



**MURILO AUGUSTO TEIXEIRA BIAGI SILVA**

**USO DE MATRIZES DE CO-OCORRÊNCIAS  
PARA CLASSIFICAÇÃO AUTOMÁTICA DE  
IMAGENS DE CANA- DE-AÇÚCAR.**

**LAVRAS – MG**

**2013**

**MURILO AUGUSTO TEIXEIRA BIAGI SILVA**

**USO DE MATRIZES DE CO-OCORRÊNCIAS PARA CLASSIFICAÇÃO  
AUTOMÁTICA DE IMAGENS DE CANA- DE-AÇÚCAR.**

Monografia apresentada ao Colegiado do Curso de  
Ciência da Computação, para a obtenção do título  
de Bacharel em Ciência da Computação.

Orientador

Prof. Cristiano Leite de Castro

**LAVRAS – MG**

**2013**

MURILO AUGUSTO TEIXEIRA BIAGI SILVA

USO DE MATRIZES DE CO-OCORRÊNCIAS PARA CLASSIFICAÇÃO  
AUTOMÁTICA DE IMAGENS DE CANA- DE-AÇÚCAR.

Monografia apresentada ao Colegiado do Curso de  
Ciência da Computação, para a obtenção do título  
de Bacharel em Ciência da Computação.

APROVADA em 21 de agosto de 2013.

  
Prof. Gabriel Chaves Afonso Coutinho

  
Prof. Wilian Soares Lacerda

UFLA

  
Prof. Cristiano Leite de Castro  
(Orientador)

LAVRAS – MG

2013

*Dedico este trabalho à minha família que sempre me apoiou e acreditou em mim.*

## **AGRADECIMENTOS**

Agradeço à meus pais, Waldemir e Paula, que me incentivaram ao longo de todos esses anos a nunca desistir e fazer sempre aquilo que eu gosto. Agradeço a minha irmã, Eduarda, que sempre acreditou em mim e no meu potencial. Aos meus amigos de minha cidade natal que me ajudaram em tempos difíceis. Aos meus amigos de Lavras também, por estarem sempre me motivando a lutar pelo que eu acredito. Ao meu orientador, por todo o auxílio e orientação que me deu, desde o começo desse trabalho até a sua conclusão.

## RESUMO

Esse trabalho é um estudo e aplicação de uma técnica para extração de características nas imagens de textura de cana, com o objetivo de automatizar o processo de separação da cana nas indústrias de acordo com o tipo de corte (manual ou mecânico) do pedaço que passa pela esteira. É feita uma contextualização da área de processamento de imagens e análise de texturas, além da apresentação de alguns métodos para extração de características. Será utilizado o método conhecido como Grey-Level Co-occurrence Matrix (GLCM) e esse trabalho apresentará as características, além das vantagens e desvantagens do método. Vários testes serão feitos sobre duas bases de imagens diferentes, incluindo a própria base de imagens de cana-de-açúcar e uma análise mais detalhada será sobre os resultados obtidos.

**Palavras-Chave:** Análise de Textura, Extração de Características, Grey-Level Co-occurrence Matrix (GLCM), Métodos Estatísticos, Classificação de Imagens.

## **ABSTRACT**

This work is a study and application of a technique for features extraction on sugarcane texture images in order to automate the process of separation of sugarcane on industries according to the type of cut (manual or mechanic) that passes through the mat. It made a contextualization of the area of image processing and texture analysis, besides the presentation of some methods for feature extraction. Will be used the method known as Grey-Level Co-occurrence Matrix (GLCM) and this work presents the characteristics, besides the benefits and disadvantages of the method. Several tests will be made on two different images databases, including the very basis of sugarcane images and a more detailed analysis will be made on the results.

**Keywords:** Texture Analysis, Feature Extraction, Grey-Level Co-occurrence Matrix (GLCM), Statistical Methods, Image Classification.

## SUMÁRIO

<b>1</b>	<b>Introdução</b>	<b>12</b>
1.1	Justificativa . . . . .	14
1.2	Objetivos . . . . .	15
1.2.1	Objetivo Geral . . . . .	15
1.2.2	Objetivos Específicos . . . . .	15
<b>2</b>	<b>Textura de imagens e suas características</b>	<b>16</b>
2.1	Introdução . . . . .	16
2.2	Análise de Texturas . . . . .	21
2.3	Abordagem Descritiva . . . . .	26
2.3.1	Métodos Estatísticos . . . . .	27
2.3.2	Métodos Espectrais . . . . .	29
2.4	Abordagem Genérica . . . . .	31
2.4.1	Modelos sintáticos . . . . .	31
2.4.2	Modelos probabilísticos . . . . .	32
2.5	Máquinas de Vetores de Suporte . . . . .	32
<b>3</b>	<b>GLCM - Grey-Level Co-occurrence Matrix</b>	<b>35</b>
3.1	Uma breve introdução . . . . .	35
3.2	Variações da GLCM . . . . .	40
3.3	GLCM aplicada em texturas de cana . . . . .	45

<b>4</b>	<b>Metodologia</b>	<b>47</b>
<b>5</b>	<b>Testes e Resultados</b>	<b>54</b>
5.1	Base Outex . . . . .	54
5.2	Base Cana . . . . .	62
<b>6</b>	<b>Conclusão</b>	<b>68</b>

## LISTA DE FIGURAS

1.1	Trabalhador no corte de cana. Retirada de: <a href="http://www.cutsp.org.br">http://www.cutsp.org.br</a> . . .	13
1.2	Máquina para corte de cana. Retirada de: <a href="http://www.esalq.usp.br">http://www.esalq.usp.br</a> . . .	13
2.1	Representação bidimensional de uma imagem digital. Retirada de (QUEIROZ; GOMES, 2006) . . . . .	18
2.2	Exemplo de textura rugosa. Retirada de: <a href="http://pinheiropintor.no.comunidades.net">http://pinheiropintor.no.comunidades.net</a>	19
2.3	Exemplo de textura lisa. Retirada de: <a href="http://fvc pinturas.blogspot.com.br/">http://fvc pinturas.blogspot.com.br/</a>	19
2.4	Etapas do processamento digital de imagens. Retirada de: <a href="http://www.cbpf.br/">http://www.cbpf.br/</a>	21
2.5	Textura com várias regiões. a) Uma imagem consistindo de regiões diferentes de textura. b) A rotulação das regiões de textura, dis- tinguindo cada uma delas. c) O resultado de uma segmentação na imagem que detecta as fronteiras entre os vários tipos de regiões de textura na imagem. Retirada de (CHEN <i>et al.</i> , 1999) . . . . .	23
2.6	Algoritmo para classificação de textura. Adaptado de (SALEM; NASRI, 2010) . . . . .	25
2.7	Estrutura geral da análise de texturas. . . . .	26
2.8	Principais relacionamentos entre vários métodos estatísticos para ex- tração de características. Adaptado de (CHEN <i>et al.</i> , 1999) . . . . .	29
2.9	Funcionamento de uma SVM. Retirada de (GUNN, 1998) . . . . .	33
2.10	Funcionamento de um kernel SVM. Retirada de <a href="http://www.imtech.res.in/">http://www.imtech.res.in/</a>	34
3.1	Processo de construção de uma matriz GLCM a partir da imagem. Re- tirada de: <a href="http://www.mathworks.com/help/images/ref/graycomatrix.html">http://www.mathworks.com/help/images/ref/graycomatrix.html</a>	36

3.2	Representação em alto nível do algoritmo para a técnica GLCM. . . .	40
3.3	Representação da estrutura GLCLL. Retirada de (CLAUSI; ZHAO, 2002) . . . . .	41
3.4	Representação da estrutura GLHS. Retirada de (CLAUSI; ZHAO, 2002)	42
4.1	Exemplos das imagens de corte manual de cana já em níveis de cinza e normalizadas. . . . .	52
4.2	Exemplos das imagens de corte mecânico de cana já em níveis de cinza e normalizadas. . . . .	53
5.1	Gráfico que mostra a média da quantidade de acertos para cada <i>offset</i> usando 4 características sobre a base <i>Outex</i> . . . . .	58
5.2	Gráfico que mostra a média da quantidade de acertos para cada <i>offset</i> usando 8 características sobre a base <i>Outex</i> . . . . .	59
5.3	Gráfico que mostra a média da quantidade de acertos com validação cruzada (VC) sobre a base <i>Outex</i> . . . . .	61
5.4	Gráfico que mostra a média da quantidade de acertos para cada <i>offset</i> usando 4 características sobre a base de Cana. . . . .	64
5.5	Gráfico que mostra a média da quantidade de acertos para cada <i>offset</i> usando 8 características sobre a base de Cana. . . . .	65
5.6	Gráfico que mostra a média da quantidade de acertos com validação cruzada (VC) sobre a base de Cana. . . . .	67

## LISTA DE TABELAS

3.1	Tabela com as estatísticas definidas por Haralick. . . . .	39
3.2	Tabela com as complexidades de tempo da técnica GLCM e algumas de suas variações. . . . .	44
4.1	Exemplo de imagens da base <i>Outex</i> . . . . .	48
5.1	Tabela com as bases utilizadas para os experimentos. . . . .	55
5.2	Tabela com as taxas de acertos para a base <i>Outex</i> , utilizando quatro características. . . . .	56
5.3	Tabela com as taxas de acertos para a base <i>Outex</i> , utilizando oito características. . . . .	57
5.4	Tabela com os resultados usando validação cruzada na base de imagens <i>Outex</i> . . . . .	60
5.5	Tabela com as taxas de acertos para a base de Cana, utilizando quatro características. . . . .	62
5.6	Tabela com as taxas de acertos para a base de Cana, utilizando oito características. . . . .	63
5.7	Tabela com os resultados usando validação cruzada (VC) na base de Cana. . . . .	66

## 1 INTRODUÇÃO

O corte e colheita da cana são atividades comuns e a base para a obtenção de produtos como açúcar e álcool. O corte pode ser de dois tipos: manual e mecânico. No corte manual, é feita a queima da cana para remover folhagem e eliminar insetos peçonhentos. Geralmente a mão de obra é cara, principalmente se não houver a queima da cana antes. Além disso, existem riscos de acidentes, doenças e outros empecilhos, reduzindo a quantidade de mão de obra qualificada que existe. É bem comum os trabalhadores no corte da cana sofrerem cortes e lesões, sem contar as condições precárias de trabalho. A Figura 1.1 mostra o corte manual da cana.

No corte mecânico, uma máquina conhecida como colhedora de cana, faz todo o serviço. Nesse caso, não há necessidade da queima da cana antes. Essa técnica reduz bastante os custos com a produção, mas pelo menos no Brasil, ainda precisa ser muito aprimorada. O grande problema é devido ao estrago e os danos que a máquina causa aos pedaços de cana, pois a máquina não corta de fato a cana e sim, arranca ela do chão, o que pode destruir ou deformar a cana. Em alguns casos, o corte mecanizado pode diminuir a longevidade da cana e diminuir a produtividade. A Figura 1.2 mostra o corte mecânico da cana.

Depois que a cana é cortada, ela é levada para a indústria onde sofrerá o processo de moagem. A máquina responsável por esse processo precisa receber a cana na íntegra e sem deformações, que é o caso da cana cortada pelo homem. No caso particular da cana cortada por máquina, é necessário ter uma pessoa próxima à máquina para “ajeitar” a cana de forma que a moagem não seja prejudicada. Em algumas situações, a cana cortada pela máquina deve passar por todo um pré-processamento antes de ser enviada para a moagem.

De todas as etapas envolvidas na obtenção do açúcar ou do álcool a partir da cana, a que mais interessa é a etapa de moagem. Uma cana bem cortada e livre de



**Figura 1.1:** Trabalhador no corte de cana. Retirada de: <http://www.cutsp.org.br>



**Figura 1.2:** Máquina para corte de cana. Retirada de: <http://www.esalq.usp.br>

impurezas, ao passar pela moagem, terá um rendimento muito maior do que uma cana que não passou por todo um preparo ou já chegou deformada na indústria.

## 1.1 Justificativa

É importante que na indústria de processamento de cana, seja feita a separação entre os dois tipos de cana que existem: cana cortada pelo homem e cana cortada pela máquina.

A cana cortada pelo homem geralmente tem um corte mais correto e menor quantidade de sujeira, devido à queima antes do corte. Logo, não há necessidade de lavagem e nem de "ajeitar" a cana na máquina para um processo de moagem mais eficiente. Por outro lado, a cana cortada pela máquina geralmente vem defeituosa, com sujeiras ou até folhagens e, portanto, deve passar por uma preparação antes de ser enviada para a moagem.

Tendo isso em mente, é de fundamental importância estudar a viabilidade de automatização do processo de separação da cana. A mão de obra humana é mais cara e está sujeita a erros, já que depende do julgamento que a pessoa faz a respeito da cana, apenas pela observação. O que realmente conta é a experiência do indivíduo que está fazendo a separação e com isso existe a possibilidade de erros. Se a pessoa está cansada ou distraída, esse julgamento pode ser incorreto, levando a uma separação incorreta. Em um processo automatizado, falhas desse tipo são mais difíceis de ocorrerem, pois o computador não cansa e seu julgamento é sempre baseado em fatos.

Outra vantagem de se usar um computador para fazer a separação da cana está na segurança. O uso do computador minimiza os riscos de acidentes com pessoas, o que poderia trazer mais prejuízos para a indústria canavieira. O computador pode apresentar problemas com o tempo, mas o preço de uma manutenção pode ser mais barato do que o custo com os funcionários na empresa.

## 1.2 Objetivos

### 1.2.1 Objetivo Geral

O estudo aqui proposto visa analisar imagens de pedaços de cana que passam por uma esteira e são obtidas através de uma câmera. O objetivo é fazer o reconhecimento dessas imagens através de processamento de imagens e visão computacional e determinar qual o tipo de corte que corresponde àquela imagem de cana. Para conseguir isso, será estudada uma técnica de extração de características que permite descrever uma imagem através de detalhes na sua textura. Será avaliado a capacidade de discriminação dessa técnica frente a classificadores baseados em Máquinas de Vetores de Suporte.

### 1.2.2 Objetivos Específicos

- Fazer um estudo do método GLCM (Grey-Level Co-occurrence Matrix), entendendo seus parâmetros e suas principais características.
- Aplicar o método GLCM sobre o base *benchmark Outex*, para avaliar a capacidade de discriminação do método e tentar encontrar o melhor conjunto de características.
- Aplicar o método GLCM e explorar sua eficiência sobre uma base de imagens de cortes de cana.
- A partir dos resultados dos testes, discutir as principais vantagens e desvantagens do método GLCM.

## 2 TEXTURA DE IMAGENS E SUAS CARACTERÍSTICAS

### 2.1 Introdução

Antes de falar sobre a imagem digital e suas características, é necessário criar uma distinção entre uma função contínua e uma função discreta. Uma função contínua está definida em toda a reta do conjunto dos reais, onde para cada valor de  $x$ , existe um único valor de  $f(x)$ . Uma função discreta é uma função definida sobre o conjunto dos inteiros. O computador só armazena bits, um valor que pode ser 0 ou 1, e esses bits podem ser combinados para formar palavras que podem assumir valores maiores.

Assim, um computador sempre trabalha com números inteiros ou uma aproximação dos reais, conhecida como ponto flutuante. Conseqüentemente, não é possível representar uma função contínua em um computador, só é possível fazer uma simulação dela. A função contínua precisa primeiramente ser transformada em uma função discreta e esse processo é chamado de discretização ou digitalização (SCURI, 2002). Nesse processo, valores pontuais de  $x$  são selecionados e os valores de  $f(x)$  correspondentes são armazenados. Um problema desse processo é que geralmente há perda de informações, devido a quantidade de valores que são descartados ao longo do eixo  $x$ .

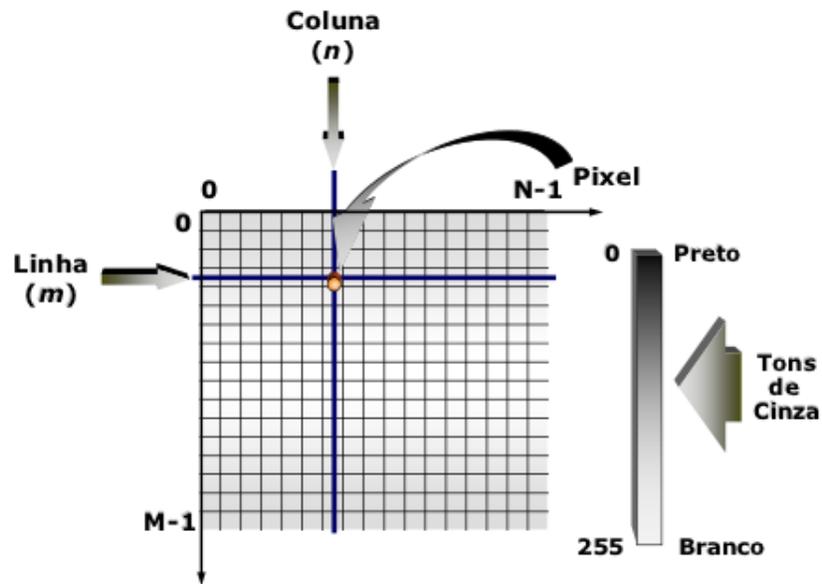
Uma imagem pode ser definida por uma função  $f(x,y)$ , em que  $x$  e  $y$  são coordenadas espaciais no plano, e o valor de  $f$  em qualquer par de coordenadas representa a intensidade ou nível de cinza da imagem nesse ponto. Quando  $x$ ,  $y$  e os valores de intensidade são todos discretos, temos o que chamamos de imagem digital. Uma imagem digital é uma representação numérica de uma imagem que pode ser processada e analisada por um computador (GONZALEZ; WOODS,

1992). Para se obter a imagem digital, deve-se realizar o processo de digitalização como já foi citado anteriormente, e esse processo constitui basicamente de duas partes: amostragem e quantização. A amostragem consiste em discretizar as coordenadas espaciais, enquanto a quantização consiste em amostrar os valores de brilho ou intensidade da imagem.

Toda imagem digital é composta por um conjunto de elementos conhecidos como *pixels* (*picture element*), que é o menor componente da imagem. Cada *pixel* possui uma cor ou uma intensidade dentro da imagem e é a variação nos valores desses *pixels* que gera as regiões que percebemos em uma imagem (GONZALEZ; WOODS, 1992). Uma imagem binária possui dois valores diferentes para os *pixels*, 0 ou 1. Uma imagem em níveis de cinza, possui *pixels* com valores de 0 a 255. Já uma imagem RGB também possui *pixels* variando de 0 a 255, só que em cada uma das componentes, ou seja, para cada *pixel*, há um valor de 0 a 255 em cada uma das três componentes de cor do mesmo (*red, green, blue*).

A Figura 2.1 mostra a representação comum de uma imagem digital. Cada um dos pontos na grade corresponde a um *pixel* da imagem e possui seu próprio valor de intensidade. Para localizar um *pixel*, basta fornecer o índice  $m$  que corresponde a linha e o índice  $n$  que corresponde a coluna do mesmo. A varredura da imagem geralmente é feita da esquerda para a direita e de cima para baixo, que é um padrão conhecido como *raster*. Isso é feito como uma forma de aproveitamento da memória cache do computador na hora de obter cada um dos *pixels* correspondentes a imagem, evitando acessos desnecessários à memória principal. A intensidade de um *pixel* no ponto  $(x,y)$ , pode ser decomposta em: (i) componente de iluminação (quantidade de luz incidente sobre o ponto) e (j) componente de reflectância (quantidade de luz refletida pelo ponto).

Uma imagem digital também possui outras características como a cor, luminosidade, textura, entre outras. O trabalho presente se preocupa com apenas



**Figura 2.1:** Representação bidimensional de uma imagem digital. Retirada de (QUEIROZ; GOMES, 2006)

uma dessas características que é a textura. É comum ouvir falar sobre textura e associá-la com o toque. Geralmente ela está ligada às características de uma dada superfície (se ela é lisa, rugosa, macia, áspera). Segundo Materka, texturas são padrões visuais complexos que são compostos por entidades que a caracterizam, como brilho, cor, tamanho, etc.

A textura pode ser percebida tanto de forma visual quanto de forma tátil. Algumas texturas não possuem qualidades táteis, mas apenas óticas, como as linhas de uma folha de papel. Uma textura rugosa ao toque possui uma grande diferença entre pontos baixos e altos na superfície, ou seja, é possível sentir as elevações na superfície quando ela é tocada. A Figura 2.2 mostra um exemplo de uma textura rugosa. Já uma textura lisa tem poucas diferenças entre pontos altos e baixos, o que passa a sensação de ela ser plana ao toque, como pode ser visto na Figura 2.3 que mostra um exemplo de uma textura lisa.



**Figura 2.2:** Exemplo de textura rugosa. Retirada de: <http://pinheirpintor.no.comunidades.net>



**Figura 2.3:** Exemplo de textura lisa. Retirada de: <http://fvc pinturas.blogspot.com.br/>

As texturas também podem ser classificadas em naturais e artificiais. Uma textura natural é aquela que resulta da intervenção natural do meio ambiente ou que representa os aspectos e formas de coisas que existem na natureza. Uma textura artificial é aquela que resulta da intervenção do homem através do uso de materiais e instrumentos devidamente manipulados.

Texturas de imagens funcionam quase da mesma forma que texturas de objetos reais, sendo que os pontos altos e baixos são valores de luminosidade ou brilho (também chamados de níveis de cinza). Quando uma textura de imagem

é decomposta, ela possui duas dimensões básicas sobre as quais ela é descrita. A primeira dimensão é para descrever as primitivas que compõem a imagem e a segunda dimensão se concentra na organização espacial das primitivas chamadas de *tonal*. Primitivas *tonal* são regiões na textura com propriedades *tonal* (HARALICK, 1979). Assim, uma imagem de textura pode ser descrita pelo número e tipo de suas primitivas e pela organização espacial que existe entre elas.

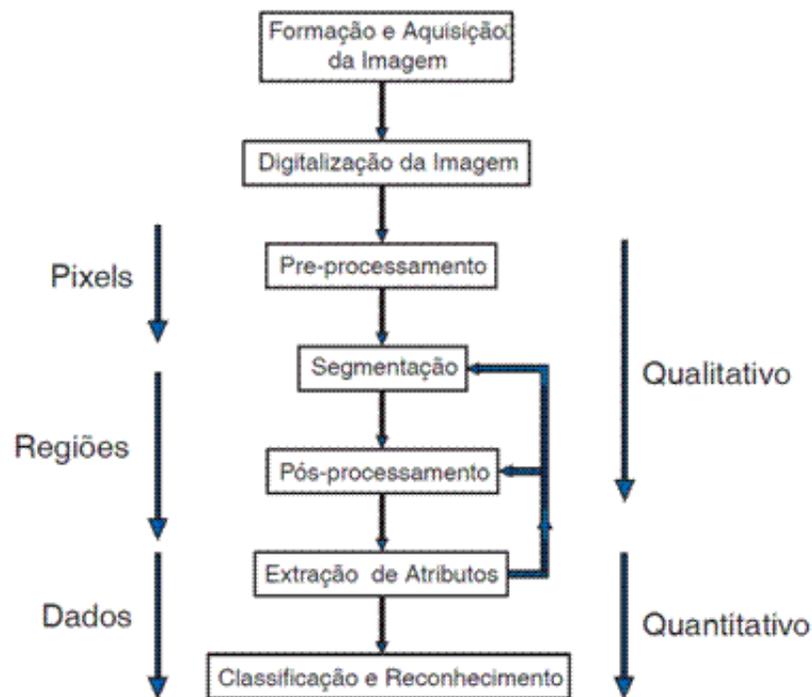
Uma textura de imagem possui uma série de qualidades visíveis que desempenham um papel fundamental na distinção das texturas. Várias características são usadas para descrever uma textura, tais como finura, grossura, suavidade, granulação, entre outras (HARALICK, 1979). Cada uma dessas características se traduz em uma propriedade das primitivas *tonal* e as interações espaciais entre as primitivas. Existe uma relação entre os padrões de textura e de tom, que diz que quando uma área da imagem tem uma pequena variação de primitivas *tonal*, a propriedade dominante dessa área é o tom e quando tiver uma grande variação de primitivas, a propriedade dominante é a textura.

Textura de imagem é útil em uma série de aplicações e tem sido alvo de muitas pesquisas ao longo dos anos (CHEN *et al.*, 1999). Uma aplicação comum é reconhecer regiões dentro de uma imagem usando as características da textura.

A textura é a dica visual mais importante para se perceber regiões homogêneas dentro de uma imagem. Em uma imagem, a textura pode ser vista como o padrão de cores que cobre sua superfície, além de outras características intrínsecas na imagem, como o brilho e a distribuição de formas. O olho humano consegue reconhecer formas e padrões facilmente, assim podemos usar essa capacidade para distinguir imagens diferentes. Já um computador precisa de um conjunto de regras que permita simular a visão humana e a capacidade de detectar os padrões nos objetos que estão sendo visualizados.

## 2.2 Análise de Texturas

O processamento digital de imagens é um campo muito vasto e cheio de sub-áreas que vem sendo assunto de muitas pesquisas ao longo dos anos e tem trazido muitas contribuições nas mais diversas áreas. De uma forma resumida, o processamento digital de imagens se refere ao processamento de imagens por um computador digital.



**Figura 2.4:** Etapas do processamento digital de imagens. Retirada de: <http://www.cbpf.br/>

Esse processamento envolve uma série de atividades, como mostra a Figura 2.4. Tudo começa com a aquisição da imagem através de algum dispositivo de captura (câmeras, satélites, sensores de infra-vermelho, etc.). Depois a imagem passa por um pré-processamento, onde filtros são aplicados para criar uma representação

mais apropriada da imagem e mais fácil para o computador trabalhar. Um exemplo seria a aplicação de filtros para eliminar os ruídos e as distorções geométricas na imagem.

A próxima etapa é a análise da imagem, que envolve extrair características ou atributos que representem bem a imagem (CHEN *et al.*, 1999). Em seguida, objetos precisam ser separados do fundo e destacados através de um processo de segmentação. Por fim, vem a etapa de classificação que é considerada a de mais alto nível e envolve reconhecer, verificar ou inferir a identidade dos objetos na imagem a partir das características e representações obtidas pelas etapas explicadas anteriormente (QUEIROZ; GOMES, 2006).

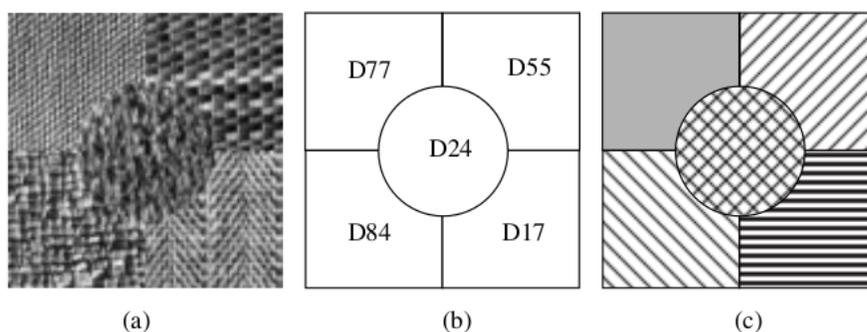
Não serão explicados em detalhes cada uma dessas etapas, mas algumas serão mais enfatizadas, pois são necessárias no presente trabalho, começando com a etapa de análise ou mais especificamente, a análise de texturas de uma imagem.

A análise de texturas é uma área que vem crescendo bastante e que possui uma grande variedade de aplicações. Ela procura derivar uma descrição eficiente de uma dada imagem a partir de um conjunto de características que representam essa imagem, permitindo que operações matemáticas sejam aplicadas sobre essa descrição, para comparar e transformar texturas (ZHANG; TAN, 2002).

É possível também encontrar os contornos de uma textura, mesmo que não seja feita a classificação da mesma. Esse é um tipo de problema que os pesquisadores tem focado muito e é conhecido como segmentação de textura (ZHANG, 2001). O objetivo é encontrar um mapa de contorno que separa visualmente cada uma das regiões homogêneas na textura. Um exemplo de aplicação da segmentação é o reconhecimento de faces humanas.

A Figura 2.5 mostra três situações. Na primeira, temos a imagem original composta por cinco regiões diferentes de textura. Observe que é possível perceber visualmente aonde termina uma textura e aonde começa outra. Na segunda, temos

a classificação dessas regiões, dando um rótulo para cada uma delas. Na terceira, temos a segmentação, destacando as fronteiras que dividem essas cinco regiões de textura.



**Figura 2.5:** Textura com várias regiões. a) Uma imagem consistindo de regiões diferentes de textura. b) A rotulação das regiões de textura, distinguindo cada uma delas. c) O resultado de uma segmentação na imagem que detecta as fronteiras entre os vários tipos de regiões de textura na imagem. Retirada de (CHEN *et al.*, 1999)

Em algumas aplicações as texturas já possuem um papel fundamental, como é o caso do sensoriamento remoto. Em outros casos, como a inspeção de superfície, novas aplicações são descobertas e estudadas conforme as técnicas são aprimoradas. Algumas das aplicações mais conhecidas para a textura são: inspeção automatizada, processamento de imagens médicas, processamento de documentos e o próprio sensoriamento remoto (CHEN *et al.*, 1999). A inspeção procura detectar defeitos em imagens têxteis e também desgaste nas pinturas de carpetes e carros, por exemplo.

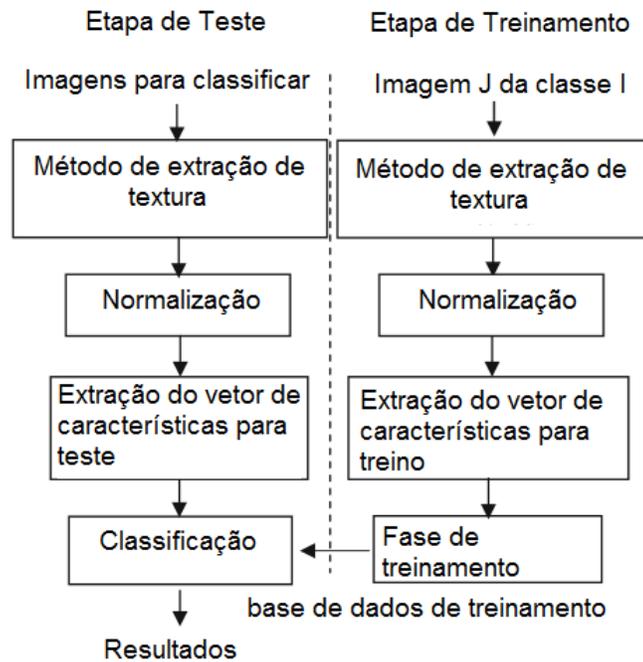
Para as imagens médicas, as aplicações envolvem extração automática de características que são usadas em várias tarefas de classificação, como distinguir um tecido normal de um anormal. Outra área em que as texturas são muito utilizadas é a análise de imagens de documentos e o reconhecimento ótico de caracteres. Por fim temos o sensoriamento remoto, que envolve a análise do uso de terras, onde regiões homogêneas com tipos diferentes de terreno precisam ser classificadas.

Análise de texturas é dividida em dois estágios. O primeiro estágio é o de extração de características (*feature extraction*), que tem como objetivo caracterizar uma textura com base nas medidas das características (*features*) obtidas (GIMEL'FARB; ZHOU, 2008). É importante selecionar *features* que não variam com transformações nas imagens, como translação, rotação e escala (CHORAS, 2007). A *feature* é definida como uma função de uma ou mais medidas, sendo que cada uma descreve uma propriedade quantificável de um objeto.

Deve-se notar também que as características que descrevem uma imagem não precisam necessariamente serem obtidas a partir da textura da imagem. Existem outras fontes na imagem que permitem a análise e obtenção dessas características, como as próprias regiões homogêneas ou as bordas que as separam e em alguns casos, até as cores na imagem podem ser utilizadas como forma de representação da mesma. Um detalhe importante é saber selecionar as melhores características para descrever uma imagem, que é algo que depende do tipo de imagem analisada.

O segundo estágio é o de classificação de textura, no qual um classificador é desenvolvido para determinar a classe que cada textura de entrada pertence, usando como base o valor das *features* extraídas no primeiro estágio (ZHANG; TAN, 2002). Para fazer essa classificação, podem ser usados classificadores de padrões conhecidos, como os classificadores Bayesianos, as *Support Vector Machine* (GUNN, 1998) ou as Redes Neurais Artificiais.

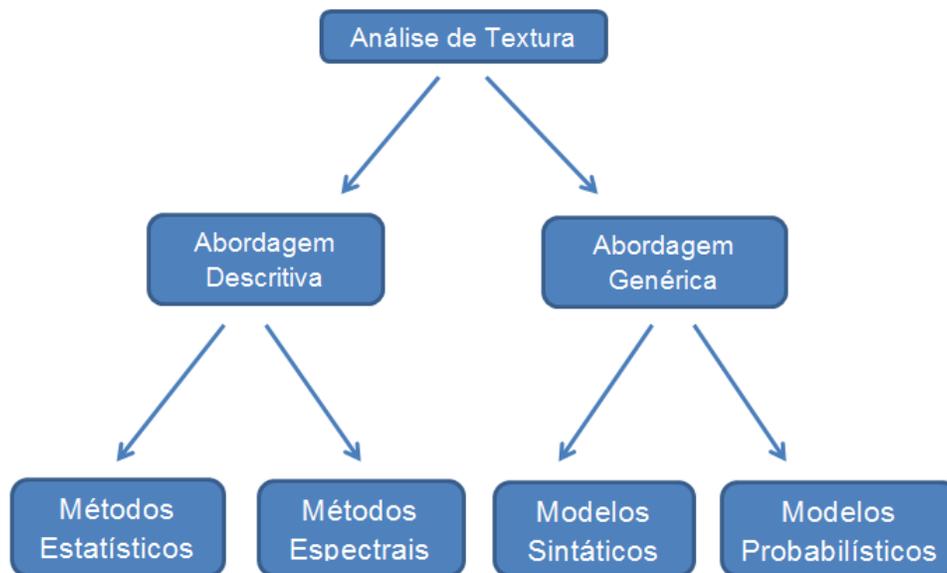
Uma das vantagens da análise de texturas é que ela consegue representar uma textura complexa como um pequeno conjunto de características, diminuindo assim as dimensões envolvidas e permitindo um processamento mais rápido e automático das texturas. Uma quantidade de características muito grande ou características não muito representativas podem piorar a representação da textura e gerar classificações incorretas.



**Figura 2.6:** Algoritmo para classificação de textura. Adaptado de (SALEM; NASRI, 2010)

Existem vários métodos desenvolvidos para análise de texturas, mas no geral, existem duas abordagens mais utilizadas: abordagem descritiva e abordagem genérica (GIMEL'FARB; ZHOU, 2008). A Figura 2.6 demonstra cada uma das etapas do processo de análise de textura. Nela, pode-se ver que existe uma separação entre a fase de treinamento e a fase de teste, pois uma parte das imagens da base de dados é usada para treinar o classificador e a outra parte é usada para testes. Essa separação é um requisito do aprendizado supervisionado e será utilizado no presente trabalho.

O fluxograma mostrado na Figura 2.7 contém um resumo de cada um dos tópicos que serão abordados nas seções seguintes, para facilitar o entendimento sobre análise de textura, que envolve uma grande quantidade de abordagens e métodos.



**Figura 2.7:** Estrutura geral da análise de texturas.

## 2.3 Abordagem Descritiva

A abordagem descritiva gera uma descrição quantitativa de uma textura com base em um conjunto de características (*features*) mensuráveis. Nesse caso, uma textura é quantificada por um vetor de características, sendo que cada característica tem o seu próprio valor correspondente (sua medida). O valor ou medida de cada característica representa o quanto ela está presente na textura.

É importante saber quais e quantas características são mais relevantes para um dado problema, pois são essas características que irão representar a textura (HARALICK; SHANMUGAM; DINSTEIN, 1973). Um conjunto muito pequeno de características pode gerar uma representação insuficiente, invalidando os resultados obtidos na classificação dos tipos de textura. Um conjunto muito grande de características exige uma quantidade maior de cálculos a serem feitos e uma demanda maior de tempo, mas pode gerar resultados mais precisos.

Não existe uma regra geral que determina quais as melhores características, pois como já foi comentado, tudo depende do tipo de textura em questão. Um dado conjunto pode funcionar bem para um caso, mas pode não ser muito útil para outro caso. O desafio, portanto, consiste em encontrar o conjunto ótimo que representa a textura da melhor forma possível. No geral, é recomendado não utilizar uma quantidade muito grande de características para descrever as texturas (HARALICK; SHANMUGAM; DINSTEN, 1973). Existem muitas características definidas na literatura e cada uma delas possui um significado diferente para a imagem em si.

Definido o conjunto de características a ser utilizado, é necessário aplicar um método para extrair essas características. Os métodos para extração de características podem ser divididos em dois grupos: métodos estatísticos e métodos espectrais (MATERKA; STRZELECK, 1998). Dependendo do tipo de método utilizado, as características são extraídas de fontes diferentes como será visto, mas de uma forma geral, todos os métodos usam como entrada uma representação digital da imagem.

### **2.3.1 Métodos Estatísticos**

Na abordagem estatística, uma textura é definida por estatísticas extraídas de um conjunto de propriedades locais da imagem (GIMEL'FARB; ZHOU, 2008). Os métodos estatísticos coletam sinais estatísticos da imagem como descritores para a mesma. Esses métodos analisam a distribuição espacial dos níveis de cinza, calculando características (*features*) em cada *pixel* da imagem. A distribuição e organização espacial dos níveis de cinza em uma imagem é uma das formas de se definir a textura da mesma.

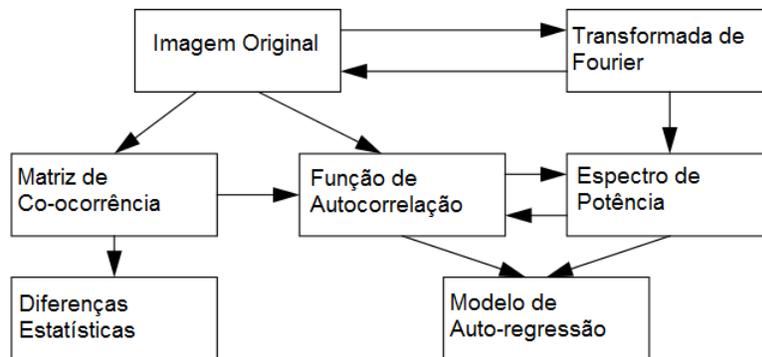
As estatísticas podem ser de primeira ordem, segunda ordem ou ordens superiores. Estatísticas de primeira ordem, tais como média e o desvio padrão, descre-

vem as propriedades de apenas um *pixel*, sem se preocupar com o relacionamento entre os *pixels* na imagem (SRINIVASAN; SHOBHA, 2008).

Dentre as estatísticas de primeira ordem, podemos citar os histogramas de níveis de intensidade. Assumindo que a imagem é representada por uma função  $f(x,y)$  de duas variáveis  $x$  e  $y$ , os seus valores são números discretos que representam a intensidade do *pixel* em um ponto  $(x,y)$  da imagem. O histograma de níveis de intensidade é uma função que mostra o número de *pixels* para cada nível de intensidade ao longo de toda a imagem (KIM *et al.*, 2013). Esse histograma contém um resumo de toda a informação estatística sobre a imagem e analisando a sua forma é possível obter várias informações sobre as características da imagem. Além disso, é possível extrair *features* a partir dos histogramas para descrever a imagem, tais como a média e a variância.

Estatísticas de segunda ordem, tais como a GLCM, diferenças de níveis de cinza e LBP (*Local Binary Pattern*), se preocupam com o relacionamento espacial entre dois *pixels* ou a co-ocorrência entre eles. Assim, uma imagem que contenha uma mesma quantidade de diferentes níveis de cinza que outra, diferenciando-se somente em termos do posicionamento dos mesmos, irá apresentar, conforme a abordagem de primeira ordem, um mesmo histograma (GIMEL'FARB; ZHOU, 2008). Utilizando estatísticas de segunda ordem, é possível diferenciar essas imagens. A Figura 2.8 mostra o relacionamento entre vários dos métodos estatísticos conhecidos.

O método GLCM, a ser detalhado mais adiante, consiste em uma matriz que representa a quantidade de co-ocorrências entre certos valores de níveis de cinza em uma imagem, dado um relacionamento espacial entre os *pixels* (uma distância e um ângulo). Foi uma das primeiras técnicas de extração de características de segunda ordem desenvolvidas e serviu como base para o surgimento de muitas outras.



**Figura 2.8:** Principais relacionamentos entre vários métodos estatísticos para extração de características. Adaptado de (CHEN *et al.*, 1999)

A técnica LBP foi descrita pela primeira vez em 1994. É totalmente invariante às mudanças monotônicas de níveis de cinza, como variações na luminosidade, além de ser uma técnica computacionalmente simples, o que permite sua aplicação em sistemas de tempo real (HEIKKILÄ; PIETIKÄINEN; SCHMID, 2009).

Existem outras técnicas estatísticas de segunda ordem que não serão descritas aqui para não fugir do foco do presente trabalho. As estatísticas de ordem superiores não são muito usadas na prática, devido à alta complexidade computacional envolvida. Existem alguns estudos que tentam viabilizar o uso dessas estatísticas de ordens maiores do que dois, mas em problemas reais elas não são utilizadas.

### 2.3.2 Métodos Espectrais

Os métodos espectrais envolvem o domínio da frequência na imagem, onde as características estão ligadas às estatísticas obtidas pelos filtros aplicados (GIMEL'FARB; ZHOU, 2008). Extração de características sobre o domínio de frequência tem várias vantagens. Primeiro, um filtro sobre a imagem é seletivo, ele destaca algumas características enquanto suprime outras. Segundo, a estrutura periódica da

textura na imagem pode ser representada em certa frequência dentro do domínio espectral.

Modelos dessa abordagem usam banco de filtros que fazem a transformação do domínio espacial no domínio da frequência e vice-versa (GIMEL'FARB; ZHOU, 2008). Um banco de filtros é composto por vários filtros paralelos, cada um ajustado com uma dada frequência espacial, orientação ou escala. Assim como nos métodos estatísticos, as medidas das *features* obtidas podem ser usadas para descrever a textura da imagem e usadas como entrada para classificação ou segmentação.

Um dos problemas dos métodos espectrais é em relação a seleção dos filtros, que geralmente é feita de forma heurística, sem garantias de que vão gerar resultados bons. Assim, a descrição da imagem obtida pode misturar tanto *features* importantes quanto *features* irrelevantes. Alguns pesquisadores tem procurado estabelecer o papel e importância dos bancos de filtros na análise de texturas.

Vários métodos espectrais têm sido desenvolvidos, tais como Wavelets (WOUWER; SCHEUNDERS; DYCK, 1999), transformada de Fourier (TIMM; MARTINETZ, 2010), filtros de Gabor, entre outros. As wavelets são funções matemáticas que separam os dados em diferentes componentes de frequência e estudam cada um dos componentes com uma resolução combinada a sua escala. As wavelets foram aplicadas com sucesso na compressão, melhoria, análise, classificação, caracterização e recuperação de imagens.

As transformadas de Fourier permitem decompor um sinal em seus componentes de frequência (senos e cossenos). Ela primeiro translada uma função no domínio do tempo para uma função no domínio da frequência. O sinal pode então ser analisado pela frequência, pois os coeficientes da transformada representam a contribuição dos senos e cossenos em cada uma das frequências.

Tanto wavelets quanto transformadas de Fourier podem ser vistas como uma modificação do espaço de funções para um domínio diferente. Para as transformadas de Fourier, o novo domínio contém funções bases que são senos e cossenos. Para as wavelets, o novo domínio contém funções mais complexas chamadas wavelets. Além desses métodos, existem muitos outros na literatura, mas eles não serão explicados e nem detalhados no presente trabalho.

## **2.4 Abordagem Genérica**

Uma abordagem genérica cria um modelo probabilístico ou geométrico para descrição da textura. Dois tipos de abordagens são os modelos sintáticos e os modelos probabilísticos.

### **2.4.1 Modelos sintáticos**

Modelos sintáticos analisam a geometria das texturas através de técnicas analíticas espaciais. Análise fractal e abordagem estrutural são dois métodos conhecidos para criar modelos sintáticos para texturas (LU; FU, 1978).

Um fractal consegue medir a complexidade geométrica de uma textura, o que pode ser usado para descrever vários padrões na mesma (GIMEL'FARB; ZHOU, 2008). Como a maioria das texturas envolvem padrões com certo grau de auto similaridade, métodos fractais permitem medir esses padrões para gerar descritores de textura.

A abordagem estrutural procura derivar representações geométricas baseadas na ideia da textura como uma organização espacial de elementos (GIMEL'FARB; ZHOU, 2008). A estrutura e organização dos elementos é que define os modelos que serão gerados usando essa abordagem. A organização espacial dos elementos

de textura é definida por um conjunto de regras de posicionamento (*placement rules*).

## 2.4.2 Modelos probabilísticos

Modelos probabilísticos tentam generalizar a abordagem de descrição, derivando um modelo a partir da distribuição conjunta das características selecionadas da imagem.

Um modelo de probabilidade assume que as texturas são geradas por um processo aleatório e descreve a textura através do modelo de um processo gerador (BRADY; JERMYN; ZERUBIA, 2003). Existem duas etapas para gerar um modelo de probabilidade, chamadas de criação e identificação. A criação envolve selecionar as características da textura para o modelo e criar funções de modelo paramétricas. Identificação envolve estimar parâmetros de modelos desconhecidos para uma certa textura. Essa última etapa é resolvida usando técnicas de inferência estatísticas.

Um dos modelos de probabilidade mais bem sucedidos são os campos aleatórios de Markov-Gibbs (BOYKOV; VEKSLER; ZABIH, 1998). Outro desenvolvimento recente de modelos de probabilidade é a combinação da teoria de filtros com a modelagem de textura.

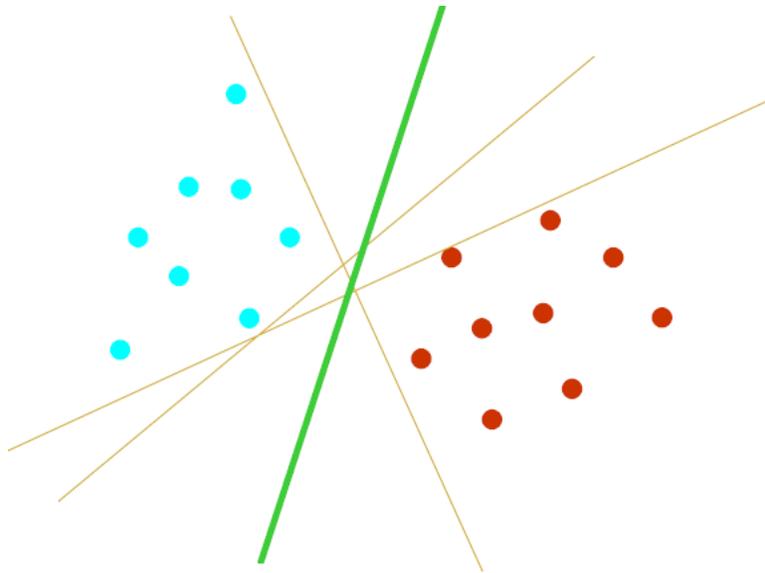
## 2.5 Máquinas de Vetores de Suporte

A partir do conjunto de características obtidos para cada imagem, é possível criar uma matriz de treinamento que será usada como entrada para o classificador. Neste trabalho, será utilizado o classificador conhecido como *Support Vector Machine* (SVM) (GUNN, 1998). Uma SVM é uma técnica de aprendizado de máquina muito utilizada em aplicações de classificação e regressão. Uma tarefa de

classificação envolve separar dados em conjunto de treinamento e de teste. Cada instância no conjunto de treino possui um valor alvo, que seria sua classe ou *label*, além de vários atributos que ajudam a distinguir uma instância de outra. O objetivo da SVM é produzir um modelo que prevê os valores-alvo dos dados de teste apenas utilizando seus atributos.

Dado um conjunto de treinamento formado por pares de instâncias  $(x_i, y_i), i = 1, \dots, l$ , onde  $x_i \in R^n$  e  $y \in \{1, -1\}^l$ , uma SVM procura uma solução ótima que minimize o valor de:  $\frac{1}{2}w^T w + C \sum_{i=1}^l \xi$ .

Observe que um dos parâmetros presentes na fórmula é o  $C$  ou *BoxConstraint*. Esse parâmetro impede o custo da função de tender a menos infinito.

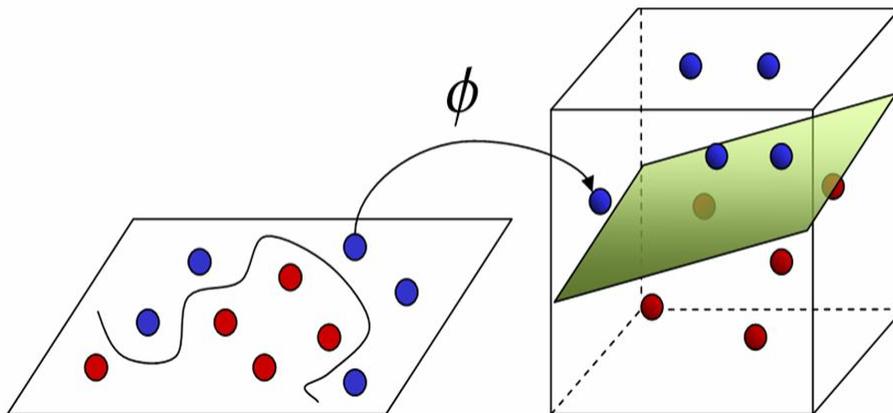


**Figura 2.9:** Funcionamento de uma SVM. Retirada de (GUNN, 1998)

A SVM cria uma superfície de separação, tal que a margem de separação entre as duas classes seja máxima. A construção dessa superfície é feita a partir dos vetores de suporte, que são estruturas encontradas durante a fase de treinamento. A Figura 2.9 mostra um exemplo onde a SVM encontra a superfície ótima de

separação entre duas classes. Observe que a distância da superfície até os dados mais próximos de cada uma das classes é praticamente a mesma.

É muito comum problemas de classificação envolverem dados que não são linearmente separáveis e uma SVM é capaz de lidar com problemas desse tipo, através de funções auxiliares conhecidas como *kernels*, que mapeiam os dados de entrada para um espaço dimensional maior. Os *kernels* mais utilizados são os Polinomiais, os Gaussianos ou RBF (*Radial Basis Function*) e os Sigmoidais e para cada um deles, existe um conjunto de parâmetros que deve ser especificado. A Figura 2.10 mostra como uma função de kernel converte os dados do espaço de entrada para um espaço de maior dimensão onde os dados se tornam linearmente separáveis.



**Figura 2.10:** Funcionamento de um kernel SVM. Retirada de <http://www.imtech.res.in/>

Uma das vantagens da SVM é a possibilidade de realizar validação cruzada, onde os melhores parâmetros, dependendo do tipo de *kernel*, são encontrados de forma empírica. Assim, não há necessidade do usuário testar manualmente cada combinação de valores possíveis para os parâmetros. A SVM geralmente possui uma boa capacidade de generalização e um tempo menor de execução que algumas outras técnicas de aprendizagem, por isso ela foi escolhida para este trabalho.

## 3 GLCM - GREY-LEVEL CO-OCCURRENCE MATRIX

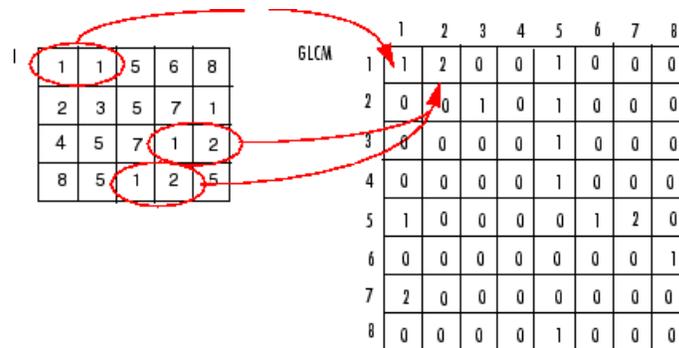
### 3.1 Uma breve introdução

A GLCM (Grey-Level Co-occurrence Matrix) é uma técnica utilizada dentro da área de análise de texturas, que foi desenvolvida na década de 70 pelo pesquisador Robert M. Haralick (HARALICK; SHANMUGAM; DINSTEIN, 1973). É um método estatístico para extração de características que usa estatísticas de segunda ordem, pois analisa as co-ocorrências existentes entre pares de *pixels*, ou seja, ela não analisa cada *pixel* individualmente mas sim, conjuntos de *pixels* relacionados através de algum padrão.

Uma GLCM é uma matriz sempre quadrada que mantém informações das intensidades relativas dos *pixels* em uma dada imagem. Ela calcula as probabilidades de co-ocorrências entre dois níveis de cinza  $i$  e  $j$ , dada uma certa distância  $s$  (também conhecida como *pps* ou *pixel pair spacing*) e uma orientação  $\Theta$  (tetha), que pode assumir os valores de  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$  e  $135^\circ$  (BARALDI; PARMIGGIANI, 1995). Existe uma matriz de co-ocorrência para cada relacionamento espacial possível (combinações da distância  $s$  e da orientação  $\Theta$ ). Todas as informações sobre a textura de uma imagem vão estar contidas nessa matriz.

Um elemento da matriz GLCM contém uma probabilidade, ou seja, a frequência com que um dado *pixel* de nível  $i$  está na vizinhança de um *pixel* de nível  $j$ . Um cálculo simples para essa probabilidade seria o número de vezes com que tal co-ocorrência acontece dividido pela número total de combinações possíveis (HARALICK, 1979). É possível estabelecer o relacionamento entre três ou mais *pixels*, só que não é muito eficiente em termos computacionais, portanto, essa abordagem não é utilizada na prática.

Um exemplo de uma matriz GLCM obtida a partir dos *pixels* de uma imagem pode ser visto na Figura 3.1. Observe que o relacionamento estabelecido é (1,1), ou seja, um *pixel* adjacente horizontalmente a uma distância unitária. O elemento na linha 1 e coluna 1 da matriz GLCM tem o valor 1, o que indica que existe apenas uma única situação na imagem onde um *pixel* de nível de intensidade 1 é adjacente horizontalmente a um outro *pixel* de nível de intensidade 1. O elemento na linha 1 e coluna 2 da matriz GLCM tem o valor 2, o que indica que existem duas ocorrências na imagem onde um *pixel* de nível de intensidade 1 é adjacente horizontalmente a um *pixel* de nível de intensidade 2. Esse processo é repetido até que a matriz GLCM esteja completa.



**Figura 3.1:** Processo de construção de uma matriz GLCM a partir da imagem. Retirada de: <http://www.mathworks.com/help/images/ref/graycomatrix.html>

Uma análise computacional da obtenção da matriz GLCM a partir da representação matricial da imagem mostra que o relacionamento escolhido (*offsets*) não interfere na complexidade do algoritmo. Para um dado *offset*, a imagem deve ser percorrida *pixel a pixel* para verificar se algum *pixel* possui o relacionamento estabelecido e incrementar essa quantidade na matriz GLCM. Logo, se imagem possui  $N$  *pixels*, a complexidade será  $O(N)$ , independente do *offset* que foi escolhido.

Assim, uma outra forma de entender um elemento da matriz GLCM, é pensar nele como a quantidade de vezes que um *pixel* de nível  $i$  aparece a uma distância  $s$  e a uma orientação  $\Theta$  de um *pixel* de nível  $j$ . Tanto o  $i$  quanto o  $j$  variam de

zero até a quantidade de níveis de cinza, também chamada de quantização. Essas quantidades devem ser transformadas em probabilidades antes de se fazer o cálculo das características. O processo de transformar uma matriz GLCM em uma aproximação de uma tabela de probabilidades é chamado de normalização (HARALICK, 1979). Este processo é apenas uma aproximação, pois uma verdadeira probabilidade só trabalha com valores contínuos e os níveis de cinza são valores inteiros, portanto discretos.

É importante notar que a imagem a partir da qual será obtida a matriz GLCM, deve ser primeiramente convertida para uma imagem em nível de cinza. Uma imagem em nível de cinza, geralmente é composta por tons de cinza, variando entre o preto com a menor intensidade e o branco com a maior intensidade. O tamanho da matriz GLCM depende da quantidade de níveis de cinza. Ela possui um número de linhas e um número de colunas igual a quantidade de níveis de cinza.

Em relação a quantidade de níveis de cinza, isso varia muito de uma imagem para outra. Uma imagem binária possui dois níveis de cinza (0 e 1). Uma imagem em níveis de cinza, pode possuir quaisquer quantidade de níveis de cinza variando de 0 a 255. Geralmente são utilizados 4 ou 8 níveis de cinza, mas esse valor pode variar dependendo da situação.

O algoritmo para geração de uma matriz GLCM a partir de uma imagem em nível de cinza, faz uma varredura na matriz que representa a imagem e determina, para cada combinação de níveis  $i$  e  $j$ , a quantidade de co-ocorrências que existem entre eles, usando o relacionamento espacial definido como base.

Uma vez que a matriz é preenchida com base em uma imagem já convertida para níveis de cinza, a técnica aplica estatísticas ao longo de toda a matriz para calcular as características mais relevantes da textura. Uma das abordagens é percorrer a matriz da esquerda para a direita e de cima para baixo, analisando elemento por

elemento, independente do elemento ser zero ou não. Para cada um dos elementos, é aplicado estatísticas que gerarão as características de interesse.

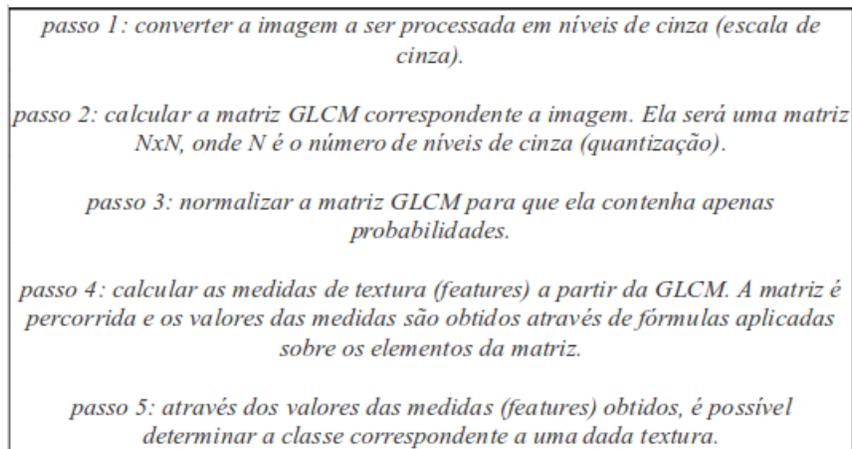
Haralick estabeleceu 14 características significativas sendo que a quantidade de características usadas em um problema varia de acordo com as especificações do mesmo (HARALICK; SHANMUGAM; DINSTEIN, 1973). Nem sempre será necessário a utilização de todas essas características, o que na verdade, pode piorar o desempenho do método ao invés de melhorar. O que se deve ter em mente é que o conjunto de características que será escolhido para representar a imagem tem que ser preciso o suficiente para que seja possível diferenciar, por exemplo, duas imagens apenas pelos padrões que existem na superfície de suas texturas. Na Tabela 3.1, são mostradas as principais características definidas por Haralick.  $G$  é a quantidade de níveis de cinza na imagem.  $C_{ij}$  é a entrada da matriz GLCM na linha  $i$  e coluna  $j$ . O  $p_x(i)$  corresponde a  $i$ -ésima entrada na matriz de probabilidades obtida somando os elementos na linha de  $p(i, j)$ . Note também que  $(\mu_x, \mu_y)$  e  $(\sigma_x, \sigma_y)$  representam média e desvio padrão, respectivamente, para a linha  $i$  e coluna  $j$  da matriz GLCM.

Tendo calculado todas as características para cada *pixel* da imagem, usando a matriz GLCM, essas características serão utilizadas como uma forma de descrever aquela imagem. A partir da descrição dessa imagem, é possível classificá-la dentro de um conjunto, que é a etapa final do processo de análise de texturas, ou seja, a etapa de classificação.

Uma representação do algoritmo em alto nível para a técnica GLCM pode ser vista na Figura 3.2. O primeiro passo descreve o pré-processamento que deve ser feito com a imagem. O segundo passo é o cálculo da matriz GLCM, que depende da imagem de entrada e dos parâmetros  $s$  e  $\Theta$ . Lembrando que a quantidade de matrizes GLCM depende da quantidade de relacionamentos utilizado no método. O terceiro passo é a normalização da matriz, como já foi discutido antes. O quarto

**Tabela 3.1:** Tabela com as estatísticas definidas por Haralick.

Estatística	Equação
Dissimilarity	$\sum_{i,j=1}^G C_{ij}  i-j $
Uniformity	$\sum_{i,j=1}^G C_{ij}^2$
Entropy	$-\sum_{i,j=1}^G C_{ij} \log C_{ij}$
Contrast	$\sum_{i,j=1}^G C_{ij} (i-j)^2$
Correlation	$\sum_{i,j=1}^G \frac{(i-\mu_x)(j-\mu_y)C_{ij}}{(\sigma_x)(\sigma_y)}$
Variance	$\sum_{i,j=1}^G (i-\mu)^2 C_{ij}$
Inverse Difference Moment	$\sum_{i,j=1}^G \frac{1}{1+(i-j)^2} C_{ij}$
Sum Average	$\sum_{i=2}^{2G} i p_{x+y}(i)$
Sum Variance	$\sum_{i=2}^{2G} (i - \text{Sum Entropy})^2 p_{x+y}(i)$
Sum Entropy	$-\sum_{i=2}^{2G} p_{x+y}(i) \log p_{x+y}(i)$
Difference Variance	variância de $p_{x-y}$
Difference Entropy	$-\sum_{i=0}^{G-1} p_{x-y}(i) \log p_{x-y}(i)$
Maximal Correlation Coefficient	$\sum_k \frac{p(i,k)p(j,k)}{p_x(i)p_y(k)}$



**Figura 3.2:** Representação em alto nível do algoritmo para a técnica GLCM.

passo é o cálculo das características a partir da matriz GLCM. O quinto passo é a utilização do vetor de características gerado para tentar determinar a classe de cada uma das imagens do problema.

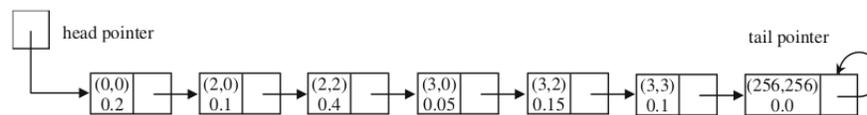
## 3.2 Variações da GLCM

Um dos principais problemas relacionados à técnica GLCM é o fato da matriz obtida a partir da imagem ser esparsa, ou seja, conter uma quantidade grande de zeros. Durante o cálculo das características, algumas fórmulas são aplicadas sobre as probabilidades definidas por cada elemento da matriz. Se uma probabilidade é zero, o resultado da aplicação da fórmula também será zero, o que não gera informação útil (CLAUSI; ZHAO, 2002). Levando em consideração que a quantidade de níveis de cinza pode ser muito grande, o tamanho da matriz, conseqüentemente, poderá ser muito grande também, levando a um número alto de computações e uma alta demanda de memória.

Algumas soluções foram implementadas para esse problema, levando ao desenvolvimento de algumas variações da técnica GLCM. Uma primeira ideia seria

criar uma estrutura híbrida de lista ligada e tabela *hash*, gerando a técnica conhecida como GLCHS (gray level co-occurrence hybrid structure). Nessa técnica, as probabilidades de co-ocorrência diferentes de zero são mantidas em uma lista ligada que está sempre ordenada, que é conhecida como GLCLL (gray level co-occurrence linked list) (CLAUSI; ZHAO, 2002).

Na GLCLL, a lista é mantida ordenada, para se obter um acesso mais eficiente aos pares de co-ocorrências. Como o acesso a uma lista geralmente toma um tempo  $O(n)$ , sendo  $n$  igual ao número de elementos na lista, uma boa solução seria usar uma estrutura de tabela *hash* que indexa cada elemento da lista, permitindo um acesso quase instantâneo aos mesmos. Assim, é usada a técnica conhecida como GLCHS que mistura tabela *hash* com lista encadeada. Esse método não exige ordenação da lista encadeada, pois o acesso aos seus elementos é feito diretamente através do *hash*. A Figura 3.3 mostra uma representação da GLCLL.

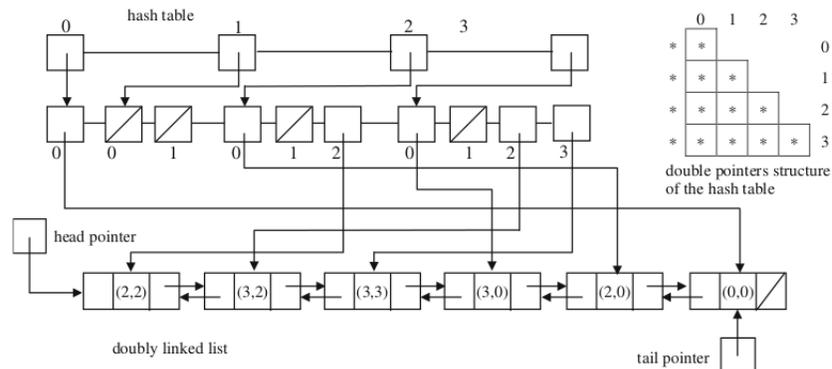


**Figura 3.3:** Representação da estrutura GLCLL. Retirada de (CLAUSI; ZHAO, 2002)

Na hora de capturar as características da imagem, uma janela deslizante é utilizada. Essa janela começa no canto superior esquerdo da imagem. As probabilidades são determinadas e as características são obtidas usando as estatísticas. Depois, a janela desloca uma coluna pra direita e as probabilidades da nova coluna adicionada na janela são inseridas na GLCLL, enquanto as probabilidades da coluna que passou são excluídas. Para esses novos pares, novos nós são criados na lista. Chegou no final da linha, a janela desliza uma linha pra baixo. Em linha par, ela move da esquerda para a direita, enquanto em linha ímpar, ela move da direita para a esquerda.

Uma tabela *hash* é uma estrutura que armazena elementos em posições fáceis de serem calculadas, através do valor ou representação do elemento. A tabela *hash* rapidamente transforma um elemento no endereço dele, através de uma função conhecida como *Hash*. Para fazer o acesso a tabela *hash*, os pares de níveis de cinza são utilizados como entrada, e cada elemento na tabela *hash* tem um ponteiro para um elemento na lista (CLAUSI; ZHAO, 2002).

Para fazer o cálculo das características de textura, basta percorrer a lista ordenada e aplicar as estatísticas sobre os seus elementos. Observe que um dos gargalos está no tamanho da lista ligada  $L$ , mas ainda sim é uma grande melhoria em relação ao método GLCM tradicional, que tinha de percorrer toda a matriz para obter as características. Outro gargalo é o tamanho da janela, pois quanto maior for essa janela, maior será a quantidade de probabilidades de co-ocorrências utilizadas nos cálculos e maior será o processamento realizado. A estrutura completa do GLHS pode ser vista na Figura 3.4.



**Figura 3.4:** Representação da estrutura GLHS. Retirada de (CLAUSI; ZHAO, 2002)

Uma outra variação da GLCM, que é uma evolução da GLCHS, é uma técnica que combina a estrutura híbrida em GLCHS com histogramas de soma e diferença (CLAUSI; ZHAO, 2003). Essa técnica é conhecida como GLCIA (grey level co-occurrence integrated algorithm) e é uma combinação do melhor das técnicas

GLCHS e GLCHH (grey level co-occurrence hybrid histogram). Uma vantagem do GLCHH é que ele usa uma matriz de uma única dimensão para calcular as características.

Os histogramas de soma e diferença são implementados usando um *framework* GLCHS. A implementação resultante é a GLCHH que reduz a demanda computacional do GLCHS sozinho. O problema é que o GLCHH não é apto a ser usado como base para os cálculos de todas as características definidas por Haralick, ou seja, ele não funciona para todas as características. Para essas outras características, o GLCHS é usado. No final temos um novo método GLCIA que é a combinação da GLCHS com a GLCHH.

A tabela *hash*, nesse caso, é dimensionada para o número de índices do vetor que representa o histograma tanto de soma quanto de diferença. Somente uma tabela *hash* de uma dimensão é criada que é acessada por uma chave única (seja soma ou diferença). Novamente, o ganho tanto em termos de memória quanto de computações é grande se comparada com a técnica GLCM original.

Outra variação é feita usando diferenças de níveis de cinza e suas distribuições para descrever a textura (OJALA *et al.*, 2001). As diferenças com sinal descrevem a textura de uma forma mais compacta e eficiente do que as co-ocorrências. Outra vantagem é que as diferenças não são afetadas pelas variações da luminosidade presente na imagem. A variação de luminosidade pode ser um empecilho para muitos dos algoritmos de análise de textura, mas felizmente, existem técnicas para lidar com esse tipo de problema.

Assim pode-se ver que existem muitas variações da técnica GLCM que tentam eliminar alguns gargalos que a técnica original possui. Mas o interessante é notar que até mesmo a versão proposta por Haralick é eficiente para imagens em uma escala aceitável, pois ela exige apenas um *loop* sobre a matriz para a obtenção das características, o que gera uma complexidade de  $O(G^2)$ , sendo  $G$  o número

**Tabela 3.2:** Tabela com as complexidades de tempo da técnica GLCM e algumas de suas variações.

Método	Ordem
GLCM	$O(n) + O(\alpha G^2)$
GLCLL	$O(nL) + O(\alpha L)$
GLCHS	$O(n) + O(\alpha L)$
GLCHH com histograma de soma	$O(n) + O(\alpha L_s)$
GLCHH com histograma de diferença	$O(n) + O(\alpha L_d)$

de níveis de cinza. As outras variações discutidas são melhores ainda, pois ao invés de iterar sobre a matriz toda, itera apenas sobre uma lista, seja com ou sem uma estrutura *hash*. A complexidade cai de  $O(G^2)$  para  $O(G)$ , que é considerado aceitável para a maior parte dos problemas.

A Tabela 3.2 mostra uma comparação das complexidades de tempo envolvidas em algumas das técnicas apresentadas nessa seção. Observe que o primeiro termo representa a complexidade para criar as probabilidades e o segundo termo representa a complexidade para o cálculo das características. Em quase todos os casos, o primeiro termo é  $O(n)$  devido a varredura na imagem para a criação da GLCM, sendo necessário passar por cada um dos  $n$  *pixels* na imagem. Em relação ao segundo termo, o  $\alpha$  indica uma ordem genérica para cada método.  $L$  é o tamanho da lista ligada e  $G$  é quantidade de níveis de cinza da imagem. O método GLCM requer, como já foi dito,  $O(G^2)$  pois deve percorrer toda a matriz. Os outros métodos dependem do tamanho da lista ligada, gerando as diferenças entre eles.

Ainda é possível melhorar mais a complexidade do método e diminuir seu tempo de execução. Por isso que a técnica GLCM continua sendo foco de estudos até os dias de hoje. Mesmo o método possuindo um bom desempenho, se a escala da imagem for muito grande, a geração da matriz será mais lenta, pois será percorrido uma quantidade muito maior de *pixels*.

A tendência agora é seguir a linha do paralelismo para futuras implementações, ou seja, tentar fazer proveito da programação paralela e distribuída para acelerar o processo de geração da matriz GLCM e dos cálculos das características. Uma aplicação em tempo real depende quase que totalmente da rapidez de resposta do sistema, o que em certos problemas pode ser crítico e isso é um motivo para se estar sempre tentando melhorar a técnica GLCM, assim como outras técnicas de extração de características.

### **3.3 GLCM aplicada em texturas de cana**

A técnica GLCM foi uma das pioneiras na área de extração de características, servindo como base para diversas outras técnicas que, ou são melhorias diretas ou são abordagens diferentes mas com o mesmo propósito que a técnica GLCM. A técnica foi desenvolvida sem pensar em uma aplicação específica e sim com o intuito de ser a mais genérica possível. A simplicidade dessa técnica juntamente com o baixo custo computacional, permite que ela possa ser utilizada em uma vasta gama de situações, sendo que o usuário deve apenas ter o trabalho de determinar uma dada configuração de parâmetros para sua aplicação específica. Esse foi um dos motivos que incentivou a escolha da GLCM ao invés de outras técnicas para extração de características.

Outro motivo que contribuiu para a escolha do método foi o fato de ele geralmente apresentar resultados bons nos tipos mais variados de problemas. As

texturas de cana, assim como qualquer tipo de textura, possuem padrões que podem ser observados visualmente e que fornecem dicas de quais relacionamentos entre *pixels* são mais comuns na imagem. Isso é um fator relevante para o método GLCM, que precisa dessas informações a respeito das imagens para obter resultados melhores. Em alguns tipos de textura é bem mais difícil perceber esses padrões e o uso do método GLCM se torna mais empírico. Não se sabe a princípio quais relacionamentos seriam mais adequados, o que não é o caso das imagens de cana, que possui padrões visíveis e mais fáceis de serem percebidos.

A aplicação de separação das imagens de cana é algo que deve ser feito em tempo real, pois a cana estará passando em uma esteira e no instante em que a imagem for obtida, ela deve ser processada e classificada, ativando algum mecanismo para direcionar corretamente a cana. O fato de ser em tempo real, exige que o método gere uma resposta correta dentro de um tempo aceitável, ou seja, não são permitidos atrasos.

O método GLCM pode ser utilizado em uma série de aplicações tais como a classificação de padrões em geleiras, a classificação de tecidos de fábrica (SALEM; NASRI, 2010), ou também para monitorar as condições de ferramentas através da análise da textura em sua superfície (DUTTA *et al.*, 2012). Pode-se perceber a variedade de campos que o método consegue atingir, mesmo sendo relativamente simples.

## 4 METODOLOGIA

O trabalho aqui desenvolvido visa fazer uso de processamento de imagens e visão computacional para automatizar o processo de separação de cana de acordo com o tipo de corte, através da metodologia de análise de textura nas imagens de cana. Para extrair as características que irão representar a textura em cada uma das imagens, foi escolhido o método estatístico GLCM, explicado anteriormente na Seção 3.1. A versão utilizada do método é a que encontra-se disponível no *toolbox* de Digital Image Processing do software *MatLab*<sup>1</sup>, cuja implementação é baseada no artigo original do Haralick (HARALICK; SHANMUGAM; DINSTEIN, 1973) de 1973.

Todas as imagens passaram por um estágio de pré-processamento que consistiu em: transformação das imagens para níveis de cinza e normalização das imagens, para remover a influência da luminosidade sobre as mesmas. As imagens da base de dados *Outex* estão todas na mesma escala, para garantir resultados consistentes. No caso das imagens de cana, foram recortadas imagens de mesma escala a partir de imagens de alta dimensão. Além disso, todas as características extraídas são invariantes a certas transformações, como translação e rotação na imagem (DUTTA *et al.*, 2012).

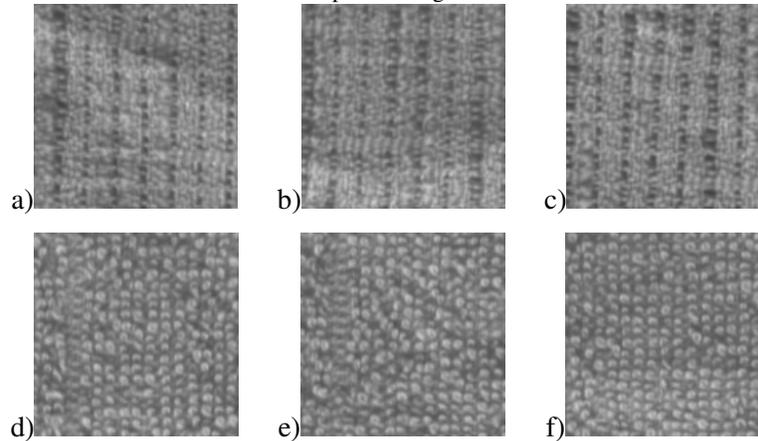
Tanto os testes sobre a base de imagens *Outex* quanto os testes sobre a base *Cana*, foram realizados em uma mesma máquina com as seguintes especificações:

- Processador Intel Core i5 com clock de 2,4GHz
- Memória Ram de 6GB (+1GB de Memória Dedicada)
- HD de 750GB
- Sistema Operacional Windows 7 Premium - 64 bits

---

<sup>1</sup><http://www.mathworks.com/products/matlab/>

**Tabela 4.1:** Exemplo de imagens da base *Outex*



Para testar a eficiência do método GLCM, primeiramente foram feitos testes sobre uma base de dados retirada do *Outex*<sup>2</sup>. *Outex* é um *framework* desenvolvido para avaliação de algoritmos de classificação e segmentação de texturas em imagens digitais. Ele apresenta um conjunto bem amplo de problemas de classificação e segmentação encapsulados nos chamados *test suites*, além de ser desenvolvido abertamente com a ajuda de vários grupos de pesquisa.

A base de imagens *Outex* é composta por duas classes que totalizam 400 imagens. Dessas 400 imagens, 200 são da classe 1 e 200 são da classe 2. Como essas imagens serão utilizadas para treinar e testar um classificador, foi definido que para cada uma das classes, 10% formaria o conjunto de treinamento e o restante, 90%, formaria o conjunto de teste. As imagens estão no formato *.ras* (*The sun raster format image*) e na escala 128x128. Todas as imagens da base *Outex* já se encontram em níveis de cinza. A Tabela 4.1 contém exemplos de imagens da base *Outex*. As imagens *a*, *b* e *c* são da classe 1 e as imagens *d*, *e* e *f* são da classe 2.

A função que gera a matriz GLCM no *toolbox* de processamento de imagens do Matlab é a *graycomatrix* que recebe a imagem *I* como parâmetro, além de parâ-

<sup>2</sup><http://www.outex.oulu.fi/>

metros adicionais como o conjunto de *offsets* e a quantização da matriz, ou seja, a quantidade de níveis de cinza. Para imagens binárias, o número de níveis de cinza geralmente é dois e para imagens de intensidade, geralmente é oito. Essa função retorna uma *struct* que contém a matriz GLCM da imagem correspondente. Um dos principais parâmetros na técnica GLCM são os *offsets*, que já foram explicados na definição da técnica. É possível definir qualquer conjunto de *offsets* para gerar as matrizes GLCM. Nesse trabalho, foi definido a utilização de cinco *offsets* para tentar abranger os mais variados tipos de relacionamentos espaciais entre os *pixels* da imagem.

O primeiro offset é composto por quatro relacionamentos definidos por [0 1; -1 1; -1 0; -1 -1], ou seja, é utilizado um relacionamento de uma única unidade de distância em cada um dos quatro ângulos que o parâmetro  $\Theta$  pode assumir. Ele geralmente é um *offset* razoável, pois relacionamentos com distâncias muito grande, ou seja, com um valor do parâmetro  $s$  muito grande, podem gerar resultados piores do que relacionamentos entre *pixels* mais próximos. O segundo *offset* procura priorizar relacionamentos na horizontal, somente aumentando a distância entre os *pixels* na horizontal e ele é definido por [0 1; 0 2; 0 3; 0 -1; 0 -2; 0 -3]. Observe que ele abrange os dois sentidos na horizontal e usa os valores 1,2 e 3 para o parâmetro  $s$ . O terceiro *offset* prioriza relacionamentos na vertical, somente aumentando a distância entre os *pixels* na vertical e é definido por [-1 0; -2 0; -3 0; 1 0; 2 0; 3 0]. Observe que ele abrange os dois sentidos na vertical assim como o caso anterior e usa os valores 1,2 e 3 para o parâmetro  $s$ . O quarto *offset* procura priorizar relacionamentos tanto na horizontal quanto na vertical e é definido por [0 1; 0 2; -1 0; -2 0; 0 -1; 0 -2; 1 0; 2 0]. Observe que ele abrange os dois sentidos em cada uma das direções e usa os valores 1 e 2 para o parâmetro  $s$ . O último *offset* procura priorizar relacionamentos nas diagonais e é definido por [-1 1; -2 2; -3 3; -1 -1; -2 -2; -3 -3]. Observe que ele usa as duas diagonais superiores e usa os valores 1, 2 e 3 para o parâmetro  $s$ .

Outra função do toolbox de processamento de imagens do *MatLab* utilizado nesse trabalho é a função *graycoprops* que gera o valor de cada uma das características para a GLCM passada como parâmetro. O *MatLab* trabalha com apenas quatro características que são: *Contrast*, *Correlation*, *Energy* e *Homogeneity*. No artigo do Haralick são definidas 14 características diferentes para representação das texturas, conforme foi apresentado na seção 3.1, mas a maioria dos trabalhos na literatura utilizam um número menor de características. Por esse motivo, foi implementado algumas dessas características para expandir a capacidade do método disponível no *MatLab*. No total ficaram oito características, sendo as quatro já citadas e mais quatro que são: *Maximum*, *Inverse Difference Moment (IDM)*, *Inverse Difference (INV)* e *Entropy*.

Cada característica possui um significado diferente. *Contrast* representa uma medida da intensidade de contraste de um *pixel* e seu vizinho sobre toda a imagem. *Correlation* representa o quão correlacionado um *pixel* é com o seu vizinho sobre toda a imagem. *Energy* representa a quantidade de variação na intensidade em uma dada região da imagem. *Homogeneity* representa a proximidade da distribuição dos elementos na GLCM com a sua diagonal. *Maximum* é o maior elemento da matriz, ou seja, a maior quantidade de co-ocorrências na imagem. *Entropy* é o contrário de *Energy* e indica o quão normal é a distribuição dos níveis de cinza na imagem. Cada uma dessas características e a fórmula para calculá-las a partir da matriz GLCM foram apresentadas na seção 3.1.

A extração de características é apenas uma parte do processo de análise de texturas. Através das características obtidas é possível diferenciar as várias classes de textura e dizer para qual classe uma dada imagem de textura pertence. A quantidade de valores de características depende da quantidade de matrizes GLCM que uma imagem possui e isso, por sua vez, depende da quantidade de *offsets* utilizados na geração da GLCM. Uma imagem que possui um *offset* que define quatro relacionamentos, possuirá quatro GLCM associadas a ela e assim, para cada caracte-

terística, teremos quatro valores, sendo cada um associado a uma GLCM diferente da imagem.

Então, com a extração das características, cada imagem irá gerar um vetor de características que contém todos os valores de características para a mesma. A matriz de treinamento é formada combinando esses vetores de características. Essa matriz é usada como entrada para o classificador SVM.

Foi utilizado a *LibSVM*<sup>3</sup> que é uma biblioteca de código aberto, voltada para a classificação, regressão e estimativa de distribuição utilizando *Support Vector Machines*. Totalmente desenvolvida na linguagem C++, embora ela utilize uma API da linguagem C. O objetivo da biblioteca é facilitar o uso da SVM nas mais variadas aplicações. O funcionamento é dividido em dois passos: primeiro, treina-se um conjunto de dados para obter um modelo e segundo, usa-se o modelo para prever informações de um conjunto de teste.

Vários testes foram feitos usando a base *Outex* e variando os *offsets* e os números de características. Para a SVM, foi escolhido o *kernel* linear com variações no parâmetro *C*, também conhecido como *BoxConstraint*. Foi escolhido sete valores para esse parâmetro: 1, 5, 16, 32, 64, 128 e 256. A escolha desses valores foi uma tentativa de abranger uma variedade maior de valores para se poder ter uma noção de como a variação desse parâmetro pode afetar o resultado final.

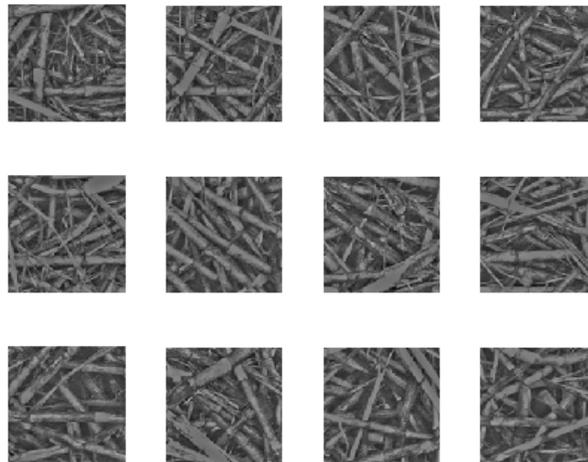
A próxima etapa é a aplicação da técnica GLCM nas imagens de cana que é o foco desse trabalho. As imagens de cana, diferente das imagens da base *Outex*, tiveram que ser transformadas para níveis de cinza e passaram por um processo de normalização para remover influências da luminosidade. Todas as imagens de cana foram recortadas de imagens de alta dimensão, gerando imagens na escala 128x128. Como já dito antes, existem duas classes de interesses para as imagens de cana, que são as canas cortadas manualmente e as canas cortadas me-

---

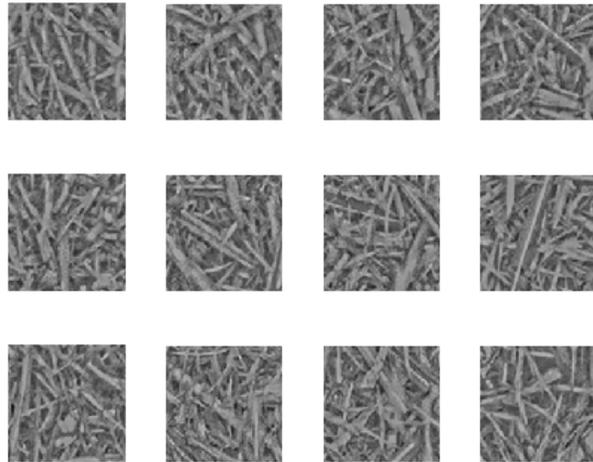
<sup>3</sup><http://www.csie.ntu.edu.tw/~cjlin/libsvm/>

canicamente. A classe das canas cortadas manualmente contém um total de 148 imagens, sendo 74 de treinamento e 74 de teste. A classe das canas cortadas mecanicamente contém um total de 657 imagens, sendo 329 de treinamento e 328 de teste. Percebe-se que foi usado uma proporção de 50% para treinamento e 50% para teste. Embora a quantidade de imagens nesta base de dados esteja desbalanceada, os resultados obtidos não serão afetados, pois a SVM consegue contornar esse problema devido a sua alta capacidade de generalização. A Figura 4.1 mostra exemplos de cortes de cana manual e a Figura 4.2 mostra exemplos de cortes de cana mecânico.

A próxima seção discute os resultados obtidos com as imagens das bases *Outex* e *Cana*, além de avaliar o método GLCM de uma forma geral, apresentando os prós e os contras desse método e em quais situações ele seria mais apropriado.



**Figura 4.1:** Exemplos das imagens de corte manual de cana já em níveis de cinza e normalizadas.



**Figura 4.2:** Exemplos das imagens de corte mecânico de cana já em níveis de cinza e normalizadas.

## 5 TESTES E RESULTADOS

### 5.1 Base Outex

Primeiramente foram feitos testes sobre as imagens na base *Outex*, como visto na Metodologia. Dentre os parâmetros do GLCM, os que mais importam são os *offsets* e as características que serão extraídas. Como já foi visto, definiu-se um conjunto de cinco *offsets* diferentes, cada um focando em certas direções no relacionamento entre os *pixels*. Para as características, todas as oito escolhidas foram retiradas do artigo do Haralick (HARALICK; SHANMUGAM; DINSTEIN, 1973) pelo fato de serem mais comumente usadas.

O impacto das características pode ser percebido quando mudamos a quantidade de quatro para oito e isso também será discutido nessa seção. Uma das tarefas mais difíceis quando se trata de um problema de extração de características em textura de imagens, é definir quais características representam mais fielmente as imagens correspondentes, pois isso é um fator determinante no momento da classificação dessas imagens. Como o objetivo é atingir uma classificação ótima com a menor taxa de erros possíveis, torna-se evidente que devemos testar com conjuntos diferentes de características para descobrir qual seria mais apropriado.

Os *offsets* possuem um grande impacto em um problema de extração de características, pois não é qualquer matriz GLCM que representa bem uma imagem. Em algumas situações, uma matriz GLCM não abrange os relacionamentos de *pixels* mais importantes da imagem e assim não é capaz de diferenciar corretamente certas classes de imagens. Uma característica interessante sobre os *offsets* e que será discutida ainda nessa seção é que os *offsets* dependem muito dos padrões de formas na textura da imagem, ou seja, é bem improvável encontrar um *offset* gené-

rico que seja ideal para qualquer tipo de textura de imagem. Geralmente, os *offsets* tem que ser adaptados para o tipo de problema que a pessoa se depara.

Os cinco *offsets* escolhidos são:

1.  $[0\ 1; -1\ 1; -1\ 0; -1\ -1]$  - relacionamentos unitários
2.  $[0\ 1; 0\ 2; 0\ 3; 0\ -1; 0\ -2; 0\ -3]$  - relacionamentos horizontais
3.  $[-1\ 0; -2\ 0; -3\ 0; 1\ 0; 2\ 0; 3\ 0]$  - relacionamentos verticais
4.  $[0\ 1; 0\ 2; -1\ 0; -2\ 0; 0\ -1; 0\ -2; 1\ 0; 2\ 0]$  - relacionamentos horizontais e verticais
5.  $[-1\ 1; -2\ 2; -3\ 3; -1\ -1; -2\ -2; -3\ -3]$  - relacionamentos diagonais

Os experimentos sobre a base *Outex* serviram mais como uma tentativa de avaliar as capacidades do método GLCM e seu potencial na extração de características e classificação de imagens.

A Tabela 5.1 mostra cada uma das bases utilizadas. Observe que foram feitos experimentos com 4 e com 8 características. E para cada um desses dois casos, foi utilizado uma variação de cinco *offsets*, como já foi discutido na seção 4. As bases são apenas uma forma de simplificar a representação e facilitar no entendimento dos resultados obtidos.

**Tabela 5.1:** Tabela com as bases utilizadas para os experimentos.

Nº de Características / Offsets	1	2	3	4	5
4	B <sub>14</sub>	B <sub>24</sub>	B <sub>34</sub>	B <sub>44</sub>	B <sub>54</sub>
8	B <sub>18</sub>	B <sub>28</sub>	B <sub>38</sub>	B <sub>48</sub>	B <sub>58</sub>

Os experimentos com a SVM foram feitos com o uso do *kernel* linear apenas. Não houve necessidade de fazer experimentos com um *kernel* mais sofisticado, como um *kernel* RBF, pois a taxa de 100% de acertos já tinha sido atingida com o uso do *kernel* linear. As imagens na base *Outex* não são linearmente separáveis, mas uma escolha de características adequadas para o método GLCM permitiu a separação linear dessas imagens em duas classes. Além do *kernel*, um parâmetro importante característico de qualquer SVM é o parâmetro  $C$ , que já foi citado anteriormente e que foi variado durante os experimentos sem a validação cruzada. Também foram feitos testes com *kernel* linear sobre validação cruzada, que tenta empiricamente encontrar o melhor valor para  $C$ .

As Tabelas 5.2 e 5.3 apresentam todos os valores de taxas de acertos obtidos para os experimentos com *kernel* linear sem validação cruzada e variando o valor de  $C$ . A primeira tabela mostra os resultados para quatro características e a segunda tabela mostra os resultados para oito características. Os valores adotados para o parâmetro  $C$  foi uma decisão parcialmente arbitrária, pois a ideia era tentar abranger a maior quantidade de valores possíveis. Além disso, quase todos os valores são potências de dois, não que isso seja um requisito, mas foi um palpite razoável e os resultados comprovam isso.

**Tabela 5.2:** Tabela com as taxas de acertos para a base *Outex*, utilizando quatro características.

Base / Parâmetro C	1	5	16	32	64	128	256
B <sub>14</sub>	78.3333%	78.3333%	81.3889%	90.5556%	93.0556%	93.0556%	91.6667%
B <sub>24</sub>	83.0556%	97.5%	98.3333%	98.6111%	97.5%	97.5%	95.8333%
B <sub>34</sub>	87.5%	88.8889%	93.3333%	94.4444%	95.2778%	96.9444%	96.9444%
B <sub>44</sub>	78.6111%	81.3889%	89.4444%	95.5556%	96.1111%	94.7222%	90.2778%
B <sub>54</sub>	97.5%	98.8889%	99.1667%	99.1667%	99.1667%	99.1667%	<b>100%</b>

**Tabela 5.3:** Tabela com as taxas de acertos para a base *Outex*, utilizando oito características.

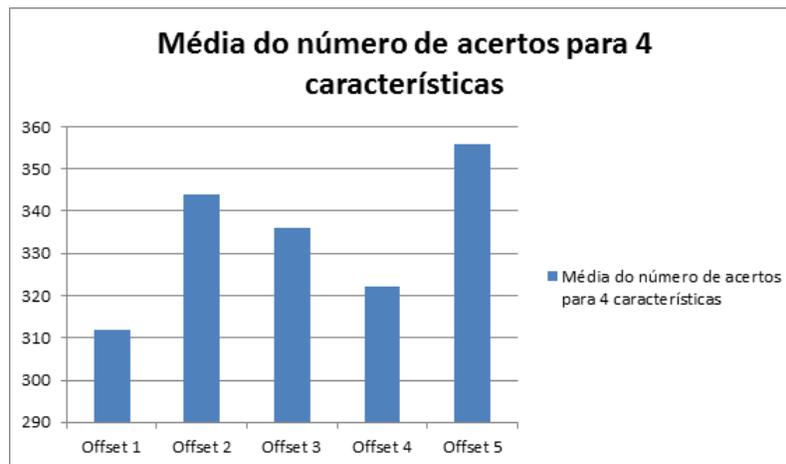
Base / Parâmetro C	1	5	16	32	64	128	256
B <sub>18</sub>	65.8333%	76.1111%	77.5%	80.2778%	78.6111%	80.5556%	92.7778%
B <sub>28</sub>	86.3889%	89.4444%	83.6111%	84.1667%	90%	89.1667%	89.1667%
B <sub>38</sub>	97.2222%	97.2222%	96.6667%	96.9444%	97.2222%	97.2222%	97.2222%
B <sub>48</sub>	85.5556%	86.9444%	91.3889%	90%	92.7778%	92.7778%	92.7778%
B <sub>58</sub>	98.0556%	<b>98.8889%</b>	98.8889%	98.6111%	97.7778%	97.7778%	97.7778%

Podemos ver pelas tabelas que a melhor taxa de acertos foi de 100%, utilizando 4 características e o quinto *offset*, com *C* valendo 256. Também pode-se notar que é justamente com esse quinto *offset* que se obtém os melhores resultados no geral, tanto com 4 quanto com 8 características, ou seja, as imagens da base *Outex* utilizadas possuem muitos padrões nas diagonais. A pior taxa de acertos obtida foi de 65.8333%, quando usamos o primeiro *offset* com 8 características e o *C* valendo 1. Podemos ver também que o primeiro *offset* é o que gerou os piores resultados tanto para 4 quanto para 8 características, o que prova que a distância no relacionamento entre os *pixels* é um fator relevante, ou seja, nem sempre a distância de um *pixel* é suficiente. Dependendo da imagem, os padrões são mais extensos e usar uma distância de um *pixel* faz com que partes de um mesmo padrão sejam analisados separadamente e não como uma única forma que caracteriza a textura.

Quanto ao tempo de execução, os resultados foram bem melhores que o esperado. Para realizar todo o processo, desde a extração das características até a classificação sobre uma única imagem, o tempo foi instantâneo. Para todo o conjunto de imagens, o tempo médio foi de um minuto, que é um tempo aceitável para aplicações em tempo real. Pode-se ver que o método GLCM é rápido para fazer a extração das características de uma imagem, mas deve se considerar também as configurações da máquina que estava sendo utilizada para realizar os testes.

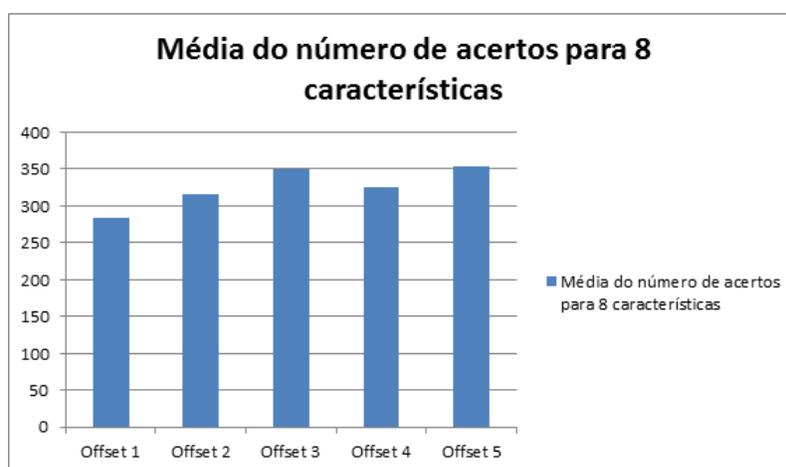
Em relação a quantidade de características, pode-se ver que o aumento das características na maior parte dos casos apenas piorou os resultados. Isso pode ser explicado pelo fato das imagens da base *Outex* serem mais simples e assim, não há necessidade de tantas características para uma representação fiel das imagens. É claro que o *offset* interfere também, mas de qualquer forma, jamais se deve supor que aumentando a quantidade de características sempre gerará resultados melhores. E essa é uma das maiores dificuldades em se tratando de problemas de classificação de imagens através das características de textura, ou seja, é difícil saber ao certo quais são as melhores características para um dado problema. A descoberta é em boa parte das vezes feita de forma empírica.

Outro detalhe que pode ser observado nas Tabelas 5.2 e 5.3 é que a variação no parâmetro  $C$  causa um impacto grande sobre os resultados, por exemplo, nas linhas correspondente as bases  $B_{34}$ ,  $B_{54}$  e  $B_{18}$ , o aumento de  $C$  fez com que as taxas de acertos só aumentassem. Mas em alguns casos, o aumento de  $C$  pode piorar o resultado, como pode-se ver na base  $B_{44}$  por exemplo, a partir de  $C$  valendo 64.



**Figura 5.1:** Gráfico que mostra a média da quantidade de acertos para cada *offset* usando 4 características sobre a base *Outex*.

Os gráficos nas Figuras 5.1 e 5.2 mostram uma média da quantidade de acertos para cada um dos *offsets*. O primeiro gráfico mostra os resultados apenas para



**Figura 5.2:** Gráfico que mostra a média da quantidade de acertos para cada *offset* usando 8 características sobre a base *Outex*.

quatro características. Podemos ver que o quinto *offset*, que prioriza relacionamentos na diagonal, foi o que gerou a maior quantidade de acertos e o primeiro *offset*, que estabelece relacionamentos unitários nas quatro principais direções foi o que gerou a menor quantidade de acertos. O segundo gráfico mostra os resultados para oito características. Pode-se ver que o terceiro e o quinto *offset* foram os que geraram as maiores quantidades de acertos, quase possuindo a mesma eficiência. O primeiro *offset* novamente foi o que gerou os piores resultados.

**Tabela 5.4:** Tabela com os resultados usando validação cruzada na base de imagens *Outex*.

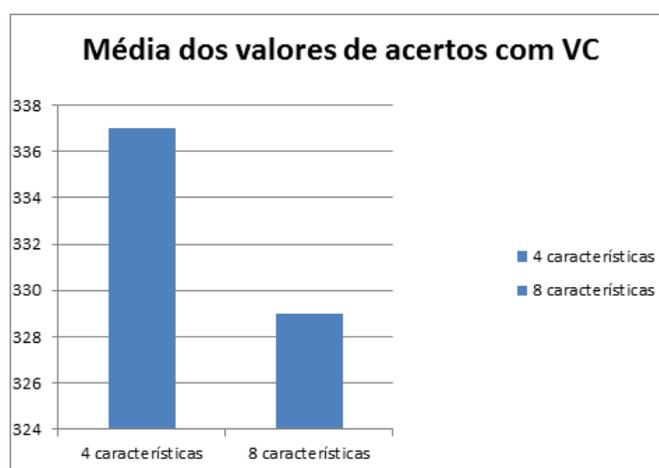
Base	Acurácia
B <sub>14</sub>	86.3889%
B <sub>24</sub>	95%
B <sub>34</sub>	96.9444%
B <sub>44</sub>	90.2778%
B <sub>54</sub>	<b>100%</b>
B <sub>18</sub>	80.5556%
B <sub>28</sub>	89.1667%
B <sub>38</sub>	97.2222%
B <sub>48</sub>	92.7778%
B <sub>58</sub>	97.7778%

A Tabela 5.4 mostra as taxas de acertos para a validação cruzada sobre a base de dados *Outex*, tanto com 4 quanto com 8 características.

Podemos ver que a melhor taxa de acertos foi de 100% com 4 características sobre o quinto *offset*. Novamente, o quinto *offset* que prioriza relacionamentos nas diagonais gerou os melhores resultados. A pior taxa de acertos foi de 80.5556% com 8 características sobre o primeiro *offset*, o que é consistente com os resultados obtidos sem validação cruzada. Observe que o primeiro *offset* é o que gerou os piores resultados novamente, enquanto o quinto *offset* foi o que gerou os melhores resultados, tanto com 4 quanto com 8 características.

Uma coisa interessante de se reparar é o fato de que em algumas situações na validação cruzada, o resultado foi pior que o resultado correspondente sem a validação cruzada. Por exemplo, para a base  $B_{14}$ , a maior taxa de acertos sem a validação cruzada foi de 93.0556%, para  $C$  valendo 64. Já na validação cruzada, a melhor taxa de acertos encontrada para a base  $B_{14}$  foi de 86.3889%, que é bem menor que a taxa de acertos sem validação cruzada.

Esses resultados não fazem muito sentido e uma das causas disso é devido ao fato da base *Outex* possuir um total de apenas 40 imagens de treinamento, que é uma quantidade pequena para a validação cruzada, pois nela, esse conjunto de treinamento será particionado para criar um novo conjunto de treinamento e um novo conjunto de teste sobre o qual a validação cruzada irá trabalhar. Um particionamento irá reduzir mais ainda a quantidade de imagens de treinamento, além de diminuir a abrangência das imagens de teste, correndo o risco de quase todas serem de uma mesma classe.



**Figura 5.3:** Gráfico que mostra a média da quantidade de acertos com validação cruzada (VC) sobre a base *Outex*.

O gráfico na Figura 5.3 mostra a média da quantidade de acertos com validação cruzada. Podemos ver que há uma diferença notável quando se muda de quatro para oito características e nesse caso, a quantidade de acertos piora quando se aumenta a quantidade de características. Como já foi comentado antes, uma justificativa para esse resultado é o fato das imagens da base *Outex* serem mais simples e terem padrões mais fáceis de serem descritos, sem a necessidade de usar um número maior de características que acaba por piorar as representações obtidas e atrapalhar na classificação final.

## 5.2 Base Cana

Depois dos experimentos com a base de imagens *Outex*, foram feitos os experimentos na base de imagens de cana. Assim como na base *Outex*, não houve necessidade de fazer testes com um *kernel* mais sofisticado, como um *kernel* RBF, pois a taxa de 100% de acertos já tinha sido atingida com o uso do *kernel* linear. Também foram feitos experimentos com *kernel* linear sobre validação cruzada, que tenta empiricamente encontrar o melhor valor para *C*.

As Tabelas 5.5 e 5.6 mostram as taxas de acertos obtidos para os experimentos com *kernel* linear sem validação cruzada e variando o valor de *C*. Da mesma forma que na base *Outex*, os resultados foram divididos em duas tabelas, sendo a primeira com 4 características e a segunda com 8 características. Os *offsets* continuam sendo os mesmos, assim como os valores do parâmetro *C*.

Quanto ao tempo de execução, não houve diferença significativa em relação à base *Outex*. Para realizar todo o processo, desde a extração das características até a classificação sobre uma única imagem, o tempo foi instantâneo. Para todo o conjunto de imagens, o tempo médio foi de um minuto e meio, que continua sendo um tempo aceitável para aplicações em tempo real.

**Tabela 5.5:** Tabela com as taxas de acertos para a base de Cana, utilizando quatro características.

Base / Parâmetro C	1	5	16	32	64	128	256
B <sub>14</sub>	84.5771%	89.5522%	95.0249%	96.5174%	97.5124%	99.005%	99.2537%
B <sub>24</sub>	83.5821%	99.7512%	<b>100%</b>	100%	100%	100%	100%
B <sub>34</sub>	85.8209%	95.0249%	97.5124%	98.2587%	98.01%	98.2587%	96.7662%
B <sub>44</sub>	86.3184%	94.0299%	99.2537%	99.7512%	<b>100%</b>	100%	99.7512%
B <sub>54</sub>	82.3383%	97.5124%	99.005%	99.2537%	99.2537%	99.005%	98.7562%

**Tabela 5.6:** Tabela com as taxas de acertos para a base de Cana, utilizando oito características.

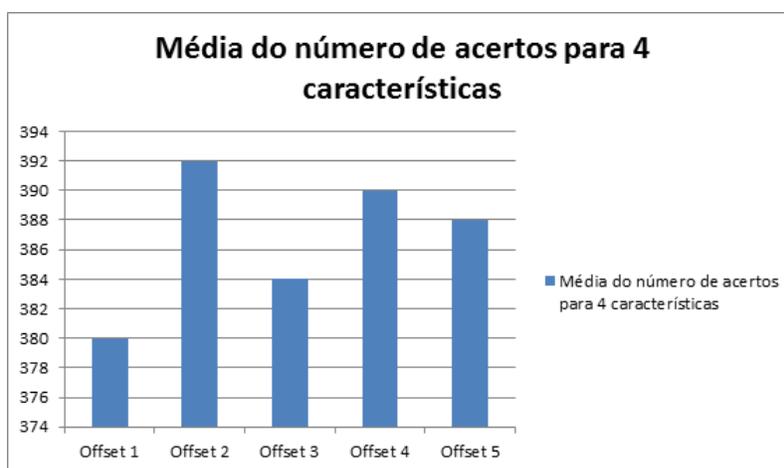
Base / Parâmetro C	1	5	16	32	64	128	256
B <sub>18</sub>	91.0448%	95.7711%	96.2687%	97.5124%	98.7562%	99.005%	99.2537%
B <sub>28</sub>	95.5224%	99.005%	<b>100%</b>	99.7512%	99.005%	99.005%	99.7512%
B <sub>38</sub>	94.0299%	97.2637%	98.01%	98.01%	98.5075%	98.5075%	98.7562%
B <sub>48</sub>	95.0249%	97.0149%	99.7512%	<b>100%</b>	100%	100%	100%
B <sub>58</sub>	93.7811%	98.01%	98.2587%	98.7562%	98.7562%	99.005%	98.2587%

Podemos ver que foi atingido a taxa de 100% de acertos nas bases  $B_{24}$ ,  $B_{44}$ ,  $B_{28}$  e  $B_{48}$ . Também se nota que no geral, os melhores resultados foram para o segundo *offset*, ou seja, as imagens de cana possuem muitos padrões na horizontal. O quarto *offset* também apresentou resultados bons tanto com 4 quanto com 8 características. Esse *offset* prioriza relacionamentos tanto horizontais quanto verticais, o que mostra novamente a prevalência de padrões horizontais na imagem, mesmo existindo muitos padrões verticais. A pior taxa de acertos foi de 82.3383%, usando 4 características sobre o quinto *offset* com  $C$  valendo 1. E nessa mesma base, se aumentarmos o  $C$  de 1 para 5, há um aumento significativo na taxa de acertos, mostrando mais uma vez o impacto que esse parâmetro possui sobre os resultados. Os piores resultados foram obtidos com o primeiro *offset*, assim como na base *Outex*. É imprescindível saber escolher uma distância adequada nos relacionamentos entre os *pixels* e isso só pode ser percebido, analisando visualmente a imagem e sua textura.

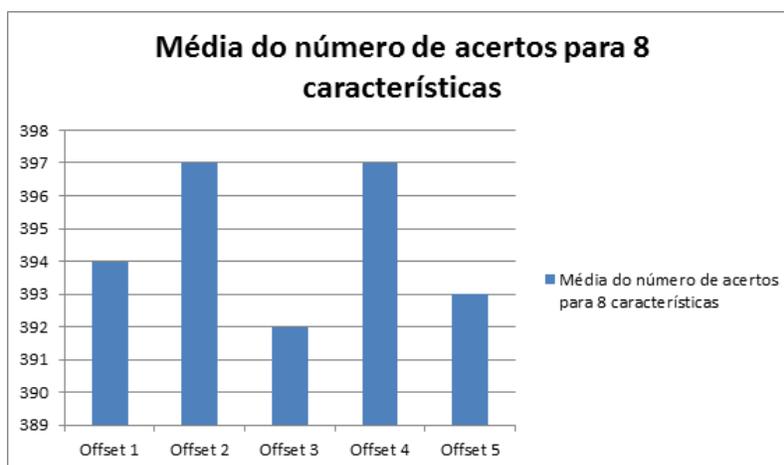
Em relação as características, enquanto na base *Outex* o aumento das características piorava os resultados, isso já não acontece com as imagens de cana. De acordo com os resultados, pode-se ver que aumentando a quantidade de características de 4 para 8, houve uma melhora nas taxas de acertos para quase todos os *offsets*. Isso se deve ao fato das imagens de cana serem bem mais complexas, com uma quantidade maior de padrões. Um conjunto muito pequeno de caracte-

rísticas pode não dar conta de representar as imagens de forma adequada. Mas é importante perceber que quantidade não é sinônimo de qualidade, ou seja, não basta escolher uma quantidade grande de características se essas não representam bem as imagens como deveriam. As características precisam estar de acordo com o seu tipo de problema e o único jeito de descobrir quais são as mais adequadas é ir testando várias combinações de características diferente, como já tinha sido explicado antes.

No geral, os resultados com as imagens de cana foram bem melhores que os resultados com a base *Outex*. O fato de uma SVM com *kernel* linear atingir 100% de acertos sobre as imagens de cana sem tantos esforços, comprova que a eficiência está na técnica GLCM e a forma como ela extrai as características que representam as texturas das imagens. Apesar da técnica GLCM ser uma das pioneiras na área de análise de texturas, nossos resultados comprovam que ela obtém ótimos resultados e tem uma alta capacidade de adaptação para vários tipos de problemas, bastando saber escolher os melhores parâmetros, ou seja, os melhores *offsets* e as melhores características que representarão a imagem.



**Figura 5.4:** Gráfico que mostra a média da quantidade de acertos para cada *offset* usando 4 características sobre a base de Cana.



**Figura 5.5:** Gráfico que mostra a média da quantidade de acertos para cada *offset* usando 8 características sobre a base de Cana.

Os gráficos nas Figuras 5.4 e 5.5 mostram uma média da quantidade de acertos para cada um dos *offsets*. O primeiro gráfico mostra os resultados para apenas quatro características. Podemos ver que o segundo *offset*, que prioriza os relacionamentos horizontais foi o que gerou as maiores quantidades de acertos, enquanto que o primeiro *offset* gerou as piores quantidades. O segundo gráfico mostra os resultados para oito características. Nesse caso, o segundo e o quarto *offset* ficaram empatados, gerando a maior quantidade de acertos, enquanto o terceiro *offset*, que prioriza relacionamentos verticais, foi o que gerou a pior quantidade de acertos e não o primeiro como na base *Outex*.

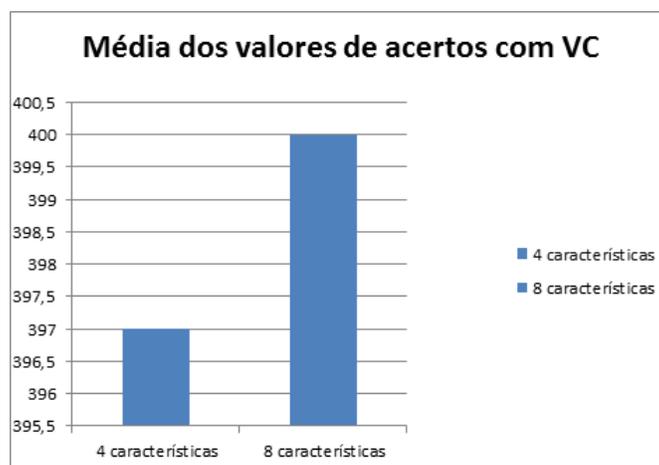
A Tabela 5.7 mostra as taxas de acertos para a validação cruzada (VC) sobre a base de Cana, tanto com 4 quanto com 8 características.

**Tabela 5.7:** Tabela com os resultados usando validação cruzada (VC) na base de Cana.

Base	Acurácia
B <sub>14</sub>	98.5075%
B <sub>24</sub>	99.005%
B <sub>34</sub>	98.7562%
B <sub>44</sub>	99.7512%
B <sub>54</sub>	98.2587%
B <sub>18</sub>	99.5025%
B <sub>28</sub>	99.005%
B <sub>38</sub>	99.2537%
B <sub>48</sub>	<b>100%</b>
B <sub>58</sub>	99.2537%

Podemos ver que a melhor taxa de acertos foi de 100%, usando 8 características sobre o quarto *offset*, que prioriza relacionamentos horizontais e verticais. Embora esse resultado não seja coerente com aqueles obtidos sem validação cruzada, ele evidencia o fato de as imagens de cana possuírem muitos padrões tanto horizontais quanto verticais. A pior taxa de acertos foi de 98.2587%, usando 4 características sobre o quinto *offset*, que prioriza relacionamentos nas diagonais. Nesse caso, nota-se que o quarto *offset* foi o que gerou os melhores resultados tanto com 4 quanto com 8 características.

Novamente, podemos ver que em alguns casos a taxa de acertos obtida com a validação cruzada para uma dada base é menor que a maior taxa de acertos sem validação cruzada para aquela mesma base. O problema torna a se repetir para a base de cana, mesmo ela possuindo uma quantidade bem maior de imagens para treinamento do que a base *Outex*, logo, é provável que a implementação da validação cruzada não está sendo eficiente, ou talvez a heurística de selecionar valores de  $C$  no intervalo de  $2^{-5}$  a  $2^{15}$  esteja ignorando outros valores que poderiam ser importantes.



**Figura 5.6:** Gráfico que mostra a média da quantidade de acertos com validação cruzada (VC) sobre a base de Cana.

O gráfico na Figura 5.6 mostra a média da quantidade de acertos com validação cruzada. Podemos ver que na base de Cana, o resultado inverte se comparado com a base *Outex*. Quando o número de características aumenta de quatro para oito, a quantidade média de acertos também aumenta. O melhor resultado nesse caso é com oito características, o que comprova que para as imagens de cana que são mais complexas, é necessário uma maior quantidade de características para se obter uma representação eficiente.

Conclui-se a partir dos resultados obtidos neste Capítulo, tanto para a base de imagens *Outex* quanto para a base de Cana, que a GLCM é uma técnica que funciona bem e que o seu desempenho depende muito dos parâmetros escolhidos. Não existe uma fórmula ou uma regra fixa que determina quais parâmetros serão os melhores em um certo caso. Isso depende muito do problema, da textura das imagens, da quantidade de imagens e até do próprio classificador utilizado para classificar os resultados.

## 6 CONCLUSÃO

Esse trabalho apresentou uma técnica para automatizar o processo de separação de cana-de-açúcar nas indústrias, como uma forma de melhorar a produtividade, diminuir os gastos e gerar mais lucros para a indústria de cana. Como foi visto, é vantajoso fazer essa separação dos pedaços de cana por um computador ao invés de uma pessoa, pois o computador não cansa, seu julgamento tende a ser mais preciso e não há riscos de acidente com funcionários. Basicamente, isso resume a proposta e a justificativa do trabalho aqui apresentado.

Foi feita uma breve introdução a área de processamento digital de imagens e uma de suas sub-áreas mais importantes, que é a análise de texturas de uma imagem como forma de identificação e classificação da mesma. Não foi possível entrar em muitos detalhes dentro dessa ampla área que é o processamento digital de imagens, mas foi possível perceber a sua importância para muitos problemas reais e o grande número de aplicações que é possível desenvolver usando os conceitos e técnicas dessa área.

A base para esse trabalho foi a seleção de um método para extração de características de uma imagem. Na literatura, existem vários métodos para todos os tipos de aplicações. Dentre eles, foi escolhido o GLCM, pelo fato de ser um método simples, eficiente e com uma capacidade de se adaptar facilmente aos mais variados tipos de problemas. Explicou-se o funcionamento do método, suas vantagens e desvantagens, além de apresentar variações do mesmo que até hoje são tópicos de pesquisa dentro da análise de texturas.

Um dos objetivos desse trabalho foi mostrar o quão eficiente o método GLCM pode ser e se realmente ele é uma boa escolha para problemas que envolvem extração de características de imagens. Foi visto que uma das dificuldades do método GLCM está em determinar os parâmetros ideais, o que causa um impacto no re-

sultado final da classificação. Para os *offsets*, é interessante analisar visualmente a imagem e tentar deduzir quais seriam os melhores *offsets*, com base nas formas e padrões existentes nas imagens. Claro que isso não garante que os *offsets* escolhidos serão os melhores, ou seja, é necessário que uma investigação empírica seja realizada.

Quanto a quantidade e quais características utilizar, tudo vai depender das imagens e da dificuldade em se encontrar uma boa representação para elas. Algumas imagens exigem menos características, outras exigem mais. Além disso, algumas características são mais representativas do que outras e isso deve ser percebido para se obter o melhor resultado.

O foco do trabalho não foi desenvolver o método GLCM e sim entender as suas propriedades, entender como tornar ele mais eficiente e como ele deveria ser utilizado em cada tipo de problema. Para se ter uma noção mais detalhada sobre o método, foram feitos vários testes utilizando duas bases de imagens. A primeira base foi a *Outex*, que foi usada como forma de averiguar o método e a segunda base foi a de cana que é a base de interesse para esse trabalho.

Uma conclusão que se chegou a partir desses testes é que o desempenho do método GLCM depende da aplicação. Como foi visto, ele obteve melhores resultados para a base de imagens de cana do que a base *Outex*, mas não se pode deixar de considerar o classificador SVM utilizado e o tamanho dos conjuntos para treinamento e teste. São fatores que também interferem, mas não tanto quanto o próprio método GLCM e seus parâmetros. Como já foi comentado antes, o tempo de processamento em todos os testes foi satisfatório, podendo a GLCM ser usada em aplicações que exijam resposta imediata.

A partir de todos os testes e observações, pode-se concluir que o método GLCM desempenha bem seu papel como um método de extração de características de imagens. Além disso, os resultados obtidos para as imagens da base Cana

foram satisfatórios, logo, o método GLCM foi uma ótima escolha como ferramenta para o problema de classificação de imagens de cana-de-açúcar.

Para trabalhos futuros, poderia ser desenvolvido uma implementação em *hardware* do método GLCM, pois ele trabalha fazendo cálculos sobre uma matriz, o que é possível de se implementar em hardware sem muitas dificuldades. Uma outra opção seria uma implementação paralela quando houvesse mais de uma matriz GLCM para uma dada imagem. Nesse caso, a geração de cada matriz GLCM seria feita de forma paralela, assim como o processo de extração das características.

## REFERÊNCIAS BIBLIOGRÁFICAS

BARALDI, A.; PARMIGGIANI, F. An investigation of the textural characteristics associated with gray level cooccurrence matrix statistical parameters. *Geoscience and Remote Sensing, IEEE Transactions on*, v. 33, n. 2, p. 293–304, 1995. ISSN 0196-2892.

BOYKOV, Y.; VEKSLER, O.; ZABIH, R. Markov random fields with efficient approximations. In: *Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on*. [S.l.: s.n.], 1998. p. 648–655. ISSN 1063-6919.

BRADY, K.; JERMYN, I.; ZERUBIA, J. Texture analysis: an adaptive probabilistic approach. In: *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*. [S.l.: s.n.], 2003. v. 2, p. II–1045–8 vol.3. ISSN 1522-4880.

CHEN, C.; PAU, L.; WANG, P.; WANG, S. *Handbook of Pattern Recognition & Computer Vision*. World Scientific, 1999. ISBN 9789810230715. Disponível em: <<http://books.google.com.br/books?id=pCghq7-i1oMC> Acessado em: julho de 2013>.

CHORAS, R. S. Feature extraction for cbir and biometrics applications. In: *Proceedings of the 7th Conference on 7th WSEAS International Conference on Applied Computer Science - Volume 7*. Stevens Point, Wisconsin, USA: World Scientific and Engineering Academy and Society (WSEAS), 2007. (ACS'07), p. 1–9. ISBN 978-960-6766-18-3. Disponível em: <<http://dl.acm.org/citation.cfm?id=1348171%-.1348172> Acessado em julho de 2013>.

CLAUSI, D. A.; ZHAO, Y. Rapid extraction of image texture by co-occurrence using a hybrid data structure. *Computers Geosciences*, v. 28, n. 6, p. 763 – 774, 2002. ISSN 0098-3004. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S009830040100108X> Acessado em: junho de 2013>.

CLAUSI, D. A.; ZHAO, Y. Grey level co-occurrence integrated algorithm (glcia): a superior computational method to rapidly determine co-occurrence probability texture features. *Computers Geosciences*, v. 29, n. 7, p. 837 – 850, 2003. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S009830040300089X> Acessado em: junho de 2013>.

DUTTA, S.; DATTA, A.; CHAKLADAR, N. D.; PAL, S.; MUKHOPADHYAY, S.; SEN, R. Detection of tool condition from the turned surface images using an accurate grey level co-occurrence technique. *Precision Engineering*, v. 36, n. 3, p. 458 – 466, 2012. ISSN 0141-6359. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0141635912000505> Acessado em: julho de 2013>.

GIMEL'FARB, G.; ZHOU, D. Texture analysis by accurate identification of a generic markov-gibbs model. In: BUNKE, H.; KANDEL, A.; LAST, M. (Ed.). *Applied Pattern Recognition*. Springer Berlin Heidelberg, 2008, (Studies in Computational Intelligence, v. 91). p. 221–245. ISBN 978-3-540-76830-2. Disponível em: <[http://dx.doi.org/10.1007/978-3-540-76831-9\\_9](http://dx.doi.org/10.1007/978-3-540-76831-9_9) Acessado em: junho de 2013>.

GONZALEZ, R. C.; WOODS, R. E. *Digital Image Processing*. 2nd. ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1992. ISBN 0201508036.

GUNN, S. R. *Support Vector Machines for Classification and Regression*. [S.l.], 1998.

HARALICK, R. Statistical and structural approaches to texture. *Proceedings of the IEEE*, v. 67, n. 5, p. 786–804, 1979. ISSN 0018-9219.

HARALICK, R.; SHANMUGAM, K.; DINSTEN, I. Textural features for image classification. *Systems, Man and Cybernetics, IEEE Transactions on*, SMC-3, n. 6, p. 610–621, 1973. ISSN 0018-9472.

HEIKKILÄ, M.; PIETIKÄINEN, M.; SCHMID, C. Description of interest regions with local binary patterns. *Pattern Recogn.*, Elsevier Science Inc., New York, NY, USA, v. 42, n. 3, p. 425–436, mar. 2009. ISSN 0031-3203. Disponível em: <<http://dx.doi.org/10.1016/j.patcog.2008.08.014> Acessado em: julho de 2013>.

KIM, D.; KIM, M.; KIM, B.; LEE, K. Histograms of local intensity differences for pedestrian classification in far-infrared images. *Electronics Letters*, v. 49, n. 4, p. 258–260, 2013. ISSN 0013-5194.

LU, S.; FU, K. A syntactic approach to texture analysis. *Computer Graphics and Image Processing*, v. 7, n. 3, p. 303 – 330, 1978. ISSN 0146-664X. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0146664X7880001X> Acessado em: junho de 2013>.

MATERKA, A.; STRZELECK, M. *Texture analysis methods - a review*. [S.l.], 1998.

OJALA, T.; VALKEALAHTI, K.; OJA, E.; PIETIKÄINEN, M. Texture discrimination with multidimensional distributions of signed gray-level differences. *Pattern Recognition*, v. 34, n. 3, p. 727 – 739, 2001. ISSN 0031-3203. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0031320300000108> Acessado em: junho de 2013>.

QUEIROZ, J. E. R. de; GOMES, H. M. Introdução ao processamento digital de imagens. *RITA*, v. 13, n. 2, p. 11–42, 2006.

SALEM, Y. B.; NASRI, S. Automatic recognition of woven fabrics based on texture and using svm. *Signal, Image and Video Processing*, Springer-Verlag, v. 4, n. 4, p. 429–434, 2010. ISSN 1863-1703. Disponível em: <<http://dx.doi.org/10.1007/s11760-009-0132-5> Acessado em: maio de 2013>.

SCURI, A. E. *Fundamentos da Imagem Digital*. Dissertação (Mestrado) — PUC - RJ, 2002.

SRINIVASAN, G.; SHOBHA, G. Statistical texture analysis. *proceedings of world academy of science, engg & tech*, v. 36, 2008.

TIMM, F.; MARTINETZ, T. Statistical fourier descriptors for defect image classification. In: *Pattern Recognition (ICPR), 2010 20th International Conference on*. [S.l.: s.n.], 2010. p. 4190–4193. ISSN 1051-4651.

WOUWER, G. Van de; SCHEUNDERS, P.; DYCK, D. V. Statistical texture characterization from discrete wavelet representations. *Image Processing, IEEE Transactions on*, v. 8, n. 4, p. 592–598, 1999. ISSN 1057-7149.

ZHANG, J.; TAN, T. Brief review of invariant texture analysis methods. *Pattern Recognition*, v. 35, n. 3, p. 735 – 747, 2002. ISSN 0031-3203. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0031320301000747> Acessado em: junho de 2013>.

ZHANG, Y. J. A review of recent evaluation methods for image segmentation. In: *Signal Processing and its Applications, Sixth International, Symposium on*. 2001. [S.l.: s.n.], 2001. v. 1, p. 148–151 vol.1.