

Moab Mariz Meira



Classificação de Texturas usando o Algoritmo  
Lempel-Ziv-Welch

João Pessoa

21 de julho de 2008

Moab Mariz Meira

**Classificação de Texturas usando o Algoritmo Lempel-Ziv-Welch**

Dissertação apresentada ao Programa de Pós-Graduação em Informática do Centro de Ciências Exatas e da Natureza da Universidade Federal da Paraíba como requisito parcial para obtenção do grau de Mestre em Informática.

Orientador: Leonardo Vidal Batista.

DI – Departamento de Informática, CCEN – Centro de Ciências Exatas e da Natureza, UFPB – Universidade Federal da Paraíba

João Pessoa

21 de julho de 2008

# Resumo

Este trabalho apresenta um novo e eficiente método de classificação de texturas usando o algoritmo de compressão sem perdas *Lempel-Ziv-Welch* (LZW). Na fase de aprendizagem, o LZW constrói dicionários para as estruturas horizontal e vertical de cada classe de textura. Na fase de classificação, amostras de texturas são codificadas com o LZW no modo estático, usando os dicionários construídos na fase anterior. Uma amostra é associada à classe cujo dicionário conduz à melhor taxa de codificação. O classificador foi avaliado para vários tamanhos do conjunto de treinamento e das amostras de treinamento, e sob diferentes condições de iluminação das texturas. O método proposto atinge 100% de acerto em alguns experimentos usando amostras de texturas do álbum *Brodatz*. Comparações diretas com outros trabalhos indicam a superioridade do método sobre outros métodos de alto desempenho.

# Sumário

|   |           |
|---|-----------|
| <b>RESUMO</b>                                 | <b>2</b>  |
| <b>SUMÁRIO</b>                                | <b>3</b>  |
| <b>LISTA DE FIGURAS E GRÁFICOS</b>            | <b>5</b>  |
| <b>LISTA DE TABELAS</b>                       | <b>6</b>  |
| <b>LISTA DE SIGLAS E ABREVIATURAS</b>         | <b>7</b>  |
| <b>1 INTRODUÇÃO</b>                           | <b>8</b>  |
| <b>2 FUNDAMENTAÇÃO TEÓRICA</b>                | <b>18</b> |
| 2.1 ENTROPIA                                  | 18        |
| 2.2 O ALGORITMO LZW                           | 20        |
| 2.3 EQUALIZAÇÃO DE HISTOGRAMA                 | 26        |
| <b>3 MATERIAIS E MÉTODOS</b>                  | <b>32</b> |
| 3.1 FASE DE TREINAMENTO                       | 32        |
| 3.2 FASE DE CLASSIFICAÇÃO                     | 33        |
| 3.3 ROTAÇÃO                                   | 34        |
| 3.4 BANCO DE TEXTURAS                         | 35        |
| 3.5 VERSÕES DO CLASSIFICADOR                  | 37        |
| 3.5.1 EXPERIMENTOS SEM VARIAÇÃO DE ILUMINAÇÃO | 37        |

---

|            |  |           |
|------------|--|-----------|
| 3.5.2      | EXPERIMENTOS COM VARIAÇÃO UNIFORME DE ILUMINAÇÃO     | 39        |
| 3.5.3      | EXPERIMENTOS COM VARIAÇÃO NÃO UNIFORME DE ILUMINAÇÃO | 40        |
| 3.5.4      | EXPERIMENTOS CBIR                                    | 41        |
| 3.5.5      | EXPERIMENTOS COM ROBUSTEZ À ROTAÇÃO                  | 43        |
| <b>3.6</b> | <b>RECURSOS EMPREGADOS</b>                           | <b>44</b> |
| 3.6.1      | SOFTWARE   | 44        |
| 3.6.2      | HARDWARE   | 44        |
| <b>4</b>   | <b>RESULTADOS</b>                                    | <b>46</b> |
| <hr/>      |  |           |
| 4.1        | EXPERIMENTOS SEM VARIAÇÃO DE ILUMINAÇÃO              | 46        |
| 4.2        | EXPERIMENTOS COM VARIAÇÃO UNIFORME DE ILUMINAÇÃO     | 48        |
| 4.3        | EXPERIMENTOS COM VARIAÇÃO NÃO UNIFORME DE ILUMINAÇÃO | 50        |
| 4.4        | EXPERIMENTOS CBIR                                    | 52        |
| 4.5        | EXPERIMENTOS COM ROBUSTEZ À ROTAÇÃO                  | 54        |
| <b>5</b>   | <b>DISCUSSÃO E CONCLUSÕES</b>                        | <b>57</b> |
| <hr/>      |  |           |
| <b>6</b>   | <b>REFERÊNCIAS BIBLIOGRÁFICAS</b>                    | <b>61</b> |
| <hr/>      |  |           |
| <b>A</b>   | <b>ARTIGOS PUBLICADOS</b>                            | <b>67</b> |
| <hr/>      |  |           |

## Lista de Figuras e Gráficos

|              |    |
|--------------|----|
| Fig. 1.....  | 9  |
| Fig. 2.....  | 10 |
| Fig. 3.....  | 26 |
| Fig. 4.....  | 27 |
| Fig. 5.....  | 28 |
| Fig. 6.....  | 29 |
| Fig. 7.....  | 30 |
| Fig. 8.....  | 33 |
| Fig. 9.....  | 34 |
| Fig. 10..... | 36 |
| Fig. 11..... | 37 |
| Fig. 12..... | 39 |
| Fig. 13..... | 43 |
| Fig. 14..... | 48 |
| Fig. 15..... | 49 |
| Fig. 16..... | 50 |
| Fig. 17..... | 51 |
| Fig. 18..... | 53 |
| Fig. 19..... | 53 |
| Fig. 20..... | 54 |
| Fig. 21..... | 55 |
| Fig. 22..... | 55 |

# Lista de Tabelas

|               |    |
|---------------|----|
| TABELA 1..... | 25 |
| TABELA 2..... | 46 |
| TABELA 3..... | 47 |
| TABELA 4..... | 48 |
| TABELA 5..... | 52 |
| TABELA 6..... | 56 |

## Lista de Siglas e Abreviaturas

|             |  |
|-------------|--|
| ACGMRF      | Anisotropic Circular Gaussian Markov Random Fields |
| AR          | Average Recognition Rate                           |
| CBIR        | Content-based Image Retrieval                      |
| CBIR-LZW    | Classificador CBIR                                 |
| CBIR-LZW-HE | Classificador CBIR com Equalização de Histograma   |
| CCR         | Correct Classification Rate                        |
| CLZW        | Classificador sem Equalização de Histograma        |
| CLZW-GHE    | Classificador com Equalização de Histograma Global |
| CLZW-LHE    | Classificador com Equalização de Histograma Local  |
| CLZW-ROT    | Classificador Robusto à Rotação                    |
| GLCM        | Gray Level Co-occurrences Matrices                 |
| HTML        | HyperText Markup Language                          |
| LVQ         | Learning Vector Quantization                       |
| LZ78        | Lempel-Ziv 78                                      |
| LZW         | Lempel-Ziv-Welch                                   |
| MPEG-7      | Multimedia Content Description Interface           |
| MRF         | Markov Random Fields                               |
| PPM         | Prediction by Partial Matching                     |
| RC          | Razão de Compressão                                |
| SVM         | Support Vector Machine                             |

# 1

## Introdução

Forma, cor, textura e movimento são os atributos fundamentais usados pelo sistema visual humano e sistemas de visão computacional para segmentação, classificação e interpretação de cenas [Porat e Zeevi 1989].

Textura é uma propriedade de uma determinada região da imagem suficientemente grande para demonstrar sua natureza [Parker 1997]. O entendimento dessa região como textura depende do ponto de vista do observador: uma folha pode ser considerada um objeto, entretanto, a folhagem de uma árvore pode ser vista como uma textura [Forsyth e Ponce 2002].

Uma definição popular, porém vaga, diz que as texturas são padrões espaciais formados por repetições de alguns sub-padrões básicos, denominados de *textons* [Baheerathan *et al.* 1999]. Os textons se repetem de forma mais ou menos precisa dentro de um mesmo tema. Cor, tamanho, formato e orientação dos textons podem variar sobre a região.

Textons são microestruturas fundamentais em imagens naturais genéricas, constituindo um dos elementos básicos da percepção visual [Zhu *et al.* 2002] e está para a caracterização de texturas assim como o fonema está para a o reconhecimento de voz [Malik *et al.* 1999]. Outras definições matematicamente mais precisas foram estudadas: textons são definidos como pequenos *templates* que consis-

tem em um número variável de imagens baseadas em algumas configurações geométricas e fotométricas [Zhu *et al.* 2002], ou como combinações freqüentemente ocorrentes de filtros de saídas orientados [Malik *et al.* 1999] .

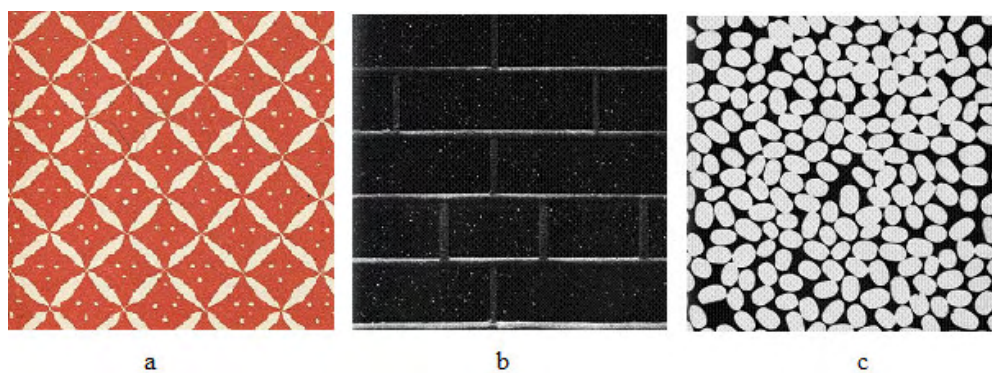
A Figura 1 apresenta amostras de algumas classes de texturas tais como: madeira, tijolos, tecido e água. Estas amostras foram obtidas de um dos bancos de dados de imagens de texturas mais utilizados na literatura, o *VisTex* [VisTex].



**Fig. 1.** Exemplos de texturas.

Texturas podem ser divididas, quanto à regularidade de algumas de suas dimensões (cor, intensidade, geometria), em três tipos: texturas regulares, aquelas com padrões periódicos como papel de parede; texturas quase regulares, que são texturas com pequenas distorções estruturais geométricas, mas com consideráveis diferenças estatísticas em cor e intensidade em relação ao padrão de texturas regulares; e texturas irregulares, onde se pode encontrar deformações tanto estruturais quanto na cor e intensidade comparadas às texturas regulares [Liu e Lin 2003].

Exemplos de cada um dos tipos de texturas descritas anteriormente, podem ser vistos na Figura 2. A imagem de textura regular foi retirada do banco de dados de texturas da *Pennsylvania State University* [CMU NRT], enquanto as outras são do álbum *Brodatz* [Brodatz 1966].



**Fig. 2.** (a) regular, (b) quase regular, (c) irregular.

Em algumas áreas, tais como sensoriamento remoto [Dell'Acqua e Gamba 2003, Augusteijn *et al.* 1995], diagnósticos médicos baseados em imagens [Southard e Southard 1996], automação [Kumar e Pang 2002] e reconhecimento biométrico [Jain e Prabhakar 2004, He *et al.* 2004], tem sido de grande interesse o desenvolvimento de métodos para reconhecimento de padrões que se baseiam em texturas.

A classificação de texturas consiste em associar uma amostra desconhecida de uma textura  $\mathbf{x}$  a uma de  $N$  classes de texturas  $C_i$ ,  $i = 1, 2, \dots, N$ . A caracterização da classe é feita através da discriminação de características ou modelos estruturais associados à textura e a sua classificação depende de alguma medida de similaridade entre  $\mathbf{x}$  e cada uma das classes.

Nas classificações supervisionadas, amostras conhecidas de texturas da classe  $C_i$ ,  $i = 1, 2, \dots, N$  estão disponíveis. Estas amostras pré-classificadas são usadas para construir modelos ou definir características discriminantes para cada classe, em um processo conhecido como treinamento ou aprendizagem. Para desenvolver classificadores de texturas eficientes, é importante descobrir atributos adequados ou modelos precisos para caracterizar cada classe de textura.

Quando o ser humano observa uma imagem, pode facilmente conectar regiões que têm níveis de cinza, cores ou padrões de repetição similares, geralmente associados a classes de objetos na imagem. De fato, mesmo com variação de luz, podem-se distinguir facilmente texturas.

Embora sejam intuitivamente reconhecíveis pelo sistema visual humano, as texturas não são facilmente caracterizadas formalmente. O problema reside na dificuldade intrínseca de definir o que é mais relevante na caracterização de uma textura, pelo fato da resposta depender de considerações de percepção subjetivas e aplicações específicas. Portanto, a extração de características de texturas e modelagem tende a ser difícil e dependente de aplicação.

Selecionar características inadequadas pode degradar o desempenho da classificação, e selecionar características demais, mesmo apropriadas, gera consequências negativas, tais como incremento no tempo de processamento e requerimentos de memória, além de degradação da precisão devido a um fenômeno estatístico conhecido como “maldição da dimensionalidade” [Duda *et al.* 2000].

As primeiras técnicas de extração de características de texturas, tais como a GLCM (*Gray Level Co-occurrences Matrices*) [Bovik 2000], eram em geral focadas em estatísticas espaciais. Mais recentemente, técnicas baseadas em modelos estatísticos, como MRF (*Markov Random Fields*) [Duda *et al.* 2000, Cohen *et al.* 1991], e técnicas de processamento de sinais, como transformadas *wavelets* e de *Gabor* têm sido investigadas [Randen e Husøy 1999]. Comparações extensivas entre os vários métodos de extração de características concluíram que não há método claramente superior aos demais em todas as situações reais [Randen e Husøy 1999].

Após definir as características ou modelos que diferenciam as classes de texturas, um método de classificação deve ser escolhido. Um grande número de classificadores tem sido proposto nas últimas décadas [Duda *et al.* 2000]. Um classificador SVM (*Support Vector Machine*) denominado *fused SVM*, com características geradas por transformadas *wavelet* discretas por quadro obteve desempenho superior aos classificadores SVM simples, classificadores de *Bayes* usando distância de *Bayes* e distância de *Mahalanobi*, e classificadores LVQ (*Learning Vector Quantization*) [Li *et al.* 2003].

O desenvolvimento de métodos de classificação de texturas eficazes e com baixa complexidade computacional é uma linha de pesquisa muito útil [Randen e Husøy 1999]. Em aplicações tais como inspeção industrial, o *throughput* é muito alto e requer um método rápido e suporte de hardware especial [Kumar e Pang 2002].

Os modernos algoritmos de compressão sem perdas têm sido aplicados em problemas de classificação devido à sua capacidade de construir modelos estatísticos precisos, em alguns casos com baixos requisitos computacionais [Bell *et al.* 1990]. Uma sólida fundamentação teórica para o uso do LZ78 [Ziv e Lempel 1978] e outros algoritmos de compressão baseados em dicionários [Bell *et al.*

1990] para classificação está bem estabelecida [Ziv 1988, Ziv e Merhav 1993]. Entretanto, a despeito da fundamentação teórica, a aplicação de esquemas de compressão em problemas práticos de classificação parece ainda largamente inexplorada.

Um problema em potencial associado com algoritmos de compressão sem perdas baseados em dicionário para classificação de texturas é o fato de que estes métodos procuram, no dicionário, por cadeias iguais àquelas da mensagem a ser comprimida. Um dicionário construído de forma precisa para uma dada classe de textura pode apresentar um desempenho fraco quando usado para comprimir uma nova amostra da mesma classe que apresente desvios em seus níveis de cinza causados por ruídos na digitalização ou variação na iluminação.

Até mesmo quando os desvios nos níveis de cinza são sutis tem-se um desempenho ruim. Isso indica que o modelo construído pode não ser capaz de descrever adequadamente a nova amostra da textura, conseqüentemente resultando numa classificação pouca precisa.

Duas possíveis soluções para este problema são:

1. Adoção de um compressor com perdas baseado em dicionário [Finamore e Leister 1996], menos sensível a pequenos desvios nos níveis de cinza.
2. Redução desses desvios pelo uso de técnicas de processamento de imagens antes da utilização de um algoritmo de compressão sem perdas baseado em dicionário.

A equalização de histograma é uma operação não-linear bastante conhecida que gera uma distribuição aproximadamente uniforme dos níveis de cinza na escala disponível [Bovik 2000]. Este procedimento tende a agrupar valores similares de nível de cinza pouco freqüentes em um mesmo valor, dessa forma reduzindo os pequenos desvios nos níveis de cinza que impedem que se localizem casa-

mentos exatos na fase de busca dos algoritmos de compressão sem perdas baseados em dicionário. A operação também diminui a probabilidade do classificador discriminar classes de texturas pela média ou variância dos níveis de cinza, ao invés das propriedades naturais das texturas [Randen e Husøy 1999]. Isso permite uma avaliação mais precisa das capacidades do método para diferenciação dos atributos da textura.

Em geral, as texturas a serem classificadas podem estar em diferentes escalas, orientações e condições de iluminação (níveis de cinza). Por esta razão, tem sido elevado o esforço para desenvolvimento de classificadores invariantes a uma, duas ou, mais raramente, todas essas três alterações [Ojala *et al.* 2002].

Chen e Kundu [Chen e Kundu 1994] abordaram o problema de variação nos níveis de cinza usando equalização de histograma global, mas esta solução é muito limitada, pelo fato de não corrigir variações não-uniformes. Por outro lado, a equalização de histograma local, que é aplicada como uma janela deslizante de dimensões pré-definidas, tem sido usada para reduzir o efeito da iluminação não-uniforme [Chen e Kundu 1994, Gonzales e Woods 1992]. Entretanto, a equalização de histograma local tende a provocar ruídos, especialmente para janelas de dimensões reduzidas [Gonzales e Woods 1992].

Outros compressores sem perdas são utilizados em problemas de classificação, um deles que tem obtido bons resultados é o PPM. O compressor sem perdas PPM (*prediction by partial matching*), é uma técnica de modelagem estatística de contexto finito que pode ser vista como uma combinação de alguns modelos de contexto de ordem fixa para prever o próximo caractere na seqüência de entrada [Cleary *et al.* 1995]. As probabilidades de predição em cada contexto no modelo são calculadas através da contagem de freqüência que são adaptativamente atualizadas; e o símbolo que realmente aparece é codificado de acordo com a sua distri-

buição estimada, condicionada ao contexto, usando codificação aritmética [Cleary *et al.* 1995].

De acordo com alguns autores, quando aplicado a tarefas específicas de reconhecimento de texto, o PPM categoriza corretamente a maioria dos documentos, mas não obtém uma precisão competitiva comparado com os esquemas de aprendizagem de máquina encontrados no estado da arte [Frank *et al.* 2000]. Por outro lado, outros trabalhos concluíram que o PPM foi aplicado com sucesso em vários problemas de classificação de textos, com desempenho comparável a outros métodos no estado da arte [Teahan e Harper 2001]. Uma das principais desvantagens do PPM deriva da exigência do algoritmo por altos requisitos computacionais [Bell *et al.* 1990].

Um classificador baseado em modelos construídos por algoritmos de compressão tem algumas vantagens em relação aos métodos clássicos de aprendizagem de máquina: como não há seleção de características, nenhuma informação é descartada, já que os modelos descrevem cada classe como um todo [Frank *et al.* 2000]; não há necessidade de se fazer suposições sobre as probabilidades de distribuição das classes de textura; a capacidade de construção de modelos adaptativos dos algoritmos de compressão oferece uma maneira uniforme de classificar diferentes tipos de fontes [Frank *et al.* 2000]; o método de classificação é muito simples [Teahan e Harper 2001].

Uma das aplicações de um classificador de imagens está associada à recuperação de imagens baseada em conteúdo. A associação de palavras-chave é o paradigma mais tradicional. Nesta abordagem, as imagens são associadas manualmente às palavras-chave (informações textuais relevantes sobre o conteúdo das imagens). Elas podem então ser recuperadas por meio da análise de nomes de arquivos; *tags* ‘META’ e os atributos ‘ALT’ das *tags* ‘IMG’ nos códigos HTML das páginas; termos próximos às imagens; natureza e orientação dos sites e prove-

dores, etc.. Entretanto, existem três dificuldades principais nesta abordagem: o alto esforço manual utilizado no processo de associação, as diferenças nas interpretações dos conteúdos das imagens e as inconsistências das palavras-chave associadas a diferentes indexadores [Faloutsos *et al.* 1993, Flickner *et al.* 1995]. À medida que o tamanho dos repositórios de imagens cresce, essa abordagem torna-se inviável.

O grande crescimento de tecnologias na área de imagem digital nos últimos anos trouxe a necessidade de se superar as dificuldades dessa abordagem. Um mecanismo alternativo de recuperação de imagens baseada em conteúdo (CBIR - *content-based image retrieval*) foi proposto na década de 1990 e desde então muitos esforços têm sido feitos no seu desenvolvimento. As ferramentas CBIR extraem e analisam características pictóricas tais como forma, cor e textura [Zibreira 2000]. Esta abordagem tem a vantagem de obter características automaticamente.

O paradigma de recuperação de imagens CBIR tem atraído grande atenção, variando desde pesquisas governamentais [Jain 1993, Jain *et al.* 1995] e industriais [Bach *et al.* 1996, Dowe 1993, Flickner *et al.* 1995], a universidades [Huang *et al.* 1996, Ma e Manjunath 1999, Mandal *et al.* 1997]. A ISO/IEC definiu o MPEG-7 [ISO/IEC *et al.* 1997a, ISO/IEC *et al.* 1997b, ISO/IEC *et al.* 1997c] para englobar uma interface descritora de conteúdo multimídia padronizado. Muitos sistemas CBIR para uso comercial têm sido desenvolvidos recentemente [Bach *et al.* 1996, Dowe 1993, Faloutsos *et al.* 1993].

Nos trabalhos de Flickner e Addis [Flickner *et al.* 1995, Addis 2002], ferramentas que abordam alguns aspectos relevantes do CBIR são apresentados. Essas ferramentas foram aplicadas na construção de um museu na Web e de aplicações de galeria na Web respectivamente. As ferramentas permitiam a busca por trabalhos artísticos através de uma seleção de cores predominantes a partir de uma paleta ou através do desenho de formas em uma tela.

O objetivo geral do trabalho proposto é apresentar um novo e eficiente método para classificação de texturas usando o *Lempel-Ziv-Welch* (LZW) [Welch 1984], um algoritmo de compressão sem perdas. Para validar essa proposta foram definidos os seguintes objetivos específicos:

- Avaliar o classificador para vários tamanhos de amostras de treinamento e conjuntos de treinamento;
- Demonstrar a importância da técnica de equalização de histograma no desempenho global do método;
- Comparar o trabalho com outros métodos de classificação de texturas no estado da arte, utilizando os mesmos corpos de testes;
- Tornar o classificador invariante à rotação e às condições de iluminação das texturas (variação dos níveis de cinza da imagem).

Este capítulo introduziu e definiu um dos atributos fundamentais para sistemas de visão computacional: textura, além de exemplos da aplicação da classificação de texturas na solução de problemas em diversas áreas. O processo de classificação, envolvendo os estágios de treinamento e teste, foi detalhado abordando-se desde as dificuldades envolvidas, tais como a necessidade de invariância às diversas alterações, e suas devidas soluções até a fundamentação para o uso de algoritmos de compressão sem perdas como procedimento de extração de características das texturas. Alguns métodos no estado da arte para caracterização e classificação de texturas também foram citados.

O texto está dividido nos seguintes tópicos: o capítulo 2 apresenta alguns conceitos fundamentais; o terceiro capítulo descreve o método detalhadamente; o capítulo 4 demonstra a avaliação empírica do classificador proposto e o capítulo 5 discute os resultados e apresenta as conclusões.

## Fundamentação Teórica

Informações sobre entropia e processos de *Markov* são abordadas nesse capítulo de fundamentação teórica. Além disso, o algoritmo LZW e a técnica de Equalização de Histograma, utilizados no método em estudo nessa dissertação, são discutidos em detalhes.

### 2.1 Entropia

Seja  $S$  uma fonte de informação estacionária discreta que gera mensagens sobre um alfabeto finito  $A = \{a_1, a_2, \dots, a_M\}$ . A fonte escolhe sucessivos símbolos de  $A$  de acordo com alguma distribuição de probabilidade que depende, em geral, de símbolos selecionados anteriormente. Uma mensagem genérica é modelada como um processo estocástico estacionário  $\mathbf{x} = \dots x_{-2} x_{-1} x_0 x_1 x_2 \dots$ , com  $x_i \in A$ . Seja  $\mathbf{x}^n = x_1 x_2 \dots x_n$  uma mensagem de tamanho  $n$ . Se a cardinalidade de  $A$  é  $|A| = M$ , a fonte pode gerar  $M^n$  diferentes mensagens de tamanho  $n$ . Seja  $\mathbf{x}_i^n$ ,  $i = 1, 2, \dots, M^n$  a  $i$ -ésima mensagem, de acordo com alguma ordem. Assumindo que a fonte segue uma distribuição de probabilidade  $P$ , então  $\mathbf{x}_i^n$  é gerado com probabilidade  $P(\mathbf{x}_i^n)$ .

Seja

$$G_n(P) = -\frac{1}{n} \sum_{i=1}^{M^n} P(\mathbf{x}_i^n) \log_2 P(\mathbf{x}_i^n) \quad (1)$$

em que  $G_n(P)$  decresce monotonicamente com  $n$  [Shannon 1948]. Assim, pode-se definir a entropia da fonte  $H(P)$  dada por:

$$H(P) = \lim_{n \rightarrow \infty} G_n(P) \quad \text{bits/símbolo.} \quad (2)$$

Uma formulação alternativa para  $H(P)$  usa probabilidades condicionais. Seja  $P(\mathbf{x}_i^{n-1}, a_j)$  a probabilidade da seqüência  $\mathbf{x}_i^n = (\mathbf{x}_i^{n-1}, a_j)$ , resultante da concatenação de  $\mathbf{x}_i^{n-1}$  com o símbolo  $x_n = a_j$ , e  $P(a_j | \mathbf{x}_i^{n-1}) = P(\mathbf{x}_i^{n-1}, a_j)P(\mathbf{x}_i^{n-1})$  a probabilidade do símbolo  $x_n = a_j$  condicionada à ocorrência de  $\mathbf{x}_i^{n-1}$ . A entropia da aproximação de  $n$ -ésima ordem para  $H(P)$  [Shannon 1948] é:

$$F_n(P) = - \sum_{i=1}^{M^n} \sum_{j=1}^M P(\mathbf{x}_i^{n-1}, a_j) \log_2 P(a_j | \mathbf{x}_i^{n-1}) \quad \text{bits/símbolo.} \quad (3)$$

$F_n(P)$  decresce monotonicamente com  $n$  [Shannon 1948] e a entropia da fonte é:

$$H(P) = \lim_{n \rightarrow \infty} F_n(P) \quad \text{bits/símbolo.} \quad (4)$$

A Equação 4 envolve a estimação de probabilidades condicionadas a uma seqüência infinita de símbolos anteriores. Na prática é assumida memória finita, e as fontes são modeladas por um processo de *Markov* de ordem  $n-1$ , de forma que  $P(a_j | \dots x_{-1}x_0x_1 \dots x_{n-1}) = P(a_j | x_1, \dots x_{n-1})$ . Neste caso,  $H(P) = F_n(P)$ .

O conceito de entropia como uma medida de informação é central para a Teoria da Informação [Shannon 1948], e a compressão de dados provê uma perspectiva intuitiva deste conceito. Baseando-se nesse fato, define-se a taxa de codificação de um esquema de codificação como o número médio de bits por símbolo que este usa para codificar a saída da fonte. Um compressor sem perdas é um esquema de codificação unicamente decodificável cujo objetivo é atingir a menor taxa de codificação possível. Por esta razão, a taxa de compressão de qualquer procedimento de codificação unicamente decodificável é sempre maior que ou

igual à entropia da fonte [Shannon 1948]. Métodos ótimos de codificação têm uma taxa de codificação igual ao limite teórico mais baixo  $H(P)$ , atingindo assim compressão máxima.

Para processos de *Markov* de ordem  $n-1$ , a codificação ótima é obtida se e somente se um símbolo  $x_n = a_j$  ocorrendo depois de  $\mathbf{x}_i^{n-1}$  é codificado com  $-\log_2 P(a_j | \mathbf{x}_i^{n-1})$  bits [Bell *et al.* 1990, Shannon 1948]. Entretanto, pode ser impossível estimar precisamente a distribuição condicional  $P(\cdot | \mathbf{x}_i^{n-1})$  para valores grandes de  $n$ , devido ao crescimento exponencial do número de contextos diferentes, que trazem problemas bem conhecidos, como a diluição de contexto [Bell *et al.* 1990].

## 2.2 O Algoritmo LZW

Mesmo que o modelo  $P$  da fonte seja geralmente desconhecido, é possível construir um esquema de codificação baseado em algum modelo probabilístico  $Q$ , implícito ou explícito, que se aproxima de  $P$ . Quanto mais  $Q$  se aproximar de  $P$ , menor é a taxa de codificação obtida por um esquema de codificação ótimo.

Objetivando atingir baixas taxas de codificação, os modernos compressores sem perdas têm como base a construção de sofisticados modelos que seguem o verdadeiro modelo da fonte. Compressores estatísticos codificam mensagens de acordo com um modelo estatístico estimado para a fonte. Compressores baseados em dicionário substituem cadeias de símbolos da mensagem a ser codificada por índices de um dicionário de cadeias, que geralmente é construído de maneira adaptativa durante o processo de codificação. Quando *parsing* guloso é usado, em cada passo o codificador procura, no dicionário corrente, pela maior cadeia que corresponda à próxima seqüência de símbolos na mensagem, e substitui esta seqüência pelo índice da maior cadeia encontrada anteriormente no dicionário.

Compressores baseados em dicionário com *parsing* guloso, como o LZW, são largamente empregados porque combinam eficiência computacional com baixas taxas de codificação. Pode-se provar que cada compressor baseado em dicionário com *parsing* guloso tem um codificador estatístico equivalente que atinge exatamente a mesma compressão [Bell *et al.* 1990]. Em codificadores baseados em dicionário, o mesmo embute um modelo estatístico implícito para a fonte.

Em 1978, Ziv e Lempel definiram um esquema de codificação simples capaz de explorar as dependências estatísticas entre os símbolos produzidos por uma fonte de informação. O algoritmo deu origem a uma grande família de compressores práticos, de bom desempenho e baixos requisitos computacionais, amplamente utilizados até hoje [Bell *et al.* 1990].

Considere uma fonte de informação que produza seqüências de símbolos selecionados de um conjunto  $A = \{a_0, a_1, \dots, a_{M-1}\}$ , denominado alfabeto da fonte. Uma *frase* é qualquer seqüência de símbolos selecionados de  $A$ . O tamanho de uma frase é o número de símbolos que a constituem. Neste sentido, uma lista de frases constitui um *dicionário*.

No LZ78 [Ziv e Lempel 1978], o dicionário inicialmente contém apenas uma frase vazia ou nula, ou seja, uma frase de tamanho zero. Esta frase ocupa o índice zero no dicionário. Seja  $\mathbf{x} = x_0x_1x_2\dots$  a seqüência de símbolos a ser codificada, e  $f$  uma frase inicialmente vazia. Os símbolos de  $\mathbf{x}$  são seqüencialmente concatenados a  $f$  até que a concatenação de um dado símbolo  $x_i$  produza uma frase que não se encontre no dicionário. A frase  $f$  é então codificada como um par  $(d, x_i)$ , onde  $d$  é o índice da última frase coincidente no dicionário, e  $f$  é adicionada ao dicionário. A frase  $f$  é novamente anulada e o processo de codificação se repete a partir do símbolo  $x_{i+1}$ . Em muitas implementações, quando há  $D$  frases no dicio-

nário,  $d$  é representado, pela codificação binária ajustada [Ziv e Lempel 1978], com  $\lceil \log_2 D \rceil$  ou com  $\lceil \log_2 D \rceil - 1$  bits, dependendo do valor de  $D$ .

No LZW [Welch 1984], derivado do algoritmo LZ78, o dicionário é inicialmente carregado com todas as frases possíveis de tamanho um. Assim, para um alfabeto  $A = \{a_0, a_1, \dots, a_{M-1}\}$ , o dicionário contém inicialmente  $M$  frases de tamanho unitário. Como no LZ78, os símbolos de  $\mathbf{x}$  são seqüencialmente concatenados a  $f$  até que a concatenação de um dado símbolo  $x_i$  produza uma frase que não se encontre no dicionário. A frase  $f$  é então codificada como o índice  $d$  da última frase coincidente no dicionário e  $f$  é adicionada ao dicionário. A frase  $f$  é novamente anulada e o processo de codificação se repete a partir do símbolo que interrompeu a coincidência,  $x_i$  [Bell *et al.* 1990]. Observe-se que, no algoritmo LZW, os códigos das frases são unicamente os índices das frases coincidentes, representados usando a codificação binária ajustada.

Para descrever em detalhes o LZW, define-se uma variável, inicialmente nula, chamada de *frase*. Na descrição dos processos de codificação e decodificação, que se segue, o símbolo + denota concatenação de *strings*.

### Codificação LZW

1. Pré-carregue o dicionário com todas as *strings* de tamanho 1
2.  $frase := ''$
3.  $s :=$  próximo símbolo do arquivo
4.  $frase + s$  encontra-se no dicionário?
  - Sim:  
 $frase := frase + s$   
Vá para 3
  - Não:  
Envie índice de  $frase$  para o arquivo codificado  
Insira  $frase + s$  no dicionário

$frase := s$

Vá para 3

### Decodificação LZW

1. Pré-carregue o dicionário com todas as *strings* de tamanho 1
2.  $Cod :=$  Primeiro índice da mensagem codificada
3. Envie *string* de número  $Cod$  para a saída
4.  $Cod\_Anterior := Cod$
5.  $Cod :=$  Próximo índice da mensagem codificada
6.  $Cod$  presente no dicionário?

- Sim:

Envie *string* de índice  $Cod$  para a saída

$frase :=$  *string* de índice  $Cod\_Anterior$

$s :=$  Primeiro símbolo da *string* de índice  $Cod$

Adicione  $frase + s$  ao dicionário

$Cod\_Anterior := Cod$

- Não:

$frase :=$  *string* de índice  $Cod\_Anterior$

$s :=$  Primeiro símbolo de  $frase$

Envie  $frase + s$  para a saída e a inclua no dicionário.

$Cod\_Anterior := Cod$

7. Vá para 5

Exemplo: Codificar “Segmentos são os componentes do programa. Finalmente, programas são compostos tam” usando o LZW. Considere que a fonte gera apenas as letras e sinais de pontuação presentes na mensagem, e que o dicionário não precisa ser ordenado.

Dicionário:

|       |        |         |          |          |
|-------|--------|---------|----------|----------|
| 0 ' ' | 10 'i' | 20 'Se' | 30 'sã'  | 40 'on'  |
| 1 ';  | 11 'l' | 21 'eg' | 31 'ão'  | 41 'ne'  |
| 2 '.' | 12 'm' | 22 'gm' | 32 'o '  | 42 'ent' |
| 3 'F' | 13 'n' | 23 'me' | 33 ' o'  | 43 'te'  |
| 4 'S' | 14 'o' | 24 'en' | 34 'os ' | 44 'es'  |
| 5 'a' | 15 'p' | 25 'nt' | 35 ' c'  | 45 's d' |
| 6 'c' | 16 'r' | 26 'to' | 36 'co'  | 46 'do'  |
| 7 'd' | 17 's' | 27 'os' | 37 'om'  | 47 'o p' |
| 8 'e' | 18 't' | 28 's ' | 38 'mp'  | 48 'pr'  |
| 9 'g' | 19 'ã' | 29 ' s' | 39 'po'  | ...      |

Saída codificada: 4, 8, 9, 12, 8, 13, 18, 14, 17, 0, 17, 19, 14, 0, 27, 0, 6, 14, 12, 15, 14, 13, 24, 18, 8, 28, 7, 32, 15, ...

Alguns experimentos, com ambos os compressores (LZ78 e LZW), foram feitos antes do desenvolvimento do sistema de classificação adotado. Essa etapa serviu para ratificação do algoritmo LZW como algoritmo de compressão a ser usado na modelagem.

Os experimentos foram realizados utilizando-se um dos corpos de teste de compressão mais conhecidos na literatura: *Calgary Corpus* [Bell *et al.* 1990]. Trata-se de um conjunto formado pelos mais diversos tipos de arquivos (textos, arquivos representando códigos executáveis, dados geofísicos e figuras). Apesar de ser bastante utilizado em experimentos de compressão de textos [Bell *et al.* 1990], decidiu-se utilizá-lo na comparação entre os dois compressores já citados devido ao seu universo de possibilidades.

A razão de compressão (RC) é definida como a razão entre o número de bits usados na representação do sinal original e o número de bits na representação do sinal comprimido.

Como observado na Tabela 1, vemos que o LZW, com uma RC média de 2,59:1, obtém um desempenho superior ao LZ78 (RC média de 2,11:1). Dessa

forma, elege-se o LZW como algoritmo a ser adotado, uma vez que a RC superior mostra que este algoritmo constrói um modelo mais preciso para compressão do corpo de testes que o LZ78.

O LZW atinge um desempenho assintoticamente ótimo para fontes de *Markov*, visto que a taxa de codificação tende à entropia da fonte à medida que o tamanho da mensagem a ser codificada tende ao infinito [Savari 1997]. Isto significa que o algoritmo progressivamente aprende um modelo melhor para a fonte durante o processo de codificação, e um modelo perfeito para a fonte é aprendido quando uma mensagem infinita, produzida por uma fonte estacionária, for codificada. Na prática, uma vez que as mensagens a serem comprimidas são finitas, o LZW aprende somente um modelo aproximado para a fonte.

TABELA 1.

RC obtida pelo LZ78 e LZW em função dos arquivos contidos no *Calgary Corpus*

| Nome do Arquivo | RC   |      |
|-----------------|------|------|
|                 | LZ78 | LZW  |
| Bib             | 2,06 | 2,59 |
| book1           | 2,01 | 2,56 |
| book2           | 2,07 | 2,65 |
| Geo             | 1,48 | 1,39 |
| News            | 1,83 | 2,25 |
| obj1            | 1,47 | 1,60 |
| obj2            | 1,73 | 1,96 |
| paper1          | 1,75 | 2,23 |
| paper2          | 1,85 | 2,39 |
| paper3          | 1,72 | 2,21 |
| paper4          | 1,60 | 2,01 |
| paper5          | 1,55 | 1,91 |
| paper6          | 1,70 | 2,15 |
| Pic             | 7,24 | 8,54 |
| Progc           | 1,73 | 2,20 |
| Progl           | 2,11 | 2,78 |
| Progp           | 2,07 | 2,73 |
| Trans           | 1,99 | 2,54 |
| RC Média        | 2,11 | 2,59 |

## 2.3 Equalização de Histograma

Se o nível de cinza  $l$  ocorre  $n_l$  vezes em uma imagem com  $n$  pixels, podemos estimar a probabilidade de ocorrência do nível  $l$  na imagem por intermédio de sua frequência relativa:

$$P(l) = \frac{n_l}{n} \quad (5)$$

Assim, o histograma da imagem é uma representação gráfica de  $n_l$  ou de  $P(l)$  em função de  $l$  como visto na Figura 3.

A análise do histograma permite conhecer a distribuição de níveis de cinza de uma imagem. A Figura 4 apresenta histogramas típicos de imagens escuras, claras, e de níveis intermediários.

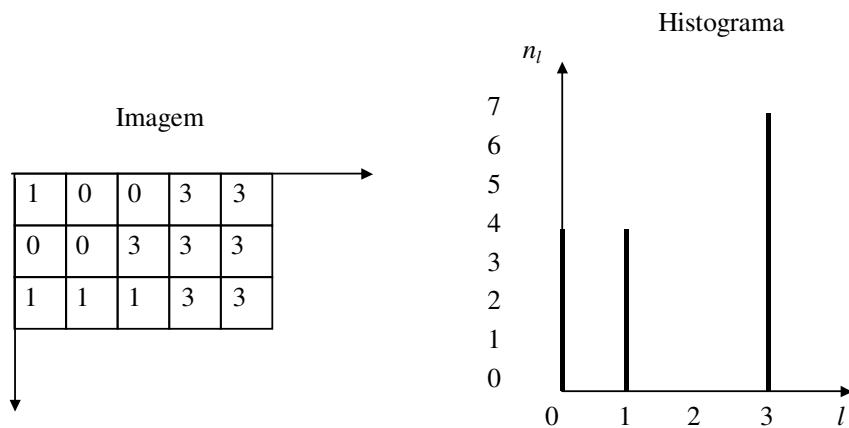


Fig. 3. Representação de uma imagem e seu histograma.

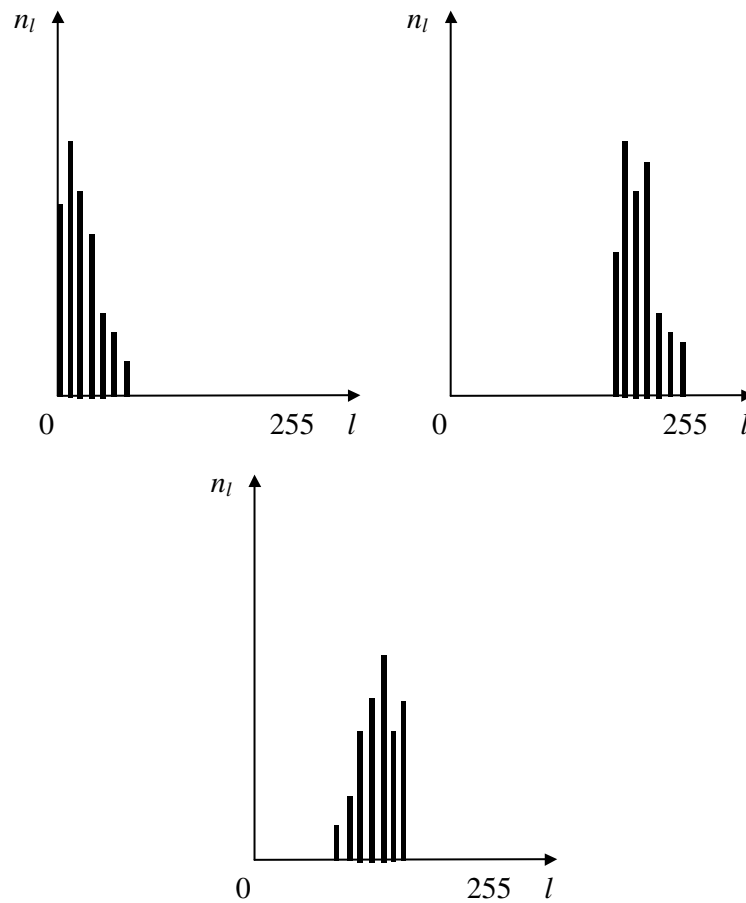


Fig. 4. Histogramas típicos de imagens

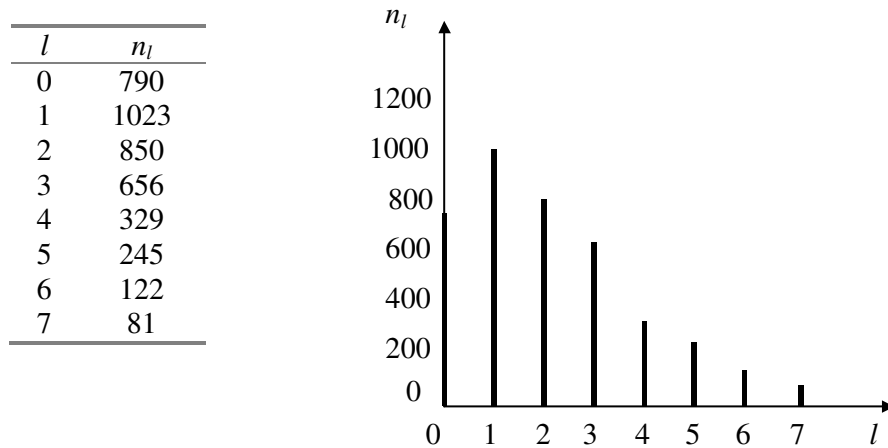
Em todos os casos, a faixa de níveis disponíveis foi inadequadamente utilizada. As técnicas de *equalização de histograma* procuram aumentar a uniformidade da distribuição de níveis de cinza. Estas técnicas são úteis, por exemplo, para realçar as diferenças de tonalidade na imagem e levam, em muitos casos, a um aumento substancial no nível de detalhes perceptíveis.

Um procedimento simples para a equalização do histograma de uma imagem  $R \times C$  com  $L$  níveis de cinza é proporcionado pela transformação:

$$s = T(r) = \text{round}\left(\frac{L-1}{RC} \sum_{l=0}^r n_l\right) \quad (6)$$

onde  $r$  representa os níveis de cinza da imagem original,  $s$  representa os níveis de cinza da imagem equalizada e  $\text{round}$  significa arredondamento para o inteiro mais próximo.

Como exemplo, consideremos uma imagem de resolução  $64 \times 64$  e  $L = 8$  níveis de cinza, com o histograma apresentado na Figura 5.

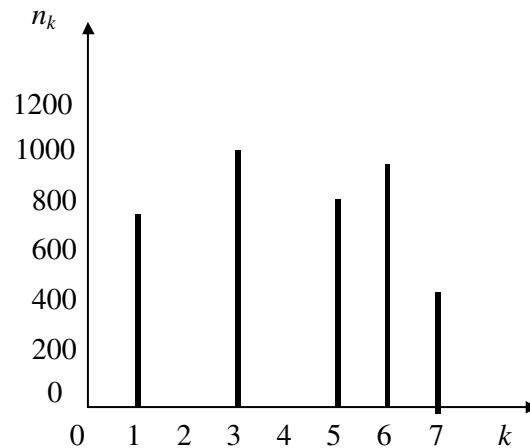


**Fig. 5.** Níveis de cinza e histograma de uma imagem.

A transformação de equalização de histograma fornece os resultados:

$$\begin{aligned} r = 0 &\rightarrow s = \text{round}(790 \times 7 / 4096) = 1 \\ r = 1 &\rightarrow s = \text{round}(1813 \times 7 / 4096) = 3 \\ r = 2 &\rightarrow s = \text{round}(2663 \times 7 / 4096) = 5 \\ r = 3 &\rightarrow s = \text{round}(3319 \times 7 / 4096) = 6 \\ r = 4 &\rightarrow s = \text{round}(3648 \times 7 / 4096) = 6 \\ r = 5 &\rightarrow s = \text{round}(3893 \times 7 / 4096) = 7 \\ r = 6 &\rightarrow s = \text{round}(4015 \times 7 / 4096) = 7 \\ r = 7 &\rightarrow s = \text{round}(4096 \times 7 / 4096) = 7 \end{aligned}$$

Vê-se que o nível de cinza 0 é transformado em 1; o 1 torna-se 3; o 2 torna-se 5; os níveis 3 e 4 são transformados em 6 (que conterà, portanto,  $656 + 329 = 985$  pixels); e o 5, o 6 e o 7 são transformados em 7 (que conterà 448 pixels). O histograma equalizado é apresentado na Figura 6 a seguir.

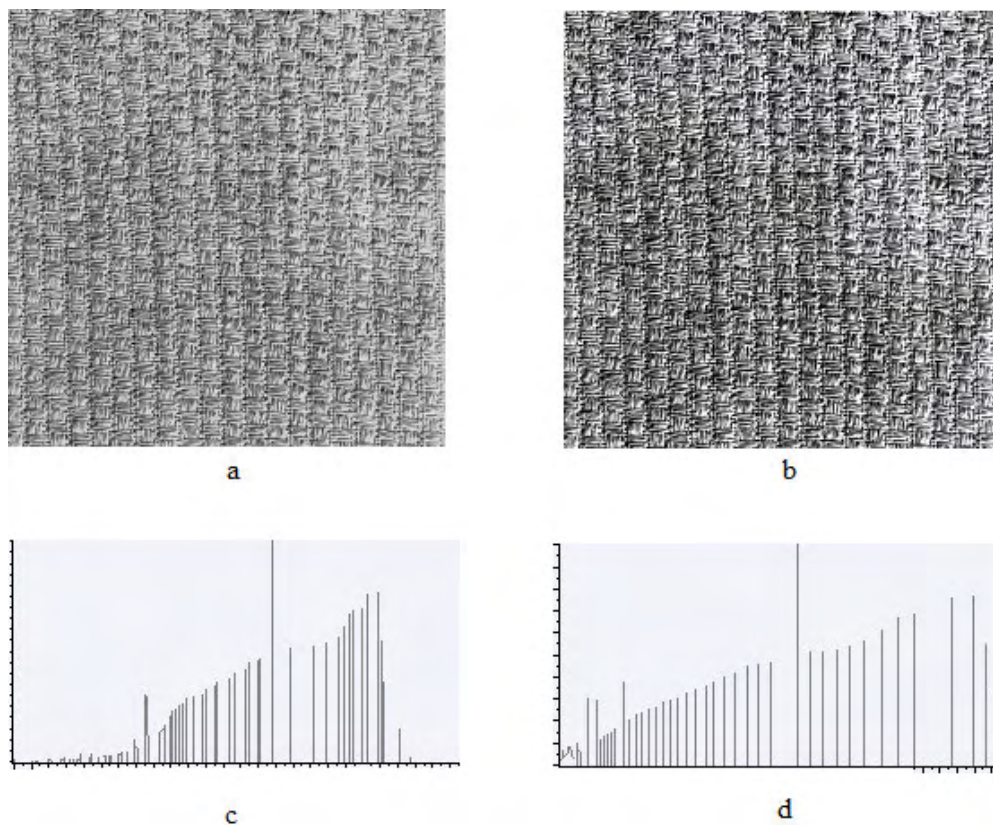


**Fig. 6.** Histograma equalizado de uma imagem.

Observe-se que os níveis de cinza não ocorrem todos com a mesma frequência na imagem resultante, o que caracterizaria uma equalização perfeita, mas o histograma original, característico de uma imagem preponderantemente escura, é transformado em um histograma característico de uma imagem com níveis mais uniformemente distribuídos. Esse resultado mostra a importância da equalização de histograma para o método em estudo: a eliminação de níveis pouco frequentes, que são agrupados em um único nível, resultando em uma diminuição das variações aleatórias causadas por ruído. Dessa forma, espera-se uma melhora significativa no desempenho do classificador com o uso desta técnica.

A Figura 7 mostra uma imagem de textura em sua forma original (Figura 7.a) e depois de ter seu histograma equalizado (Figura 7.b). Além das imagens de textura é possível observar os seus histogramas: Figuras 7.c (histograma da textu-

ra original) e Figura 7.d (histograma da textura equalizada). Nota-se a melhor distribuição dos níveis de cinza depois da equalização e o menor número de níveis de cinza presentes na imagem processada, em relação à imagem original.



**Fig. 7.** Efeitos da aplicação de equalização de histograma em amostra de uma textura. (a) textura original, (b) textura equalizada, (c) histograma da textura original, (d) histograma da textura equalizada.

Uma particularização da técnica de equalização de histograma também é utilizada nesta dissertação. Na equalização de histograma local, apenas uma região da imagem, por vez, sofre o processo de equalização. Blocos distintos de tamanho  $n \times n$  são separados e têm seus histogramas equalizados. Esse processo é utilizado em uma das versões do classificador onde experimentos com variação de iluminação nas texturas são realizados.

O capítulo que se encerra trouxe informações sobre entropia e processos de Markov; o algoritmo LZW e a técnica de Equalização de Histograma. O algoritmo de compressão baseado em dicionário LZW teve o seu processo de codificação exposto em detalhes, com um exemplo prático. Os resultados dos experimentos de compressão do LZW e o seu sucessor, LZ78, foram confrontados, concluindo o LZW como algoritmo para construção dos modelos estruturais associados à textura. Por fim, a apresentação da técnica de Equalização de Histograma utilizado para melhor distribuir os níveis de cinza da textura, a fim de diminuir eventuais efeitos de ruído, o que gera um melhor desempenho do classificador.

Os materiais utilizados nos experimentos para teste, bem como o detalhamento do método proposto e as diversas versões do esquema de classificação visando medir o desempenho do classificador são os temas do próximo capítulo.

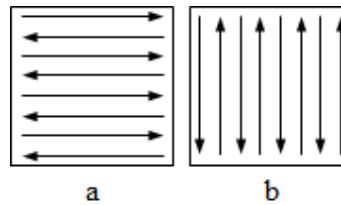
## Materiais e Métodos

Devido a sua capacidade de construir modelos precisos, os modernos compressores sem perda podem ser usados como classificadores baseados em modelos. Qualquer compressor sem perda eficiente poderia ser usado, mas o algoritmo LZW foi escolhido por causa de seu bom compromisso entre eficiência na codificação e requisitos computacionais [Bell *et al.* 1990]. Cinco versões do classificador foram estudadas. A primeira versão usa equalização de histograma global no treinamento e na classificação. O desempenho do método sob efeito de variação uniforme de iluminação nas texturas é testado na segunda versão. A terceira modalidade usa equalização de histograma local, também nas duas fases do processo, no intuito de obter melhores resultados mesmo com variação não uniforme de iluminação sendo aplicada nas amostras. Uma abordagem de recuperação de imagens baseada em conteúdo é o tema de estudo da quarta versão do esquema de classificação. Por fim, a quinta versão verifica a invariância do classificador à rotação.

### 3.1 Fase de Treinamento

Na fase de treinamento, o número  $N$  de classes é definido, e o conjunto de treinamento  $T_i$  de amostras (imagens de tamanho  $n \times n$ ) conhecidas de texturas da classe  $C_i$ , é selecionado, onde  $i = 1, 2, \dots, N$ .

A estratégia adotada consiste de dois trajetos: horizontal e vertical. No primeiro, o algoritmo LZW comprime seqüencialmente todas as amostras em  $T_i$ , seguindo o trajeto horizontal mostrado na Figura 8.a, e o dicionário resultante  $H_i$  é mantido como um modelo para a estrutura horizontal das texturas em  $C_i$ ,  $i = 1, 2, \dots, N$ . Em seguida, o algoritmo LZW comprime todas as amostras em  $T_i$ , seguindo o trajeto vertical mostrado na Figura 8.b, e o dicionário resultante  $V_i$  é mantido como um modelo para a estrutura vertical das texturas em  $C_i$ ,  $i = 1, 2, \dots, N$ .



**Fig. 8.** Trajeto (a) horizontal; e (b) vertical.

### 3.2 Fase de Classificação

Na fase de classificação, o LZW opera no modo estático. Neste modo, um dos dicionários gerados na fase de treinamento é usado e nenhuma nova cadeia é adicionada ao dicionário durante o processo de codificação.

Uma amostra de textura  $\mathbf{x}$  de tamanho  $n \times n$  de uma classe desconhecida é codificada pelo algoritmo LZW com o dicionário estático  $H_i$ , seguindo o trajeto horizontal mostrado na Figura 8.a, e a taxa de codificação correspondente  $h_i$  é registrada,  $i = 1, 2, \dots, N$ . Então o algoritmo LZW com o dicionário estático  $V_i$  codifica  $\mathbf{x}$ , seguindo o trajeto vertical mostrado na Figura 8.b, e a taxa de codificação correspondente  $v_i$  é registrada,  $i = 1, 2, \dots, N$ .

Define-se a taxa de codificação média,  $r_i$ , como:

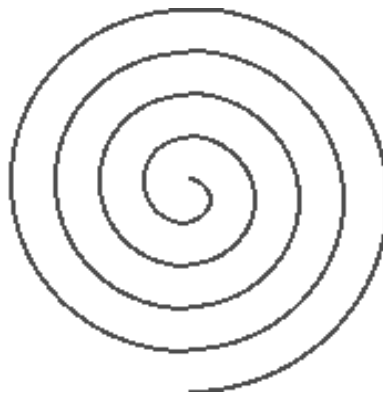
$$r_i = \frac{h_i + v_i}{2} \quad (7)$$

A amostra  $\mathbf{x}$  é associada à classe  $C_i$  se  $r_i < r_j$ ,  $j = 1, 2, \dots, N$ ,  $j \neq i$ . O raciocínio é que se  $\mathbf{x}$  é uma amostra da classe  $C_i$ , os dicionários  $H_i$  e  $V_i$  provavelmente terão embutidos o modelo que melhor descreva suas estruturas horizontal e vertical, assim obtendo a menor taxa de codificação.

### 3.3 Rotação

Na versão do classificador robusto à rotação os dicionários são criados de maneira diferente da apresentada nas seções 3.1 e 3.2.

Na fase de aprendizagem, o algoritmo LZW comprime seqüencialmente as amostras, seguindo uma espiral como ilustrado na Figura 9, e o dicionário resultante é mantido como um modelo para a estrutura robusta à rotação das texturas.



**Fig. 9.** Percurso em espiral usado para a criação do dicionário.

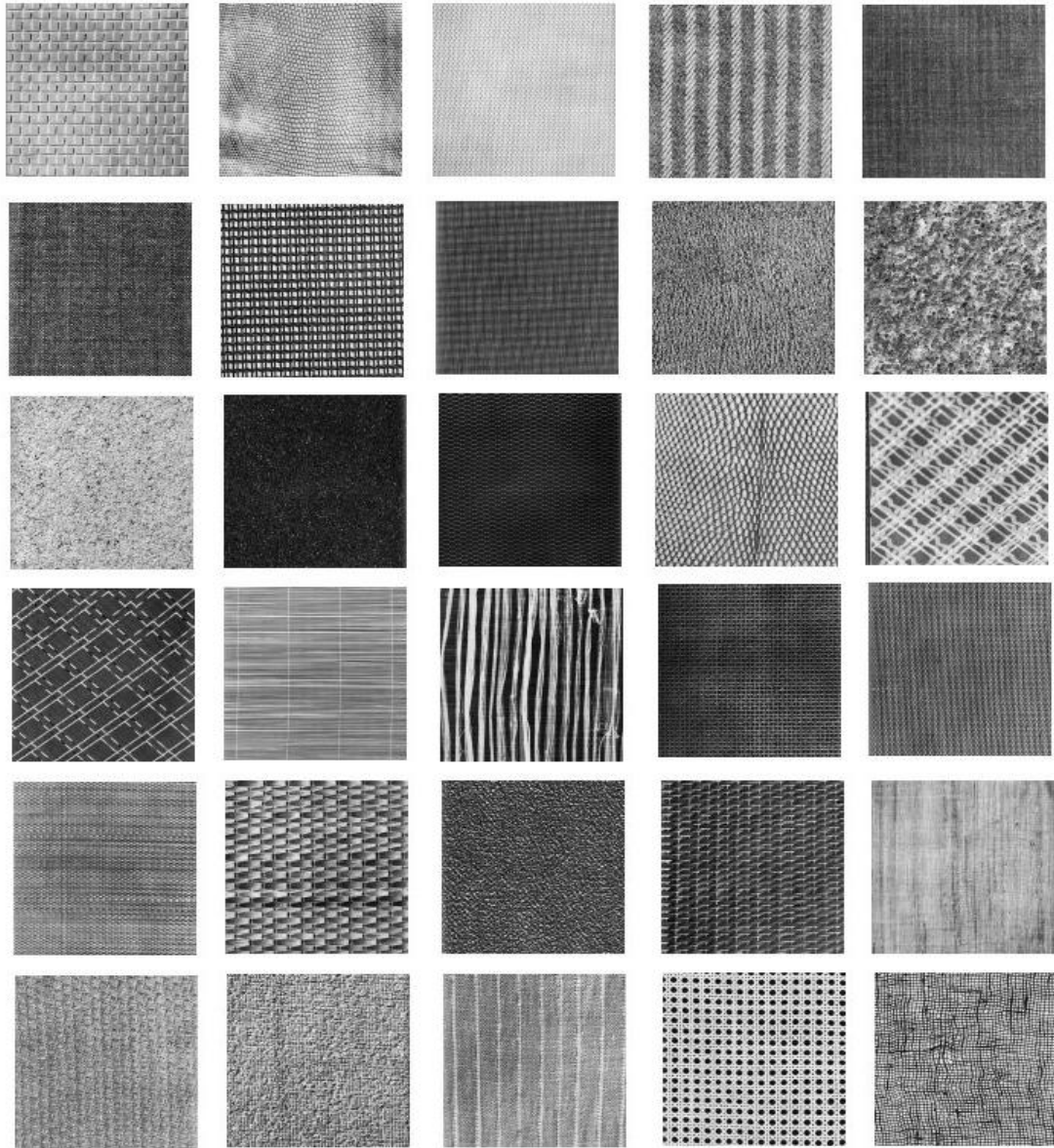
Na fase de classificação, uma amostra de textura de uma classe desconhecida é codificada pelo algoritmo LZW, seguindo a espiral mostrada na Figura 9, com os dicionários criados, e as taxas de codificação correspondentes são registradas. A amostra é então associada à classe cujo dicionário obteve a melhor taxa de compressão. O algoritmo que percorre a amostra em forma de espiral é o mesmo utilizado em [Deng e Clausi 2004].

### 3.4 Banco de Texturas

O banco de texturas utilizado nos experimentos é o *Brodatz* [Brodatz 1966]. As texturas deste álbum foram escolhidas devido à sua grande utilização pela comunidade científica, o que permite comparações diretas entre o classificador desenvolvido e diversos outros métodos da literatura.

A Figura 10 mostra as trinta texturas naturais do álbum *Brodatz* que foram selecionadas para avaliar o desempenho do método proposto em todas as versões, com exceção da que trata de CBIR, onde todo o corpo de *Brodatz* é testado. Esse corpo de teste é o mesmo usado em [Li *et al.* 2003], permitindo assim uma comparação direta com alguns classificadores no estado da arte. Em todos os experimentos, cada textura constitui uma classe separada. Todas as texturas têm 640 x 640 *pixels*, com 8 bits/*pixel*.

Podemos notar na Figura 11 uma imagem do banco *Brodatz* tão irregular que não poderia ser considerada textura, de acordo com as definições mais aceitas para imagens de texturas. Há outras imagens no álbum de *Brodatz* que se encaixariam nesse exemplo. Essas imagens altamente irregulares têm um efeito negativo na taxa de reconhecimento [Xu 2000].



**Fig. 10.** Texturas de *Brodatz* usadas nos experimentos. Da direita para esquerda e de cima para baixo: D1, D3, D6, D11, D16, D17, D20, D21, D24, D28, D29, D32, D34, D35, D46, D47, D49, D51, D52, D53, D55, D56, D57, D65, D78, DD82, D84, D85, D101, D104.



Fig. 11. Textura irregular de *Brodatz D45*.

## 3.5 Versões do Classificador

### 3.5.1 Experimentos sem variação de iluminação

O desempenho da versão do classificador com equalização de histograma global, a partir de agora identificado como CLZW-GHE, foi cuidadosamente avaliado na ausência de variação de iluminação. O classificador LZW sem equalização de histograma será identificado como CLZW. Cada textura de *Brodatz* passou pelo processo de equalização de histograma global e foi particionada em imagens não sobrepostas de tamanho  $n \times n$ , que por sua vez foram tomadas como amostras da textura. As amostras foram separadas em dois conjuntos distintos, uma para treinamento e outro para testar a eficiência do classificador.

É importante notar que somente com conjuntos distintos para treinamento e teste é possível alcançar resultados precisos. Não obstante, como observado em [Randen e Husøy 1999] e em [Li *et al.* 2003], o uso de conjuntos sobrepostos é bastante comum na literatura de classificação de texturas. Nestes casos os resultados divulgados são normalmente superestimados e não refletem uma situação realista.

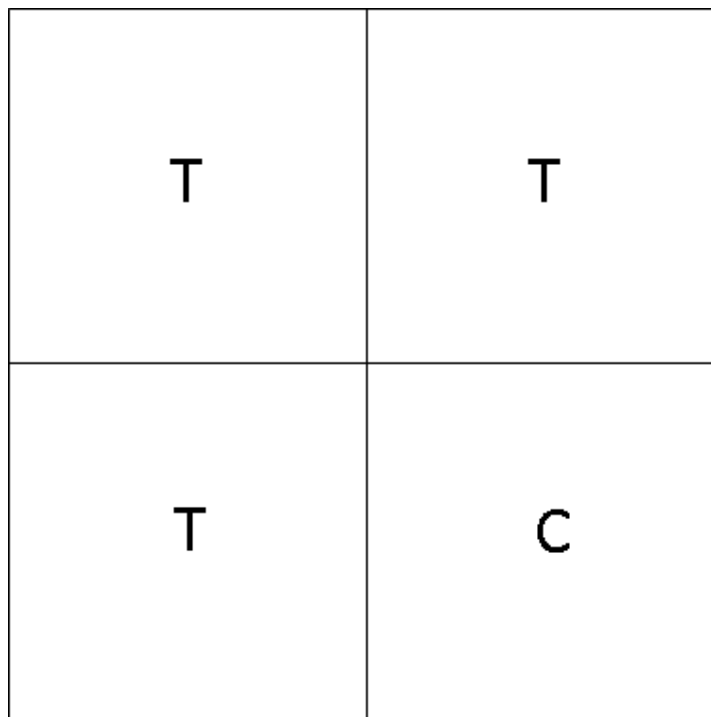
A precisão da classificação é avaliada pela taxa de classificação correta (*correct classification rate – CCR*):

$$CCR = \frac{c}{t} \times 100 \% \quad (8)$$

onde  $c$  é o número de amostras de textura corretamente classificadas e  $t$  é o número de amostras classificadas.

No primeiro experimento desta versão do classificador, o conjunto de treinamento compreende os primeiros três quadrantes de cada textura, e o último quadrante da textura corresponde ao conjunto de teste para classificação, como visto na Figura 12.

O segundo experimento tem como conjunto de treinamento os primeiros blocos da textura, no sentido horizontal (da esquerda para a direita) e o restante da imagem servindo como conjunto de classificação.



**Fig. 12.** Três primeiros quadrantes (representados pela letra T) utilizados para treinamento e o quarto quadrante (representado pela letra C) usado na classificação.

### 3.5.2 Experimentos com variação uniforme de iluminação

Experimentos com variação uniforme de iluminação foram realizados para mensurar a robustez do método proposto a esse tipo de problema. Além destes, outros experimentos com variação de iluminação foram realizados e serão descritos em breve.

O conjunto de treinamento e de classificação é o mesmo utilizado na versão anterior e já visto na Figura 12. Cada textura foi dividida em blocos de 32 x 32 *pixels*, servindo como amostras nas fases de aprendizagem e teste de desempenho.

Antes da fase de classificação uma variação de brilho uniforme foi aplicada ao quarto quadrante (quadrante de teste), multiplicando-se cada *pixel* da imagem por um fator  $\alpha$  e somando-se a esse resultado outro fator  $\beta$ :

$$g(i, j) = f(i, j) \cdot \alpha + \beta \quad (9)$$

Após simular uma variação uniforme de iluminação, efetua-se uma normalização do desvio padrão e da média dos valores dos *pixels* da imagem a ser classificada:

$$g(i, j) = \mu_0 + \left[ \frac{\sigma_0}{\sigma} \cdot (f(i, j) - \mu) \right] \quad (10)$$

Esse processo de normalização ameniza os efeitos provocados pela variação uniforme de iluminação nas texturas. Após essa operação a fase de classificação pode ser iniciada.

### 3.5.3 Experimentos com variação não uniforme de iluminação

O desempenho da versão com equalização de histograma local, a partir de agora identificado como CLZW-LHE, foi avaliado sob variações não uniformes de iluminação.

O conjunto de treinamento compreende os primeiros três quadrantes de cada textura, e o último quadrante da textura corresponde ao conjunto de teste, tal como já visto na Figura 12. Nestes experimentos as imagens de texturas foram particionadas em amostras de tamanho fixo 32x32.

Antes da equalização de histograma local, uma variação não uniforme de iluminação foi simulada no quadrante de teste, multiplicando-se cada *pixel* da imagem por um fator que assume o valor  $1 - a$  na primeira linha do quadrante de teste,  $0 \leq a \leq 0,2$ , e cresce linearmente até o valor  $1 + a$ , na última linha do quadrante. Quando  $a = 0$ , não há variação de nível de cinza, enquanto com  $a = 0,2$ , a primeira linha do quadrante é multiplicada por 0.8 e a última linha é multiplicada por 1.2, produzindo uma variação de 50% no fator multiplicativo da primeira linha até a última. Depois de aplicar a variação de iluminação, a equalização de histograma local é aplicada no quadrante de teste, e em seguida a classificação é executada.

De maneira análoga à normalização, utilizada na versão anteriormente descrita, aqui se faz uso da equalização de histograma local para se obter bons resultados mesmo com variações não uniformes sendo aplicadas nesta versão do classificador.

### 3.5.4 Experimentos CBIR

O álbum completo de *Brodatz* [Brodatz 1966] foi utilizado para analisar o desempenho desta modalidade do classificador. Todas as 112 texturas têm 640 x 640 *pixels*, com 8 bits/*pixel*. Este corpo de teste é o mesmo usado em [Xu 2000], permitindo assim uma comparação direta com outro sistema CBIR encontrado na literatura.

A fim de avaliar mais uma vez os efeitos da equalização de histograma, testes foram feitos com e sem a aplicação desta operação. A partir de agora, a versão com e sem equalização de histograma será identificada como CBIR-LZW-HE e CBIR-LZW, respectivamente. Cada textura de *Brodatz* foi particionada em imagens não sobrepostas de 128 x 128, atuando como amostras da textura. Nove des-

As amostras (para cada textura) foram armazenadas no banco de dados à medida que seus dicionários LZW correspondentes eram construídos como descrito na seção 3.1, e as amostras restantes foram usadas como amostras para consulta nos testes de precisão da versão em estudo. Portanto, o banco de dados contém 112 classes de texturas, a partir de agora identificadas como  $C_i$ ,  $i = 0, 1, \dots, 111$ , e cada classe  $C_i$  contém nove amostras, identificadas como  $s_{ij}$ ,  $i = 0, 1, \dots, 111$ ,  $j = 0, 1, \dots, 8$ , resultando num total de  $9 \times 112 = 1008$  amostras de texturas. Dada uma amostra  $\mathbf{x}$  da classe  $C_i$ , recuperar uma das nove amostras  $s_{ij}$ ,  $j = 0, 1, \dots, 8$  como resultado da consulta é considerado como sucesso.

Um razoável número de medidas de precisão é utilizado na literatura. No intuito de comparar diretamente os resultados do estudo em questão com aqueles apresentados em [Xu, 2000], a medida de precisão adotada foi a taxa de reconhecimento médio (*average recognition rate* -  $AR(c)$ ). Considere que para cada consulta o sistema retorne as  $c$  amostras de texturas mais semelhantes do banco de dados, de acordo com os modelos LZW construídos na fase de aprendizagem. Para uma amostra da classe  $C_i$  usada numa consulta, temos  $h(c)$  como o número de amostras de  $C_i$  dentro do conjunto de  $c$  amostras recuperadas. A taxa de reconhecimento  $R(c)$  é então definida como a razão entre  $h(c)$  e o número total de amostras da classe  $C_i$  armazenadas no banco de dados. Nos experimentos aqui apresentados, tem-se:

$$R(c) = \frac{h(c)}{9} \quad (11)$$

A taxa de reconhecimento médio,  $AR(c)$ , é o valor da média de  $R(c)$  para todos os testes de consulta realizados. Partindo da definição, entende-se que  $AR(c)$  melhora à medida que  $c$  cresce e sabendo-se que no banco de dados há 1008 amostras de texturas, teríamos  $AR(c) = 1$  para  $c = 1008$ . Quanto mais eficiente for o

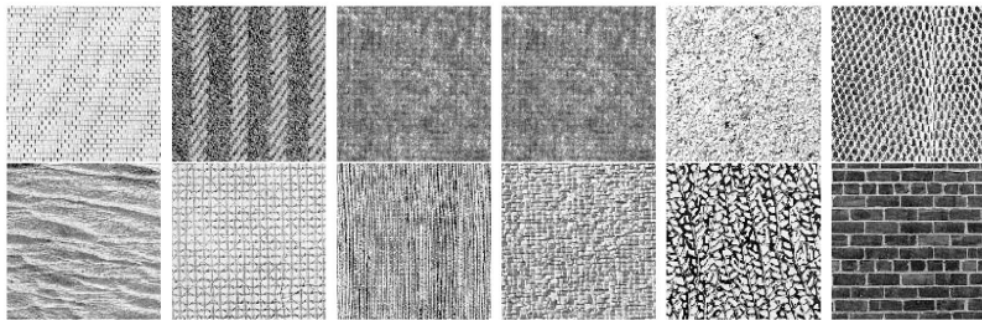
mecanismo de recuperação, menores serão os valores de  $c$  necessários para se atingir  $AR(c) = 1$ . No caso de um banco de dados contendo 9 amostras/classe, um sistema CBIR ideal obteria  $AR(9) = 1$ .

### 3.5.5 Experimentos com Robustez à Rotação

A versão do classificador robusto à rotação, CLZW-ROT, teve seu desempenho avaliado através de dois experimentos.

No primeiro experimento, o conjunto de treinamento abrange os primeiros três quadrantes de cada textura, e o conjunto de teste refere-se ao último quadrante da textura, já visto na Figura 12. Amostras não sobrepostas de tamanho  $n \times n$  de cada textura de *Brodatz*, foram criadas através de partição. As amostras foram separadas em dois conjuntos diferentes, um para a fase de aprendizagem e outro para testar a eficiência do classificador.

No segundo experimento foram utilizadas 12 texturas de *Brodatz* com 8 bits/pixel, apresentadas na Figura 13. As mesmas usadas em [Deng e Clausi 2004]. Todas as texturas têm 256x256 pixels de dimensão.



**Fig. 13.** 12 Texturas de *Brodatz* utilizadas no segundo experimento de invariância à rotação. Da direita para esquerda e de cima para baixo: D006, D011, D019, D021, D029, D036, D037, D052, D079, D084,

D087, D095.

O conjunto de teste foi obtido rotacionando as imagens em 11 ângulos: 15, 30, 45, 60, 75, 90, 105, 120, 135, 150 e 165 graus. Apenas a região central de 128x128 *pixels* foi usada na fase de treinamento. O restante da textura forma o conjunto de teste a ser rotacionado e classificado. Apesar do uso de conjuntos sobrepostos para treinamento e teste já ter sido considerado neste trabalho, aqui se abre uma exceção, pois estes conjuntos são os mesmos usados em [Deng e Clausi 2004], o que permite uma comparação direta com alguns classificadores no estado da arte. Nesse experimento o tamanho da amostra foi fixado em 32x32, tamanho utilizado em [Deng e Clausi 2004].

## 3.6 Recursos Empregados

No desenvolvimento do classificador foram utilizados apenas um software e uma máquina, descritos a seguir:

### 3.6.1 Software

Para desenvolver e testar o método exposto aqui foi utilizado o ambiente de desenvolvimento Borland C++ Builder 6. O sistema operacional em uso durante o processo de desenvolvimento foi o Windows XP Professional.

### 3.6.2 Hardware

O equipamento utilizado tratou-se de um Pentium 4 HT 2.8 Ghz, com memória RAM de 1Gb.

Foram demonstrados os dois estágios fundamentais do método proposto: treinamento e classificação. Em cada uma dessas fases o algoritmo de compressão

segue um trajeto específico, com o objetivo de se criar um modelo preciso baseado nas estruturas horizontal e vertical da textura. Tem-se ainda uma mudança do trajeto na versão do classificador robusto à rotação, visando novamente à criação de um modelo preciso, dessa vez em fora de espiral.

No capítulo seguinte são mostrados os resultados das várias versões do classificador apresentadas neste capítulo de materiais e métodos. As diversas versões foram pensadas no intuito de aferir a capacidade do esquema de classificação em certas particularidades: soluções para alguns problemas, como por exemplo, variação de níveis de cinza; e aplicações práticas como CBIR.

## Resultados

Neste capítulo são expostos os resultados de todas as versões do método estudado. Os resultados são apresentados e comparados (na forma de tabelas e gráficos) com outros classificadores encontrados na literatura. Algumas representações gráficas demonstram os efeitos das técnicas envolvidas na solução de problemas relacionados às variações de iluminação, tanto uniforme quanto não uniforme.

### 4.1 Experimentos sem variação de iluminação

Para efeito comparativo, os testes foram feitos com e sem equalização de histograma global tanto na fase de treinamento quanto na fase de classificação.

No primeiro experimento, o efeito do tamanho das amostras na *CCR* foi analisado para  $n = 4, 8, 16, 32$ . Os resultados são mostrados na Tabela 2.

**TABELA 2.**  
*CCR obtida pelo CLZW-GHE, CLZW em função de vários tamanhos de amostra.*

| Tamanho da amostra ( $n \times n$ ) | <i>CCR (%)</i> |          |
|-------------------------------------|----------------|----------|
|                                     | CLZW           | CLZW-GHE |
| 4 x 4                               | 80.6           | 99.9     |
| 8 x 8                               | 97.9           | 100      |
| 16 x 16                             | 99.7           | 100      |
| 32 x 32                             | 100.0          | 100      |

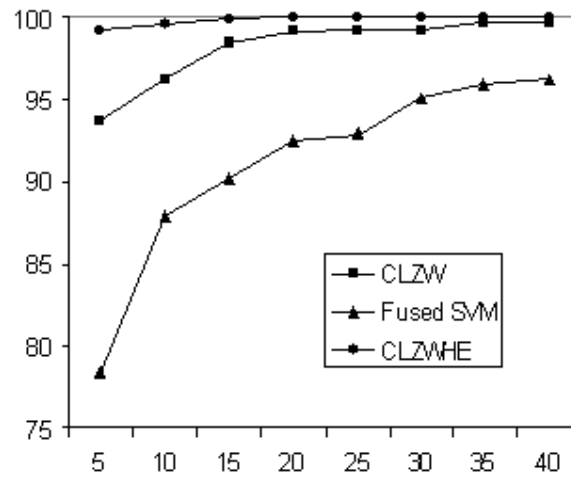
No segundo experimento, o tamanho da amostra foi fixado em 32x32. Conseqüente, existem, em cada classe, 400 amostras da textura, não sobrepostas. Dessa vez foi avaliado o efeito do tamanho do conjunto de treinamento na precisão do classificador. Foram, então, definidas as seguintes proporções para análise dos resultados deste experimento: 1.25% (5 amostras), 2.5% (10 amostras), 3.75% (15 amostras), 5% (20 amostras), 6.25% (25 amostras), 7.5% (30 amostras), 8.75% (35 amostras) e 10% (40 amostras). Em cada caso todas as amostras que não foram usadas na fase de treinamento são usadas na classificação.

A Tabela 3 resume os resultados dos classificadores CLZW-GHE e CLZW, comparando-os com os resultados do SVM simples, do *fused* SVM, do classificador de *Bayes* usando distância de *Bayes*, e do classificador LVQ [Li *et al.* 2003]. A Figura 14 apresenta uma comparação gráfica do CCR obtido pelo CLZW-GHE, CLZW e o *fused* SVM em função do tamanho do conjunto de treinamento.

**TABELA 3.**

CCR obtida pelo CLZW-GHE, CLZW e outros classificadores, para vários tamanhos de conjunto de treinamento.

| Número de amostras de treinamento por classe | CCR (%)            |      |             |                  |      |          |
|--|--------------------|------|-------------|------------------|------|----------|
|  | <i>Bayes-Bayes</i> | LVQ  | SVM simples | <i>Fused</i> SVM | CLZW | CLZW-GHE |
| 5  | 79.5               | 69.7 | 78.2        | 78.4             | 93.8 | 99.3     |
| 10   | 86.5               | 80.0 | 87.5        | 87.9             | 96.2 | 99.6     |
| 15   | 87.9               | 83.9 | 89.6        | 90.2             | 98.4 | 99.9     |
| 20   | 90.3               | 87.0 | 91.8        | 92.5             | 99.2 | 100      |
| 25   | 91.2               | 88.2 | 91.4        | 92.9             | 99.3 | 100      |
| 30   | 91.8               | 89.6 | 94.6        | 95.1             | 99.3 | 100      |
| 35   | 92.1               | 90.2 | 94.9        | 95.9             | 99.7 | 100      |
| 40   | 92.7               | 90.4 | 95.8        | 96.3             | 99.7 | 100      |



**Fig. 14.** CCR obtida pelo CLZW-GHE, CLZW e *fused SVM* em função do tamanho do conjunto de treinamento.

## 4.2 Experimentos com variação uniforme de iluminação

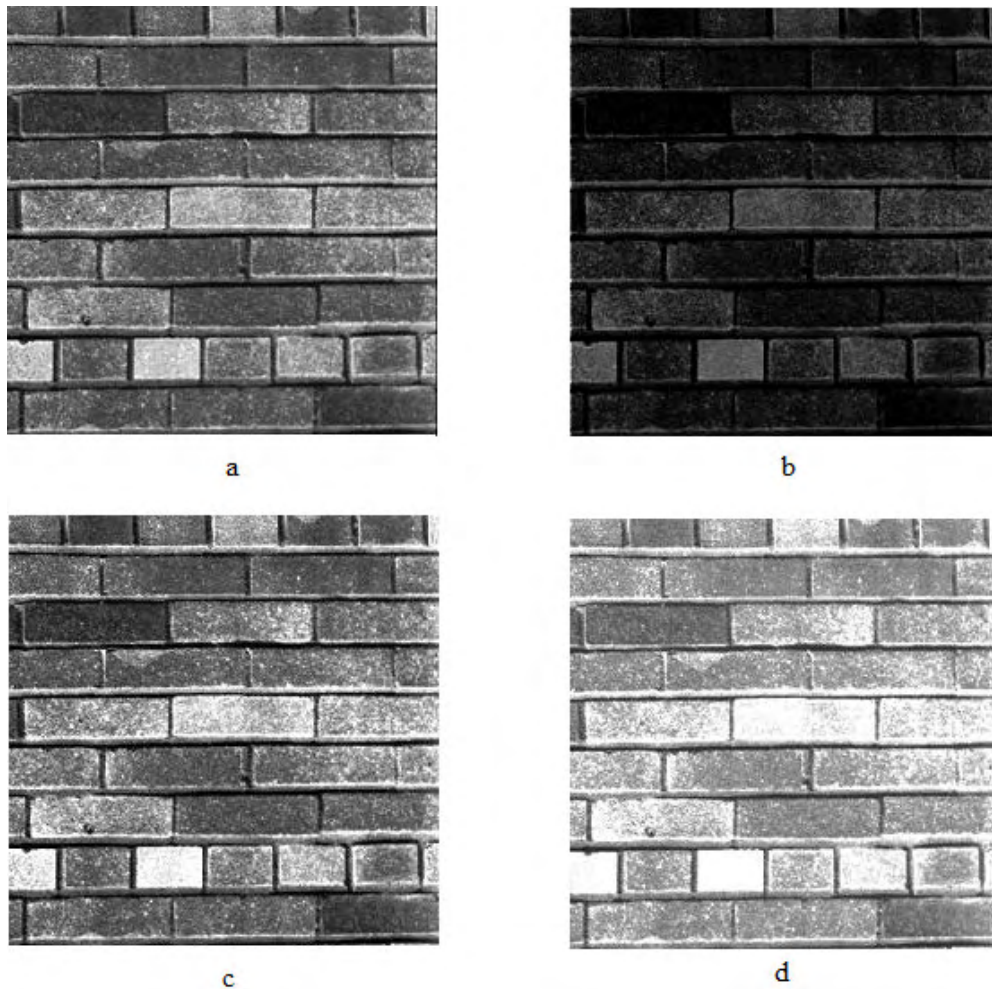
O efeito da variação uniforme dos níveis de cinza na CCR foi analisado para  $\alpha = 0,6, 0,7, 0,9, 1,0, 1,1, 1,2, 1,3, 1,4$  e  $\beta = -32, -16, -8, 0, 8, 16, 32$ . Os resultados são mostrados na Tabela 4.

**TABELA 4.**

CCR obtida pelo classificador para diversos valores de  $\alpha$  e  $\beta$ .

| $\alpha / \beta$ | -32   | -16   | -8    | 0     | 8     | 16    | 32    | Média |
|------------------|-------|-------|-------|-------|-------|-------|-------|-------|
| 0,6              | 69,20 | 88,87 | 91,07 | 91,40 | 91,40 | 91,40 | 91,40 | 87,82 |
| 0,7              | 77,73 | 85,47 | 90,97 | 91,67 | 91,63 | 91,57 | 91,57 | 88,66 |
| 0,8              | 82,53 | 92,20 | 92,20 | 92,13 | 92,13 | 92,13 | 92,13 | 90,78 |
| 0,9              | 83,63 | 92,07 | 91,63 | 91,97 | 91,77 | 91,77 | 91,60 | 90,63 |
| 1,0              | 85,57 | 91,60 | 91,90 | 91,90 | 91,83 | 92,73 | 91,50 | 91,00 |
| 1,1              | 87,73 | 92,20 | 92,13 | 92,00 | 92,30 | 89,87 | 77,20 | 89,06 |
| 1,2              | 89,60 | 91,93 | 90,83 | 87,67 | 80,50 | 72,90 | 63,03 | 82,35 |
| 1,3              | 86,83 | 84,37 | 77,43 | 69,57 | 61,93 | 57,73 | 49,90 | 69,68 |
| 1,4              | 79,87 | 68,30 | 61,83 | 57,10 | 52,87 | 48,33 | 41,03 | 58,48 |
| Média            | 82,52 | 87,45 | 86,67 | 85,05 | 82,93 | 80,94 | 76,60 | 83,16 |

Como forma de ilustrar os efeitos dessa variação uniforme a Figura 15 representa um quadrante de uma imagem de textura sobre quatro variações diferentes de iluminação seguindo a Equação 9.



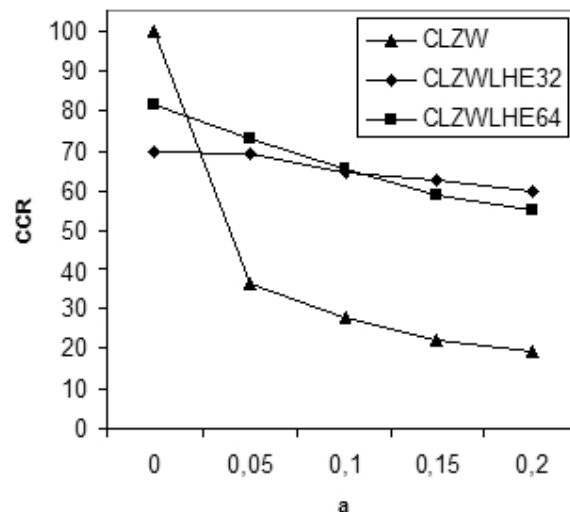
**Fig. 15.** Efeitos da aplicação de variação uniforme de iluminação em amostras de uma mesma classe de textura. (a)  $\alpha = 1$  e  $\beta = 0$ , (b)  $\alpha = 0,6$  e  $\beta = -32$ , (c)  $\alpha = 1,1$  e  $\beta = -16$ , (d)  $\alpha = 1,4$  e  $\beta = 32$ .

A Figura 15.a mostra uma imagem de textura em sua forma original, ou seja, com  $\alpha = 1$  e  $\beta = 0$ . Pode-se ver na Figura 15.b o efeito da variação nos níveis de cinza da textura quando  $\alpha = 0,6$  e  $\beta = -32$ . São aplicados, na Figura 15.c,

os valores 1,1 e -16, em  $\alpha$  e  $\beta$ , respectivamente. Para  $\alpha = 1,4$  e  $\beta = 32$  o resultado é apresentado na Figura 15.d.

### 4.3 Experimentos com variação não uniforme de iluminação

A CCR obtida pelo CLZW-LHE e pelo CLZW são apresentadas na Figura 16, para dois tamanhos de janela usados na equalização de histograma local, 32 x 32 e 64 x 64, em função de vários valores de  $a$ .

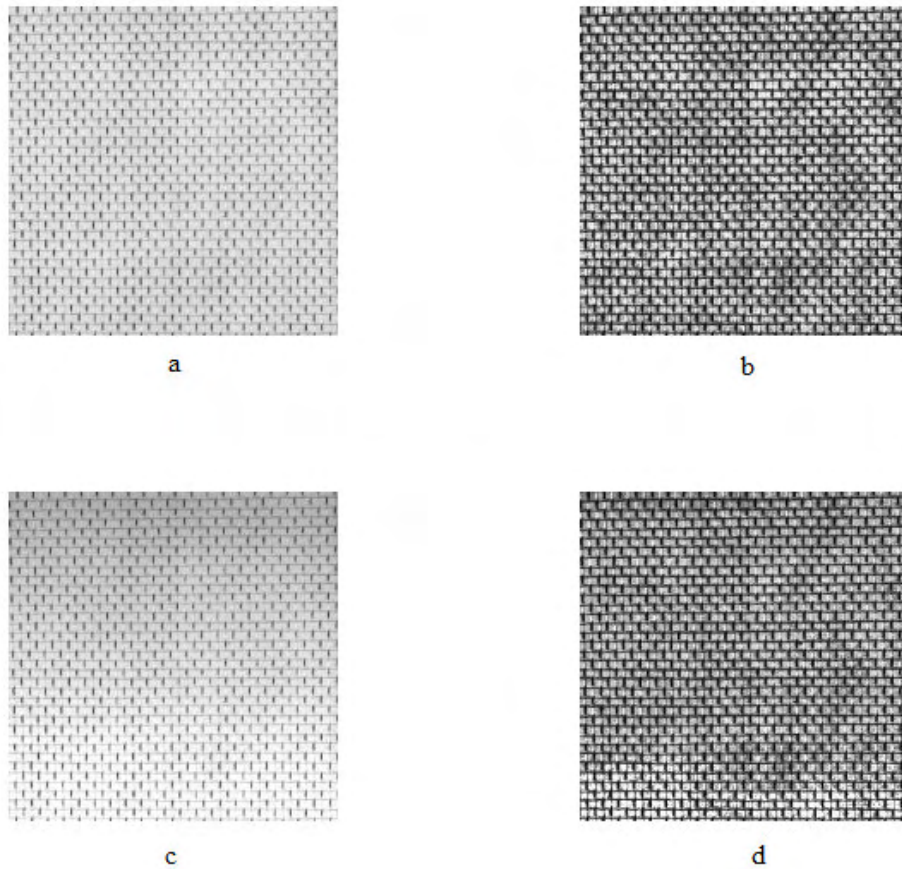


**Fig. 16.** CCR obtida pelo CLZW, CLZW-LHE 32 e CLZW-LHE 64 em função dos valores de  $a$ .

Para efeito de exemplo temos a Figura 17 que contém uma mesma amostra de uma classe de textura em diferentes condições. Nesta imagem observa-se o efeito da equalização de histograma local sobre uma variação não uniforme de iluminação.

Na Figura 17.a temos a amostra original, sem modificações. A Figura 17.b representa a mesma amostra com o seu histograma equalizado localmente. Por sua

vez, a Figura 17.c apresenta a amostra com os seus níveis de cinza alterados pela aplicação da variação não uniforme de iluminação. Por fim, tem-se a Figura 17.d onde os dois processos, equalização de histograma local e variação não uniforme de iluminação, são aplicados. Nas Figuras 17.c e 17.d utiliza-se  $a = 0,1$  na variação não uniforme de iluminação.



**Fig. 17.** Efeitos da aplicação da equalização de histograma local e variação de iluminação em amostras de uma mesma classe de textura.

É visualmente perceptível que, entre as amostras que sofreram equalização de histograma, representadas nas Figuras 17.b e 17.d, há mais semelhanças e menos complexidade na comparação do que entre as Figuras 17.a e 17.c, que não passaram pelo mesmo processo de equalização.

Essas amostras, juntamente com os dados observados na Figura 16, demonstram o efeito favorável da equalização de histograma local no combate à variação de brilho, tornando o método proposto capaz de obter bons resultados na classificação mesmo com mudanças nos níveis de cinzas das amostras causadas por condições de iluminação não-uniformes.

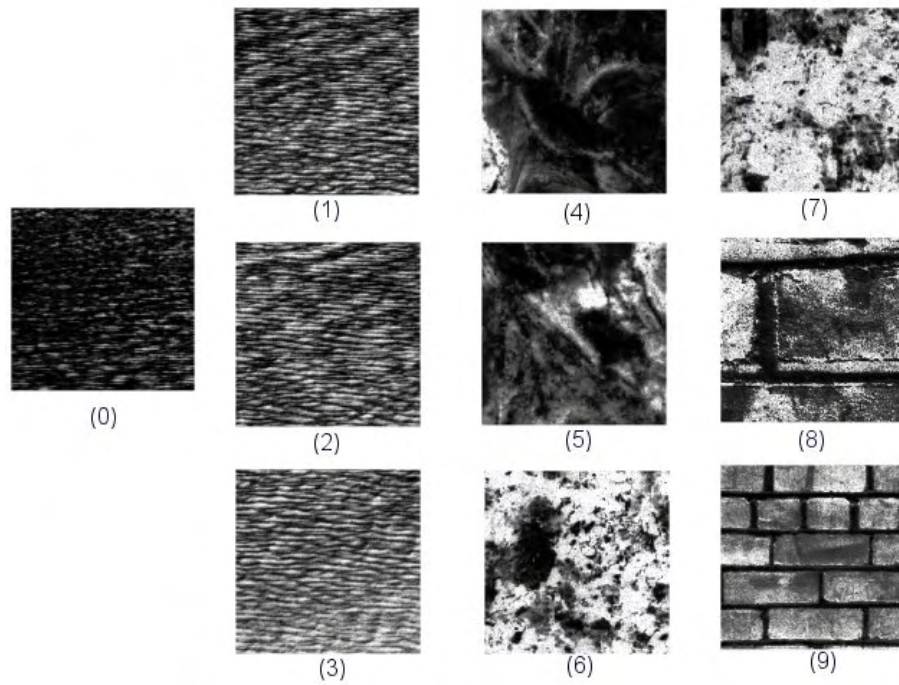
#### 4.4 Experimentos CBIR

Os resultados obtidos em função de alguns valores de  $c$ , com e sem aplicação de equalização de histograma, são apresentados na Tabela 5.

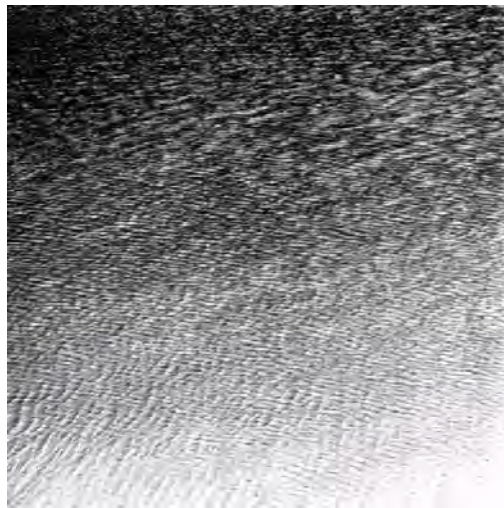
**TABELA 5.**  
*AR(c)* obtida pelo CBIR-LZW, CBIR-LZW-HE em função de  $c$ .

| $c$ | $AR(c) (%)$ |             |
|-----|-------------|-------------|
|     | CBIR-LZW    | CBIR-LZW-HE |
| 9   | 79.97       | 99.39       |
| 15  | 87.07       | 99.39       |
| 20  | 91.19       | 99.49       |
| 25  | 92.79       | 99.49       |
| 30  | 93.39       | 99.59       |
| 35  | 94.39       | 99.59       |
| 40  | 94.59       | 99.59       |

A Figura 18 demonstra um exemplo de uma consulta que resultou em uma baixa  $AR(9)$ , e a Figura 19 mostra a textura de *Brodatz* de onde foi retirada a amostra para consulta.



**Fig. 18.** Resultado de uma consulta com  $AR(9) = 0.3333$  utilizando CBIR-LZW-HE. A amostra 0 é a amostra para consulta, a amostra 1 é a primeira amostra recuperada, a amostra 2 é a segunda, e assim por diante.



**Fig. 19.** Textura D38, de onde foi retirada a amostra para consulta da Figura 18.

A Figura 20 compara o desempenho desta versão do classificador com o esquema M2, que obteve o melhor desempenho dentre as quatro variações do método desenvolvido por [Xu 2000].

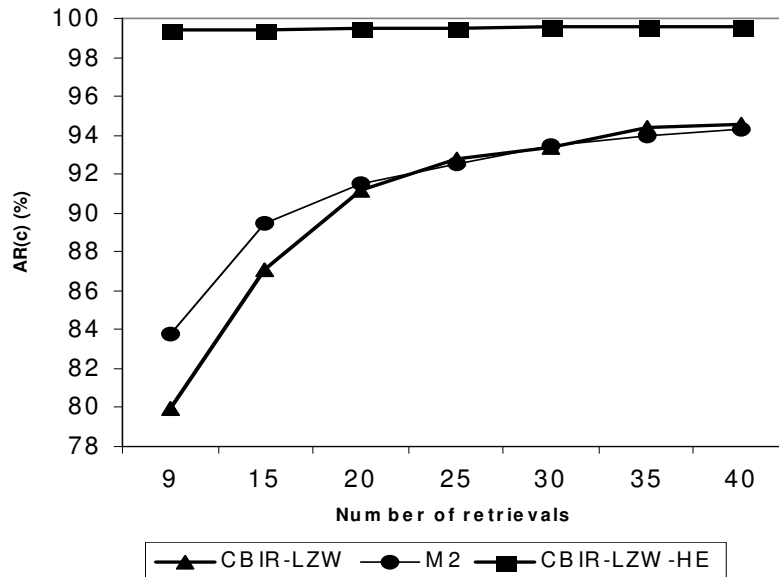


Fig. 20.  $AR(c)$  obtida pelo CBIR-LZW, CBIR-LZW-HE e M2 em função de  $c$

## 4.5 Experimentos com robustez à Rotação

No primeiro experimento, o efeito do tamanho das amostras na  $CCR$  foi analisado para  $n = 8, 16, 32$ , em função do ângulo de rotação das amostras a serem classificadas. Os resultados são ilustrados na Figura 21.

Para o segundo experimento, os resultados são mostrados na Figura 22 e um confronto direto com o método ACGMRF (*Anisotropic Circular Gaussian Markov Random Fields*) desenvolvido em [Deng e Clausi 2004] pode se visto na Tabela 6, textura a textura.

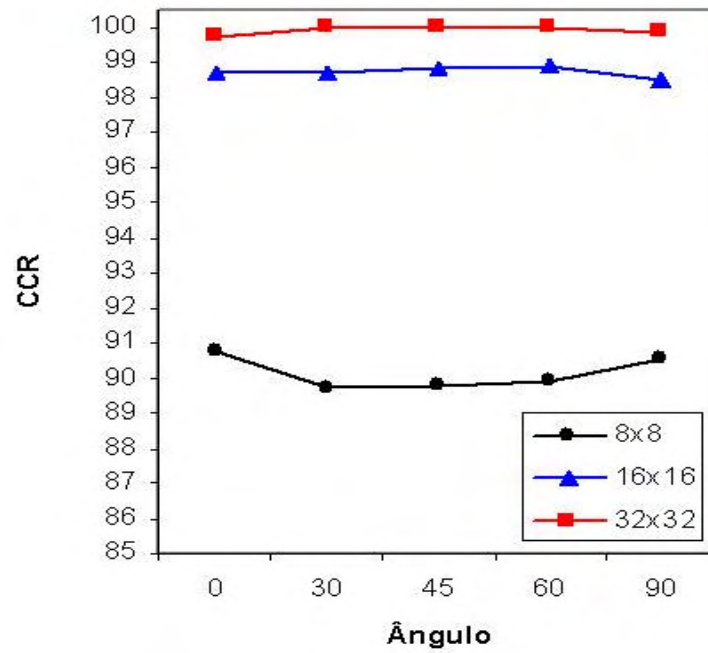


Fig. 21. CCR obtida pelo classificador proposto em função do ângulo.

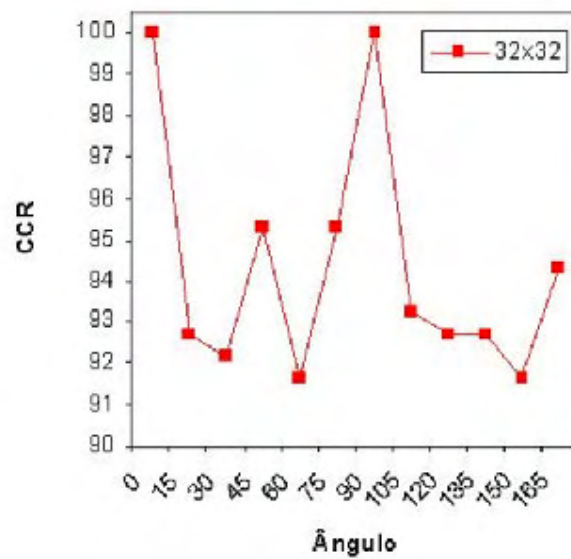


Fig. 22. CCR obtida pelo classificador proposto em função do ângulo.

**TABELA 6.***CCR obtida pelo ACGMRF, e pelo CLZW-ROT, para cada classe de textura.*

| Classe | CCR (%) |          |
|--------|---------|----------|
|        | ACGMRF  | CLZW-ROT |
| D06    | 82.30   | 100      |
| D11    | 81.30   | 96.35    |
| D19    | 87.5    | 100      |
| D21    | 81.30   | 100      |
| D29    | 75.00   | 79.16    |
| D36    | 84.30   | 93.75    |
| D37    | 88.50   | 93.87    |
| D52    | 78.10   | 94.79    |
| D79    | 82.30   | 89.06    |
| D84    | 74.00   | 93.18    |
| D87    | 82.30   | 94.22    |
| D95    | 88.5    | 97.35    |
| Média  | 82.1    | 94.31    |

Os gráficos e tabelas vistos nesse capítulo ilustram claramente os resultados dos experimentos. Realizados em comparação com diversos métodos no estado da arte, esses resultados comprovam a eficiência do esquema de classificação em estudo. As discussões e conclusões a respeito do desempenho alcançado são encontradas no capítulo 6, a seguir.

## Discussão e Conclusões

Este trabalho propôs um novo, simples e eficiente esquema de classificação de texturas usando equalização de histograma (global e local) e o algoritmo LZW. Foram desenvolvidas cinco versões do esquema e todas apresentaram resultados satisfatórios.

Avaliando-se os efeitos do tamanho da amostra sobre a precisão da classificação, a Tabela 2 mostra que a equalização de histograma global tem um impacto determinante no desempenho do classificador. CLZW-GHE atingiu 100% de acertos com amostras pequenas como 8 x 8 *pixels*, enquanto CLZW obteve 100% apenas com 32 x 32 ou com amostras maiores. A melhoria derivada do uso da equalização de histograma para a amostra de tamanho 4 x 4 foi impressionante.

O segundo experimento da primeira versão, resumido na Tabela 3 e na Figura 14, mostra a superioridade do classificador sobre os classificadores SVM, simples e *fused*, o classificador de *Bayes* usando distância de *Bayes* e o classificador LVQ. Esta superioridade ainda é mais destacada para conjuntos de treinamento de tamanho muito pequeno. Com apenas 5 amostras de treinamento por classe, o CLZW-GHE obteve  $CCR = 99.3\%$  e o CLZW atingiu  $CCR = 93.8\%$ , enquanto o terceiro melhor método neste caso, o *Bayes-Bayes* obteve apenas um  $CCR = 79.5\%$ . Isto mostra a capacidade de aprendizagem e generalização do

método para conjuntos de treinamento muito pequenos. Isto é corroborado notando-se que o classificador em estudo atingiu  $CCR = 100\%$  com apenas 20 amostras de treinamento.

Nos experimentos com variação uniforme de iluminação é possível observar que, para valores de  $\alpha$  maiores do que 1,0 a degradação na  $CCR$  é mais forte do que para valores menores do que 1,0. Analisando os valores de  $\beta$ , observa-se que essa degradação também ocorre para valores maiores e menores do que 0, só que de forma mais sutil, já que  $\beta$  é um fator que é somado ao valor do *pixel* e  $\alpha$  é um fator multiplicador.

Nos experimentos com variação não uniforme de iluminação, torna-se claro na Figura 16 que a equalização de histograma local é eficiente ao corrigir essas variações, especialmente com janelas menores, mas quando não há tanta variação a equalização de histograma global se mostra muito mais eficiente. De maneira geral, essa versão do classificador obteve um desempenho mais alto com relação à versão com o uso de variação uniforme de iluminação. Isso se deve principalmente ao uso da equalização de histograma.

A Tabela 5 mostra o forte impacto positivo do uso da equalização de histograma no desempenho do classificador. CBIR-LZW-HE obteve  $AR(9) = 0.9939(99.39\%)$ , enquanto o CBIR-LZW atingiu  $AR(40) = 0.9459(94.59\%)$ .

A Figura 20 compara o desempenho do classificador na versão robusta à rotação com M2 [Xu 2000], sobre o mesmo banco de dados. A comparação mostra a superioridade do CBIR-LZW-HE sobre o M2. Esta superioridade é ainda mais visível no caso mais restrito ( $c = 9$ ), com o CBIR-LZW-HE atingindo  $AR(9)$  aproximadamente 15% acima do valor obtido pelo método M2.

Como já observado antes, muitas imagens de *Brodatz* são tão irregulares

que não podem ser consideradas como imagens de textura, de acordo com definições aceitas para texturas. O trabalho de Xu [Xu 2000] avaliou os efeitos destas peculiaridades do álbum de *Brodatz* dividindo as texturas em dois bancos de dados separados: um contendo somente texturas visualmente homogêneas e outra contendo o restante do álbum, texturas visualmente não homogêneas. Através da análise da taxa de reconhecimento destes dois bancos de dados, eles concluíram que imagens irregulares têm um efeito negativo na taxa de reconhecimento. Por outro lado, os resultados do CBIR-LZW-HE, com  $AR(c)$  próximo de 100%, mesmo para valores baixos de  $c$ , indicam que o método é de certa forma, robusto a essas irregularidades apresentadas.

A Figura 18 apresenta um exemplo de consulta com uma  $AR(9)$  (33,33%) particularmente baixa. Como pode ser visto na Figura 19, a textura de *Brodatz* D38, da qual a amostra para a consulta foi retirada, apresenta uma forte variação não uniforme de iluminação, a ponto de saturação no canto inferior direito, tornando o reconhecimento uma tarefa complexa.

Observando a Figura 21, vê-se a importância do tamanho das janelas também nesta versão do classificador com invariância à rotação. Temos um considerável avanço nos resultados quando passamos de  $n = 8$  para  $n = 16$  e finalmente  $n = 32$ . O excelente resultado obtido com  $n = 32$  reforçou a idéia de realizar o segundo experimento com este valor fixo de janela nas duas fases do processo de classificação, treinamento e teste.

A Figura 22 ilustra os resultados do CLZW-ROT em função dos ângulos utilizados na rotação das amostras para formação do conjunto de teste do segundo experimento dessa versão do classificador. Nesta figura se vê que alguns ângulos apresentam um grau de dificuldade maior do que os demais, como por exemplo, 165 e 75 graus, o que impede uma taxa média de classificação correta ( $CCR$ ) mais alta.

No segundo experimento do classificador invariante à rotação, o classificador foi comparado com [Deng e Clausi 2004]. A Tabela 6 mostra uma comparação direta e apresenta um excelente resultado comparativo, em que o CLZW-ROT obtém uma  $CCR = 94.31\%$  enquanto o método ACGMRF atinge apenas uma  $CCR = 82.10\%$ . Um bom exemplo desta comparação pode ser visto nos resultados da textura de *Brodatz* D21 onde esta versão do método apresenta 100% de acerto na classificação e o ACGMRF mostra apenas 81.3% de classificação correta.

## Referências Bibliográficas

- Addis, M., L. P. M. K. (2002). Artiste image retrieval system puts european galleries in the picture. *Cultivate Interactive*, (7).
- Augusteijn, M. F., Clemens L. E. and Shaw K. A., “Performance evaluation of texture measures for ground cover identification in satellite images by means of a neural network classifier,” *IEEE Transactions on Geoscience and Remote Sensing*, v. 33, n. 3 (1995), 616-626.
- Bach, J. R., Fuller, C., Gupta, A., Hampapur, A., B. Horowitz, R. H., Jain, R., and Shu, C. F. (1996). The virage image search engine: An open framework for image management. In Proc. SPIE Storage and Retrieval for Image and Video Databases IV, San Jose, CA, USA
- Baheerathan, S., Albregtsen, F., Danielsen, H. E., “New Texture Features Based on the Complexity Curve,” *Pattern Recognition*, v. 32, n. 4 (1999), 605-618.
- Bell, T. C., Cleary, J. G., Witten J. H., *Text Compression*, Englewood Cliffs: Prentice-Hall, 1990.
- Bovik, A. (ed.), *Handbook of Image and Video Processing*, San Diego: Academic Press, 2000.
- Brodatz, P., *Textures: A Photographic Album for Artists and Designers*, New York: Dover, 1966.

- Chen, J., Kundu, A. Rotation and gray scale transform invariant texture identification using wavelet decomposition and hidden markov model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16:208–214, 1994.
- Cleary, J. G., Teahan, W. J., Witten, I. H. Unbounded length contexts for PPM, *Proceedings of the Conference on Data Compression*, p.52, March 28-30, 1995
- CMU NRT Database. <http://vivid.cse.psu.edu/texturedb/gallery/>
- Cohen, F. S., Fan, Z., Patel, M. A.: Classification of Rotated and Scaled Textured Images Using Gaussian Markov Random Field Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 13, Issue 2 (1991) 192–202
- Dell'Acqua, F., Gamba, P., “Texture-based Characterization of Urban Environments on Satellite SAR Images,” *IEEE Transactions on Geoscience and Remote Sensing*, v. 41, n. 1 (2003), 153-159.
- Deng, Huawu, Clausi, David A. "Gaussian MRF Rotation-Invariant Features for Image Classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 7, pp. 951-955, Jul., 2004
- Dowe, J. (1993). Content-based retrieval in multimedia imaging. In *Proc. SPIE Storage and Retrieval for Image and Video Databases*.
- Duda, R. O., Hart, P. E., Stork, D. G.: *Pattern Classification*. 2nd ed. Wiley Interscience, New York (2000)
- Faloutsos, C., Flickner, M., Niblack, W., Petkovic, D., Equitz, W., and Barber, R. (1993). Efficient and effective querying by image content. Tech. Rep., IBM Res. Rep.
- Finamore, W.A., Leister, M. A., “Lossy Lempel-Ziv algorithm for large alphabet sources and applications to image compression,” *Proc. Int. Conference on Image Processing*, v. 1 (1996), 235-238.

- Flickner, M., Sawhney, H., Niblack, W., Ashley, J., Huang, Q., Dom, B., Gorkani, M., Hafine, J., Lee, D., Petkovic, D., Steele, D., and Yanker, P. (1995). Query by image and video content: The qbic system. *IEEE Computer*
- Forsyth, David A., Ponce Jean. *Computer vision - A modern approach*. Prentice Hall, 2002
- Frank, E., Chui, C., Witten, I. H., “Text Categorization Using Compression Models,” *Proc. DCC*, Salt Lake City (2000), 555.
- Gonzalez, R. C., Woods, R. E. . *Digital Image Processing*. New York: Addison-Wesley, 1992.
- He, X., Hu, Y., Zhang, H. ., Li, M., Cheng, Q., Yan, S., “Bayesian Shape Localization for Face Recognition Using Global and Local Textures,” *IEEE Transactions on Circuits and Systems for Video Technology*, v. 14, n. 1 (2004), 102-113.
- Huang, T., Mehrotra, S., and Ramchandran, K. (1996). Multimedia analysis and retrieval system (mars) project. In *Proc. 33rd Annu. Clinic Library Appl. of Data Processing Digital Image Access and Retrieval*.
- ISO/IEC, JTC1/SC29/WG11, and N1920 (1997a). *Mpeg-7: Context and objectives (v.5)*.
- ISO/IEC, JTC1/SC29/WG11, and N1921 (1997b). *Third draft of mpeg-7 requirements*.
- ISO/IEC, JTC1/SC29/WG11, and N1922 (1997c). *Mpeg-7 applications document*
- Jain, R. (1993). NSF-ARPA workshop on visual information management systems. *SIGMOD Rec.*, 22(3):57–75

- Jain, R., Pentland, A., and Petkovic, D. (1995). NSF-ARPA workshop visual inform. management systems.
- Jain, A. K., Ross, A., Prabhakar, S., “An Introduction to Biometric Recognition,” *IEEE Transactions on Circuits and Systems for Video Technology*, v. 14, n. 1(2004), 4-20.
- Kumar, A., Pang, G. K. H., “Defect Detection in Textured Materials Using Optimized Filters Systems,” *IEEE Transactions on Man and Cybernetics, Part B*, v. 32, n. 5 (2002), 553-570.
- Li, S., Kwok, J. T., Zhu, H., Wang, Y.: Texture Classification Using the Support Vector Machines. *Pattern Recognition*, Vol. 36, Issue 12 (2003) 2883-2893
- Liu Y., Lin, W. Deformable Texture: the Irregular-Regular-Irregular Cycle. *Proceedings of the 3rd International Workshop on Texture Analysis and Synthesis (Texture 2003)*, October, 2003, pp. 65 - 70.
- Ma, W.-Y. and Manjunath, B. S. (1999). Netra: A toolbox for navigating large image databases. *Multimedia Systems*, 7(3):184–198.
- Malik, J.; Belongie, S.; Shi, J.; Leung, T., Textons, contours and regions: cue integration in Image Segmentation *Computer Vision*, 1999. The Proceedings of the Seventh IEEE International Conference on Volume 2, Issue , 1999 Page(s):918 - 925 vol.2
- Mandal, M. K., Panchanathan, S., and Aboulnasr, T. (1997). Image indexing using translation and scale-invariant moments and wavelets. In *Storage and Retrieval for Image and Video Databases (SPIE)*, pages 380–389
- Ojala, T., Pietikainen, M., Maenpaa, T., “Multiresolution gray-scale and rotation invariant texture classification with local binary patterns,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 24, n. 7 (2002), 971-987.
- Parker R. James. *Algorithms for image processing and computer vision*. New York: Wiley Computer Publishing, 1997

- Pentland, A., Picard, R. W., and Sclaroff, S. (1996). Photobook: content-based manipulation of image databases. *Int. J. Comput. Vision*, 18(3):233–254.
- Porat, M., Zeevi, Y. Y., “Localized Texture Processing in Vision: Analysis and Synthesis in the Gaborian Space,” *IEEE Transactions on Biomedical Engineering*, v. 36, n. 1 (1989), 115-129.
- Randen, T., Husøy, J. H., “Filtering for Texture Classification: a Comparative Study,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 21, n. 4 (1999), 291-310.
- Savari, S. A., “Redundancy of the Lempel-Ziv-Welch Code,” *Proc. DCC*, Salt Lake City (1997), 191-200.
- Shannon, C. E., “A Mathematical Theory of Communication,” *Bell Syst. Tech. J.*, v. 27 (1948), 379-423.
- Southard, T. E., Southard, K. A., “Detection of Simulated Osteoporosis in Maxillae Using Radiographic Texture Analysis,” *IEEE Transactions on Biomedical Engineering*, v. 43, n. 2 (1996), 123-132.
- Teahan, W. J., Harper, D. J., “Using Compression Based Language Models for Text Categorization,” *Workshop on Language Modeling and Information Retrieval* (2001), 83-88.
- VisTex - MIT Media Lab Vision Texture - VisTex Database.  
<http://vismod.media.mit.edu/vismod/imagery/VisionTexture/vistex.html>
- Welch, T. A., “A Technique for High Performance Data Compression,” *IEEE Computer*, v. 17, n. 6 (1984), 8-19.
- Xu, K., Georgescu, B., Meer, P., Comaniciu, D., “Performance Analysis in Content-Based Retrieval with Textures”, International Conference on Pattern Recognition (ICPR'00)-Volume 4, 2000 Barcelona, Spain

- Zhu, Song Chun, Guo Cheng-en, Wu Ying Nian, Wang Yizhou, What Are Textons?, Proceedings of the 7th European Conference on Computer Vision-Part IV, p.793-807, May 28-31, 2002
- Zibreira, C., *Descrição e Procura de Vídeo Baseadas na Forma*, Master thesis, Instituto Superior Técnico de Lisboa, Portugal, 2000.
- Ziv, J., Lempel, A., "Compression of Individual Sequences via Variable-Rate Coding," *IEEE Transactions on Information Theory*, v. 24, n. 5 (1978), 530-536.
- Ziv, J., "On Classification with Empirically Observed Statistics and Universal Data Compression," *IEEE Transactions on Information Theory*, v. 34, n. 2 (1988), 278-286.
- Ziv, J., Merhav, N., "A Measure of Relative Entropy Between Individual Sequences with Application to Universal Classification," *IEEE Transactions on Information Theory*, v. 39, n. 4 (1993), 1270-1279.



# Artigos Publicados

BATISTA, Leonardo Vidal ; MEIRA, M. M. ; CAVALCANTI JÚNIOR, Nicomedes Lopes . Content-based texture image retrieval using the Lempel-Ziv-Welch algorithm. In: International Conference on Computer Vision Theory and Applications - VISAPP, 2006, Setúbal. Proceedings of the International Conference on Computer Vision Theory and Applications - VISAPP, 2006, 2006. v. 1. p. 106-113.

BATISTA, Leonardo Vidal ; MEIRA, M. M. ; CAVALCANTI JÚNIOR, Nicomedes Lopes . Texture Classification using Local and Global Histogram Equalization and the Lempel-Ziv-Welch Algorithm. In: The 5th International Conference on Hybrid Intelligent Systems - HIS'05, 2005, Rio de Janeiro. Proceedings of the 5th International Conference on Hybrid Intelligent Systems - HIS'05, 2005, 2005. v. 1. p. 1-6.

BATISTA, Leonardo Vidal ; MEIRA, M. M. . Texture Classification using Local Histogram Equalization and Lempel-Ziv-Welch Algorithm. In: XVII Brazilian Symposium on Computer Graphics and Image Processing - SIBGRAPI'2004, 2004, Curitiba. Proceedings of the XVII Brazilian Symposium on Computer Graphics and Image Processing - SIBGRAPI'2004, 2004. v. 1. p. 1-3.

BATISTA, Leonardo Vidal ; MEIRA, M. M. . Texture Classification using Histogram Equalization and the Lempel-Ziv-Welch Algorithm. In: XXI Simpósio Brasileiro de Telecomunicações - SBrT'2004, 2004, Belém. Anais do XXI Simpósio Brasileiro de Telecomunicações - SBrT'2004, 2004. v. 1. p. 1-6.

BATISTA, Leonardo Vidal ; MEIRA, M. M. . Texture Classification Using the Lempel-Ziv-Welch Algorithm. Lecture Notes in Computer Science, Berlin, v. 3171, p. 444-453, 2004.