4.2  | Amostras - Beans . . . . .   | 21 |
| 4.3  | Resultado da classificação Caso 1 - Beans . . . . .  | 22 |
| 4.4  | Resultado da classificação Caso 2 - Beans . . . . .  | 22 |
| 4.5  | Mosaico de 5 Texturas . . . . .  | 23 |
| 4.6  | Amostras - Mosaico 5 . . . . .   | 23 |
| 4.7  | Resultado da classificação Caso 1 - Mosaico 5 . . . . .  | 24 |
| 4.8  | Resultado da classificação Caso 2 - Mosaico 5 . . . . .  | 24 |
| 4.9  | Imagem letra “ka” . . . . .  | 25 |
| 4.10 | Amostras - letra “ka” . . . . .  | 25 |
| 4.11 | Resultado da classificação de 4 tamanhos de janela - letra “ka” . . . . .  | 26 |
| 4.12 | Imagem Continente/Oceano . . . . .   | 27 |
| 4.13 | Amostras - Imagem Continente/Oceano . . . . .  | 27 |
| 4.14 | Imagem Continente/Oceano segmentada . . . . .  | 28 |
| 4.15 | Imagem Colorida - “Continente Verde” . . . . .   | 29 |
| 4.16 | Amostras - Imagem “Continente Verde” . . . . .   | 29 |
| 4.17 | Imagem “Continente Verde” segmentada . . . . .   | 30 |

# Lista de Abreviaturas e Siglas

|                |  |
|----------------|--|
| FANN (ou fann) | <i>Fast Artificial Neural Network</i>              |
| GLCM           | <i>Gray Level Co-Occurrence Matrix</i>             |
| HSVCM          | <i>Hue, Saturation, Value Co-Occurrence Matrix</i> |
| JavaNNS        | <i>Java Neural Network Simulator</i>               |
| JPEG           | <i>Joint Photographers Expert Group</i>            |
| PCA            | <i>Principal Component Analysis</i>                |
| PGM            | <i>Portable Graymap File Format</i>                |

# Resumo

A análise de imagens consiste em tentar identificar características e informações de interesse a fim de determinado objetivo. Junto com a computação, métodos podem ser utilizados para automatizar essa tarefa.

Nesse sentido dentro de uma imagem podemos identificar tipos e formas de agrupamento de textura, disposição dos elementos, formas, cada um com uma certa disposição de cores ou níveis de cinza.

Como objetivo podemos ter a identificação de áreas em imagens geográficas (obtidas via aérea ou por satélite), identificação de objetos em um meio industrial por robôs; na medicina e microscopia podemos ter identificação de itens em radiografias; na própria área de computação para armazenar e depois poder obter através de métodos rápidos imagens em um banco de dados.

Com isso, usando a análise de imagens, podemos atingir os diversos objetivos listados. A própria análise pode ser efetuada em etapas, nas quais há segmentação das imagens através de obtenção de amostras e a partir daí utilização de métodos para efetivar a própria segmentação.

O trabalho aqui apresenta a análise de imagens orientado a aspectos de textura usando processos computacionais, exatamente focada na área de segmentação. Cada textura com seus devidos parâmetros pode ser usada para segmentar a imagem a fim de qualquer objetivo.

Assim é demonstrado os conceitos de textura, treinamento e classificação de imagens através da computação e a influência das amostras e características nesse processo. São usados também definições e estudos de matrizes de co-ocorrência para níveis de cinza e proposto um novo método para níveis de cor. Tudo isso é testado na prática através de uma implementação de programa efetuado e os resultados apresentados.

# Capítulo 1

## Introdução

A análise de imagens em termos computacionais é uma área explorada há algum tempo, já que há várias aplicações para processamento de imagens. Reconhecimento facial, identificação de itens em processo industrial são algumas delas. Muitas vezes em áreas de risco, onde só é possível utilizar máquinas para efetuar certas tarefas (como identificar itens no fundo do mar através de imagens obtidas via um submarino não populado equipado com câmera, análise de imagens por um robô enviado a outro planeta, dentre outros), a análise de imagens de forma computacional é imprescindível.

Dentro dessa área há bastante interesse em segmentar imagens, dividindo em classes pré-definidas de acordo com a aplicação e o resultado desejado. Uma forma de segmentar imagens é dividi-las por textura.

Assim dependendo do tipo de textura pode-se analisar semanticamente uma imagem. Este trabalho teve por objetivo estudar os descritores de textura para que essa análise seja possível: a partir deles pode-se medir as características de amostras de textura para que isso sirva de base para a segmentação posterior.

No entanto, não estudamos somente os descritores: apresentamos também o estudo da classificação em geral, obtendo assim tanto o impacto de cada característica/descritor como a qualidade final da segmentação no processo de classificação.

Assim o objetivo da análise é a segmentação da imagem em regiões/classes de interesse. Cada região possui uma forma, cores ou níveis de cinza e textura característica. Essa segmentação consiste de uma classificação supervisionada, isto é, as amostras de cada classe de textura são obtidas para treinamento e posterior classificação (a obtenção das amostras antecede a classificação).

A textura contém as características de tonalidade, cores ou níveis de cinza, forma e distribuição espacial dos pixels, que determinam a sua forma, e é usando isso que pode se divi-

dir uma imagem em vários segmentos. Conforme veremos apesar de não haver uma definição formal única para textura teremos como isolar essas características e usar a favor no processo de classificação.

Na prática, pode se usar esse estudo na segmentação de imagens em geral, como nas seguintes áreas: Sensoriamento Remoto (obtenção de imagens de regiões via aérea ou por satélite de regiões, no sensoriamento remoto devemos armazenar e guardar as imagens em banco de dados para posterior busca/obtenção ou geralmente identificar pedaços de interesse na imagem. Para isso a segmentação é um método importante); Identificação Facial (as características de segmentação não somente podem ser usadas para a segmentação mas também para ferramentas de busca e segurança como na identificação computacional de rostos de pessoas); recuperação de imagens de um banco de dados ou de vídeo com certas características, medicina, microscopia, dentre outros.

Demonstramos assim os métodos computacionais envolvidos nesse processo de análise, segmentação e classificação de imagens. Revisitamos processos já conhecidos na área, como a utilização de matrizes de co-ocorrência de níveis de cinza e métodos relacionados às imagens em níveis de cinza. Apresentamos uma proposta de um novo método para classificação de imagens coloridas (baseado em um método já existente para obtenção de imagens coloridas de um banco de dados através de determinadas características) e também a implementação de um programa para análise de imagens e prova de todos os conceitos e estudos apresentados.

No capítulo 2, veremos mais os conceitos relacionados a texturas, obtenção de características e classificação. Depois no capítulo 3, explicaremos com detalhes como aplicamos os conceitos na prática e mostraremos de forma mais detalhada esses próprios conceitos. No capítulo 4 mostramos o que obtivemos de sucesso ou não na classificação/segmentação de imagens usando os conceitos estudados, demonstrados e implementados. Por final o capítulo 5 apresenta as conclusões e propõe alguns trabalhos futuros relevantes ao assunto abordado.

# Capítulo 2

## Tópicos Relacionados ao Tema

Este trabalho de pesquisa desenvolvido na área de processamento de imagens procurou buscar várias ferramentas para a solução de um problema específico: a análise de imagens baseada em descritores de textura. Muito pode se ir além da classificação de texturas baseada em amostras e uma imagem final, como vamos mostrar a seguir, que foi o foco que demos à pesquisa.

### 2.1 Textura

Definir o que é uma textura é algo difícil. Há várias formas de se definir textura. Geralmente é definida como uma primitiva (componente básico da imagem) que segue uma determinada regra ou padrão, sendo visualmente identificável como uma classe ou tipo. Porém textura poderia ser definida também quanto a outras características, como ser uma parte que se repete várias vezes em determinada imagem. O correto é que a textura é definida de acordo com a aplicação em que ela é necessária.

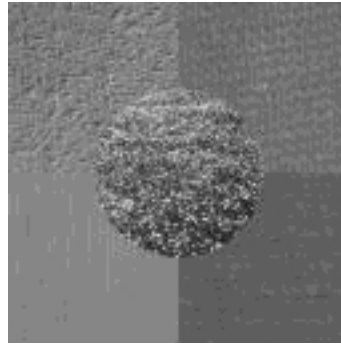
No entanto, vários pesquisadores concordam que uma textura define-se basicamente pela diferença entre pixels vizinhos e pela homogeneidade maior que a escala da imagem. Assim pode-se diferenciar as texturas pela aspereza, intensidade, etc.

Assim, a classificação de texturas basicamente consiste no processo em, a partir de determinadas amostras, encontrar características que determinem as diferenças entre elas para que as imagens formadas por conjuntos dessas amostras possam ser divididas em regiões (cada região pode ser chamada também como classe).

As características podem levar em conta vários fatores, como desde as mais simples (diferentes níveis de cinza ou de cores) até as mais complexas determinadas a partir de métodos estatísticos, como contraste, variância, etc. Geralmente essas características usadas dentro de diferentes métodos de classificação.



**Figura 2.1:** Amostras



**Figura 2.2:** Imagem original a ser classificada

As aplicações da divisão de uma imagem em regiões de textura são várias. Poderíamos por exemplo usar isso em um processo industrial, uma empresa textil, onde determinada máquina tem que costurar determinada parte de uma roupa onde o tecido apresenta um padrão diferente dos outros. Na robótica, para identificação através de uma câmera dos padrões do ambiente. Enfim, são várias áreas possíveis de utilização, conforme foi especificado na proposta.

O trabalho consistiu nessa área: implementação, testes e estudos das características e alguns métodos relacionados para classificação de texturas em imagens.

Dentre os métodos estudados e utilizados, estão a matriz de co-ocorrência de níveis de cinza (GLCM - Grey Level Co-Occurrence Matrix) e características conforme propostas por Haralick [5], além de características adicionais, para imagens em tons de cinza. Para imagens coloridas, foi utilizado o método hsvcm baseado na proposta de Dacheng [2], na qual usa o método semelhante à matriz de co-ocorrência de haralick, no entanto com várias modificações para representar características de texturas com cores no formato HSV (HSVCM - Hue, Saturation, Value Co-occurrence Matrix).

Utilizando esses métodos é possível obter características baseadas em probabilidade que definem vários aspectos/padrões de textura, com isso é possível determinar para cada classe de textura seus dados relacionados para que depois possa ser identificada em determinada imagem para ser classificada. Os dados de características de textura são geralmente treinados em um classificador para que este possa "aprender" para depois usar dados da imagem real para classificação. Dentre os classificadores implementados, testados e estudados estão as redes neurais (foram usadas duas implementações de rede neural) e a distância mínima Euclidiana.

## 2.2 Métodos

### 2.2.1 GLCM - Grey Level Co-Occurrence Matrix

A matriz de co-ocorrência de níveis de cinza definida por Haralick [5] mede a diferença entre dois pixels de distância  $d$  em determinado ângulo em termos probabilísticos. É determinado que 45 graus é uma variação melhor para medir as diferenças, então são gerados quatro matrizes de variações nos ângulos 0, 45, 90 e 135. Cada matriz mede a probabilidade de ocorrência dos níveis de cinza entre dois pixels no ângulo representado.

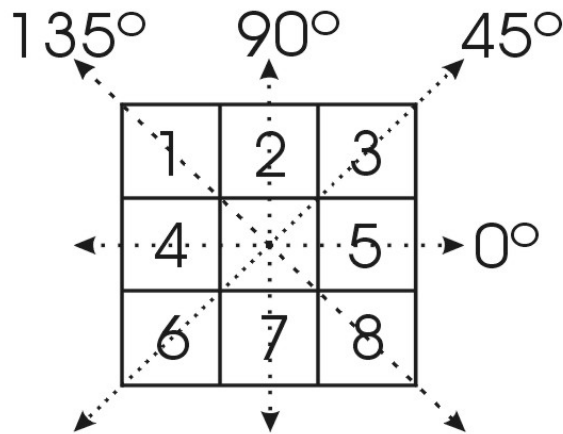
Para entender melhor o que é ou como funciona uma matriz de co-ocorrência, temos basicamente o conceito de vizinhança de pixels. Considerando uma certa imagem, para cada pixel dela temos pixels vizinhos. Assim, sendo um pixel qualquer na coordenada  $(x, y)$ , podemos ter como vizinhos desse pixel para uma distância 1 (ou seja, pixels imediatamente vizinhos ao pixel em questão):  $(x - 1, y)$ ,  $(x, y - 1)$ ,  $(x - 1, y - 1)$ ,  $(x + 1, y)$ ,  $(x, y + 1)$ ,  $(x + 1, y + 1)$ ,  $(x - 1, y + 1)$ ,  $(x + 1, y - 1)$ .

Poderíamos ter também nesse caso uma vizinhança maior do que 1: definindo  $d$  como a distância de vizinhança, poderíamos definir os vizinhos de distância “ $d$ ” do pixel  $(x, y)$  como:  $(x - d, y)$ ,  $(x, y - d)$ ,  $(x - d, y - d)$ ,  $(x + d, y)$ ,  $(x, y + d)$ ,  $(x + d, y + d)$ ,  $(x - d, y + d)$ ,  $(x + d, y - d)$ .

A matriz de co-ocorrência contém as transições de valores de cada pixel com relação aos seus pixels vizinhos, em determinada direção, conforme pode ser visto na figura 2.3. Como exemplo, se temos uma imagem com 2 níveis de cinza que variam entre 0 e 1, então para cada pixel com valor 0 ou 1, um vizinho seu também pode ter o valor 0 ou 1. Se o pixel sendo olhado tem valor 0, e o vizinho tem valor 1, então houve variação 0,1 entre os dois pixels. Como a variação ocorre entre o pixel considerado e seu vizinho, mas temos vários vizinhos, então analisamos as variações para cada direção:

- Na horizontal ou ângulo 0 grau: consideramos variações entre os vizinhos  $(x - 1, y)$  e  $(x + 1, y)$ ;
- Na diagonal de ângulo 45 graus: consideramos variações entre os vizinhos  $(x - 1, y + 1)$  e  $(x + 1, y - 1)$ ;
- Na vertical ou ângulo 90 graus: consideramos variações entre os vizinhos  $(x, y - 1)$  e  $(x, y + 1)$ ;
- Na diagonal de ângulo 135 graus: consideramos variações entre os vizinhos  $(x - 1, y - 1)$  e  $(x + 1, y + 1)$ .

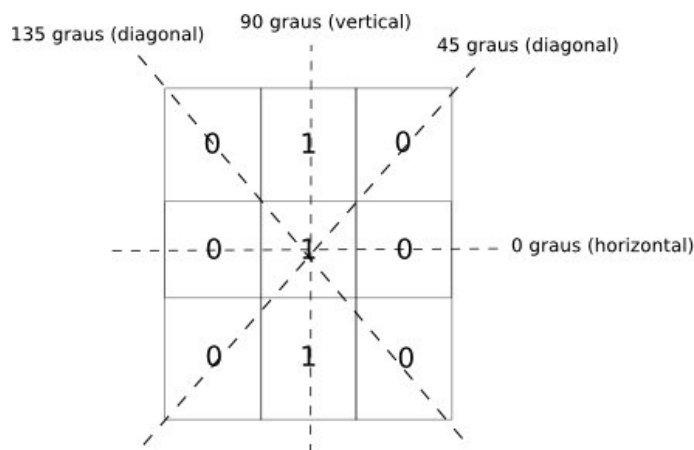
Assim, conforme exemplificado na figura 2.4, pegamos um pixel (o central, de coordenadas



**Figura 2.3:** Direções para analisar transições de valores para gerar a matriz de co-ocorrência

$x,y$  quaisquer) e seus vizinhos. Para este pedaço conforme mostra a figura, temos as seguintes variações/transições dos níveis de cinza:

- Na horizontal ou 0 graus:  $(x,y) \rightarrow (x-1,y) = 1 \rightarrow 0$  e  $(x,y) \rightarrow (x+1,y) = 1 \rightarrow 0$ ;
- Na diagonal de 45 graus:  $(x,y) \rightarrow (x-1,y+1) = 1 \rightarrow 0$  e  $(x,y) \rightarrow (x+1,y-1) = 1 \rightarrow 0$ ;
- Na vertical ou 90 graus:  $(x,y) \rightarrow (x,y-1) = 1 \rightarrow 1$  e  $(x,y) \rightarrow (x,y+1) = 1 \rightarrow 1$ ;
- Na diagonal de 135 graus:  $(x,y) \rightarrow (x-1,y-1) = 1 \rightarrow 0$  e  $(x,y) \rightarrow (x+1,y+1) = 1 \rightarrow 0$ .



**Figura 2.4:** Pedaço de uma imagem com 2 níveis de cinza, Para o pixel central de valor 1 é apresentado seus vizinhos de distância 1 ao redor

Para cada ângulo é gerado uma matriz de co-ocorrência, já que cada matriz mede as variações em cada direção (ou seja, são 4 direções = 4 matrizes resultantes). Aproveitando o pedaço de imagem apresentado na figura 2.4, se considerássemos ela como uma imagem completa, olhando para cada pixel sem considerar borda, teríamos as seguintes matrizes de co-ocorrência:

| 0 graus |   |   | 45 graus |   |   | 90 graus |   |   | 135 graus |   |   |
|---------|---|---|----------|---|---|----------|---|---|-----------|---|---|
|         | 0 | 1 |          | 0 | 1 |          | 0 | 1 |           | 0 | 1 |
| 0       | 0 | 6 | 0        | 0 | 4 | 0        | 8 | 0 | 0         | 0 | 4 |
| 1       | 6 | 0 | 1        | 4 | 0 | 1        | 0 | 4 | 1         | 4 | 0 |

Como se pode notar, a matriz de co-ocorrência na sua forma normal sempre tem o tamanho  $n \times n$ , onde  $n$  é o número de níveis de cinza usados para representar a imagem. Acima são mostrados o número de transições de um nível de cinza para outro, no caso é número de vezes da transição do nível da linha atual para a coluna atual. Agora atenção, porque as matrizes acima não são as matrizes de co-ocorrência de níveis de cinza (GLCM) propriamente ditas: as matrizes acima na prática não são representadas dessa forma, deixamos elas com os números de transições diretamente para fins didáticos. As matrizes de co-ocorrência após serem contados os números das transições tem esses números normalizados em termos probabilísticos, ou seja, cada transição tem a probabilidade de ocorrer de 0% a 100%. As matrizes acima normalizadas ficariam da seguinte forma (note que por não considerarmos a borda aqui entre cada direção nunca temos um número de transições necessariamente igual entre elas):

| 0 graus |     |     | 45 graus |     |     | 90 graus |      |      | 135 graus |     |     |
|---------|-----|-----|----------|-----|-----|----------|------|------|-----------|-----|-----|
|         | 0   | 1   |          | 0   | 1   |          | 0    | 1    |           | 0   | 1   |
| 0       | 0   | 0.5 | 0        | 0   | 0.5 | 0        | 0.75 | 0    | 0         | 0   | 0.5 |
| 1       | 0.5 | 0   | 1        | 0.5 | 0   | 1        | 0    | 0.25 | 1         | 0.5 | 0   |

No processo para classificação usando a GLCM, são usadas amostras, cada uma representando uma das classes de textura. Dentro das amostras percorre-se usando uma janela de tamanho configurável (exemplo, 5x5 pixels, 12x12 pixels). Para cada pixel percorrido na imagem, é determinado suas características, usando-se a GLCM para o cálculo. Assim, para cada janela obtida, um valor de característica é obtido, e esse dado pode ser utilizado para treinar um classificador para posterior classificação das imagens compostas das amostras treinadas.

Assim, cada característica é um resultado estatístico calculado a partir da matriz de probabilidades (a GLCM). As características abordadas em nossa pesquisa e implementadas no programa *kimageprocess* [1] foram as seguintes: Segundo Momento Angular (Angular Second Moment), Tendência de agrupamento (Cluster Tendency), Contraste (Contrast), Correlação (Correlation), Diferença de Entropia (Difference Entropy), Diferença de Variância (Difference Variance), Dissimilaridade (Dissimilarity), Energia (Energy), Entropia (Entropy), Inércia (Inertia), Medidas de Informação da Correlação (Information Measures of Correlation), Momento Diferencial Inverso (Inverse Differential Moment), Kurtosis, Máximo Coeficiente de Correlação (Maximal Correlation Coefficient), Máxima Probabilidade (Maximum Probability), Média (Mean), Falta de Simetria (Skewness), Desvio Padrão (Standard Deviation), Soma da Média (Sum Average), Soma da Entropia (Sum Entropy), Soma da Variância (Sum Variance) e Variância (Variance).

### 2.2.2 HSVCM - Hue, Saturation, Value Co-Occurrence Matrix

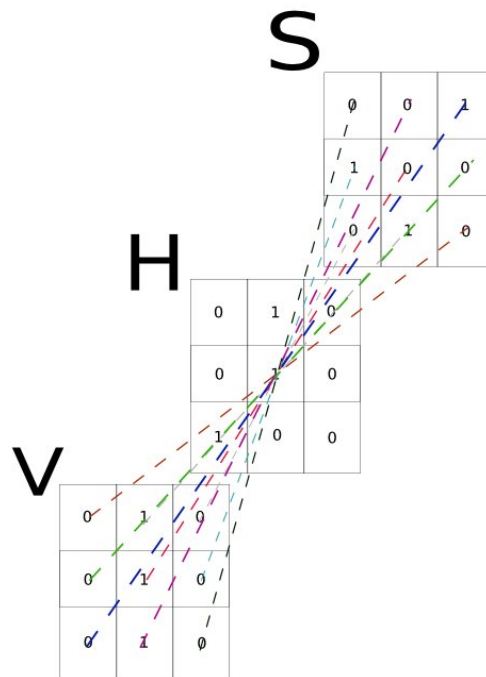
Para imagens coloridas é necessário usar outro método para classificação de texturas. Na verdade poderia-se converter uma imagem colorida em imagem com apenas tons de cinza, no entanto detalhes importantes definidos apenas pelas cores poderiam ser perdidos. Assim um método que leve em conta as cores foi definido e usado durante a pesquisa: é um método baseado no armazenamento de imagens coloridas e suas características para pesquisa e obtenção em um banco de dados, proposto por Dacheng [2].

O método usado foi originalmente proposto para obtenção de imagens armazenadas baseando-se no conteúdo, porém suas características e definições adaptados encaixam perfeitamente também na classificação de imagens coloridas baseando-se na textura.

A HSVCM é uma matriz de probabilidades assim como a GLCM. No entanto, como incluíse as cores, precisamos definir uma matriz 3D de probabilidades, de modo que podemos usar determinado sistema de cores. Sistemas de cores são definidos com algum propósito, e alguns deles como o RGB, LUV ou o HSV. Por exemplo, o RGB (Red Green Blue) não representa fielmente as cores como o esperado pela visão dos seres humanos (já que não há informação de vivacidade, brilho, etc., são apenas os valores das cores misturadas). HSV ou LUV são bons sistemas de cores, o que levou Dacheng em seu artigo a escolher o HSV.

Uma matriz 3D possui muito mais dados: enquanto tínhamos somente que usar quatro ângulos em direções 2D na GLCM com um plano de cores (os níveis de cinza), na HSVCM temos um espaço determinado por três planos; representamos a variação dos valores dos pixels em nove direções considerando os valores nas três camadas de cores (H, S e V), resultando em nove matrizes (veja as direções mostradas na figura 2.5), ou seja, nesse método para imagens coloridas consideramos 9 direções que resultam em 9 matrizes analogamente ao que ocorria na GLCM onde eram quatro direções resultando em quatro matrizes como foi explicado, e o mesmo método de geração das matrizes é usado. Tomando o valor do plano H de um pixel, teremos 9 valores nos planos S e V incluindo os do próprio pixel e de seus vizinhos. Ligando retas entre o valor do plano H do pixel e dos pixels vizinhos temos 9 direções como mostra a figura 2.5, e usando essas direções geramos as matrizes de co-ocorrência que podemos usar depois para treinamento e classificação.

Assim, usando essa matriz de co-ocorrência junto com algumas características adaptadas a ela, podemos usá-la na classificação de imagens coloridas. Implementamos e testamos junto com a HSVCM quatro características: Segundo Momento Angular (Angular Second Moment), Contraste (Contrast), Correlação (Correlation) e Entropia (Entropy).



**Figura 2.5:** Direções para analisar transições de valores para gerar a matriz de co-ocorrência com os valores dos canais HSV

## 2.3 Classificadores

### 2.3.1 Redes Neurais

As redes neurais compreendem algo próximo a uma implementação artificial de uma rede de neurônios. A partir de uma entrada de dados e resultados esperados podemos treinar essa rede para que ela aprenda o que é o certo que esperamos. Assim, as redes neurais encaixam perfeitamente no treinamento/aprendizado de características de texturas para que após ela possa corretamente classificar imagens em classes de textura, subdividindo e delineando as regiões esperadas.

Utilizamos implementações já prontas de redes neurais para alimentar com os dados das características obtidas através do cálculo com as matrizes de probabilidades (GLCM e HSVCM). Os testes efetuados foram feitos com o JavaNNS (Java Neural Network Simulator) [6], lwnn (Lightweight Neural Network) [4] e fann (Fast Artificial Neural Network) [3].

### **2.3.2 Distância Mínima Euclidiana**

O classificador de distância mínima euclidiana foi implementado no kimageprocess com o intuito de compará-lo com o classificador por rede neural, a fim de pelo menos verificar como outro classificador se comportaria em comparação. A classificação por distância mínima euclidiana consiste em tentar detectar as classes de textura na imagem por distância entre pixels. Os pixels são classificados em agrupamento. Cada pixel é adicionado ao agrupamento que apresenta a menor distância.

# Capítulo 3

## Método Desenvolvido

A classificação de imagens é um assunto que pode ser abordado sob vários focos diferentes dependendo do fim a ser atingido. Um desses focos é a classificação de imagens utilizando descritores de textura. A abordagem utilizada consiste em primeiro extrair amostras representativas das texturas a serem classificadas e realizar um treinamento supervisionado no classificador escolhido. Para este trabalho foram implementados três classificadores: dois baseados em redes neurais e um baseado em distância mínima euclideana.

Após ter realizado o treinamento do classificador, a imagem que será classificada é submetida ao classificador que decidirá à qual das texturas cada pixel pertence.

Como forma de extraírmos as características de cada textura, utilizamos a abordagem proposta por Haralick [5], na qual calculamos uma matriz de co-ocorrência de níveis de cinza, GLCM (Gray-Level Co-Occurrence Matrix), e à partir das informações coletadas calculamos alguns aspectos relativos à essa textura, como contraste, correlação, entropia, etc.

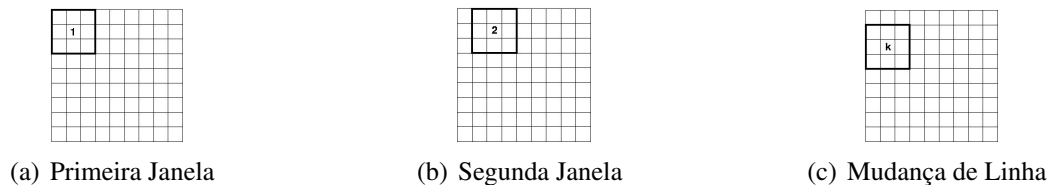
O método citado limita-se à análise de texturas representadas por tons de cinza. Como forma de expandir o trabalho e avaliar novos métodos, foi também implementada a abordagem proposta por Dacheng [2], a qual analisa os aspectos de textura de imagens utilizando o modelo de cores HSV e matriz de co-ocorrência em três dimensões (cada uma das dimensões representando uma banda do modelo HSV). Assim como o método utilizado para texturas em tons de cinza, neste método calculamos informações a partir da matriz de co-ocorrência e utilizamos essas informações como uma forma de descrever uma textura.

Diversos fatores são determinantes para uma adequada classificação de imagens. Dentre eles temos o tamanho da janela escolhida, a quantidade e a qualidade das características escolhidas da textura, o uso ou não de normalização dos dados obtidos, o método de normalização utilizado, e o classificador utilizado.

### 3.1 Extração de amostras

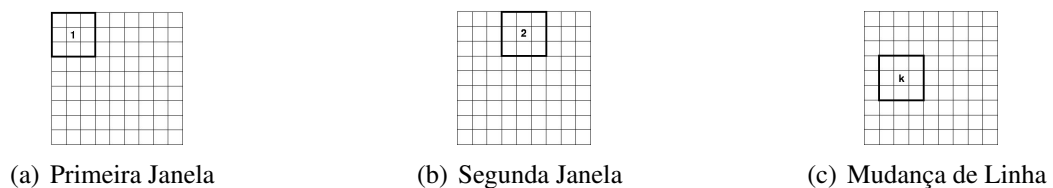
Dada uma textura a ser aprendida pelo classificador, é necessário utilizar uma amostra dessa textura para a extração das informações relevantes sobre ela. Tendo coletado essa amostra, escolhemos uma janela de tamanho definido (que deverá ter o mesmo tamanho da janela utilizada para a classificação da imagem posteriormente) e um algoritmo para percorrer essa amostra a fim de obter as informações necessárias. O algoritmo para percorrer amostras pode ser:

- **Seqüencial:** A primeira janela selecionada neste algoritmo é posicionada no canto superior esquerdo da amostra. A próxima janela se dá pelo deslocamento da janela à direita em um pixel. Ao encontrar o final da linha, a janela é deslocada em um pixel para baixo e retornada à coluna mais à esquerda da amostra. A figura 3.1 demonstra de forma visual o funcionamento do algoritmo.



**Figura 3.1:** Algoritmo Seqüencial de Iteração em Amostras

**Cascata:** Assim como o algoritmo seqüencial, a primeira janela está situada no canto superior esquerdo da amostra. A diferença neste algoritmo é que as janelas são todas disjuntas. A próxima janela a ser selecionada sofrerá um deslocamento à direita de exatamente a largura da janela. Com isto consegue-se a extração de janelas que representem toda a amostra. Ao chegar no final da linha, a janela é deslocada para baixo tantos pixels quanto o comprimento da janela, e ao invés de retornar para a primeira coluna, ele retorna para a segunda coluna. Na próxima mudança de linha irá retornar para a terceira coluna e assim por diante até que esse deslocamento atinja a largura da janela. Nesse caso o retorno será novamente para a primeira coluna. A figura 3.2 mostra o funcionamento deste algoritmo.



**Figura 3.2:** Algoritmo Cascata de Iteração em Amostras

- **Aleatório:** Neste algoritmo janelas totalmente internas à amostra são escolhidas de forma aleatória, e a partir delas são calculadas as informações necessárias.

A escolha correta das amostras de uma textura é fator decisivo na qualidade da classificação da imagem. Ao pegarmos uma amostra não-representativa da textura estaremos sujeitos a um aprendizado incorreto no classificador, e por consequência os pixels relativos à essa textura podem não ser corretamente identificados.

## 3.2 Treinamento do Classificador

O classificador é peça essencial no processo de segmentação de imagens proposto. O ajuste fino de seus parâmetros pode garantir um bom resultado, mas uma configuração mal-feita pode ser a causa do fracasso dos experimentos. A escolha de um classificador apropriado depende muito do espaço de valores a ser segmentado. Para certificar a escolha deste classificador apropriado, os testes foram realizados utilizando-se dois classificadores cujas abordagens eram significativamente diferentes. Cada uma das características extraídas da textura será utilizada como uma entrada no classificador. Todos os classificadores utilizados utilizam uma inter-relação entre as entradas como forma de identificar cada classe de textura. Com base nessa informação, é importante garantir que todas as entradas do classificador tenham o mesmo peso na hora que forem utilizadas. As características utilizadas geram faixas de valores que em muitos casos podem ser extremamente distantes (principalmente no caso de texturas coloridas), e portanto a aplicação de um algoritmo de normalização dessa faixa de valores se torna importante. Para treinar o classificador é necessário ter os valores das características de uma certa amostra devidamente colocados nas entradas do classificador, e a classe à qual esta amostra pertence posicionada nas saídas do classificador. Este utilizará essas informações para criar uma associação entre as entradas e as saídas que será utilizada posteriormente para classificar a imagem.

## 3.3 Classificação da Imagem

A classificação da imagem é feita pixel a pixel através da utilização de uma janela de tamanho definido ao redor do pixel a ser classificado. A partir desta janela é calculada a matriz de co-ocorrência, seja ela colorida ou de tons de cinza, e as características da textura. Os valores calculados são utilizados como entrada nos classificadores que, através da abordagem que cada um utiliza, informam a classe que mais se aproxima dos valores informados. Para cada classe existente será atribuído um tom de cinza de tal forma que a diferenciação das classes seja visualmente possível. Dado o resultado do classificador, a cor definida para a classe que mais se aproximou dos valores de entrada será atribuída a esse pixel na imagem de saída. É importante observar que se os valores utilizados no treinamento haviam sido normalizados, a mesma normalização deverá ser aplicada aos valores a serem classificados.

## 3.4 Ferramenta Desenvolvida - KImageProcess

Baseados nessa extensa combinação de fatores envolvidos, mostrou-se necessária a criação que uma ferramenta que tornasse simples o ajuste dos parâmetros ao realizar o treinamento e classificação de imagens. Nesse contexto foi criado o KImageProcess, tendo como principal objetivo auxiliar a execução dos testes, diminuindo o tempo gasto com atividades que não eram o foco principal do trabalho.

### 3.4.1 Recursos Utilizados

A possibilidade de testarmos novos métodos de classificação de imagens envolvendo descritores de texturas coloridas nos fez ter uma certa preocupação em relação à maleabilidade do código fonte desenvolvido visando a fácil adição de novos recursos, evitando a necessidade de alterações significativas no programa.

Pelo conhecimento prévio dos autores sobre C++ e sobre a biblioteca QT, e pelo fato de esta fornecer diversos recursos básicos para o processamento de imagens, tais como a leitura e gravação de diferentes formatos de arquivo, o tratamento de cores tanto em RGB como em HSV, funções que facilitam o acesso aos pixels individualmente, entre outros, foram decisivos na hora de decidirmos a base do software. A escolha do uso de algumas das bibliotecas do KDE foi feita por esta fornecer recursos para a implementação de software baseado em plugins (módulos que podem ser acoplados ao programa sem a necessidade de recompilar todo o código fonte).

#### A Linguagem de Programação C++

A programação orientada à objetos possibilita a criação de ferramentas completas sem aumentar muito a complexidade do código. A utilização de conceitos como herança, templates e outros flexibiliza a criação de novos recursos utilizando código já existente.

Por outro lado, a linguagem de programação C++ é derivada da linguagem C e portanto permite acesso a recursos de mais baixo nível facilitando a otimização de código para melhor uso do processador e memória.

#### A biblioteca QT

Quando do início deste projeto havia um objetivo a ser atingido e a escolha de uma biblioteca adequada para a implementação foi fator importante. A biblioteca QT desenvolvida pela Trolltech fornece recursos que possibilitam o programador focar seus esforços na parte relevante da

implementação, não tendo que se preocupar em escrever códigos para tratamento de cadeias de caracteres, leitura e escrita de imagens, conversão de modelos de cores, etc. Com isso o tempo gasto em desenvolvimento é melhor aproveitado e mais recursos podem ser implementados. A possível perda em performance proveniente do uso de abstrações (geralmente encontradas em bibliotecas de mais alto nível) é compensada pela facilidade de criação e manutenção de código.

### **As bibliotecas do KDE**

Devido à grande quantidade de combinações de opções possíveis e da complexidade que isto implica, optou-se por criar uma interface gráfica onde todas as opções pudessem ser configuradas e armazenadas de forma simples e rápida, sem a necessidade do uso de diversas opções de linha de comando.

A possibilidade de salvar as configurações usadas para cada teste em específico facilita para a posterior reprodução dos testes realizados, o que no ambiente acadêmico é de extrema importância.

A escolha do KDE como base de desenvolvimento teve outros fatores importantes. Um deles foi a existência de um mecanismo de plugins que pôde ser usado para a fácil extensão da implementação inicial do software.

### **3.4.2 Organização da Ferramenta**

Nesta seção é apresentada a organização da ferramenta, bem como uma explicação sobre a sua utilização.

Todo o programa foi desenvolvido com base na idéia de projetos, nos quais são armazenadas as opções utilizadas para cada teste, as amostras selecionadas e a imagem a ser classificada. As alterações realizadas em todas as interfaces do programa são salvas num arquivo de projeto e podem ser restauradas posteriormente.

Cada uma das etapas do processo de segmentação de imagens possui variantes que precisam ser testadas a fim de tentar obter melhores resultados. A forma mais flexível encontrada para suprir essa necessidade foi a utilização de uma arquitetura de plugins na qual fosse possível a adição de novas variantes sem a necessidade de modificações no código fonte principal.

Essa arquitetura de plugins se divide em três níveis. No primeiro deles, existem os plugins que implementam os métodos de classificação. Para este trabalho foram utilizados dois métodos diferentes baseados em matrizes de co-ocorrência, um deles implementando o método proposto por Haralick [5] para imagens em tons de cinza, e o outro implementando o método proposto por Dacheng [2] para imagens coloridas.

O segundo nível é composto pelas características referentes à cada método. Para o primeiro método (gldm) foram implementadas as características propostas por Haralick [5] e características desenvolvidas por outros pesquisadores, e para o segundo método foram incluídas apenas as características propostas por Dacheng [2]. As características podem ser dependentes umas das outras. Neste caso o programa irá calcular apenas uma vez os dados necessários e manterá uma cópia do mesmo caso este seja referenciado novamente.

No terceiro nível tem-se os classificadores que são responsáveis por analisar os dados coletados pelas outras etapas. Foram implementados dois classificadores baseados em redes neurais artificiais, e um classificador baseado em distância mínima euclidiana. Estes classificadores podem ser utilizados com qualquer um dos dois métodos e suas respectivas características.

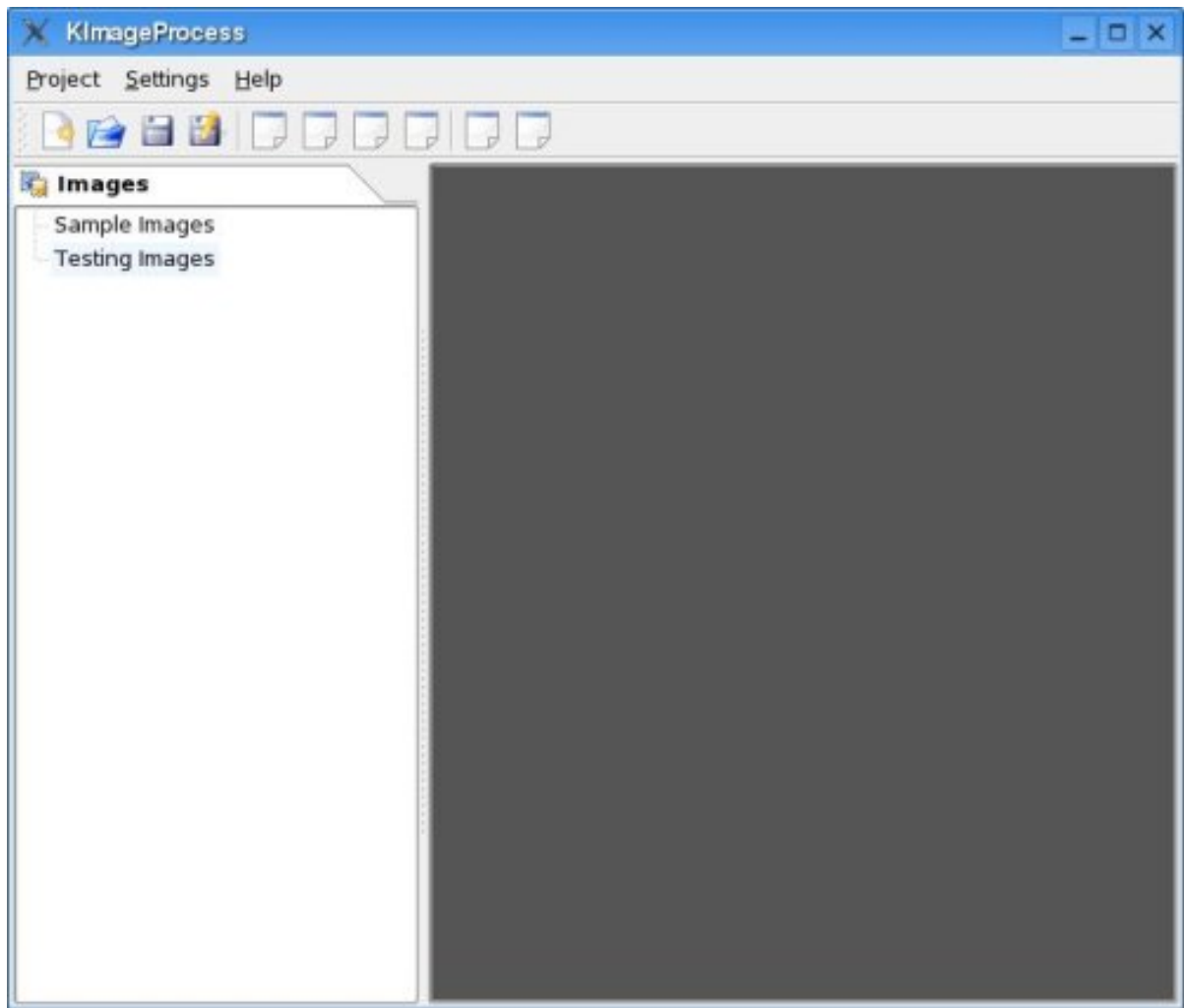
### Janela principal

Ao iniciar o programa, a janela principal (figura 3.3) é a primeira interface apresentada ao usuário. A área desta janela é dividida entre uma lista de componentes e um visualizador de imagens (utilizado para visualizar amostras e imagens presentes no projeto atual). A lista de componentes mostra as amostras e imagem adicionadas ao projeto. Ao selecionar qualquer um dos itens da lista, este é exibido no painel direito da interface. As opções mais relevantes estão dispostas numa barra de ferramentas para facilitar o acesso aos recursos mais utilizados. Outras opções são apresentadas no menu superior organizadas nos seguintes itens:

- **Projeto/Project:** Contém opções relacionadas ao projeto (salvar, abrir, fechar, criar novo) e opções para adicionar amostras e imagens.
- **Configurações/Settings:** Fornece acesso às opções do programa (seleção de método de classificação, preferências gerais, etc) e à configuração das teclas de atalho.
- **Ajuda/Help:** Arquivo de ajuda do programa e informações do projeto.

Antes de iniciar o processo de treinamento e classificação, é necessário configurar os parâmetros do programa de acordo com o teste a ser realizado. As opções configuradas nesta interface são válidas para todo o projeto sempre que aplicáveis.

A primeira opção apresentada nos permite estipular se utilizaremos janelas para treinamento e classificação ou se será utilizada a imagem inteira para a extração de informações. No mesmo painel temos também a opção de configurar o tamanho da janela. Embora o programa permita ter janelas assimétricas e/ou com tamanhos pares, é aconselhável o uso de janelas ímpares e simétricas, pois tem-se o pixel central cercado pela mesma quantidade de vizinhos em todos os lados.



**Figura 3.3:** *Janela principal*

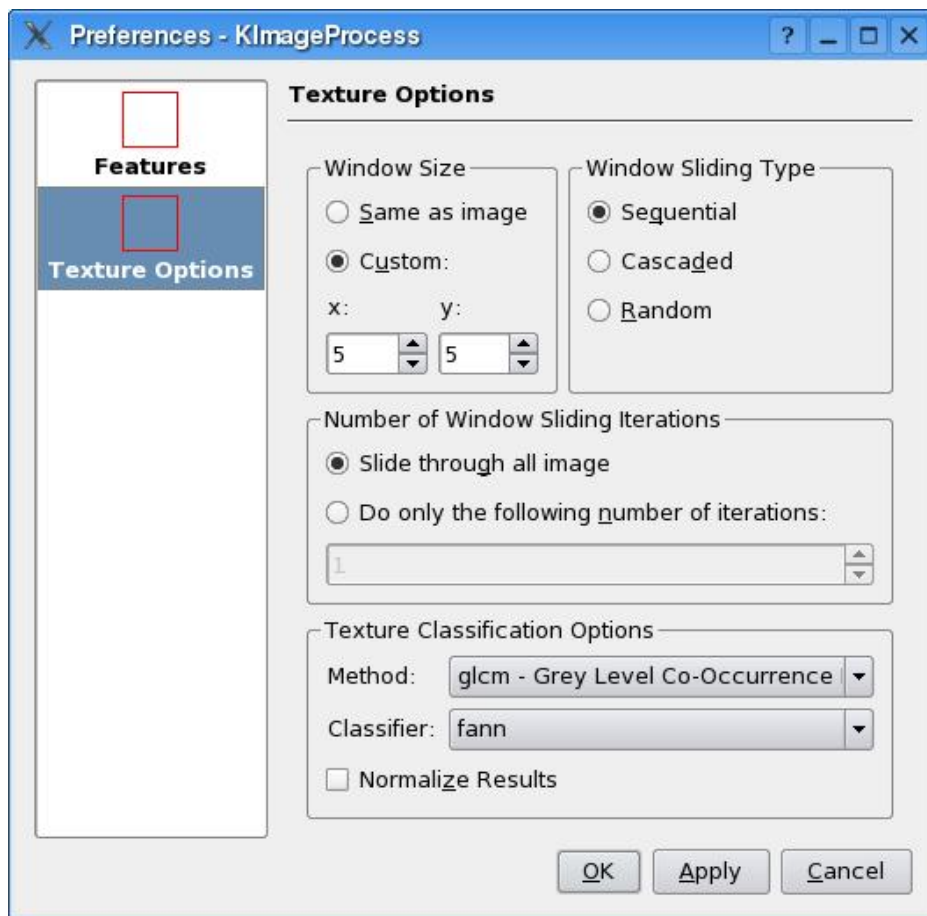
Na mesma interface configuramos também o algoritmo de percurso de janelas (utilizado apenas para as amostras de treinamento, visto que na classificação é necessário caminhar por toda a imagem). Um dentre os três diferentes percursos pode ser utilizado: *Seqüencial*, *Cascata* ou *Aleatório*

Para os dois primeiros algoritmos apresentados, com a finalidade de reduzir o custo computacional desta etapa do processo, foi adicionada a opção de limitar o número de iterações a serem utilizadas. Como efeito colateral temos que a amostra não será lida por inteiro para a extração de características, sendo que sempre a parte superior será utilizada (a menos que seja configurado um número extremamente grande de iterações que faça com que o percurso atinja o fim da imagem). Existe a possibilidade de selecionarmos o método que será utilizado para a classificação de imagens. No decorrer do projeto identificamos a necessidade de flexibilização para adição de novos métodos, o que nos levou a implementar os métodos como plugins no programa. Novos métodos podem ser criados sem a necessidade de recompilação de todo o programa. Métodos obsoletos também podem ser removidos sem a necessidade de alterações no código fonte principal. A próxima opção da interface se refere justamente à seleção do

método a ser utilizado.

Os classificadores (redes neurais, distância mínima, etc) também foram implementados como plugins, e com isto a adição de novos classificadores se tornou simples e rápida, possibilitando a realização de experimentos com diversos classificadores de redes neurais e com um classificador de distância mínima. Isto também abre espaço para trabalhos futuros envolvendo novos métodos e novos classificadores.

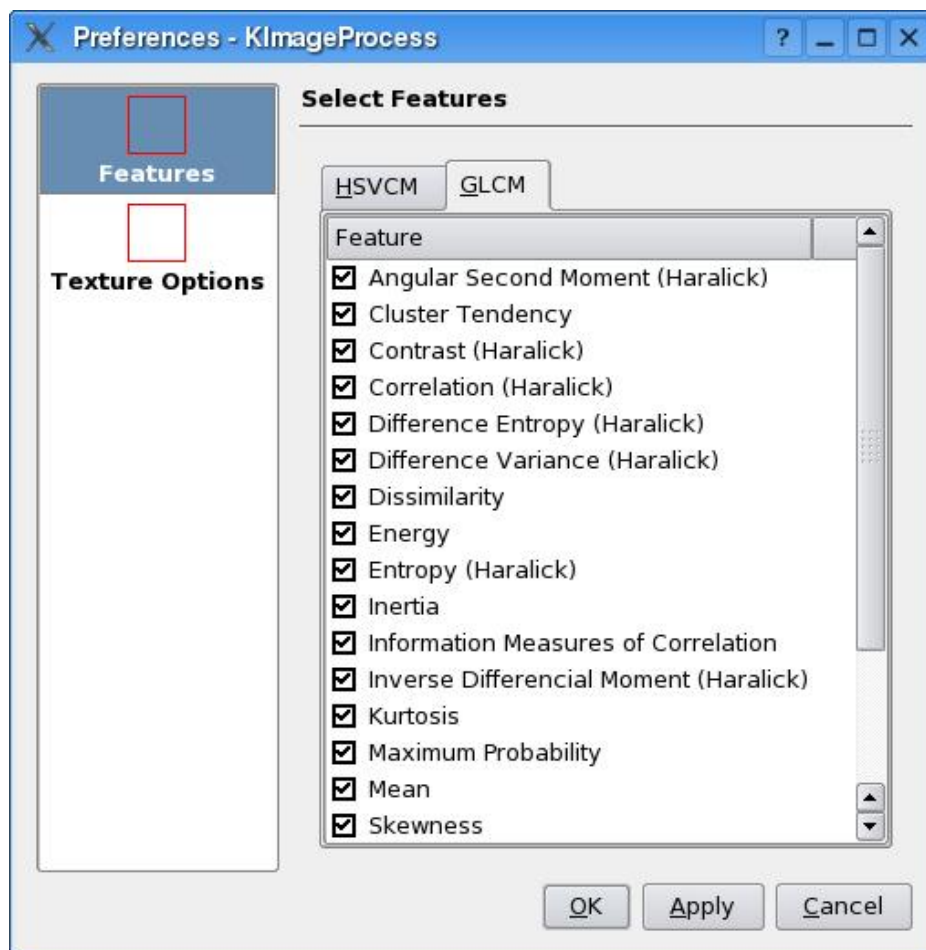
A última opção apresentada permite configurar se será utilizada a normalização dos dados extraídos para o correto balanceamento dos pesos de cada característica apresentada ao classificador.



**Figura 3.4:** *Opções de Textura*

A interface de seleção de características é o local no qual são selecionadas as características de textura a serem utilizadas para classificar a imagem. Para cada um dos métodos utilizados (GLCM e HSVCM), são apresentadas suas respectivas características.

Um dos motivos da existência desta interface é o fato de o custo computacional dos cálculos ser relativamente alto e portanto selecionar as características mais representativas é recomendável. Alguns dos resultados também indicaram que o uso de todas as características disponíveis faz com que a performance do classificador piore devido ao excesso de informação.



**Figura 3.5:** Seleção de Características

# Capítulo 4

## Resultados

Os resultados obtidos foram através da obtenção das características, treinamento usando as características, e classificação com base nos dados de treinamento. Vamos mostrar várias imagens, suas amostras e a configuração da classificação.

Conforme mostrado, a classificação ocorre da seguinte forma resumidamente: divide-se a imagem em regiões de textura. Cada região possui uma amostra. A partir das amostras treina-se o classificador desejado, e então com o classificador treinado põe-se à prova a imagem formada por texturas das quais as amostras foram obtidas. O classificador identifica as regiões da imagem, segmentando-a.

Cada imagem de resultado na realidade representa cada região de textura delineada por uma cor (exemplo: região 1 da imagem ganha a cor preta e região 2 cor branca): ou seja, ela significa apenas a identificação de cada classe de textura da imagem original.

### 4.1 Imagem dos feijões (beans)

A imagem 4.1 é uma das que possui uma classificação simples (geralmente qualquer classificador obtém bons resultados com ela), já que pode ser decomposta em apenas 2 regiões: uma é o fundo branco da imagem, a outra são as sementes de feijão.

#### 4.1.1 Resultado Caso 1

Configurações:

- janela tamanho 5x5 pixels



**Figura 4.1:** *Imagem dos feijões (beans)*



(a) amostra da região/classe 1

(b) amostra da região/classe 2

**Figura 4.2:** *Amostras - Beans*

- treinamento usando 30 iterações percorrendo a amostra em cascata
- normalização utilizada
- características usadas: segundo momento angular, contraste, correlação, dissimilaridade e variância
- classificador: rede neural (fann)

O resultado apresentado na imagem 4.3 foi satisfatório, onde percebe-se claramente a divisão das duas regiões da imagem.

#### 4.1.2 Resultado Caso 2

Configurações:

- janela tamanho 5x5 pixels
- treinamento usando 30 iterações percorrendo a amostra em cascata
- normalização utilizada



**Figura 4.3:** *Resultado da classificação Caso 1 - Beans*

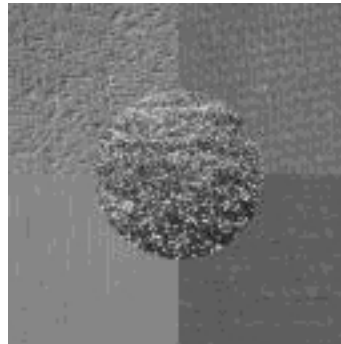
- características usadas: tendência de agrupamento, inércia, kurtose, probabilidade máxima e falta de simetria (skewness)
- classificador: rede neural (fann)



**Figura 4.4:** *Resultado da classificação Caso 2 - Beans*

O caso do resultado apresentado na imagem 4.4 apenas variou as características em relação ao caso 1. Vemos aqui que com a imagem dos feijões não apenas usando as características principais propostas por Haralick se obtém sucesso: com outras características foi possível identificar as regiões da imagem com sucesso.

## 4.2 Mosaico de 5 Texturas



**Figura 4.5:** *Mosaico de 5 Texturas*

Na seção sobre os tópicos abordados já mostramos a imagem do mosaico de 5 texturas. Ela é uma das imagens teste com uma complexidade maior. Aliado ao fato de a imagem possuir uma definição não muito boa para cada textura faz com que a classificação se torne mais difícil, e assim pondo à prova qualquer classificador baseado na análise dos descritores de textura.



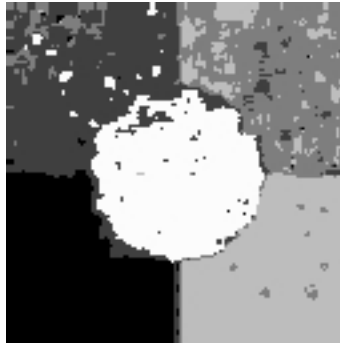
**Figura 4.6:** *Amostras - Mosaico 5*

### 4.2.1 Resultado Caso 1

Configurações:

- janela tamanho 5x5 pixels
- treinamento usando 30 iterações percorrendo a amostra em cascata
- normalização utilizada
- características usadas: segundo momento angular, contraste, correlação, dissimilaridade e variância
- classificador: rede neural (fann)

O resultado (figura 4.7) não é perfeito, no entanto percebe-se claramente que a segmentação representa fielmente a estrutura da imagem original.

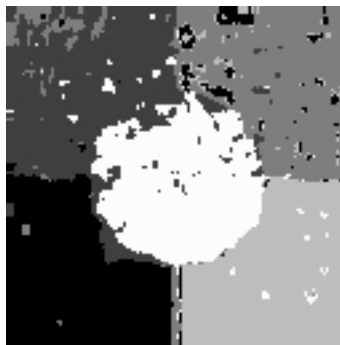


**Figura 4.7:** *Resultado da classificação Caso 1 - Mosaico 5*

## 4.2.2 Resultado Caso 2

Configurações:

- janela tamanho 5x5 pixels
- treinamento usando 30 iterações percorrendo a amostra em cascata
- normalização utilizada
- características usadas: tendência de agrupamento, inércia, kurtose, probabilidade máxima e falta de simetria (skewness)
- classificador: rede neural (fann)



**Figura 4.8:** *Resultado da classificação Caso 2 - Mosaico 5*

A mesma experiência realizada com a imagem dos feijões é demonstrada aqui (figura 4.8), só que agora com o mosaico de 5 texturas que possui uma estrutura mais complexa. O uso de outras características num treinamento de mesma configuração demonstrou que não há mudanças significativas, que por exemplo deixem a classificação com mais exatidão. Pequenos erros ainda são visíveis por todo o resultado, no entanto a segmentação pode ser considerada como satisfatória.

### 4.3 Imagem letra “ka”

A imagem 4.9 mostra a letra também possui uma composição simples necessitando de apenas duas amostras, já que a segmentação é em duas regiões.



**Figura 4.9:** Imagem letra “ka”



(a) amostra da região/classe 1

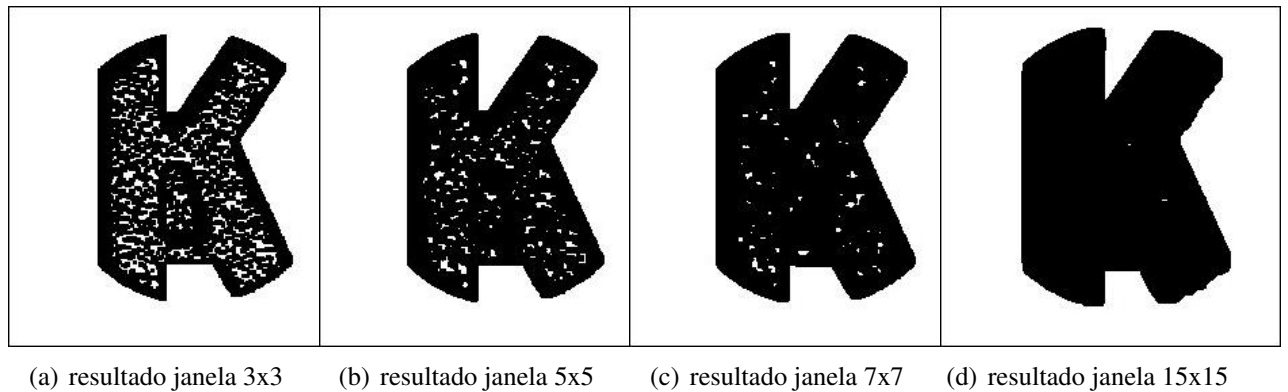
(b) amostra da região/classe 2

**Figura 4.10:** Amostras - letra “ka”

Foram feitos quatro testes com essa imagem, sempre usando as mesmas configurações alterando apenas o tamanho da janela de treinamento. Podemos verificar nos resultados de forma breve assim o impacto dos tamanhos de janela na classificação.

As configurações usadas foram as seguintes:

- janelas de tamanho 3x3, 5x5, 7x7 e 15x15 pixels
- treinamento usando 30 iterações percorrendo a amostra de forma sequencial
- normalização utilizada
- características usadas: segundo momento angular, contraste, desvio padrão, média e correlação (PCA)
- classificador: rede neural (javanns)



**Figura 4.11:** Resultado da classificação de 4 tamanhos de janela - letra “ka”

#### 4.4 Imagem Continente/Oceano (Sensoriamento Remoto)

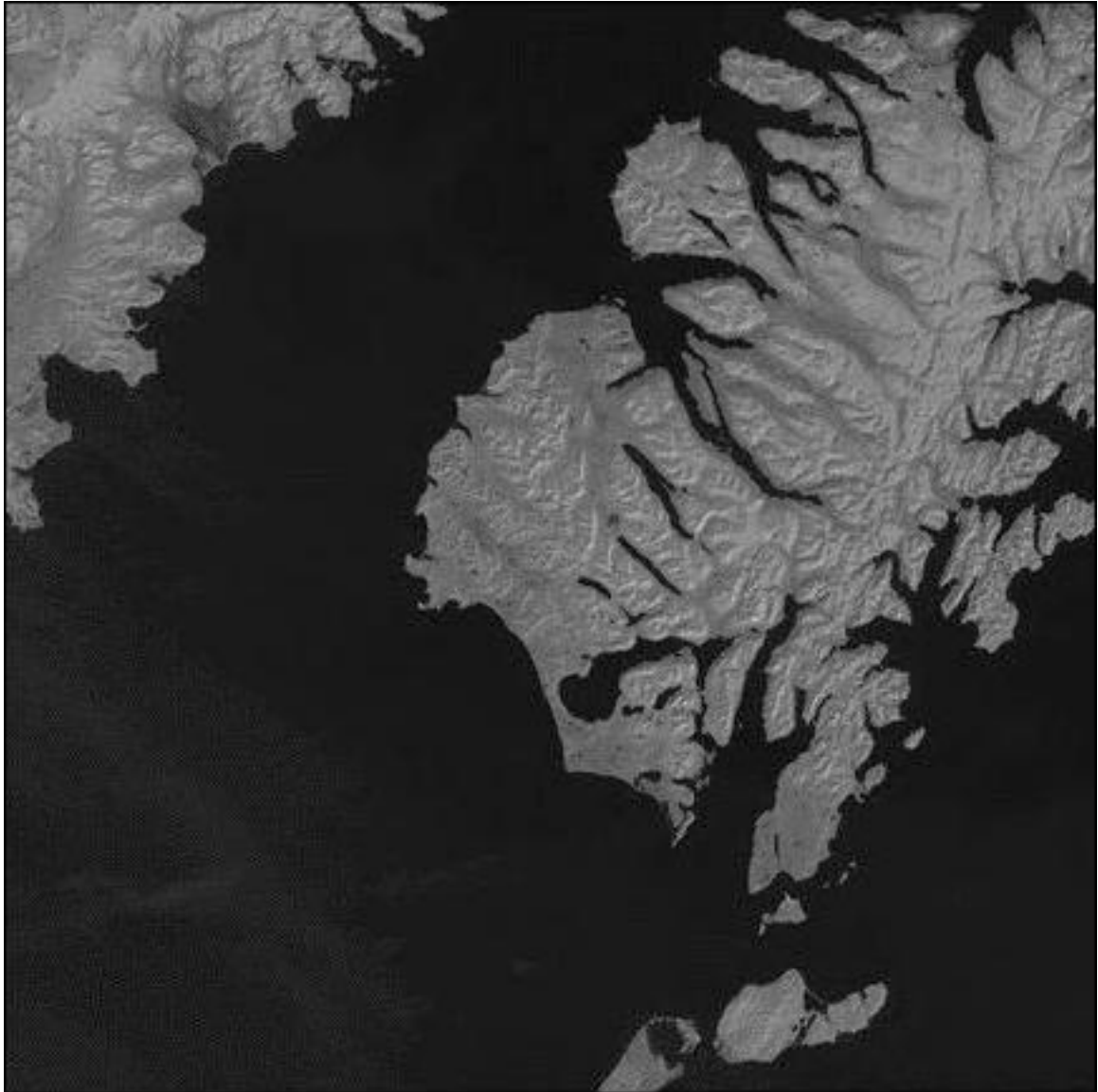
Em sensoriamento remoto geralmente deseja-se segmentar imagens, verificando áreas com determinado objetivo. Resolvemos mostrar uma aplicação simples aqui, que é saber as regiões de um mapa o que é continente e o que é oceano. O mapa/imagem 4.12 usado foi obtido através do satélite LANDSAT e corresponde à região de Kodiak Island, situada no estado do Alaska nos EUA.

A segmentação foi feita usando-se janela 3x3, imagem percorrida em cascata em 30 interações, classificador rede neural javanns; as características selecionadas foram segundo momento angular, média, desvio padrão, contraste e correlação, com normalização sendo usada.

#### 4.5 Imagem Colorida - “Continente Verde”

Com base no método de Haralick para imagens em tons de cinza e de Dacheng para armazenamento de imagens coloridas, o kimageprocess também efetua a segmentação de texturas coloridas; no caso aqui apresentamos outra imagem para sensoriamento, segmentando continente e oceano.

Para a imagem colorida “Continente Verde” obtida via satélite, foram usadas as características adaptadas para a matriz de co-ocorrência 3D (HSVCM) segundo momento angular, contraste, correlação e entropia; as amostras foram percorridas de maneira sequencial por 30 interações usando uma janela 3x3. Foi usado o classificador por rede neural (fann) e normalização dos dados de entrada da rede.



**Figura 4.12:** *Imagem Continente/Oceano*



(a) amostra da região marítima

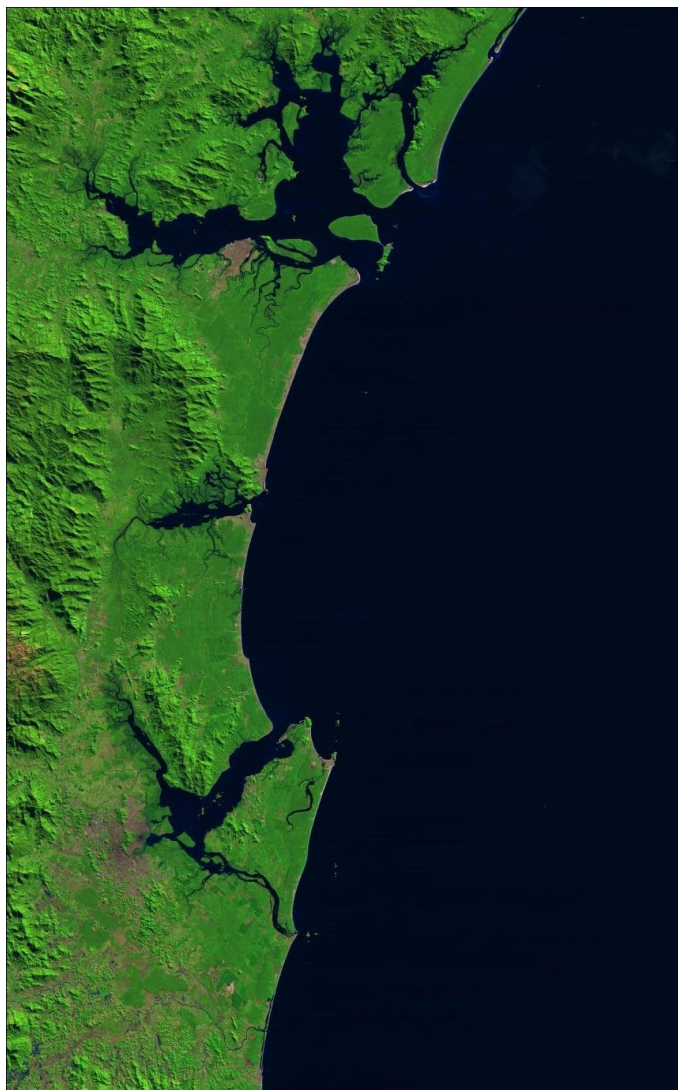


(b) amostra da região continental

**Figura 4.13:** *Amostras - Imagem Continente/Oceano*



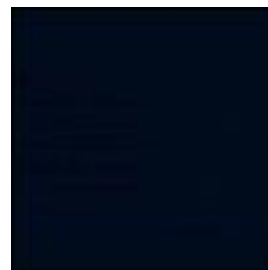
**Figura 4.14:** *Imagem Continente/Oceano segmentada*



**Figura 4.15:** *Imagem Colorida - “Continente Verde”*

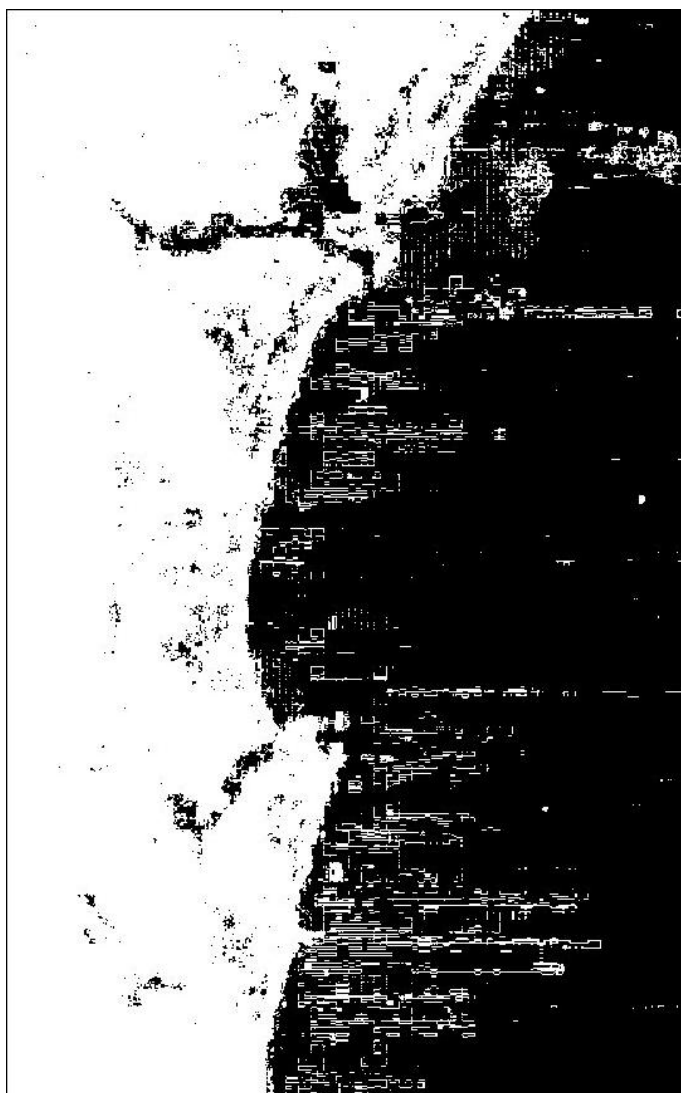


(a) amostra da região de continente



(b) amostra da região oceânica

**Figura 4.16:** *Amostras - Imagem “Continente Verde”*



**Figura 4.17:** Imagem “Continente Verde” segmentada

# Capítulo 5

## Conclusões e Trabalhos Futuros

Neste trabalho foram apresentados dois métodos de segmentação de imagens através do uso de descritores de textura.

A grande quantidade de parâmetros que quando modificados produzem diferentes resultados foi o fator motivante para o desenvolvimento de uma ferramenta que tornasse possível a realização de diferentes combinações desses parâmetros de forma fácil e prática, visando sempre a avaliação e refinamento dos métodos utilizados.

A influência dos classificadores no resultado final da classificação é extremamente grande. Simples alterações na topologia da rede neural, por exemplo, fizeram com que uma imagem fosse ou não corretamente segmentada. O classificador de distância mínima euclidiana não trouxe bons resultados, e consideramos que isso ocorreu pelo fato de este ser simples demais para conseguir representar o espaço de valores utilizado pelos descritores de textura.

O balanceamento da escala de valores de cada característica através do uso da média também foi fator decisivo principalmente quando do uso do método que chamamos de HSVCM (proposto por Dacheng [2]) no qual tínhamos para cada uma das características escalas de valores que eram extremamente distantes e isto fazia com que o peso de cada característica no resultado final fosse desigual.

O processo de segmentação de imagens coloridas apesar de eficiente tem um custo computacional elevado, o que inviabiliza seu uso em algumas aplicações. Um possível trabalho futuro é a adaptação do método feita visando a diminuição do custo computacional. A adaptação de outras características do método GLCM para o método HSVCM é outro trabalho que pode trazer bons resultados.

# Referências Bibliográficas

- [1] KRZESINSKI H. R. BOIKO G. P. KImageProcess: a framework to aid in the image textural features research, janeiro de 2005. Disponível em <http://kimageprocess.sf.net>.
- [2] DACHENG T. et al. A set of novel textural features based on 3d co-occurrence matrix for content-based image retrieval. *ISIF*, páginas 1403–1407, 2002.
- [3] NISSEN S. et al. Fast Artificial Neural Network Library (fann), fevereiro de 2005. Disponível em <http://leenissen.dk/fann/>.
- [4] ROSSUM P. V. et al. Lightweight Neural Network, julho de 2005. Disponível em <http://lwneuralnet.sourceforge.net/>.
- [5] SHANMUGAM K. HARALICK R. M. e DINSTEIN. Textural features for image classification. *IEEE Transactions on Systems, Man and Cybernetics*, 6(3):610–621, novembro de 1973.
- [6] University of Tübingen. Java Neural Network Simulator, março de 2005. Disponível em [http://www-ra.informatik.uni-tuebingen.de/software/JavaNNS/welcome\\_e.html](http://www-ra.informatik.uni-tuebingen.de/software/JavaNNS/welcome_e.html).