Estudo Comparativo de Descritores para Recuperação de Imagens por Conteúdo na Web

Este exemplar corresponde à redação final da Dissertação devidamente corrigida e defendida por Otávio Augusto Bizetto Penatti e aprovada pela Banca Examinadora.

Campinas, 30 de abril de 2009.

Ricardo Torres

Prof. Dr. Ricardo da Silva Torres
Instituto de Computação - Unicamp
(Orientador)

Dissertação apresentada ao Instituto de Computação, UNICAMP, como requisito parcial para a obtenção do título de Mestre em Ciência da Computação.

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA DO IMECC DA UNICAMP
Bibliotecária: Maria Fabiana Bezerra Müller – CRB8 / 6162

Penatti, Otávio Augusto Bizetto
P371e Estudo comparativo de descritores para recuperação de imagens por conteúdo na Web/Otávio Augusto Bizetto Penatti -- Campinas, [S.P. : s.n.], 2009.

Orientador : Ricardo da Silva Torres
Dissertação (Mestrado) - Universidade Estadual de Campinas, Instituto de Computação.

1.Descritores - Estudos comparativos. 2. Processamento de imagens. 3. Ferramentas de busca na web. 4. World Wide Web (Sistema de recuperação de informação. I.Torres, Ricardo da Silva. II. Universidade Estadual de Campinas. Instituto de Computação. III. Título.

(mfb/imecc)

Título em inglês: Comparative study of descriptors for content-based image retrieval on the Web

Palavras-chave em inglês (Keywords): Descriptors – Comparative studies, Image processing, Web search agents, World Wide Web (Information retrieval system).

Área de concentração: Sistemas de Informação

Titulação: Mestre em Ciência da Computação

Banca examinadora: Prof. Dr. Ricardo da Silva Torres (IC-Unicamp)
Prof. Dr. Alexandre Xavier Falcão (IC-Unicamp)
Prof. Dr. João Marcos Bastos Cavalcanti (UFAM)
Prof. Dr. Hélio Pedrini (IC-Unicamp)
Prof. Dr. Silvio Jamil Ferzoli Guimarães (PUC Minas)

Data da defesa: 13/03/2009

Programa de Pós-Graduação: Mestrado em Ciência da Computação

Instituto de Computação
Universidade Estadual de Campinas

Estudo Comparativo de Descritores para Recuperação de Imagens por Conteúdo na Web

Otávio Augusto Bizetto Penatti[1]

Abril de 2009

Banca Examinadora:

- Prof. Dr. Ricardo da Silva Torres
  Instituto de Computação - Unicamp (Orientador)
- Prof. Dr. Alexandre Xavier Falcão
  Instituto de Computação - Unicamp
- Prof. Dr. João Marcos Bastos Cavalcanti
  Departamento de Ciência da Computação - UFAM
- Prof. Dr. Hélio Pedrini (Suplente)
  Instituto de Computação - Unicamp
- Prof. Dr. Silvio Jamil Ferzoli Guimarães (Suplente)
  Instituto de Informática - PUC Minas

[1]Suporte financeiro de: Bolsa Fapesp (processo 2006/59525-1) 01/03/2007–28/02/2009

Resumo

A crescente quantidade de imagens geradas e disponibilizadas atualmente tem feito aumentar a necessidade de criação de sistemas de busca para este tipo de informação. Um método promissor para a realização da busca de imagens é a busca por conteúdo. Este tipo de abordagem considera o conteúdo visual das imagens, como cor, textura e forma de objetos, para indexação e recuperação. A busca de imagens por conteúdo tem como componente principal o descritor de imagens. O descritor de imagens é responsável por extrair propriedades visuais das imagens e armazená-las em vetores de características. Dados dois vetores de características, o descritor compara-os e retorna um valor de distância. Este valor quantifica a diferença entre as imagens representadas pelos vetores. Em um sistema de busca de imagens por conteúdo, a distância calculada pelo descritor de imagens é usada para ordenar as imagens da base em relação a uma determinada imagem de consulta.

Esta dissertação realiza um estudo comparativo de descritores de imagens considerando a Web como cenário de uso. Este cenário apresenta uma quantidade muito grande de imagens e de conteúdo bastante heterogêneo. O estudo comparativo realizado nesta dissertação é feito em duas abordagens. A primeira delas considera a complexidade assintótica dos algoritmos de extração de vetores de características e das funções de distância dos descritores, os tamanhos dos vetores de características gerados pelos descritores e o ambiente no qual cada descritor foi validado originalmente. A segunda abordagem compara os descritores em experimentos práticos em quatro bases de imagens diferentes. Os descritores são avaliados segundo tempo de extração, tempo para cálculos de distância, requisitos de armazenamento e eficácia. São comparados descritores de cor, textura e forma. Os experimentos são realizados com cada tipo de descritor independentemente e, baseado nestes resultados, um conjunto de descritores é avaliado em uma base com mais de 230 mil imagens heterogêneas, que reflete o conteúdo encontrado na Web. A avaliação de eficácia dos descritores na base de imagens heterogêneas é realizada por meio de experimentos com usuários reais. Esta dissertação também apresenta uma ferramenta para a realização automatizada de testes comparativos entre descritores de imagens.

Abstract

The growth in size of image collections and the worldwide availability of these collections has increased the demand for image retrieval systems. A promising approach to address this demand is to retrieve images based on image content (Content-Based Image Retrieval). This approach considers the image visual properties, like color, texture and shape of objects, for indexing and retrieval. The main component of a content-based image retrieval system is the image descriptor. The image descriptor is responsible for encoding image properties into feature vectors. Given two feature vectors, the descriptor compares them and computes a distance value. This value quantifies the difference between the images represented by their vectors. In a content-based image retrieval system, these distance values are used to rank database images with respect to their distance to a given query image.

This dissertation presents a comparative study of image descriptors considering the Web as the environment of use. This environment presents a huge amount of images with heterogeneous content. The comparative study was conducted by taking into account two approaches. The first approach considers the asymptotic complexity of feature vectors extraction algorithms and distance functions, the size of the feature vectors generated by the descriptors and the environment where each descriptor was validated. The second approach compares the descriptors in practical experiments using four different image databases. The evaluation considers the time required for features extraction, the time for computing distance values, the storage requirements and the effectiveness of each descriptor. Color, texture, and shape descriptors were compared. The experiments were performed with each kind of descriptor independently and, based on these results, a set of descriptors was evaluated in an image database containing more than 230 thousand heterogeneous images, reflecting the content existent in the Web. The evaluation of descriptors effectiveness in the heterogeneous database was made by experiments using real users. This dissertation also presents a tool for executing experiments aiming to evaluate image descriptors.

Agradecimentos

Eu gostaria de agradecer a todos que contribuíram para a realização deste trabalho.

Agradeço ao meu orientador pela atenção e apoio durante o desenvolvimento do projeto, assim como pelo meu crescimento intelectual e acadêmico.

Agradeço à minha família pela motivação e apoio às minhas decisões.

Agradeço aos colegas que contribuíram com idéias, implementações, críticas e sugestões durante este período intenso de estudos.

Gostaria também de agradecer à FAPESP (processo 2006/59525-1), ao CNPq (projetos WebMAPS e BioCore) e ao Instituto Virtual FAPESP Microsoft (projeto eFarms) pelo apoio financeiro.

Sumário

Resumo **vii**

Abstract **ix**

Agradecimentos **xi**

1 Introdução **1**

2 Conceitos Básicos e Trabalhos Relacionados **7**

2.1 Descritor de imagens 7

2.2 Funções de distância 8

2.2.1 Distância L1 8

2.2.2 Distância L2 9

2.2.3 Distância L_∞ 9

2.2.4 Intersecção de histogramas 9

2.2.5 Distância dLog 10

2.2.6 Distância de Hamming 10

2.2.7 Modelo vetorial 10

2.3 Sistemas de recuperação de imagens na Web 11

2.3.1 Arquitetura de sistemas de recuperação de imagens na Web 11

2.3.2 Sistemas existentes 17

2.4 Critérios de avaliação 17

2.4.1 Complexidade da extração de características 18

2.4.2 Complexidade da função de distância 18

2.4.3 Requisitos de armazenamento 19

2.4.4 Eficácia 19

2.4.5 Ambiente de validação 20

3 Estudo Teórico **21**

3.1 Descritores de cor 22

3.1.1 Taxonomia para descritores de cor . . . 22
3.1.2 Tabelas comparativas . . . 23
3.1.3 Detalhes dos descritores . . . 24
3.2 Descritores de textura . . . 31
3.2.1 Taxonomia para descritores de textura . . . 31
3.2.2 Tabelas comparativas . . . 33
3.2.3 Detalhes dos descritores . . . 34
3.3 Descritores de forma . . . 41
3.3.1 Taxonomia para descritores de forma . . . 41
3.3.2 Tabelas comparativas . . . 43
3.3.3 Detalhes dos descritores . . . 44

4 Ferramenta para automatização de testes comparativos 53
4.1 Gerenciamento de descritores . . . 53
4.2 Gerenciamento de bases de imagens . . . 55
4.3 Realização de experimentos . . . 56
4.4 Visualização de experimentos . . . 59
4.4.1 Tempos e progresso do experimento . . . 59
4.4.2 Diretório de resultados . . . 60
4.4.3 Resultados de consultas . . . 60
4.4.4 Avaliação de eficácia com usuários . . . 61
4.5 Aspectos de implementação . . . 62

5 Experimentos 67
5.1 Medidas de avaliação . . . 67
5.1.1 Eficiência na extração de vetores de características e no cálculo de distância . . . 68
5.1.2 Eficiência no armazenamento de vetores de características . . . 68
5.1.3 Eficácia . . . 68
5.2 Bases de imagens . . . 70
5.2.1 ETH . . . 70
5.2.2 Brodatz . . . 71
5.2.3 Caltech101 . . . 71
5.2.4 Yahoo . . . 72
5.3 Resultados . . . 73
5.3.1 Cor . . . 73
5.3.2 Textura . . . 79
5.3.3 Forma . . . 84
5.3.4 Avaliação na base Yahoo . . . 87

6 Conclusões **101**
6.1 Contribuições 103
6.2 Trabalhos futuros 103

Bibliografia **105**

Lista de Tabelas

3.1 Comparação entre descritores de cor: complexidade da extração de características, complexidade da função de distância e requisitos de espaço. Legenda: n=número de pixels em uma imagem, vs=tamanho do vetor de características, NC=não está claro, ND=não menciona. 46

3.2 Comparação entre descritores de cor: ambiente de testes e validação. Legenda: HET=heterogênea, HETWEB=heterogênea da Web, NU=não usou, NC=não comparou com outros descritores, IT=comparou com variações do descritor proposto, ND=não menciona. 47

3.3 Comparação entre descritores de textura: complexidade da extração de características, complexidade da função de distância e requisitos de espaço. Legenda: n=número de pixels em uma imagem, vs=tamanho do vetor de características, NC=não está claro, ND=não menciona, Estat=estatísticos, Geo=geométricos, Mod=modelos, Proc=processamento de sinais/wavelets. 48

3.4 Comparação entre descritores de textura: ambiente de testes e validação. Legenda: HET=heterogênea, HETWEB=heterogênea da Web, NU=não usou, NC=não comparou com outros descritores, IT=comparou com variações do descritor proposto, ND=não menciona. 49

3.5 Comparação entre descritores de forma: complexidade da extração de características, complexidade da função de distância e requisitos de espaço. Legenda: SD=dependente de segmentação, n=número de pixels em uma imagem, vs=tamanho do vetor de características, NC=não está claro, ND=não menciona, NSeg=independente de segmentação. 50

3.6 Comparação entre descritores de forma: ambiente de testes e validação. Legenda: HET=heterogênea, HETWEB=heterogênea da Web, NU=não usou, NC=não comparou com outros descritores, IT=comparou com variações do descritor proposto, ND=não menciona. 51

5.1 Comparação entre descritores de cor: tempos de extração de 1 vetor de características absolutos e relativos ao descritor GCH. Unidade: milissegundos 74

5.2 Comparação entre descritores de cor: tempos de 1 cálculo de distância absolutos e relativos ao descritor GCH. Unidade: microssegundos 75
5.3 Comparação entre descritores de cor: tamanho absoluto em bytes de 1 vetor de características e relativo ao descritor GCH. 77
5.4 Comparação entre descritores de textura: tempos de extração de 1 vetor de características absolutos e relativos ao descritor LBP. Unidade: milissegundos 79
5.5 Comparação entre descritores de textura: tempos de 1 cálculo de distância absolutos e relativos ao descritor LBP. Unidade: microssegundos 81
5.6 Comparação entre descritores de textura: tamanho absoluto em bytes de 1 vetor de características e relativo ao descritor LBP. 81
5.7 Comparação entre descritores de forma: tempos de extração de 1 vetor de características absolutos e relativos ao descritor EOAC. Unidade: milissegundos 84
5.8 Comparação entre descritores de forma: tempos de 1 cálculo de distância absolutos e relativos ao descritor EOAC. Unidade: microssegundos 85
5.9 Comparação entre descritores de forma: tamanho absoluto em bytes de 1 vetor de características e relativo ao descritor EOAC. 85
5.10 Comparação entre descritores na base Yahoo: tempos de extração de 1 vetor de características absolutos e relativos ao descritor GCH. Unidade: milissegundos 88
5.11 Comparação entre descritores na base Yahoo: tempos de 1 cálculo de distância absolutos e relativos ao descritor GCH. Unidade: microssegundos 90
5.12 Comparação entre descritores na base Yahoo: tamanho absoluto em bytes de 1 vetor de características e relativo ao descritor GCH. 91
5.13 Médias gerais dos valores de P_{10}, P_{20} e P_{30} para todas as consultas na base Yahoo. 94
5.14 Médias dos valores de P_{10}, P_{20} e P_{30} para a consulta 1. 94
5.15 Médias dos valores de P_{10}, P_{20} e P_{30} para a consulta 2. 95
5.16 Médias dos valores de P_{10}, P_{20} e P_{30} para a consulta 3. 95
5.17 Médias dos valores de P_{10}, P_{20} e P_{30} para a consulta 4. 96
5.18 Médias dos valores de P_{10}, P_{20} e P_{30} para a consulta 5. 96
5.19 Médias dos valores de P_{10}, P_{20} e P_{30} para a consulta 6. 97
5.20 Médias dos valores de P_{10}, P_{20} e P_{30} para a consulta 7. 97
5.21 Médias dos valores de P_{10}, P_{20} e P_{30} para a consulta 8. 98
5.22 Médias dos valores de P_{10}, P_{20} e P_{30} para a consulta 9. 98
5.23 Médias dos valores de P_{10}, P_{20} e P_{30} para a consulta 10. 98
5.24 Médias dos valores de P_{10}, P_{20} e P_{30} para a consulta 11. 99
5.25 Médias dos valores de P_{10}, P_{20} e P_{30} para a consulta 12. 99

5.26 Médias dos valores de P_{10}, P_{20} e P_{30} para a consulta 13. 99
5.27 Médias dos valores de P_{10}, P_{20} e P_{30} para a consulta 14. 100
5.28 Médias dos valores de P_{10}, P_{20} e P_{30} para a consulta 15. 100
5.29 Médias dos valores de P_{10}, P_{20} e P_{30} para a consulta 16. 100

Lista de Figuras

1.1 Exemplo de imagem com múltiplas interpretações. 2
1.2 Resultados de uma busca por "driver" no Google Images em 07/01/2009. . 3
1.3 Resultados de uma busca por "mouse" no Google Images em 07/01/2009. . 3
1.4 Resultados de uma busca por "windows" no Google Images em 07/01/2009. 4

2.1 Componentes de um descritor de imagens. 8
2.2 Arquitetura geral de um sistema de busca de imagens na Web e seus principais módulos (adaptado de [50]). 12

3.1 Taxonomia para descritores de cor. 22
3.2 Taxonomia para descritores de textura. 32
3.3 Taxonomia para descritores de forma. 42

4.1 Arquitetura da ferramenta para comparação de descritores. 54
4.2 Tela de gerenciamento de descritores. 55
4.3 Tela de gerenciamento de bases de imagens. 56
4.4 Telas de configuração do experimento a ser realizado. 57
4.5 Tela de exibição dos experimentos realizados e experimentos em andamento. 59
4.6 Tela de exibição de resultados de consultas. 60
4.7 Tela para avaliação de eficácia com usuários. 61
4.8 Modelo Entidade-Relacionamento do banco de dados da ferramenta para comparação de descritores. 63
4.9 Tela inicial da ferramenta. 64

5.1 Exemplos de imagens da base ETH. 70
5.2 Exemplos de imagens da base Brodatz. 71
5.3 Exemplos de imagens da base Caltech101. 72
5.4 Exemplos de imagens da base Yahoo. 73
5.5 Curvas de Precision $\times$ Recall para os descritores de cor. 78
5.6 Curvas de Precision $\times$ Recall para os descritores de textura. 83
5.7 Curvas de Precision $\times$ Recall para os descritores de forma. 86

5.8 20 primeiras imagens retornadas para a consulta “elephant5” usando os descritores de forma. 87
5.9 20 primeiras imagens retornadas para a consulta “ketch4” usando os descritores de forma. 88
5.10 20 primeiras imagens retornadas para a consulta “motorbike1” usando os descritores de forma. 89
5.11 20 primeiras imagens retornadas para a consulta “leopard8” usando os descritores de forma. 90
5.12 Imagens de consulta escolhidas para a avaliação de eficácia com usuários reais na base Yahoo. 92

Capítulo 1

Introdução

Nos últimos anos a quantidade de imagens digitais disponíveis vem aumentando rapidamente. Entre os principais responsáveis estão a popularização das câmeras fotográficas digitais e a disseminação da Internet de alta velocidade. Estes fatores permitiram, entre outros, que a quantidade de imagens geradas aumentasse rapidamente e que estas imagens fossem colocadas em domínio público. Tudo isso significa uma quantidade gigantesca de informação visual presente no cotidiano de um número crescente de pessoas.

A disponibilização de grande parte da informação visual gerada atualmente acontece principalmente na Web. Este cenário faz com que os sistemas de busca atuais sejam projetados para serem capazes de trabalhar com a informação distribuída na Web, seja para busca de informação textual, seja para busca de informação visual. Devido à popularização e disseminação da Web, ela se tornou uma das maiores e mais heterogêneas bases de imagens existentes.

De acordo com [66], em março de 2003, o índice do Google Image Search [1] continha 425 milhões de imagens. Em 2005, segundo o blog oficial do Google [2], esta quantidade já havia aumentado para aproximadamente um bilhão e cem milhões de imagens.

Neste cenário, existe uma grande demanda por sistemas capazes de recuperar a informação visual disponível. Os sistemas de busca de informação visual, e mais especificamente, de busca de imagens, disponíveis atualmente são, em sua quase totalidade, baseados em descrição textual. Estes sistemas tentam associar conteúdo textual (palavras-chaves) ao conteúdo de uma imagem. A principal vantagem deste tipo de abordagem é que as descrições textuais, em geral, refletem o conteúdo semântico de uma imagem [20, 21].

Contudo, existem alguns problemas com a abordagem baseada apenas em descrição textual. O primeiro deles está relacionado com o processo de descrição textual de uma imagem. Além de ser cansativo de ser realizado manualmente, torna-se inviável quando

[1] http://images.google.com
[2] http://googleblog.blogspot.com/2005/02/get-picture.html

o número de imagens é muito grande. O segundo problema refere-se à subjetividade da descrição textual, já que a interpretação do conteúdo de uma imagem pode variar de uma pessoa para outra. Além disso, existem problemas de ambigüidade e sinônimos dos termos textuais usados. Como exemplo, a imagem da Figura 1.1 pode ter significados diferentes para pessoas diferentes. Ela poderia ser descrita, por exemplo, como "praia", "verão", "férias", "mar", "céu azul", "água cristalina". O terceiro grande problema está relacionado com a ausência de descrição textual. Muitas imagens são disponibilizadas na Web sem qualquer tipo de descrição ou texto associado. Desta maneira, um sistema de busca baseado apenas em descrição textual torna-se incapaz de recuperar estas informações visuais de maneira eficaz.

Figura 1.1: Exemplo de imagem com múltiplas interpretações.

Outros exemplos de problemas com a busca de imagens baseada apenas em descrição textual são ilustrados nas Figuras 1.2, 1.3 e 1.4. Foram feitas consultas utilizando-se o buscador de imagens do Google, que é baseado em descrição textual e é um dos mais usados atualmente. Os resultados obtidos ilustram o problema da ambigüidade de termos textuais. As Figuras 1.2, 1.3 e 1.4 consideram a realização de consultas textuais envolvendo os termos "driver", "mouse" e "windows", respectivamente. No caso da Figura 1.2, o termo *driver* pode se referir tanto ao equipamento para jogar golf quanto ao motorista de um carro. A palavra *mouse* pode significar tanto o animal quanto o dispositivo usado

em computadores, como se vê na Figura 1.3. Um problema similar também pode ocorrer com a palavra *windows* (Figura 1.4). Este termo pode ser referir tanto ao sistema operacional da Microsoft quanto à janelas de edificações.

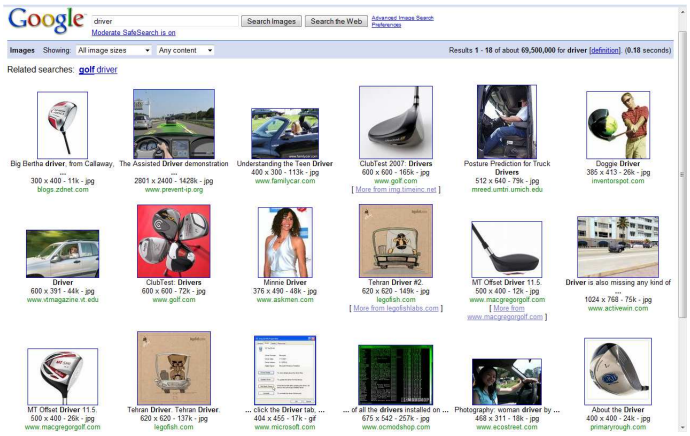


Figura 1.2: Resultados de uma busca por "driver" no Google Images em 07/01/2009.

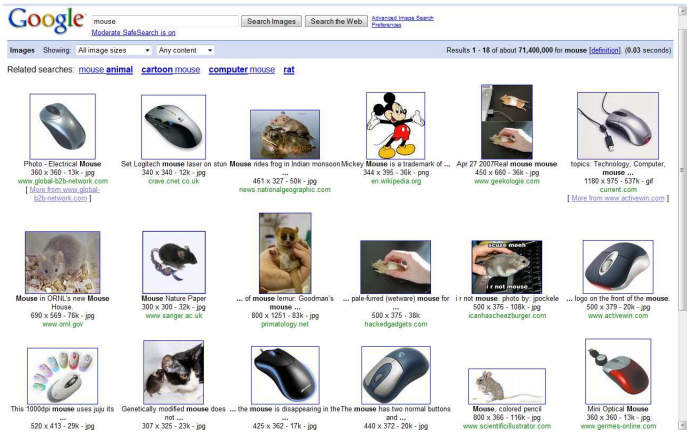


Figura 1.3: Resultados de uma busca por "mouse" no Google Images em 07/01/2009.

Uma outra abordagem para a busca de imagens pode solucionar muitos destes problemas. Trata-se da recuperação de imagens baseada em conteúdo (*Content-based Image Retrieval - CBIR*) [9, 11, 30, 88, 98, 118], que começou a ganhar força no início da década de 1990. Nesta abordagem, as imagens são identificadas a partir do seu conteúdo visual, como cor, textura e forma dos objetos. Isto permite que, em um processo de busca, imagens visualmente semelhantes sejam recuperadas sem que o processo de busca seja afetado

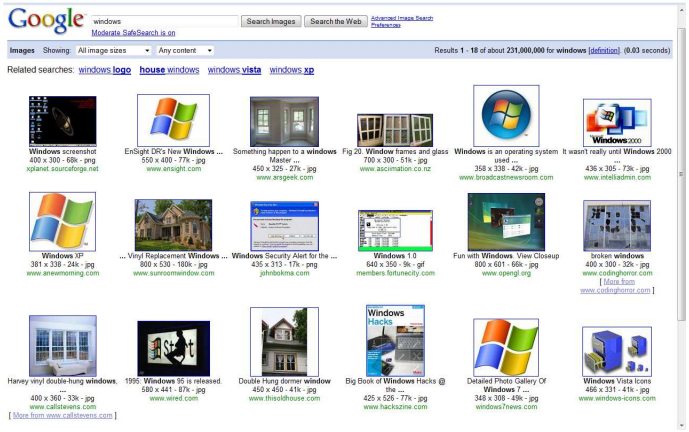


Figura 1.4: Resultados de uma busca por “windows” no Google Images em 07/01/2009.

por descrições textuais subjetivas, ambíguas ou mesmo inexistentes. Desta forma, grande parte das imagens que não poderiam ser encontradas ao se considerar apenas descrições textuais, podem ser recuperadas [23].

Um dos principais componentes de um sistema de recuperação de imagens por conteúdo é o **descritor de imagens**. Ele pode ser caracterizado por: (i) um algoritmo de extração de vetores de características das imagens e (ii) uma função de distância (medida de similaridade). Um descritor de imagens pode ser classificado dependendo do tipo de informação visual por ele considerado: cor, textura, forma e relacionamento espacial.

Neste cenário, o processo de busca ocorre da seguinte maneira. O usuário faz uma consulta ao sistema usando, por exemplo, uma imagem de exemplo. O vetor de características desta imagem é extraído e então comparado aos vetores das imagens da base, que já haviam sido extraídos previamente. O vetor extraído contém informações de propriedades visuais da imagem, como cor, forma e textura. As comparações entre os vetores retornam valores de distância que são usados para ordenar as imagens da base. Após isso, as imagens mais similares são finalmente exibidas ao usuário.

Esta dissertação tem como objetivo a apresentação de um estudo comparativo de descritores de imagens considerando a Web como cenário de uso. Este estudo considera, entre outros, a eficiência e a eficácia dos descritores quando aplicados a um ambiente extremamente heterogêneo e de dimensões gigantescas, como a Web.

As contribuições deste projeto seguem algumas das direções apontadas pelos “Grandes Desafios da Pesquisa em Computação no Brasil - 2006 - 2016” da Sociedade Brasileira de Computação (SBC). Mais especificamente, é tratado o desafio de lidar com grandes volumes de dados multimídia distribuídos. Este projeto também faz parte do Projeto

Universal: "Programação Genética aplicada à Recuperação de Imagens por Conteúdo", iniciado em março de 2007, apoiado pelo CNPq e coordenado pelos professores Ricardo da Silva Torres da Unicamp e Marcos André Gonçalves da UFMG.

Os resultados gerados durante o período do projeto já se converteram em duas publicações. Uma delas, no Simpósio Brasileiro de Computação Gráfica e Processamento de Imagens de 2008, contém um estudo comparativo de descritores de cor [85]. A outra refere-se à artigo publicado nos anais do Seminário Integrado de Software e Hardware de 2008 [26]. Neste artigo, são discutidos desafios de pesquisa relacionados à realização de busca de imagens na Web.

Nesta dissertação o estudo comparativo dos descritores é feito em dois níveis. O primeiro nível é teórico e considera a análise dos algoritmos e funções que compõem os descritores. O segundo nível avalia o desempenho dos descritores em experimentos práticos realizados em bases de imagens e utiliza alguns dos melhores descritores apresentados na análise teórica.

A dissertação está estruturada da seguinte maneira. O Capítulo 2 apresenta trabalhos relacionados e conceitos importantes na área de *CBIR*, assim como os critérios usados para comparar os descritores (Seção 2.4). O Capítulo 3 apresenta o resultado do estudo comparativo teórico realizado entre os descritores de imagens estudados. São apresentadas as principais abordagens para descrição de cor, textura e forma de uma imagem. São também apresentadas tabelas comparativas considerando os critérios apresentados na Seção 2.4 e explicações breves sobre o funcionamento de cada descritor escolhido para os experimentos práticos. O Capítulo 4 apresenta a ferramenta para a realização automatizada de testes desenvolvida neste projeto. O Capítulo 5 descreve os experimentos realizados assim como discute os resultados obtidos. Finalmente, o Capítulo 6 apresenta as conclusões e os trabalhos futuros deste projeto.

Capítulo 2

Conceitos Básicos e Trabalhos Relacionados

A recuperação de imagens baseada em conteúdo é uma abordagem alternativa à busca baseada em descrições textuais. Sua principal virtude está em considerar as características visuais das imagens no processo de indexação e recuperação. Entre as características visuais mais comumente analisadas estão cor, textura e forma dos objetos. Por meio desta abordagem, o processo de busca de imagens não é afetado pela ausência de descrições textuais ou pela existência de descrições subjetivas e ambíguas. Como conseqüência, a busca por conteúdo retorna imagens que não seriam consideradas relevantes por mecanismos de busca baseados em descrições textuais.

2.1 Descritor de imagens

Um dos principais componentes de um sistema de recuperação de imagens por conteúdo é o **descritor de imagens**. Conceitualmente, ele pode ser entendido como o responsável por quantificar o quão semelhante são duas imagens. Formalmente, um descritor de imagens D pode ser definido como uma tupla (ϵ_D, δ_D) [23], na qual

- ϵ_D é o componente responsável por realizar a caracterização de propriedades visuais de uma dada imagem de entrada em um vetor de características, como mostra a Figura 2.1. As propriedades visuais de uma imagem são processadas por meio de técnicas e algoritmos de processamento de imagens e são armazenadas em um vetor de características. Um vetor contém informações sobre os aspectos visuais de uma imagem, como informações de cor, de textura, de forma de objetos ou de relacionamento espacial entre objetos.

- δ_D é uma função responsável por comparar vetores de características. Como mostra a Figura 2.1, dados dois vetores como entrada, esta função calcula a distância entre esses vetores. A distância calculada entre os vetores é considerada como a distância entre as imagens que eles representam. Um outro termo comum para δ_D é função de similaridade. Um valor de similaridade é normalmente o inverso de um valor de distância, ou seja, quanto mais similares duas imagens, menos distantes elas são.

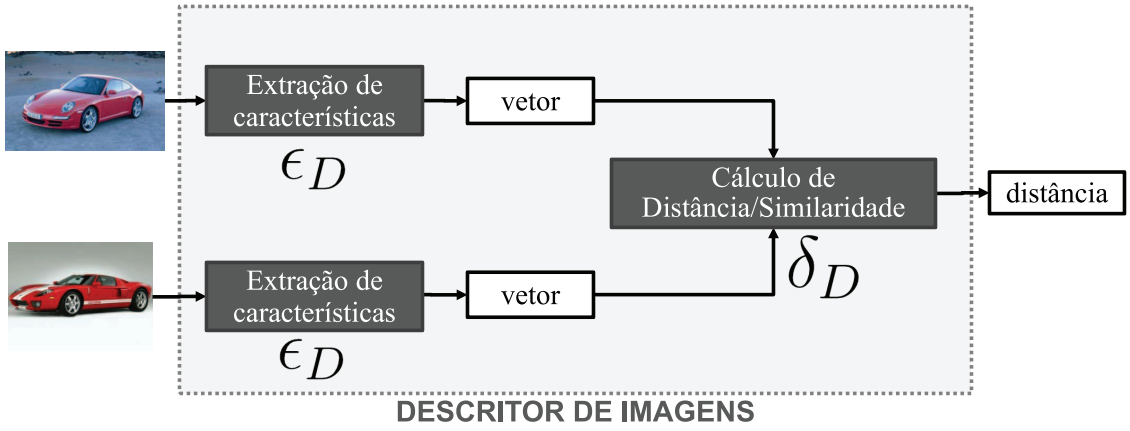


Figura 2.1: Componentes de um descritor de imagens.

2.2 Funções de distância

Funções de distância são componentes de descritores de imagens responsáveis pela comparação de vetores de características. Os valores calculados pelas funções de distância são responsáveis pela ordenação das imagens retornadas numa determinada consulta.

Funções de distância métricas respeitam as seguintes propriedades: (i) não negatividade, (ii) identidade, (iii) simetria e (iv) desigualdade triangular.

Nesta seção são explicadas algumas das funções de distância mais comuns em descritores de imagens.

2.2.1 Distância L1

A função de distância L1, também conhecida como *City-block*, é uma das funções de distância mais utilizadas. Ela calcula a distância entre dois vetores de características

acumulando as diferenças modulares entre elementos correspondentes dos vetores, como mostra a Equação 2.1 abaixo:

$$d_{L1}(V_Q, V_B) = \sum_{i=1}^{vs} |V_Q(i) - V_B(i)| \tag{2.1}$$

na qual V_Q e V_B são os vetores de características da imagem de consulta e da imagem da base, respectivamente, e vs é o tamanho dos vetores.

A função L1 é uma função métrica e necessita percorrer os vetores de características apenas uma vez para calcular a distância entre eles.

2.2.2 Distância L2

A função de distância L2 é também chamada de distância Euclidiana. Ela calcula a distância entre dois vetores de características percorrendo os vetores apenas uma vez, como mostra a Equação 2.2:

$$d_{L2}(V_Q, V_B) = \sqrt{\sum_{i=1}^{vs} (V_Q(i) - V_B(i))^2} \tag{2.2}$$

na qual V_Q e V_B são os vetores de características da imagem de consulta e da imagem da base, respectivamente, e vs é o tamanho dos vetores.

A função L2 é uma função métrica.

2.2.3 Distância L_∞

A função de distância L_∞ é uma função métrica e é dada pela Equação 2.3:

$$d_{L_\infty}(V_Q, V_B) = \max_{1 \leq i \leq vs} |V_Q(i) - V_B(i)| \tag{2.3}$$

na qual V_Q e V_B são os vetores de características da imagem de consulta e da imagem da base, respectivamente, e vs é o tamanho dos vetores.

2.2.4 Intersecção de histogramas

A função de distância de intersecção de histogramas (*hint*) foi inicialmente proposta em [101]. Seu objetivo é ser capaz de evitar influências do fundo de uma imagem no cálculo de distâncias, suportar mudanças de pontos de vista e oclusão de um objeto, e suportar variação de resolução de uma imagem.

Ela é dada pela Equação 2.4:

$$d_{hint}(V_Q, V_B) = 1 - \frac{\sum_{i=1}^{vs} \min(V_Q(i), V_B(i))}{\sum_{i=1}^{vs} V_Q(i)} \tag{2.4}$$

na qual V_Q e V_B são os vetores de características da imagem de consulta e da imagem da base, respectivamente, e vs é o tamanho dos vetores. A distância d_{hint} é normalizada pela soma dos bins do histograma da imagem de consulta.

2.2.5 Distância dLog

A função de distância *dLog*, proposta em [29], é utilizada para comparar histogramas. Sua principal vantagem é a diminuição do efeito de bins com valores muito altos no valor final da distância.

A distância dLog é calculada pela Equação 2.5:

$$d_{dLog}(V_Q, V_B) = \sum_{i=1}^{vs} |f(V_Q(i)) - f(V_B(i))| \tag{2.5}$$

$$f(x) = \begin{cases} 0, & \text{se } x = 0 \\ 1, & \text{se } 0 < x \leq 1 \\ \lceil log_2 x \rceil + 1, & \text{caso contrário} \end{cases}$$

na qual V_Q e V_B são os vetores de características da imagem de consulta e da imagem da base, respectivamente, e vs é o tamanho dos vetores.

2.2.6 Distância de Hamming

Em CBIR, a distância de Hamming é normalmente usada para comparar seqüências de bits. Dadas duas seqüências de bits, a distância é calculada contando-se os bits correspondentes que diferem de uma seqüência para outra. A distância de Hamming pode ser implementada a partir da aplicação de operações XOR entre cada par de bits e da contagem da quantidade de bits 1 no resultado, como mostra a Equação 2.6:

$$d_{Hamming}(V_Q, V_B) = \sum_{i=1}^{vs} V_Q(i) \oplus V_B(i) \tag{2.6}$$

na qual V_Q e V_B são os vetores de características da imagem de consulta e da imagem da base, respectivamente, e vs é o tamanho dos vetores.

2.2.7 Modelo vetorial

O modelo vetorial, comumente usado na recuperação de documentos textuais, também pode ser aplicado na recuperação de imagens por conteúdo [73]. Neste modelo, cada termo do vetor de características de uma imagem é considerado como uma coordenada em um espaço vs-dimensional, no qual vs é a quantidade de valores no vetor. A distância

entre dois vetores é dada pelo cosseno do ângulo formado pelos vetores no espaço vs-dimensional, como mostra a Equação 2.7:

$$d_{cosseno}(V_Q, V_B) = 1 - \frac{\sum_{i=1}^{vs} V_Q(i) * V_B(i)}{\sum_{i=1}^{vs} V_Q(i)^2 * \sum_{i=1}^{vs} V_B(i)^2} \tag{2.7}$$

na qual V_Q e V_B são os vetores de características da imagem de consulta e da imagem da base, respectivamente, e vs é o tamanho dos vetores.

2.3 Sistemas de recuperação de imagens na Web

Um sistema de busca de imagens por conteúdo para a Web deve ser cuidadosamente projetado, pois a Web apresenta características normalmente não encontradas em ambientes com conteúdo específico, como um sistema de informações geográficas ou um sistema de análise de imagens médicas, por exemplo. A seguir são descritos alguns dos principais módulos da arquitetura de um sistema de recuperação de imagens na Web [50].

2.3.1 Arquitetura de sistemas de recuperação de imagens na Web

A criação de um sistema de busca de imagens na Web exige o enfrentamento de vários desafios de pesquisa, os quais precisam de atenção especial devido à grande quantidade de informação presente na Web assim como a sua enorme heterogeneidade. Os principais componentes envolvidos estão resumidos na Figura 2.2. É preciso entender cada um desses componentes para se desenvolver um sistema de busca de imagens eficiente e eficaz. Alguns desses processos estão descritos mais detalhadamente a seguir.

Coleta de dados

Para a criação de um sistema de busca de imagens na Web alguns processos devem ser realizados antes de o sistema poder aceitar consultas dos usuários. Entre eles, estão os processos de coleta de imagens pela Web, extração de características e geração de índices. A extração de características pode ser feita por meio de descritores textuais, usando-se textos da página HTML na qual a imagem se encontra, ou por meio de descritores de imagens, que extraem as características visuais da própria imagem.

Ao se realizar a coleta dos dados na Web, os seguintes pontos devem ser considerados:

- Como percorrer a Web: normalmente utilizam-se *crawlers* que percorrem a Web automaticamente. O processo pode se iniciar em uma URL semente e então o *crawler* varre recursivamente os links dessa URL. Outra forma é usar um conjunto

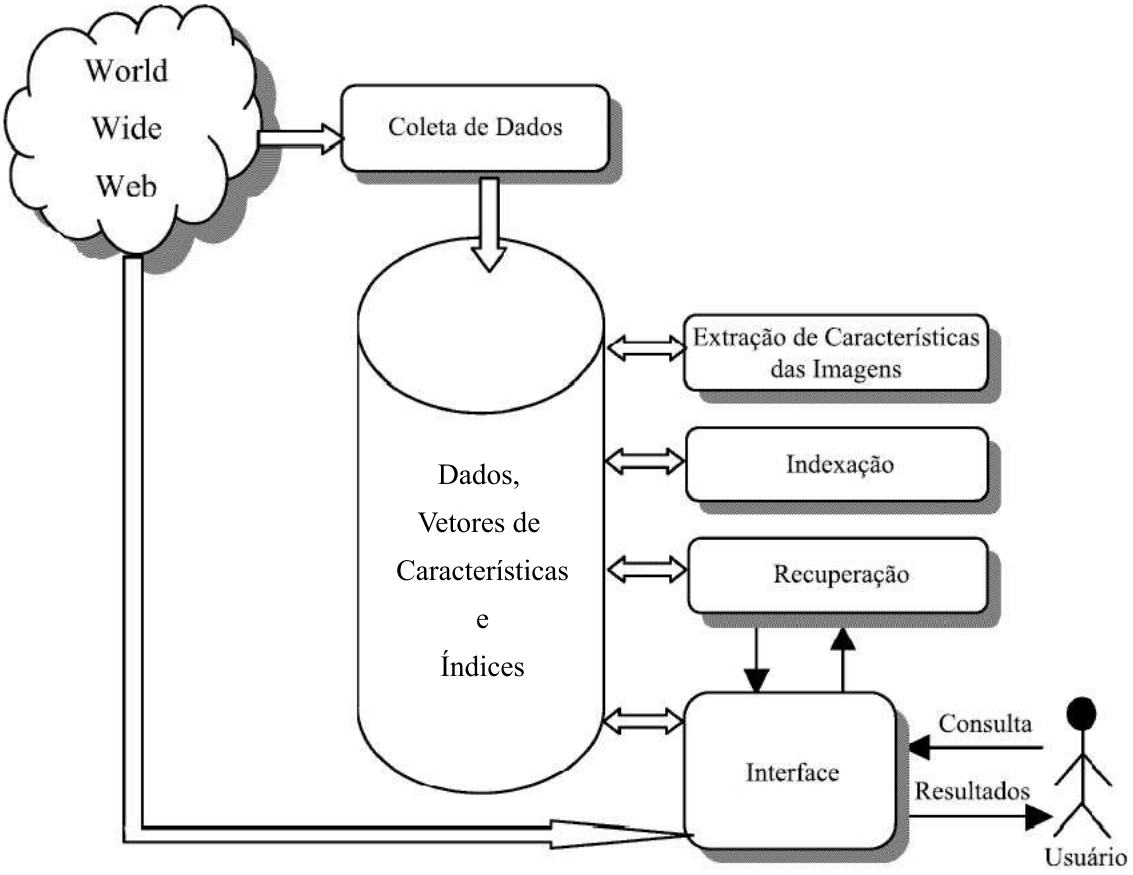


Figura 2.2: Arquitetura geral de um sistema de busca de imagens na Web e seus principais módulos (adaptado de [50]).

de URLs encontradas em sites populares, como o diretório do Yahoo![1], e percorrê-las recursivamente. Há ainda a possibilidade de particionar a Web em espaços de nomes ou países e deixar robôs explorarem esses espaços exaustivamente.

- Tratamento de *links* especiais: algumas informações somente poderão ser acessadas através de *links* especiais, como por exemplo, após a submissão de formulários. Os robôs de busca deverão, portanto, estar preparados para tratar tipos especiais de *links*.

- Armazenamento dos dados: é inviável manter as imagens coletadas em bases de dados locais já que quantidade de imagens na Web é imensa. Dessa forma, as imagens

[1] http://dir.yahoo.com

devem ser excluídas após suas características terem sido extraídas. Assim, as bases locais armazenarão apenas miniaturas das imagens, URLs, vetores de características e outras informações relevantes para o sistema.

- Atualização dos dados: o *crawler* deve percorrer a Web periodicamente para verificar atualizações, como o aparecimento de novas imagens ou a exclusão de imagens antigas, e manter os dados atualizados.
- Tempo de coleta de dados: para reduzir o tempo necessário para coletar os dados, podem ser usados vários robôs que percorrem a Web ao mesmo tempo.
- Conversão das imagens: as funções de extração de características normalmente usam um formato único de imagem como entrada; dessa maneira, as imagens coletadas devem ser convertidas do formato original, normalmente JPEG e GIF, para o formato aceito por estas funções.
- Criação de miniaturas: miniaturas das imagens devem ser geradas para que os resultados das buscas possam ser mostrados mais rapidamente aos usuários.
- Validade da imagem: é preciso saber quais imagens são válidas para o sistema de busca. Algumas imagens muito pequenas, por exemplo, podem não ser úteis para uma determinada aplicação.

Extração de características das imagens

Nesta etapa, as imagens são "digeridas" pelo sistema para que sejam extraídas suas características. As características podem ser extraídas tanto do texto próximo à imagem quanto da própria imagem.

Na primeira proposta, é analisado o texto em torno da imagem, ou seja, são extraídas informações baseadas no texto da página HTML que contém a imagem. A partir daí a recuperação das informações é textual, assim como a usada por vários buscadores de imagens, como o buscador de imagens do Google.

Na segunda proposta, são usados descritores de imagens que irão extrair propriedades visuais das imagens, como cor e textura. Dessa forma, a recuperação das imagens é feita por conteúdo. Neste módulo encaixam-se os descritores de imagens estudados neste projeto de mestrado.

Existe também a possibilidade de se combinar as duas propostas [10, 49, 95]. Além disso, é possível extrair atributos das imagens, como dimensão, tipo de arquivo, data, etc. Alguns buscadores de imagens atuais permitem refinar a busca a partir dos atributos das imagens, como o Yahoo! Image Search [2].

[2] http://images.yahoo.com

- Características textuais: a utilização de características textuais contribui para a busca de imagens por tornar possível a realização de buscas em alto nível. Trabalhos nesta área utilizam informações encontradas em textos que podem se referir ao conteúdo da imagem, como no texto HTML da página que contém a imagem, na URL, no nome do arquivo, etc. Em [20], assim como em [21], são feitos estudos sobre o uso do texto próximo à imagem para caracterizá-la. Existem, no entanto, dois problemas a respeito dessas essas abordagens. O primeiro refere-se à subjetividade relacionada à anotação de imagens. Uma imagem pode ter significados diferentes para cada pessoa e, por isso, o texto próximo à imagem pode ser a interpretação da pessoa que disponibilizou a imagem. O segundo problema refere-se à possível dificuldade em se descrever uma imagem textualmente. Uma possível solução para estes problemas seria o uso de ontologias, que, basicamente, permitem mapear termos textuais diferentes a conceitos semelhantes [38].

- Características visuais: as características visuais podem se referir a cor, textura, formas ou relacionamento espacial dos objetos de uma imagem. Um aspecto relevante é escolher quais características visuais representar. Em aplicações como as de reconhecimento de impressões digitais, características de textura podem ser suficientes. Em aplicações de reconhecimento de faces, representar formas talvez seja suficiente para uma boa eficácia. Entretanto, na Web, o cenário é muito mais complexo. Como as imagens são muito heterogêneas, a solução pode ser combinar uma grande variedade de descritores. No entanto, deve-se achar o conjunto de características que consiga abranger o maior número possível de categorias de imagens e que não degrade o desempenho do sistema. A avaliação de descritores de imagens considerando aspectos como estes consiste no enfoque deste projeto de mestrado.

Similaridade e busca exata

Busca exata (*matching*) é normalmente usada em casos que requerem uma resposta exata, ou seja, é uma operação binária. No caso da Web, uma busca de imagens genérica deverá retornar imagens que tenham certo grau de similaridade com a consulta realizada. Portanto, busca baseada em similaridade é mais adequada para a Web. Podem existir, porém, aplicações em que uma busca exata seja mais adequada, como por exemplo, a busca por imagens repetidas.

Índices

A utilização de índices é imprescindível para um sistema de buscas na Web. Por meio deles não é necessário que uma busca percorra toda a base de dados para encontrar resultados relevantes. Com isso, o tempo de resposta será muito menor. Um dos desafios da geração

de índices para um sistema de busca de imagens na Web consiste no fato de como lidar com a grande heterogeneidade e a grande quantidade de imagens na Web. Outro desafio refere-se ao problema da dimensionalidade, pois os vetores de características gerados pelos descritores de imagens normalmente possuem alta dimensionalidade e isto pode degradar o desempenho de um índice [6, 111].

Estruturas de indexação como a M-tree [19] e a Slim-tree [15] usam propriedades dos espaços métricos para a redução dos espaços de busca.

Especificação de consultas

Em um sistema de busca textual comum, o usuário realiza consultas a partir de palavras-chaves. Por outro lado, em um sistema de busca de imagens por conteúdo as consultas do usuário podem ser de diferentes formas, como, consultas com imagens de exemplo [42], consultas informando apenas as características visuais desejadas, consultas através de representações icônicas (como por exemplo, o posicionamento de objetos em um espaço) ou consultas por esboços, nas quais o usuário pode desenhar o objeto de interesse.

A utilização de consultas textuais agregadas às consultas visuais ajuda a reduzir o *semantic gap* [98], que é a dificuldade em se relacionar a interpretação do usuário às informações visuais da imagem.

Recuperação e refinamento

Recuperar consiste em fornecer os resultados da busca para o usuário. No caso de sistemas de buscas de imagens na Web, a recuperação consiste em retornar as imagens mais similares ao padrão de consulta definido pelo usuário, como por exemplo, uma imagem de consulta. Entre as técnicas normalmente usadas destacam-se a dos *k-vizinhos mais próximos* e a *consulta por abrangência*. A primeira consiste em retornar para o usuário apenas as k imagens mais próximas à consulta. A segunda técnica recupera todas as imagens que tenham uma distância menor que um valor definido como raio de busca em relação à consulta [19].

No contexto da Web estas técnicas apresentam alguns problemas. No caso da primeira, pode ser necessário explorar um grande espaço de busca para encontrar os vizinhos mais próximos devido ao tamanho da coleção de imagens e à utilização de vetores de características de grande dimensão. Outro problema é que não há garantia de que os resultados estão próximos ao padrão especificado.

Na segunda técnica, existe garantia de que os resultados estão próximos à consulta. Entretanto, dois problemas podem ocorrer: quando o raio de busca é muito pequeno, nenhuma imagem pode ser retornada; quando o raio de busca é muito grande, um número muito grande de imagens podem ser retornadas. No caso de muitos resultados, pode-se

optar por truncá-los, o que provocará uma migração para a técnica dos k-vizinhos mais próximos.

Após a recuperação das imagens, usam-se técnicas para refinamento dos resultados. A mais conhecida é a realimentação de relevância ou *relevance feedback* [129]. Por meio dela, o usuário pode indicar quais imagens são realmente relevantes e quais não devem ser consideradas. Assim o sistema pode se ajustar e fornecer resultados melhores em buscas futuras. Técnicas de refinamento, como a *relevance feedback*, ajudam o sistema a "entender" melhor as necessidades do usuário e, dessa forma, possibilitam uma redução no espaço de busca e no tempo de recuperação [34, 35, 93].

Apesar das vantagens oferecidas pela busca baseada em conteúdo, ela também possui deficiências. A principal delas está relacionada à falta de informação semântica quando se considera apenas o conteúdo visual da imagem. Este problema é conhecido como *semantic gap* [98]. Existem diversas abordagens para se tentar amenizá-lo, entre elas está o uso de informação textual em conjunto com a informação visual [3] e o uso de técnicas de refinamento, como a realimentação de relevância [34, 35, 93, 129].

Metabusca de imagens

Uma maneira de se fazer um sistema de busca de imagens na Web é utilizar metabusca, ou seja, realizar uma determinada consulta em diversos sistemas de busca existentes e depois combinar os resultados para exibir ao usuário. É uma maneira de expor os usuários a diversos sistemas de busca de forma transparente. Este tipo de busca pode amenizar o problema de cobertura da Web, que se refere à incapacidade de um sistema de busca cobrir todo o conteúdo da Web. Combinando os resultados de vários sistemas de busca, pode-se aumentar a área de cobertura da Web, já que sistemas distintos podem cobrir regiões diferentes dela.

Avaliação de desempenho

Assim que o sistema de busca estiver pronto, deve-se avaliar seu desempenho para saber sua eficiência e eficácia. A comparação de sistemas de busca de imagens na Web é difícil de ser realizada, pois as bases de imagens usadas são normalmente diferentes. Como é muito difícil cobrir toda a Web, as bases de imagens tendem a abranger espaços distintos da rede.

Apesar disso, é desejável que um sistema de busca de imagens na Web exiba os resultados mais relevantes nas primeiras posições, ou seja, que seja feito um *ranking* adequado. O sistema deve permitir também a realização de refinamento da busca.

Outros parâmetros importantes seriam o tempo de resposta e a facilidade de uso do sistema de busca.

2.3.2 Sistemas existentes

Em [50] são citados alguns sistemas de recuperação de imagens por conteúdo na Web. Dentre eles, destacam-se o ImageRover [95], o Diogenes [10], o ImageScape [62] e o Atlas WISE [49].

O ImageRover [95], desenvolvido na Universidade de Boston, EUA, representa as imagens analisando o conteúdo HTML de onde elas se encontram e analisando histogramas de cor, histogramas de orientação e textura das imagens. O ImageRover utiliza uma combinação de busca textual e visual e foi experimentado em uma base de cem mil imagens coletadas da Web.

O Diogenes [10], desenvolvido na Universidade de Illinois, EUA, é um sistema específico para busca de imagens de faces de celebridades. O usuário primeiro informa o nome da celebridade textualmente. O sistema então realiza um busca em sistemas de busca textual, como o Google, e os resultados são avaliados levando em conta o conteúdo HTML de onde as imagens foram encontradas e o conteúdo visual delas, a partir de um sistema detector de faces.

O ImageScape [62], desenvolvido na Universidade de Leiden na Holanda, realiza buscas usando consultas textuais ou por esboços e ícones semânticos. Neste último, o usuário posiciona um objeto icônico na posição em que ele deve aparecer na imagem desejada ou então faz um esboço. Este esboço é avaliado pelas suas bordas e contornos.

O Atlas WISE [49], projeto do Departamento de Matemática e Ciência da Computação da Universidade de Sherbrooke, Canadá, faz a coleta de dados da Web navegando por páginas de índices populares, como o Google e o Yahoo!. As características visuais são avaliadas por histogramas de cor e histogramas de orientação de bordas. O sistema também possui um módulo de *relevance feedback*, no qual o usuário pode informar exemplos positivos e negativos na formulação de consultas.

Uma análise mais detalhada de outros sistemas pode ser encontrada em [50].

Cabe destacar que atualmente o Google Image Search permite o refinamento dos resultados das consultas textuais usando-se características visuais. É possível escolher que os resultados sejam apenas imagens de faces, clip arts, desenhos de linha ou fotografias.

2.4 Critérios de avaliação

Nesta seção são apresentados os critérios usados para avaliar os descritores de imagens presentes neste estudo.

A escolha dos critérios foi baseada nos principais elementos envolvidos na utilização de um sistema de busca de imagens na Web. Durante um uso comum de um sistema de busca de imagens por conteúdo ocorrem as etapas a seguir:

- Um usuário define sua consulta, usando, por exemplo, uma imagem de consulta;
- O sistema deve extrair as características desta imagem de consulta;
- O vetor de características da imagem de consulta é comparado aos vetores das imagens da base;
- Os valores de distância retornados pelas comparações devem ser ordenados;
- Uma lista ordenada da imagem mais semelhante para a imagem menos semelhante em relação à consulta é exibida ao usuário.

Os critérios são os seguintes: complexidade da extração de características (Seção 2.4.1), complexidade da função de distância (Seção 2.4.2), requisitos de armazenamento (Seção 2.4.3), eficácia (Seção 2.4.4) e ambiente de validação (Seção 2.4.5).

2.4.1 Complexidade da extração de características

O algoritmo de extração de características de um descritor é usado sempre que as características de uma imagem precisam ser extraídas. Isto ocorre em dois momentos em um sistema de busca de imagens por conteúdo.

O primeiro momento ocorre quando as imagens estão sendo coletadas da Web para serem inseridas na base local. Cada imagem coletada deve ser processada pelo sistema, ou seja, seu vetor de características deve ser extraído antes que ela seja incluída na base local. Este processo acontece, em geral, em segundo plano (*off-line*).

O segundo momento ocorre quando um usuário submete uma consulta ao sistema. Considerando que a consulta é feita por meio de uma imagem de exemplo, as características desta imagem devem ser extraídas assim que a consulta é submetida. O sistema só consegue iniciar a busca depois que as características da consulta já tiverem sido extraídas. Portanto, trata-se de um processo realizado em tempo de consulta, e tem impacto no tempo de resposta do sistema.

Desta maneira, uma extração de vetores de características é realizada principalmente *off-line* e uma outra extração é realizada em tempo de consulta (*on-line*). Mesmo a parte mais custosa do uso da função de extração de vetores de características sendo feita *off-line*, uma parte importante acontece *on-line*. Daí a importância de que a extração de características seja rápida.

2.4.2 Complexidade da função de distância

A função de distância de um descritor é normalmente a responsável pela maior parte do tempo gasto pelo sistema de busca de imagens por conteúdo ao processar uma consulta.

Durante um processo de busca, a imagem de consulta deve ser comparada a uma quantidade muito grande de imagens, pois, apenas após todas as comparações serem realizadas é que se pode ordenar os resultados. Considerando o cenário da Web, nota-se que o número de imagens a ser comparado é significativamente maior. Se considerarmos a realização de uma busca seqüencial, teremos um número exorbitante de comparações a serem realizadas. Desta forma, pode-se notar a importância de que a função de distância seja bastante rápida.

A função de distância também é importante devido a questões de indexação. O uso de estruturas de indexação é muito importante em sistemas com grandes bases de dados e imprescindível em um ambiente como a Web. Sem o uso de um índice, o tempo de resposta a uma consulta é muito maior. A utilização de índices permite uma redução no espaço de busca, eliminando comparações com imagens potencialmente pouco semelhantes à consulta. Isto é possível quando o espaço de busca é métrico, ou seja, quando a função de distância de um descritor é métrica. Exemplos de funções de distâncias métricas são as funções L1 e L2.

2.4.3 Requisitos de armazenamento

Um descritor de imagem caracteriza as propriedades visuais de uma imagem por meio de vetores de características. Cada imagem da base local possui um ou mais vetores de características associados. Portanto, o espaço necessário para o armazenamento de vetores de características é proporcional à quantidade de imagens da base. Em um cenário Web, no qual existe um número muito grande de imagens, a quantidade de espaço necessária é bem elevada. Na Web, é preciso ainda lidar com a grande heterogeneidade das imagens, o que torna quase indispensável o uso de mais de um descritor de imagens. Dessa forma, o espaço de armazenamento necessário é proporcional também à quantidade de descritores usados.

Estes fatores ilustram a importância de se ter descritores capazes de gerar vetores de características compactos, pois quanto mais compactos os vetores de características gerados, menor o impacto dos requisitos de armazenamento para um sistema de busca de imagens por conteúdo na Web.

2.4.4 Eficácia

O termo eficácia refere-se à capacidade de um descritor de imagens conseguir recuperar imagens relevantes. Um descritor com alta eficácia retorna as imagens em ordem decrescente de relevância, ou seja, apresentando imagens mais relevantes nas primeiras posições e imagens menos relevantes nas últimas posições. Alta eficácia também pode indicar que não são mostradas imagens sem relevância em meio à imagens relevantes.

O sucesso de um sistema de busca de imagens é fortemente dependente da qualidade dos resultados, isto é, ele depende muito da eficácia dos descritores usados. O usuário pode tolerar um tempo de resposta maior, mas não irá tolerar a exibição de resultados sem relevância. Desta forma, a escolha dos descritores deve ponderar questões de eficácia e eficiência.

2.4.5 Ambiente de validação

Neste estudo foi considerado também como critério de avaliação o ambiente de validação do descritor. Informações deste tipo foram coletadas dos artigos nos quais o descritor é proposto ou apresentado. Por ambiente de validação de um descritor, foram consideradas as bases de imagens em que o descritor foi testado, com quais outros descritores ele foi comparado e quais medidas de desempenho foram utilizadas. Estes parâmetros possibilitam verificar, entre outros aspectos, a capacidade do descritor de lidar com grandes bases de imagens heterogêneas. Também é possível observar se um descritor apresenta vantagens em relação a descritores já existentes na literatura.

Capítulo 3

Estudo Teórico

Este capítulo apresenta o resultado do estudo comparativo teórico realizado entre os descritores de imagens estudados. Considerando alguns dos descritores de imagens existentes na literatura foi feita uma análise teórica levando-se em conta os critérios apresentados na Seção 2.4. Nesta dissertação, procurou-se considerar descritores importantes da literatura e descritores recentemente propostos. A análise feita foi baseada nos artigos nos quais cada descritor foi proposto. Em alguns casos também foram consideradas informações adicionais de estudos comparativos que incluíam o descritor analisado.

A análise dos algoritmos de extração de características e das funções de distância dos descritores foi feita utilizando-se a notação de complexidade assintótica O.

Para o algoritmo de extração de características foi considerada como operação básica a visitação de um pixel da imagem, sendo a quantidade de pixels de uma imagem igual a n. Por exemplo, um algoritmo de extração de características que possui complexidade $O(n)$ extrai o vetor de características trabalhando linearmente no tamanho da imagem. Em casos nos quais uma operação de convolução é aplicada na imagem, verificou-se que o tamanho das máscaras usadas é sempre muito menor do que o tamanho da imagem, portanto a complexidade assintótica calculada não é afetada pela operação de convolução.

Para a função de distância, a operação básica considerada é a visitação de um elemento do vetor de características. Desta forma, existem, por exemplo, funções de distância lineares ou quadráticas no tamanho do vetor.

Para a avaliação dos requisitos de armazenamento foi analisado o tamanho do vetor de características gerado por cada descritor. Esta análise considerou a quantidade de valores numéricos existentes no vetor. Em alguns casos, o vetor é formado por valores binários, sendo, portanto, seu tamanho analisado em bits. Para vários dos descritores estudados, a quantidade de valores armazenada no vetor de características depende da quantização realizada durante o processo de extração. Dessa forma, o vetor não possui um tamanho fixo.

Durante o estudo dos descritores foram encontradas dificuldades principalmente relacionadas à análise das complexidades dos algoritmos. Na grande maioria dos artigos as complexidades assintóticas não são apresentadas e, em alguns casos, nada é mencionado sobre o custo computacional do descritor. Em outros casos, as informações presentes nos artigos não foram suficientes para se analisar a complexidade assintótica dos algoritmos.

A seguir são apresentados os resultados da análise teórica para descritores de cor (Seção 3.1), textura (Seção 3.2) e forma (Seção 3.3).

3.1 Descritores de cor

Uma das propriedades visuais mais importantes identificadas em imagens pela visão humana é a cor. Isto faz com que a cor seja uma das propriedades visuais mais estudadas na literatura e uma das mais usadas em sistemas de CBIR.

A seguir são apresentadas: uma taxonomia para descritores de cor (Seção 3.1.1), tabelas comparativas entre os descritores de cor estudados (Seção 3.1.2); e são detalhados os descritores de cor implementados (Seção 3.1.3).

3.1.1 Taxonomia para descritores de cor

Na literatura existem quatro abordagens principais para a análise de cor, como mostra a Figura 3.1.

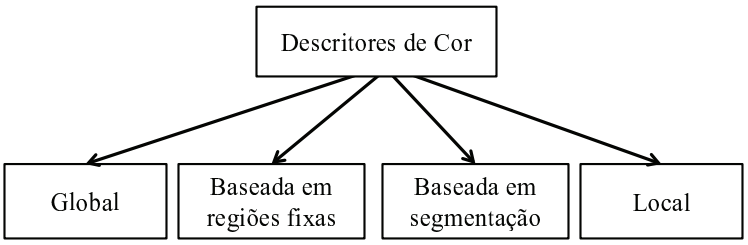


Figura 3.1: Taxonomia para descritores de cor.

A abordagem global considera a informação de cor de uma imagem globalmente. Como nenhum particionamento ou pré-processamento da imagem é necessário durante a extração de características, descritores que seguem esta abordagem normalmente apresentam algoritmos de extração de características mais simples e rápidos. No entanto, como nenhuma informação da distribuição espacial das cores é considerada, a eficácia destes descritores

pode ser baixa. Muitos dos descritores que seguem esta abordagem codificam as informações de cor em histogramas, como o *histograma global de cor* [101] e o *histograma acumulado* [99].

A abordagem baseada em regiões fixas divide a imagem em células de tamanho fixo e extrai informações de cor de cada célula separadamente. Esta abordagem codifica uma quantidade maior de informação espacial e, no entanto, descritores deste tipo normalmente geram vetores de características pouco compactos. Como exemplo, têm-se os descritores *histograma local de cor* [101] e *cell/color histogram* [28].

A abordagem baseada em segmentação divide uma imagem em regiões que podem variar em quantidade e tamanho de uma imagem para outra. Esta divisão é normalmente feita por um algoritmo de segmentação ou agrupamento de cores, o que introduz uma complexidade adicional durante o processo de extração de características. Outra forma de segmentação é a realização de um classificação dos pixels da imagem antes de se realizar a extração do vetor de características. Descritores deste tipo podem apresentar melhor eficácia, embora quase sempre apresentem maior complexidade computacional. Os descritores *color-based clustering* [27] e *dominant-color* [32,72] são exemplos de descritores de cor baseados em segmentação.

A abordagem local usa algoritmos para a extração de características que encontram pontos de interesse na imagem e os caracterizam por informações invariantes a transformações como orientação e escala. Dessa maneira, o vetor de características contém informações sobre elementos locais da imagem. Descritores desta abordagem são baseados em algoritmos como o *SIFT* [68], como os descritores *Zhang SIFT* [126] e *CSIFT* [1].

3.1.2 Tabelas comparativas

Baseando-se nos critérios apresentados na Seção 2.4, foram montadas duas tabelas em que são comparados os descritores de cor estudados. Na Tabela 3.1 são apresentadas as complexidades dos algoritmos de extração e das funções de distância dos descritores de cor estudados, assim como os tamanhos dos vetores de características gerados. Além disso, a Tabela 3.1 apresenta a classe taxonômica a que pertence cada descritor, de acordo com a Figura 3.1.

Pelas complexidades encontradas, pode-se observar que descritores como DCSD [113], DCDI [116], DCD [72], WC [75] e CBC [27] apresentam algoritmos de extração um pouco mais caros computacionalmente em comparação à grande maioria de descritores que extrai o vetor de características em complexidade $O(n)$.

Considerando as complexidades das funções de distância, nota-se que descritores como CSIFT [1], Nayar MRH [39] e DCD [72] apresentam uma complexidade maior do que a grande maioria dos descritores, sendo estes últimos lineares no tamanho do vetor para

calcular a distância. Altas complexidades destes algoritmos podem afetar o tempo de resposta a uma consulta, como explicado na Seção 2.4.

Considerando os requisitos de armazenamento, observa-se que descritores como Zhang SIFT [126], JAC [112], CCH [28], Nayar MRH [39], LCH [101] e ACC [45] geram vetores de características pouco compactos. Isto pode fazer com que os gastos com dispositivos de armazenamento sejam maiores em um sistema de CBIR, como explicado na Seção 2.4. Já os descritores CW-LUV [77] e CW-HSV [107] geram vetores bastante compactos.

Na Tabela 3.2 são apresentados os ambientes de validação de cada descritor estudado [1].

Observa-se que muitos descritores são validados em bases de dados muito restritas e de conteúdo bastante específico, como os descritores CW-LUV [77], Chaira Fuzzy [16] e CM [83]. Outros descritores são comparados apenas com variações deles próprios, como os descritores Zhang SIFT [126], CW-HSV [107] e os descritores do MPEG-7 [72]. Em outros casos a comparação é feita apenas com o histograma global de cor (GCH), como em CGCH [99], Color Blocks [63] e CCV [84]. Além disso, observa-se que muitos dos descritores da Tabela 3.2 usam o descritor GCH como base de comparação, como os descritores LeeSC [60], CGCH [99], JAC [112], CCH [28], Color Bitmap [69], Coesão [22], Color Blocks [63], BIC [29], CBC [27], CCV [84] e ACC [45].

Na literatura são encontrados vários estudos comparativos entre descritores de cor. O estudo apresentado em [8] compara os descritores de cor do padrão MPEG-7. O estudo conclui que o descritor CSD [72] é o de melhor desempenho nos experimentos realizados. Já o descritor CLD [72] foi o mais sensível a ruído.

Um outro estudo comparativo com os descritores de cor do MPEG-7 é apresentado em [81]. Foi usada uma base de imagens contendo 2445 imagens divididas em categorias semânticas. As medidas apresentadas no estudo indicaram uma melhor eficácia para o descritor CSD [72], seguido pelos descritores SCD [72], CLD [72] e DCD [72], nesta ordem. Segundo o estudo, o descritor DCD, apesar de ser o mais caro computacionalmente, apresentou resultados ruins por ser mais focado em características locais, como objetos ou regiões, e não em imagens como um todo.

3.1.3 Detalhes dos descritores

Abaixo são explicados com mais detalhes os descritores escolhidos para os experimentos práticos e são dadas as razões de cada escolha.

São os descritores: GCH [101], CGCH [99], LCH [101], CCV [84], ACC [45], JAC [112], BIC [29], CBC [27], Color Bitmap [69], CSD [72], CW-HSV [107], CW-LUV [77] e CM [83].

[1] As medidas de *Precision* e *Recall*, que podem ser traduzidas para o português como *precisão* e *revocação*, respectivamente, não foram traduzidas

Histograma global de cor - GCH

O histograma global de cor ou *GCH* [101] é o descritor mais popular da literatura. Ele é comumente usado em comparações por ser um dos mais antigos, por ser simples de implementar e por possuir eficácia aceitável em muitas aplicações. Por estes motivos o GCH foi escolhido para ser implementado e testado nos experimentos.

Seu algoritmo de extração de características quantiza o espaço de cor em uma quantidade uniforme de bins e percorre a imagem calculando a quantidade de pixels que pertence a cada bin. Dessa forma, o vetor de características é extraído percorrendo-se a imagem uma única vez. O tamanho do vetor de características gerado depende da quantização realizada. Nos experimentos foi usada uma quantização uniforme do espaço RGB em 64 bins, por isso, o vetor gerado possui 64 valores.

A função de distância utilizada pelo GCH pode variar. Em [101] é utilizada a intersecção de histogramas. Nos experimentos realizados foi utilizada a função L1.

Histograma de cor acumulado - CGCH

O histograma de cor acumulado ou *CGCH* [99] tem um algoritmo de extração de características semelhante ao do descritor GCH. A única diferença é que o valor de um bin é acumulado no bin seguinte. Desta maneira, o último bin do histograma possui a soma dos valores de todos os bins anteriores acrescido dos valores do próprio bin.

A função de distância usada também pode variar. Em [99] são usadas as funções L1, L2 e L_∞. Na versão implementada foi usada a distância L1. Foi utilizado o espaço de cor RGB quantizado em 64 bins.

O CGCH foi escolhido para os experimentos práticos pois possui um algoritmo de extração de características e uma função de distância simples. Além disso, ele também é muito difundido na literatura.

Histograma local de cor - LCH

O histograma local de cor ou *LCH* [101] é um dos descritores mais populares que segue a abordagem baseada em regiões fixas. Seu algoritmo de extração de características divide a imagem em células de tamanho fixo e calcula um histograma de cor para cada célula. Ao final, os histogramas das células são concatenados para formar um único grande histograma.

A função de distância usada em [101] é a intersecção de histogramas. Na versão implementada foi usada a função L1 e o espaço de cor RGB quantizado em 64 bins. Uma imagem de entrada do algoritmo foi dividida em 16 células (grade 4x4), por isso, o vetor de características gerado contém 1024 bins. Apesar de gerar um vetor pouco compacto,

o LCH é um dos descritores mais populares da abordagem baseada em regiões fixas e comumente usado como comparação com descritores da mesma abordagem.

Vetor de coerência de cor - CCV

O descritor vetor de coerência de cor ou *CCV* [84] é também bastante popular na literatura. Seu algoritmo de extração de características classifica os pixels da imagem em coerentes ou incoerentes. Esta classificação leva em conta se o pixel pertence ou não a uma região de cores semelhantes, chamada de região coerente. Pixels que fazem parte de uma região coerente são classificados como coerentes, caso contrário, são classificados como incoerentes. Para se calcular as regiões coerentes, a imagem é primeiramente suavizada e o espaço de cor é discretizado para que sejam eliminadas pequenas variações entre pixels vizinhos. A seguir são encontradas as componentes conexas da imagem a fim de se realizar a classificação dos pixels em coerentes ou incoerentes. Após a classificação é calculado um histograma para os pixels coerentes e um histograma para os pixels incoerentes.

Como função de distância podem-se usar diversas funções que comparam histogramas, como as funções L1 ou L2. Nos experimentos foi utilizada a função L1 com o espaço RGB quantizado em 64 bins, portanto o vetor de características gerado possui 128 valores.

Auto-correlograma de cor - ACC

O descritor auto-correlograma de cor ou *ACC* [45] tenta mapear a informação espacial das cores de uma imagem utilizando o conceito de auto-correlograma. Um auto-correlograma indica a probabilidade de se encontrar na imagem dois pixels de cor C a uma distância d um do outro. Seu algoritmo de extração realiza inicialmente uma quantização do espaço de cor e então a imagem é percorrida a fim de se calcular o auto-correlograma para cada distância escolhida. Ao final têm-se m valores de probabilidade para cada distância d considerada, onde m é igual à quantidade de cores no espaço quantizado. Portanto, o vetor de características do ACC possui $m*d$ valores. Na implementação realizada, o espaço de cor RGB foi quantizado em 64 bins e foram considerados quatro valores de distância (1, 3, 5 e 7), assim como em [45].

Em [45] é usada a função de distância L1 e uma variação dela, chamada de d_1. A função d_1 diferencia-se da função L1 apenas por uma normalização de cada subtração pela soma dos valores correspondentes. Na versão implementada, foi usada a função L1.

Apesar do vetor de características gerado pelo ACC ser pouco compacto, ele apresenta boa eficácia em seus experimentos [45] e por isso foi escolhido para implementação.

Joint auto-correlogram - JAC

O descritor JAC ou joint auto-correlogram [112] segue o mesmo princípio do descritor ACC, porém ele calcula o auto-correlograma conjunto para mais de uma propriedade da imagem. As propriedades por ele consideradas são: cor, magnitude do gradiente, *rank* e *texturedness*. A cor é considerada no espaço de cor RGB quantizado. As últimas três propriedades são calculadas a partir da imagem em níveis de cinza. A magnitude do gradiente é calculada como o máximo entre a diferença de brilho do pixel atual e seu vizinho da direita e a diferença de brilho do pixel atual e seu vizinho de baixo. O *rank* indica a variação de intensidade na vizinhança de um pixel e é representado pela quantidade de pixels da vizinhança que tem brilho maior do que o brilho do pixel atual. A *texturedness* é calculada contando-se a quantidade de pixels na vizinhança com diferença de brilho maior do que um certo limiar em relação ao pixel atual. Um auto correlograma conjunto indica, para cada distância, a probabilidade de ocorrer simultaneamente as quatro propriedades consideradas.

Em [112] o espaço RGB foi quantizado em 64 bins, os valores de magnitude do gradiente, *rank* e *texturedness* foram quantizados em 5 bins cada e foi usada uma janela de 5x5 pixels para a vizinhança durante os cálculos de *rank* e *texturedness*. Na implementação realizada, as mesmas configurações foram utilizadas exceto pelo espaço de cor, pois foi usado o espaço HSV. Além disso, foram usados quatro valores de distância no cálculo do auto correlograma (1, 3, 5 e 7).

O vetor de características gerado é pouco compacto, pois deverá conter todas as possíveis combinações entre os valores das quatro propriedades. Desta forma seriam $QC * QG * QR * QT * QD$ valores, sendo QC a quantidade de cores no espaço de cor quantizado, QG, QR e QT a quantidade de valores possíveis no espaço quantizado de magnitude do gradiente, *rank* e *texturedness*, respectivamente; e QD é a quantidade de distâncias usada. Para os valores de quantização implementados, tem-se um vetor de características com 64*5*5*5*4 valores, ou 32000 valores.

A função de distância utilizada é L1.

O JAC foi escolhido pois, apesar de gerar vetores de características grandes, ele combina informação de mais de uma propriedade da imagem, além de usar uma função de distância simples.

Border/Interior Pixel Classification - BIC

O descritor BIC ou *border/interior pixel classification* [29] classifica os pixels da imagem em pixels de borda ou pixels de interior. Seu algoritmo de extração inicialmente quantiza a imagem em 64 cores. Os pixels são então classificados pela seguinte regra: se um pixel possui a mesma cor dos seus vizinhos-4 (acima, à direita, abaixo e à esquerda) ele

é classificado como *interior*; caso contrário ele é classificado como *borda*. Ao final, são calculados dois histogramas de cor, sendo um para os pixels de borda e outro para os pixels de interior. Os dois histogramas são concatenados e armazenados como um único histograma de 128 bins.

A comparação dos histogramas é feita utilizando-se a distância *dLog*, como em [29].

A escolha do descritor BIC foi devido à sua alta eficácia em uma base de imagens heterogêneas [29], além disso, seu algoritmo de extração e sua função de distância são simples.

Color-based clustering - CBC

Em [27] é proposto um método de extração de características baseado em segmentação chamado *color-based clustering* ou CBC. Este método decompõe a imagem em componentes conexas disjuntas. Cada região possui um tamanho mínimo e uma diferença máxima de cor. Uma região é definida pela sua cor média no espaço de cor CIE Lab, pelo seu centro horizontal e vertical normalizados (h, v) e pelo seu tamanho em pixels normalizado em relação ao tamanho da imagem. Dessa forma, o vetor de características gerado possui 6 valores para cada região encontrada na imagem.

A função de distância usada em [27] considera a distância Euclidiana entre regiões de duas imagens e a distância final é calculada pela *Integrated Region Matching (IRM)*. Apesar de a extração de características e a função de distância serem caras computacionalmente, o CBC foi escolhido por seguir a abordagem baseada em segmentação.

Color Bitmap

O descritor Color Bitmap [69] é bastante simples e analisa a informação de cor da imagem globalmente e por meio de regiões fixas. Seu algoritmo de extração calcula a média e o desvio padrão dos valores da imagem para cada canal do espaço R, G e B separadamente. A seguir, a imagem é dividida em blocos sem sobreposição e a média dos valores de cada bloco também é calculada para cada canal separadamente. As três médias de cada bloco são comparadas com as três médias da imagem. Se a média do bloco for maior que a média da imagem, a posição do vetor de características referente ao bloco recebe 1, caso contrário, recebe 0. Este processo é feito para cada canal de cada bloco, portanto, o vetor de características é formado por três valores binários para cada bloco. Além disso, o vetor também armazena a média e o desvio padrão de imagem para os três canais, sendo então mais 6 valores. No final, tem-se um vetor com $3 * m$ bits mais 6 valores, sendo m a quantidade de blocos. Na versão implementada foram usados 100 blocos.

A função de distância calcula a distância em duas etapas. Na primeira etapa calcula-se a distância Euclidiana (L2) entre os valores globais (média e desvio padrão da imagem).

Na segunda etapa, usa-se a distância de Hamming para o cálculo da distância entre as características de cada bloco. Na primeira etapa existe uma normalização das médias e desvios globais pela média e desvio padrão de toda a base de imagens. Este tipo de normalização torna-se inviável em uma base de imagens da Web, na qual a quantidade de imagens é constantemente atualizada. Por isso, na versão implementada a normalização não foi realizada.

O descritor Color Bitmap foi implementado devido ao seu algoritmo de extração simples, sua função de distância rápida e por considerar as informações de cor da imagem de maneira global e por meio de regiões fixas.

Color Structure - CSD

O descritor *color structure* ou *CSD* [72] foi selecionado como um dos descritores de cor do padrão MPEG-7. Seu algoritmo de extração de características usa um elemento estruturante de 8x8 pixels para percorrer a imagem, sendo que esta deve estar no espaço de cor HMMD (hue, max, min, diff) [72]. À medida que o elemento estruturante percorre a imagem, um histograma é atualizado. Cada bin $h(m)$ do histograma é incrementado sempre que a cor m estiver dentro da janela, onde m varia de 0 a $M - 1$ e M indica a quantização do espaço de cor. O elemento estruturante pode ter amostragens diferentes dependendo do tamanho da imagem. Este tamanho de amostragem também indica qual o passo do deslocamento em pixels do elemento estruturante durante a varredura da imagem. Ao final, cada bin $h(m)$ do histograma representa a quantidade de locais em que a janela esteve na imagem nos quais havia um pixel de cor m dentro dela. Após a imagem ser totalmente percorrida o histograma é normalizado pela quantidade de locais em que o elemento estruturante fica durante o percurso. Os valores de cada bin são então quantizados não linearmente em 8 bits por bin. Na versão implementada o espaço de cor foi quantizado em 184 bins, pois os melhores resultados reportados em [72] usam esta quantização.

Segundo [72], a função de distância pode ser qualquer uma das funções L. Na versão implementada foi usada a função L1.

De acordo com [8], o descritor CSD tem melhor eficácia dentre os descritores do MPEG-7. Pelos experimentos de [72], o CSD também obtém valores melhores para a medida de eficácia ANMRR [72]. Segundo um outro estudo comparativo com os descritores de cor do MPEG-7, o descritor CSD é o que apresenta melhor eficácia [81].

ColorWavelet HSV - CW-HSV

O descritor CW-HSV [107] considera as informações de cor da imagem no domínio wavelet. Seu algoritmo de extração de características trabalha sobre uma imagem no espaço de

cor HSV e calcula um histograma de cor de 64 bins. Os coeficientes da transformada de Haar do histograma são calculados hierarquicamente pela função wavelet de Haar. Isto é feito recursivamente da seguinte maneira: o histograma de 64 bins é dividido ao meio e é calculada a soma dos valores da primeira e da segunda metade do histograma separadamente. Se a soma da primeira metade for maior que a soma da segunda metade, o primeiro bit do vetor de características é 1, senão é 0. No segundo nível, a primeira e a segunda metade são divididas ao meio. Se a soma do primeiro quarto do histograma for maior que a soma do segundo quarto, o segundo bit do vetor é 1, senão é 0. Se a soma do terceiro quarto do histograma for maior que a soma do quarto quarto, o terceiro bit do vetor é 1, senão é 0. Este processo se repete recursivamente até o último nível possível. No final tem-se um vetor de características com 63 valores binários.

Para o cálculo de distância é usada a distância de Hamming.

O principal motivo para a escolha do CW-HSV é seu vetor de características compacto e sua função de distância rápida.

ColorWavelet CIE Luv - CW-LUV

O descritor CW-LUV [77] é similar ao descritor CW-HSV. A principal diferença entre eles é o espaço de cor utilizado. O CW-LUV trabalha sobre o espaço de cor CIE Luv e calcula um histograma de 128 bins. Dessa maneira, o cálculo hierárquico dos coeficientes de Haar permite um nível a mais de recursão. Isto faz com que seja gerado um vetor de características com 127 bits.

A função de distância, assim como as razões para a escolha do CW-LUV, são as mesmas do descritor CW-HSV.

Chromaticity Moments - CM

O descritor *chromaticity moments* ou CM [83] caracteriza a imagem por valores de cromaticidade. Seu algoritmo de extração converte primeiramente a imagem para o espaço de cor CIE XYZ. Os valores de cromaticidade (x, y) são calculados como $x = \frac{X}{X+Y+Z}$ e $y = \frac{Y}{X+Y+Z}$. A partir destes valores é gerado o *traço* das cromaticidades, que indica a presença ou ausência de um valor (x, y) na imagem. Como é possível existir mais de um valor (x, y) na imagem, também é calculado um histograma de cromaticidades. O traço e o histograma de cromaticidades são usados para definir os momentos de cromaticidade. Na versão implementada, assim como em [83], 6 momentos foram calculados. Isso resulta em um vetor de características com 12 valores.

A função de distância calcula a soma dos módulos das diferenças entre os momentos correspondentes.

O descritor CM foi escolhido devido à sua função de distância rápida e seu vetor de características compacto.

3.2 Descritores de textura

Textura é uma propriedade muito importante para a caracterização e reconhecimento de imagens, o que resulta em uma grande quantidade de pesquisa envolvendo a análise de textura de imagens [71,105,115,120]. Apesar disso, não existe uma definição formal do que é textura, dadas as variadas definições existentes na literatura. É possível que se reconheça facilmente uma textura em uma imagem, como areia, folhagem, nuvens, pedras, etc. No entanto, é difícil defini-la em palavras. Em [105] é exibido um conjunto de definições para textura encontradas na literatura, entre elas: "sua estrutura é simplesmente atribuída a padrões repetitivos nos quais elementos ou primitivas são arranjados de acordo com uma regra de posicionamento", "a textura de uma imagem é descrita pelo número e tipos de suas primitivas e pela organização espacial ou layout de suas primitivas". Segundo [30], a textura pode ser descrita por propriedades espaciais, propriedades de freqüência ou propriedades perceptuais. De uma maneira geral, a textura pode ser entendida como um conjunto de variações de intensidades que formam certos padrões de repetição.

Diferentemente de cor, a textura é uma propriedade difícil de se analisar considerando apenas o valor de um único pixel. Ela ocorre principalmente devido à variação de brilho entre pixels vizinhos. Dessa maneira, é possível nomear atributos para texturas. Segundo [31], uma textura tem atributos de rugosidade, contraste, direcionalidade, regularidade, aspereza e semelhança com linhas, sendo os três primeiros os mais importantes.

A grande maioria dos descritores de textura da literatura trabalha sobre imagens em níveis de cinza. Entretanto, são poucos os que explicam qual a melhor maneira para se converter uma imagem colorida em uma imagem em níveis de cinza de forma que o descritor tenha melhor desempenho.

A seguir é apresentada uma taxonomia para descritores de textura (Seção 3.2.1), são apresentadas as tabelas comparativas entre os descritores de textura estudados (Seção 3.2.2) e são detalhados os descritores de textura implementados (Seção 3.2.3).

3.2.1 Taxonomia para descritores de textura

Existem diversas abordagens para a extração de textura de uma imagem e elas variam em alguns trabalhos encontrados na literatura [30, 31, 105]. A taxonomia apresentada na Figura 3.2 é a encontrada em [105].

Uma das maneiras mais tradicionais para se analisar a distribuição espacial dos níveis de cinza de uma imagem é a análise estatística. Dessa maneira, é possível extrair in-

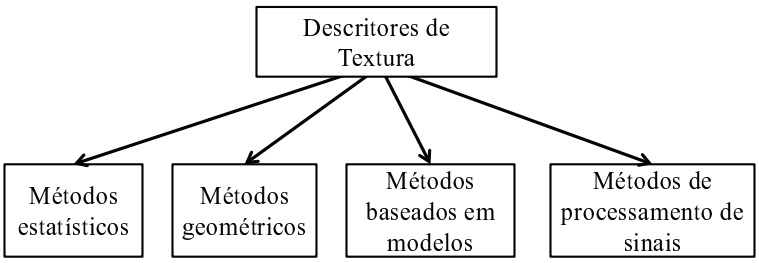


Figura 3.2: Taxonomia para descritores de textura.

formações estatísticas sobre os níveis de cinza de uma imagem, como por exemplo, a probabilidade de co-ocorrência de níveis de cinza em diferentes distâncias e orientações. Podem ser usadas, por exemplo, estatísticas de primeira ordem, que usam apenas os valores individuais dos pixels, e estatísticas de segunda ordem, que consideram propriedades de pares de pixels [105]. Métodos que utilizam histogramas também caracterizam estatisticamente propriedades de textura de uma imagem. Um dos métodos estatísticos mais populares da literatura é a matriz de co-ocorrência [41].

Métodos geométricos analisam a textura por meio de "elementos de textura" ou primitivas. Esta análise é feita considerando-se as propriedades geométricas destas primitivas, como tamanho, forma, área e comprimento. Uma vez que as primitivas estejam identificadas em uma imagem, são extraídas delas regras de posicionamento que descrevem a textura, como uma grade ou estatísticas do vetor que une os centroids das primitivas [79]. Este tipo de análise se torna mais difícil em texturas naturais pois tanto as primitivas quanto os padrões de posicionamento podem ser bastante irregulares [79]. Por exemplo, descrever um muro de tijolos por meio do tamanho e forma de um elemento *tijolo* e uma grade como regra de posicionamento pode ser simples. No entanto, descrever nuvens no céu pode ser complicado, pois um elemento *nuvem* pode ter formas e tamanhos variados e não seguir uma regra bem definida de posicionamento.

Métodos baseados em modelos constroem um modelo que pode ser usado não somente para descrever a textura, mas também para sintetizá-la. Os parâmetros do modelo capturam as qualidades perceptivas essenciais de uma textura. Por exemplo, podem ser modelados elementos de textura como: um ponto claro ou um ponto escuro, uma transição horizontal ou vertical, cantos ou retas, etc. Descritores que seguem esta abordagem são bons para texturas bem definidas. O descritor LBP [82] é um exemplo de descritor que segue esta abordagem.

Métodos de processamento de sinais realizam a caracterização de texturas por meio

de filtragens de uma imagem. Podem ser usadas tanto filtragens no domínio espacial quanto filtragens no domínio da freqüência. Propriedades são extraídas das imagens filtradas, como a densidade de bordas no caso de uma filtragem espacial, ou a média e desvio padrão dos valores no caso de uma filtragem no domínio da freqüência. Descritores baseados em wavelets e em filtros de Gabor se encaixam nesta abordagem, como por exemplo o descritor HTD [71, 114].

Dentre os descritores estudados, encontram-se descritores baseados em wavelets [55, 109], descritores baseados em filtros de Gabor [40, 71, 86, 87, 114, 117, 127], descritores que usam matrizes de co-ocorrência [41, 57] e conceitos similares [106], descritores baseados em *local binary patterns* [82, 102], entre outros.

3.2.2 Tabelas comparativas

Baseando-se nos critérios apresentados na Seção 2.4, foram montadas duas tabelas em que são comparados os descritores de textura estudados. Na Tabela 3.3 são apresentadas as complexidades dos algoritmos de extração e das funções de distância dos descritores de textura estudados, assim como os tamanhos dos vetores de características gerados. A Tabela 3.3 também apresenta a classe taxonômica a que pertence cada descritor, segundo a taxonomia da Figura 3.2.

Pelas complexidades encontradas, observa-se que vários descritores de textura possuem algoritmos de extração com complexidade alta, como os descritores HanGabor [40], HTDI [127], TBDI [61], SASI [130,131], SOH [128], MERF [117], HTD [71,114], HTDR [86], TBD [72, 114], SBP [119] e SID [121]. O uso de algoritmos de extração de características lentos pode aumentar o tempo de resposta a uma consulta, como explicado na Seção 2.4.

Descritores como Takala LBP [102], SOH [128] e Rubner EMD [87] possuem funções de distância mais complexas do que a maioria dos outros descritores, que possui função linear no tamanho do vetor de características.

Os vetores de características gerados por TBDI [61], TBD [72, 114], EHD [72] e LBP [82] são bastante compactos e podem ajudar a reduzir o impacto dos requisitos de armazenamento em um sistema CBIR.

Na Tabela 3.4 são exibidas as informações do ambiente de validação dos descritores estudados.

Pelos dados da Tabela 3.4 observa-se que a grande maioria de descritores de textura da literatura é validado na base de texturas Brodatz. Isso indica uma certa padronização do processo de validação para descritores de textura. São poucos os descritores validados em bases de imagens heterogêneas, entre eles estão TBDI [61], Takala LBP [102], QCCH [44], Rubner EMD [87] e CCOM [57].

Em relação às comparações realizadas, muitos descritores são validados com com-

parações entre descritores baseados em filtros de Gabor, como 2DWAH [53], Newsam SIFT [80], HanGabor [40], HTDI [127], TBDI [61], QCCH [44], BSpline [109], KokareCW [55], SimZernike [97], SASI [130, 131], MERF [117] e Sim DCT [96]. Isto mostra que os filtros de Gabor são atualmente um dos modelos mais bem sucedidos para a descrição de textura.

Na Tabela 3.4 ainda encontram-se descritores que não foram comparados com nenhum outro descritor ou foram comparados apenas com variações deles próprios, como é o caso dos descritores LAS [103], Rubner EMD [87], CCOM [57], TBD [72, 114], EHD [72] e GCOM [41].

Na literatura são encontrados alguns estudos comparativos entre descritores de textura. Em [4] são comparados um descritor baseado na transformada de Fourier (DFT) e outro baseado em filtros de Gabor. A comparação foi feita na tentativa de identificar qual deles era mais robusto em relação a ruído em imagens de textura. Os resultados mostraram que o descritor baseado em Fourier é melhor em imagens sem ruído e o descritor baseado em filtros de Gabor é melhor na presença de ruído.

Em [115] são comparados os três descritores de textura do MPEG-7. Em relação à extração do vetor de características, o estudo mostra que o descritor TBD [72, 114] é o mais caro computacionalmente e o descritor EHD [72] é o que tem menor custo computacional. Além disso, é concluído que o descritor HTD [71,114] captura informação global, o TBD captura informação global e local e o EHD captura apenas informação local. O estudo ainda aponta que o descritor TBD é pouco recomendado para CBIR, sendo melhor aproveitado em aplicações de navegação em bases de imagens (*browsing*) e em detecção de defeitos. Além disso, o estudo indica que o descritor HTD pode ser usado em CBIR e para segmentação de texturas e que o descritor EHD é recomendado para CBIR. O estudo mostra também que o HTD e o TBD são menos sensíveis a ruído, enquanto o EHD não é recomendado em casos com ruído. Nos experimentos realizados, o TBD apresentou baixa eficácia e o HTD mostrou-se pouco eficaz para imagens com rotação. Neste último caso, são encontradas na literatura várias versões aprimoradas do descritor HTD objetivando a invariância à rotação, como os descritores HTDR [86], HTDI [127] e HanGabor [40], presentes neste estudo.

3.2.3 Detalhes dos descritores

Para os experimentos práticos foram escolhidos os seguintes descritores de textura: LBP [82], HTD [71, 114], SASI [130, 131], CCOM [57], Unser [106], QCCH [44], LAS [103].

Abaixo são explicados cada um deles e são dadas as razões de cada escolha.

Local Binary Pattern - LBP

O descritor *local binary pattern* ou LBP [82] é um descritor relativamente recente e que ganhou popularidade rapidamente. Como pontos fortes, ele possui algoritmos simples, bons resultados de eficácia e é invariante à rotação e variações na escala de cinza.

Seu algoritmo de extração de características extrai informações de variação dos brilhos entre pixels vizinhos. Para isso, ele define uma janela com raio R e uma quantidade de vizinhos P e percorre a imagem calculando a quantidade de variações positivas ou negativas entre os brilhos dos pixels vizinhos em relação ao pixel central da janela. Para garantir invariância a variações na escala de cinza, apenas o sinal da variação de brilho calculada é considerado, sendo simplificado para 1 quando a variação é positiva e para 0 quando a variação é negativa. Assim que esses valores são calculados, é feita uma contagem de quantas transições entre 0/1 e 1/0 existem na vizinhança. A contagem garante a invariância à rotação. Se a quantidade de transições for menor ou igual a 2, o valor de LBP para aquela posição da janela é igual à quantidade de sinais 1 na vizinhança. Caso contrário, o valor de LBP é $P + 1$. Dessa forma, ao final do percurso da janela pela imagem, cada posição pela qual a janela passou possui um valor LBP entre 0 e $P + 1$. É calculado então um histograma dos valores LBP contendo $P+2$ bins. Nos testes realizados em [82] foram comparadas implementações com valores de R iguais a 1, 2 e 3 e valores de P iguais a 8, 16 e 24. Na implementação realizada, foi usado P=8 e R=1.

A função de distância do LBP é uma simplificação da estatística conhecida como *razão log-likelihood* e é uma função não métrica, pois não é simétrica. Na implementação realizada, foi implementada a função de distância L1.

Homogeneous Texture Descriptor - HTD

O descritor HTD [71, 114] é um dos descritores de textura do padrão MPEG-7 [72]. Seu algoritmo de extração do vetor de características aplica na imagem um conjunto de filtros sensíveis a escala e orientação. Depois desta etapa, são calculados a média e o desvio padrão da saída de cada filtro no domínio da freqüência. No processo de filtragem são usados filtros de Gabor. Em [114] são consideradas 4 escalas e 6 orientações diferentes, assim como na implementação realizada. Dessa maneira, o vetor de características possui 48 valores.

A função de distância do HTD calcula a diferença entre cada componente de média e desvio padrão dos vetores de características e aplica uma normalização, em cada componente, pela média e pelo desvio padrão geral da base de imagens. Este tipo de normalização torna-se inviável em uma base de imagens da Web, na qual a quantidade de imagens é constantemente atualizada. Por isso, na versão implementada, a normalização não foi realizada.

Existem diversas variações do HTD propostas na literatura (HanGabor [40], HTDI [127], HTDR [86]). O descritor HanGabor [40] propõe uma maneira de garantir invariância à rotação ou invariância à escala. A invariância à rotação é obtida somando-se as saídas dos K filtros em cada escala, sendo K a quantidade de orientações. A invariância à escala é obtida somando-se as saídas dos S filtros em cada orientação, sendo S a quantidade de escalas. HTDI [127] propõe uma versão invariante à rotação para o HTD. A invariância é alcançada calculando-se a soma das médias da saída de cada filtro em cada uma das orientações consideradas. A orientação com maior soma é a direção dominante e é usada para realizar "shifts" circulares no vetor de características. Já o descritor HTDR [86] propõe a obtenção de invariância à escala e à rotação alterando a função de distância do HTD original. A função de distância proposta realiza 12 "shifts" no vetor de características da imagem de consulta de maneira a encontrar a menor distância entre as variações.

Steerable Pyramid Decomposition invariante à escala e rotação - SID

O algoritmo de extração de características do descritor SID [121] é semelhante ao do descritor HTD. A imagem é processada por um conjunto de filtros sensíveis a escala e orientação. No descritor SID a imagem de entrada é inicialmente decomposta em duas sub-bandas usando um filtro passa-altas e um filtro passa-baixas. A seguir, a banda resultante do filtro passa-baixas é decomposta recursivamente em K sub-bandas por filtros passa-bandas e em uma sub-banda por um filtro passa-baixas. Cada etapa da recursão captura diferentes informações direcionais numa determinada escala. A média e o desvio padrão de cada sub-banda resultante são armazenados no vetor de características.

A invariância à escala e rotação é obtida separadamente. No caso da invariância à rotação, algoritmo encontra a orientação dominante somando as energias das escalas para cada orientação. A máxima energia indica a orientação dominante. O vetor de características é então deslocado circularmente de maneira a deixar a orientação dominante na primeira posição dentro de cada escala. Para a invariância à escala, o processo é análogo. O descritor SID calcula as invariâncias independentemente. Na versão implementada, porém, a invariância à escala e à rotação é calculada concomitantemente. Ou seja, o vetor gerado na versão implementada é invariante à escala e à rotação ao mesmo tempo. A versão implementada usou 2 escalas e 4 orientações, como sugerido em [121]. Isto gerou vetores de características com 16 valores.

A função de distância soma as diferenças entre as médias e desvios padrão correspondentes, aplicando uma normalização por valores relativos à base de imagens usada. Em uma base de imagens da Web, na qual a quantidade de imagens é constantemente atualizada, este tipo de normalização é inviável. Na versão implementada, nenhuma normalização foi realizada.

O descritor SBP [119] também apresentado nas tabelas comparativas do estudo teórico

desta dissertação é uma variação do descritor SID. O SBP aplica LBP nas sub-bandas resultantes do processo de decomposição de maneira a caracterizar informações locais das texturas.

O descritor SID foi escolhido por apresentar bons resultados em seus experimentos [121] e por apresentar invariância à escala e rotação.

Statistical Analysis of Structural Information - SASI

O descritor SASI (Statistical Analysis of Structural Information) [130, 131] codifica as propriedades de textura de uma imagem baseado em propriedades estruturais das texturas. Seu algoritmo de extração de características percorre a imagem usando janelas com múltiplas resoluções e orientações. Segundo [130], apesar de trabalhar no domínio espacial, o descritor SASI consegue medir informações espectrais de uma dada textura. O primeiro passo do algoritmo é escolher os tamanhos das janelas e as orientações a serem usadas. Cada janela pode ser percorrida de diferentes maneiras. Estas maneiras são determinadas por vetores chamados de *lag vectors*. A quantidade de lag vectors por janela depende da largura da janela e é igual a K. O valor de K é obtido pela fórmula $K = \lceil (S/4) \rceil + 1$, na qual S é a largura da janela em pixels [130]. Para cada janela, o algoritmo percorre a imagem calculando um valor de auto-correlação considerando diferentes direções. Ao final, tem-se uma imagem de valores de auto-correlação para cada configuração de cada janela. A média e o desvio padrão de cada imagem são calculados e armazenados no vetor de características. Os valores de média e desvio padrão do vetor de características gerado são normalizados pela média e desvio padrão globais da base de imagens usada. A normalização por valores da base de imagens é inviável em uma base de imagens da Web, na qual a quantidade de imagens é constantemente atualizada. Na versão implementada, a normalização foi feita pela média e pelo desvio padrão dos valores do vetor de características.

A implementação realizada utilizou janelas de tamanhos 3x3, 5x5 e 7x7 pixels e foram usadas 4 direções (0°, 45°, 90° e 135°). Pelos tamanhos das janelas, o somatório da fórmula apresentada na Tabela 3.3 tem valor 8 (2 [3x3] + 3 [5x5] + 3 [7x7]). Portanto, a quantidade de valores no vetor de características gerado é igual a 64.

A função de distância usada pelo SASI [130, 131] calcula um valor de similaridade, ao invés de um valor de distância, por meio de produtos internos entre os valores dos vetores de duas imagens. Para efeito de padronização, a implementação realizada converteu o valor de similaridade em um valor de distância.

O descritor SASI foi escolhido por ter alcançado melhor eficácia do que descritores baseados em filtros de Gabor, como reportado em [131].

Color co-occurrence matrix - CCOM

A matriz de co-ocorrência de níveis de cinza (GCOM) [41] foi uma das primeiras técnicas para se caracterizar as propriedades de textura de uma imagem. Sua idéia básica é codificar as variações entre os brilhos dos pixels em várias resoluções e direções diferentes em matrizes. A partir destas matrizes é possível obter informações como contraste, correlação, variância, entropia, etc.

A matriz de co-ocorrência de cor (CCOM) [57] apresenta uma abordagem semelhante à GCOM [41], tendo como vantagem a consideração da cor de uma imagem. O algoritmo de extração do vetor de características do descritor CCOM realiza uma quantização no espaço de cor da imagem e em seguida varre a imagem a fim de calcular a matriz de co-ocorrência $W(c_i, c_j, d)$. Para cada par de pixels i, j da imagem, com distância d entre si, incrementa $W(c_i, c_j, d)$ em uma unidade, onde c_i é a cor do pixel i no espaço quantizado, c_j é a cor do pixel j no espaço quantizado e d é a distância entre eles. Ao final, W é normalizada pelo tamanho da imagem. O vetor de características considera apenas valores de W que estejam no intervalo aberto entre dois limiares A_{min} e A_{max}, sendo A_{min} maior ou igual a zero. Dessa forma, o vetor de características não contém valores nulos da matriz, reduzindo o espaço de armazenamento necessário.

A função de distância do descritor CCOM soma as diferenças entre os valores de W correspondentes entre os vetores de características da imagem de consulta e da imagem da base. A distância é normalizada pela soma do valores dos dois vetores. A implementação realizada quantizou o espaço RGB em 216 bins (6 bins por canal) e usou d igual a 1, assim como em [57].

Devido à popularidade da matriz de co-ocorrência, o descritor CCOM [57] foi escolhido para implementação, pois agrega a informação de cor à matriz de co-ocorrência originalmente proposta [41]. Em [57] ainda são reportadas outras vantagens do descritor CCOM, como ser invariante a rotação, translação e reflexão, conseguir recuperar imagens a partir de subpartes de texturas e conseguir lidar bem com a heterogeneidade das imagens da base.

Unser

O descritor Unser [106] foi proposto com o objetivo de reduzir a complexidade da matriz de co-ocorrência (GCOM) mantendo bons resultados de eficácia. Sua abordagem é bastante semelhante à do descritor GCOM para a extração do vetor de características. A idéia principal é calcular um histograma de somas H_{soma} e um histograma de diferenças H_{dif} entre os pixels ao invés de calcular a matriz de co-ocorrência. O algoritmo de extração define inicialmente os ângulos a serem considerados entre os pixels vizinhos. A imagem é então percorrida e, para cada ângulo a definido, incrementam-se os dois histogramas da

seguinte maneira:

$$\begin{aligned} H_{soma}[a][f(p)+f(q)] &\leftarrow 1 \\ H_{dif}[a][f(p)-f(q)] &\leftarrow 1 \end{aligned}$$

onde $f(x)$ é valor de brilho do pixel x na imagem, p é o pixel atual e q é o pixel vizinho à distância de d pixels de p e a um ângulo a em relação a p. A distância d é fixa em um único valor. Isto gera dois histogramas de $QA * QG * 2$ valores cada, sendo QA a quantidade de ângulos considerada e QG a quantidade de níveis de cinza. Os histogramas são normalizados pela quantidade de pixels na imagem. A partir deles é possível extrair informações semelhantes às extraídas de uma matriz de co-ocorrência, como energia, contraste, etc. Para extrair estas informações deve-se, no pior caso, percorrer ambos os histogramas completamente, sendo portanto de complexidade $O(QG)$.

Em [106] não é apresentada uma função de distância específica para o descritor Unser. Na implementação realizada, foi usada a função de distância L1. Além disso, a implementação realizada usou 256 níveis de cinza, 4 valores de ângulos (0°, 45°, 90° e 135°), distância $d = 1,5$ e extraiu as seguintes informações a partir dos histogramas: média, contraste, correlação, energia, entropia, homogeneidade, probabilidade máxima e desvio padrão. Isto gerou vetores com 32 valores.

Pelos resultados experimentais reportados em [106], o descritor Unser atinge seus objetivos, pois gera vetores mais compactos, tem menos complexidade para a extração do vetor e apresenta eficácia muito semelhante à do descritor GCOM. Devido a esses motivos, o Unser foi escolhido para os experimentos práticos.

Quantized Compound Change Histogram - QCCH

O descritor Quantized Compound Change Histogram (QCCH) [44] caracteriza a informação de textura de uma imagem baseado na relação de um pixel e seus vizinhos. Sua abordagem gera uma representação invariante a rotação e translação.

O algoritmo de extração do vetor de caraterísticas define uma vizinhança quadrada $N_r(i,j)$ de raio r centralizada no pixel (i,j). O valor de cada pixel da imagem colorida RGB é convertido para cinza pela fórmula $y = 0,299*R+0,587*G+0,114*B$. A imagem é então percorrida e o valor de cinza médio $y_r(i,j)$ é calculado para cada posição da janela $N_r(i,j)$. A partir disso são calculadas as taxas de variação de cinza em 4 direções:

$$\begin{aligned} H_r^y(i,j) &= |y_r(i-r,j) - y_r(i+r,j)| \\ V_r^y(i,j) &= |y_r(i,j-r) - y_r(i,j+r)| \\ D_r^y(i,j) &= |y_r(i-r,j-r) - y_r(i+r,j+r)| \\ A_r^y(i,j) &= |y_r(i+r,j-r) - y_r(i-r,j+r)| \end{aligned}$$

É calculado então a média destas 4 taxas de variação:

$$v(i,j) \quad = (H_r^y(i,j) + V_r^y(i,j) + D_r^y(i,j) + A_r^y(i,j))/4.$$

Os valores de $v(i,j)$ são quantizados não uniformemente em T valores inteiros. O vetor de características é gerado contendo o histograma dos valores de $v(i,j)$ quantizados. Na implementação realizada $T = 40$, portanto os vetores possuem 40 valores.

A função de distância do descritor QCCH é a função L1.

O descritor QCCH foi escolhido para implementação devido a seu algoritmo de extração de características relativamente simples, sua função de distância rápida e seu vetor de características compacto. A implementação realizada seguiu as configurações reportadas em [44].

Local Activity Spectrum - LAS

O descritor *local activity spectrum* (LAS) proposto em [103] captura a atividade espacial de uma textura nas direções horizontal, vertical, diagonal e anti-diagonal separadamente. O algoritmo de extração de características é bastante simples e extrai as informações de textura usando 4 medidas de atividade espacial para cada pixel de uma imagem. A imagem é percorrida pixel a pixel e as medidas são calculadas, para o pixel (i,j), pelas fórmulas abaixo:

$$\begin{aligned}
g_1 &= |f(i-1,j-1) - f(i,j)| + |f(i,j) - f(i+1,j+1)| \\
g_2 &= |f(i-1,j) - f(i,j)| + |f(i,j) - f(i+1,j)| \\
g_3 &= |f(i-1,j+1) - f(i,j)| + |f(i,j) - f(i+1,j-1)| \\
g_4 &= |f(i,j+1) - f(i,j)| + |f(i,j) - f(i,j-1)|
\end{aligned}$$

Estes valores são usados para o cálculo de um histograma, que é chamado de *local activity spectrum*. Cada componente g_i é quantizada independentemente. Em [103], as componentes foram quantizadas em 4 bins cada, gerando um histograma de 256 bins. Na implementação realizada foi usada a mesma quantização. No entanto, os valores de cada g_i sofreram uma quantização não uniforme, para evitar a geração de um histograma com muitos valores no bin zero.

A função de distância do descritor LAS é a função L1.

O principal motivo para a escolha do LAS é sua simplicidade, tanto para extração do vetor de características, quanto para o cálculo de distância.

3.3 Descritores de forma

A identificação, reconhecimento e recuperação de formas em imagens têm utilidade em diversas aplicações, como, por exemplo, o reconhecimento de caracteres alfanuméricos, a detecção e reconhecimento de pessoas em sistemas de segurança, o rastreamento de objetos em vídeos, etc.

A caracterização de formas em aplicações de propósito geral, como a Web, pode aumentar a qualidade dos resultados obtidos. Um dos principais motivos é que grande parte da informação semântica de uma imagem está associada aos objetos nela presentes. Por exemplo, se for solicitado a algumas pessoas que selecionem, dentro de um conjunto de imagens, quais são as imagens mais semelhantes a uma determinada imagem de consulta, tendo esta imagem de consulta uma semântica forte associada aos objetos nela presentes, haverá grande chance de as pessoas ignorarem outras características, como cor, e selecionarem imagens semelhantes devido à presença de objetos semelhantes [59]. Se existirem, por exemplo, duas imagens idênticas exceto pelo fato de uma ser colorida e outra estar em níveis de cinza, é enorme a chance destas imagens serem indicadas como altamente semelhantes. Um outro fator importante no uso de formas, refere-se ao processo de reconhecimento no cérebro humano. A imagem captada pelo olho humano tem suas informações de cor e forma processadas em locais diferentes do cérebro humano [12].

Durante o processo de reconhecimento de formas, a visão humana tolera diferentes distorções e fenômenos associados à forma. Dessa maneira, descritores de forma devem ser capazes de reconhecer imagens semelhantes mesmo que existam objetos rotacionados, transladados, escalonados, com ruído, distorcidos ou parcialmente encobertos [125].

A seguir é apresentada uma taxonomia para descritores de forma (Seção 3.3.1), são apresentadas as tabelas comparativas entre os descritores de forma estudados (Seção 3.3.2) e são detalhados os descritores de forma implementados (Seção 3.3.3).

3.3.1 Taxonomia para descritores de forma

A análise de formas de uma imagem pode ser feita de maneiras diferentes [67, 89, 125]. Na Figura 3.3 é exibida uma taxonomia que engloba a grande maioria dos descritores de forma existentes.

A taxonomia exibida na Figura 3.3, proposta em [125], classifica os descritores de forma basicamente em duas abordagens: baseada em contorno e baseada em região. Ambas podem analisar uma forma globalmente ou estruturalmente (em segmentos).

A abordagem baseada em contorno é mais popular na literatura. Entretanto, ela pode sofrer efeitos negativos na presença de ruído. Além disso, não é sempre que o contorno de uma forma está presente na imagem e, em algumas aplicações, o conteúdo da forma é

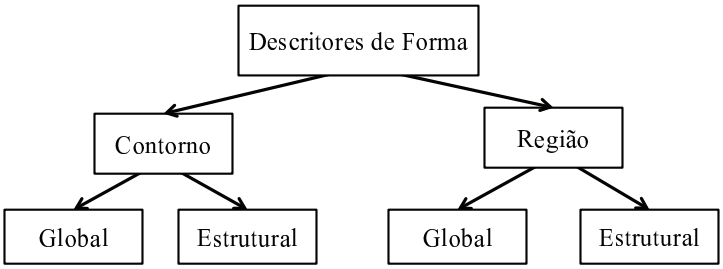


Figura 3.3: Taxonomia para descritores de forma.

mais importante do que o contorno [125]. Um exemplo de descritor baseado em contorno é o descritor *CSS* [76].

A abordagem baseada em região pode ser mais robusta na recuperação de formas por considerar a informação de toda a forma ao invés de considerar apenas a informação do contorno, podendo ser usada em aplicações de propósito geral, como a Web [125]. Um exemplo de descritor de forma baseado em região é o descritor *ZMD* [52].

Dentro destas abordagens, pode-se ainda classificar os descritores como sendo *globais* ou *estruturais*. Os descritores globais consideram o contorno ou a região da forma como um todo, sem realizar divisões em subpartes. Os descritores estruturais dividem o contorno ou a região em segmentos e possibilitam a comparação parcial de formas, conseguindo, portanto, suportar oclusão de formas. Entretanto, eles são mais complexos computacionalmente, especialmente para o cálculo de distância. Além disso, são mais sensíveis a ruído e necessitam de ajuste de parâmetros que dependem de informações das formas existentes na base de imagens, o que os torna impraticáveis em sistemas de propósito geral, como a Web.

Uma das principais questões envolvendo descritores de forma é relativa à segmentação das formas de uma imagem. A segmentação é ainda um problema em aberto. Em bases de imagens com conteúdo conhecido e controlado, é possível realizar ajustes de parâmetros de maneira a conseguir bons resultados de segmentação. No entanto, em bases de conteúdo heterogêneo, ajustes de parâmetros que satisfaçam todas as possíveis categorias de imagens são difíceis de realizar. Dessa maneira, torna-se impraticável o uso de descritores de forma que dependam de segmentação em um sistema de CBIR na Web.

Descritores de forma que dependem de segmentação necessitam que as imagens de entrada para seus algoritmos de extração do vetor de características sejam binárias, de maneira a saber claramente o que é *objeto* e o que é *fundo*. Portanto, nesta dissertação são considerados como dependentes de segmentação descritores de forma que partem do

pressuposto de que as imagens de entrada já vêm separadas em objeto e fundo.

No estudo comparativo desta dissertação, a grande maioria de descritores de forma encontrados é baseada em segmentação. A minoria que não depende de segmentação realiza um pré-processamento mais simples do que uma segmentação completa na imagem de entrada antes de extrair o vetor de características. Este pré-processamento tem como objetivo detectar bordas ou gradiente da imagem. Para isso, são normalmente usados detectores de borda como Canny e gradientes como Sobel [36]. Os descritores que não dependem de segmentação não se encaixam perfeitamente na taxonomia da Figura 3.3.

3.3.2 Tabelas comparativas

Baseando-se nos critérios apresentados na Seção 2.4, foram montadas tabelas para a comparação dos descritores de forma estudados. Na Tabela 3.5 são apresentadas as complexidades dos algoritmos de extração e das funções de distância dos descritores, assim como os tamanhos dos vetores de características gerados. A Tabela 3.5 também apresenta a classificação taxonômica de cada descritor, de acordo com a Figura 3.3.

Como mencionado na Seção 3.3.1, foram encontrados descritores de forma que dependem de segmentação para extrair o vetor de características de uma imagem. A complexidade de extração destes descritores está indicada na Tabela 3.5 como *SD*. Descritores que extraem o vetor de características sem depender de segmentação prévia têm com a complexidade analisada seguindo os padrões especificados na Seção 2.4.1.

Pela Tabela 3.5, observa-se que os descritores 2DWAH [54], Jang CO [47], MGFD [64], TSD [74], HBA [94], EOAC [70], Spytec [59] e OR [13] são independentes de segmentação para conseguir extrair o vetor de características. No entanto, alguns deles extraem o vetor de características em uma complexidade maior do que linear, como os descritores TSD [74] e OR [13], por exemplo. Foram encontrados também vários descritores de forma que se baseiam na transformada de Fourier para extrair o vetor de características, como os descritores Chen RPD [18], 2GFD [92], MGFD [64], GFDC [90], MSF [58], GFD [123], OR [13] e FD [122, 124].

As funções de distância dos descritores de forma independentes de segmentação são quase todas lineares no tamanho do vetor, exceto pelo descritor TSD [74] que possui uma função de distância quadrática no tamanho do vetor. Já entre os descritores dependentes de segmentação, existem alguns que possuem funções de distância mais caras, como os descritores CS [24], Wang CCD [110], TSDIZ [7], TSCS [7], Wang Hash [108] e CSS [76].

Em relação aos vetores de características, observa-se que os vetores gerados por alguns dos descritores dependentes de segmentação contêm uma quantidade de valores que depende dos objetos existentes na imagem, como os vetores dos descritores CS [24], Wang CCD [110], TSCS [7] e Wang Hash [108], por exemplo. Existe também uma grande

quantidade de descritores que gera vetores com tamanhos que dependem de quantizações, como os gerados pelos descritores Jang CO [47], TSD [74], Naskar RH [78], HBA [94] e EOAC [70], por exemplo.

Na Tabela 3.6 são exibidas as informações do ambiente de validação dos descritores de forma estudados.

Pelos dados exibidos na Tabela 3.6, observa-se que vários descritores são validados em bases de imagens do MPEG-7, como os descritores 2GFD [92], TAR [5], SHD [91], HBA [94], GFDC [90], GFD [123], MLEV [51], ZMD [52], TSDIZ [7] e TSCS [7], por exemplo.

Em relação às comparações realizadas na validação dos descritores, não é predominante, dentre os descritores estudados, o uso de um determinado descritor como base de comparação. Entretanto, nota-se que os descritores CSS [76] e *Moment Invariants* (MI) [43] são bastante usados nas comparações efetuadas.

Na literatura são encontrados dois estudos comparativos entre os descritores de forma CSS e FD. Em [124], o descritor FD apresentou melhor eficácia do que descritor CSS nos experimentos realizados em bases do MPEG-7. Além disso, FD foi mais rápido para extrair e calcular distâncias do que o descritor CSS. Segundo [122], o descritor FD captura informação global e local, enquanto que o descritor CSS captura apenas informação local. E ainda, o descritor FD é menos dependente de parâmetros.

3.3.3 Detalhes dos descritores

Para os experimentos práticos foram escolhidos os seguintes descritores de forma: EOAC [70], Spytec [59].

Abaixo são explicados cada um deles e são dadas as razões de cada escolha.

Edge Orientation Autocorrelogram - EOAC

O descritor *edge orientation autocorrelogram* (EOAC) [70] classifica as bordas de uma imagem baseado em dois fatores: orientação de bordas e correlação entre bordas vizinhas. Dessa forma, são incluídas informações de bordas e linhas contínuas das imagens descrevendo, de certa maneira, as propriedades de forma dos objetos de uma imagem. Este esquema tolera variações de iluminação, cor, escala e translação.

O algoritmo de extração do vetor de características consiste basicamente em 3 passos. O primeiro deles realiza o cálculo da imagem de gradiente de uma dada imagem de entrada. O gradiente pode ser calculado por um operador de gradiente como Sobel. A imagem de gradiente calculada contém um valor de magnitude e um valor de orientação para cada pixel. Neste passo, as magnitudes do gradiente que tiverem valor menor que um certo limiar são desconsideradas. O segundo passo define um esquema de quantização para

os ângulos e define as distâncias a serem consideradas no cálculo do auto-correlograma. O terceiro e último passo calcula um auto-correlograma de orientações de bordas. Este auto-correlograma possui QA*QD valores, sendo QA a quantidade de ângulos no espaço de ângulos quantizados e QD a quantidade de distâncias do auto-correlograma. O vetor de características gerado contém os valores do auto-correlograma.

A função de distância do descritor EOAC é a função L1.

A implementação realizada utilizou uma quantização de ângulos em 72 segmentos de 5° cada, 4 valores de distância (1, 3, 5, e 7), o operador de Sobel para calcular o gradiente e limiar de magnitude de gradiente igual a 25, como sugerido em [70]. Portanto, o vetor gerado possui 288 valores.

A principal razão para a escolha do EOAC é sua não dependência de segmentação para extrair o vetor de características.

Spytec

Em [59] é apresentada uma estrutura de indexação para busca por similaridade em bases de dados de alta dimensionalidade. Esta estrutura é chamada de *Spherical Pyramid-Technique* (SPY-TEC). O artigo [59] também propõe um método para extrair automaticamente características de forma de uma imagem utilizando detecção de bordas e transformada wavelet. Nesta dissertação, este método é referenciado pelo mesmo nome da estrutura de indexação proposta em [59]: *spytec*.

O algoritmo para a extração do vetor de características realizada pelo descritor Spytec parte de uma imagem colorida. O primeiro passo consiste em converter a imagem para níveis de cinza. O segundo passo extrai a informação de bordas da imagem cinza. Em seguida, aplica-se uma transformada wavelet discreta (DWT) na imagem de bordas. Os coeficientes wavelet calculados são normalizados para valores entre 0 e 1 e são guardados no vetor de características apenas os coeficientes de maior magnitude. Como a quantidade de coeficientes a ser armazenada no vetor é sempre muito menor que a quantidade de coeficientes existentes [59], não é necessário ordená-los para obter os maiores coeficientes. Dessa forma, o algoritmo de extração do Spytec possui complexidade $O(n)$.

A função de distância do descritor Spytec é a função L2.

Em relação aos passos do algoritmo de extração, o artigo menciona apenas que foi usada a transformada wavelet de Haar e que foram usados 16 coeficientes no vetor de características. Na implementação realizada, a imagem foi convertida para o espaço HSV e o canal V foi usado como a imagem em níveis de cinza. Além disso, para a extração da informação de bordas, foi usado o operador de Sobel.

O descritor Spytec foi escolhido por ser independente de segmentação para extrair o vetor de características.

Tabela 3.1: Comparação entre descritores de cor: complexidade da extração de características, complexidade da função de distância e requisitos de espaço. Legenda: n=número de pixels em uma imagem, vs=tamanho do vetor de características, NC=não está claro, ND=não menciona.

Descritor	**Extração**	**Distância**	**Espaço**	**Classe taxonômica**
DCSD [113]	O(n) + O(GLA), GLA=algoritmo GLA	O(vs)	3*QD valores, QD=qtd de cores dominantes	Baseada em segmentação
Zhang SIFT [126]	O(SIFT), SIFT=algoritmo SIFT	O(vs)	722 dimensões (sem PCA) e 188 dimensões (com PCA)	Local
CW-LUV [77]	O(n)	O(vs)	127 bits	Global
DCDI [116]	O(n) + O(LBA), LBA=algoritmo LBA	O(Nq*Ni), Nq=qtd de cores dominantes da imagem de consulta, Ni=qtd de cores dominantes da imagem da base	4*QD valores, QD=qtd de cores dominantes	Baseada em segmentação
CW-HSV [107]	O(n)	O(vs)	63 bits	Global
CSIFT [1]	NC	O(vs^2)	O(NK), NK=número de *keypoints*	Local
LeeSC [60]	O(n)	O(Nc), Nc=qtd de cores no espaço HLS	QB + 2*nb, QB=qtd de pixels de borda, nb=qtd de bins	Baseada em segmentação
SCD [72]	O(n)	O(vs)	QF valores, QF=qtd de coeficientes de Haar	Global
CSD [72]	O(n)	O(vs)	entre 32 e 184 bins (com quantização de 8 bits por bin)	Global
DCD [72]	O(n) + O(GLA), GLA=algoritmo GLA	O(Nq*Ni), Nq=qtd de cores dominantes da imagem de consulta, Ni=qtd de cores dominantes da imagem da base	4*QD valores, QD=qtd de cores dominantes	Baseada em segmentação
CLD [72]	O(n) + O(DCT), DCT=complexidade da transformada discreta cosseno	O(vs)	3*QC valores, QC=qtd de coeficientes DCT	Regiões fixas
CGCH [99]	O(n)	O(vs)	Q valores, Q=qtd de cores no espaço quantizado	Global
JAC [112]	O(n)	O(vs)	QC*QG*QR*QT*QD valores, QC=qtd de cores no espaço quantizado, QG, QR e QT=qtd valores possíveis no espaço quantizado de magnitude do gradiente, rank e texturedness, respectivamente; QD=qtd de distâncias	Global
Chaira Fuzzy [16]	O(n)	O(vs)	3*Q valores, Q=qtd de cores no espaço quantizado	Global
WC [75]	O(nlogn)	O(vs)	QW*QD*QM*QS valores, QW=quantização coeficientes wavelet, QD=qtd de distâncias autocorrelograma, QM=qtd de matrizes, QS=qtd de escalas	Global
CCH [28]	O(n)	O(vs)	QC*0.45*Q (na média) ou QC*Q (no máximo) valores, QC=qtd de células, Q=qtd de cores no espaço quantizado	Regiões fixas
Color Bitmap [69]	O(n)	O(vs)	3*m bits + 6 valores, m=qtd de blocos	Global e Regiões Fixas
CDE [100]	O(n)	O(vs)	(h_1, E_1, H, h_i, E_i, H, h_{nb}, E_{nb}), i=1..nb, nb=qtd de cores no espaço quantizado	Regiões fixas
Nayar MRH [39]	O(n*y), y=largura do filtro Gaussiano separável	O(vs)	O(t(L-1)), t=qtd de valores no espaço quantizado, L=$\log_2 \sqrt{n}$	Global
Coesão [22]	O(n)	O(vs)	Q valores, Q=qtd de cores no espaço quantizado	Baseada em segmentação
Color Blocks [63]	O(n)	O(k*m*Q), k=qtd de blocos imagem de consulta, m=qtd blocos imagem da base, Q=qtd de cores no espaço quantizado	valor ou histograma para cada bloco. Pior caso: Q*QB valores, QB=qtd de blocos da imagem	Regiões fixas
CM [83]	O(n) + O(Q^2*QM), Q=qtd de cores no espaço quantizado, QM=qtd de momentos	O(vs)	QM*2 valores	Global
BIC [29]	O(n)	O(vs)	Q*2 valores, Q=qtd de cores no espaço quantizado	Baseada em segmentação
CBC [27]	O(nlogn)	O(Q1*Q2*log(Q1*Q2)), Q1=qtd de regiões imagem de consulta, Q2=qtd de regiões imagem da base	QR*6 valores, QR=qtd de regiões	Baseada em segmentação
CCV [84]	O(n)	O(vs)	Q*2 valores, Q=qtd de cores no espaço quantizado	Baseada em segmentação
GCH [101]	O(n)	O(vs)	Q valores, Q=qtd de cores no espaço quantizado	Global
LCH [101]	O(n)	O(vs)	Q*QC valores, Q=qtd de cores no espaço quantizado, QC=qtd de células	Regiões fixas
ACC [45]	O(n)	O(vs)	Q*QD valores, Q=qtd de cores no espaço quantizado, QD=qtd de valores de distância	Global

Tabela 3.2: Comparação entre descritores de cor: ambiente de testes e validação. Legenda: HET=heterogênea, HETWEB=heterogênea da Web, NU=não usou, NC=não comparou com outros descritores, IT=comparou com variações do descritor proposto, ND=não menciona.

Descritor	**Tamanho da base**	**Características da base**	**Medidas de avaliação**	**Comparações**
DCSD [113]	5466	CCD MPEG-7	ANMRR, NMRR	DCD, CSD, SCD, CLD
Zhang SIFT [126]	ND	INRIA dataset	Recall × (1-Precision)	SIFT, IT
CW-LUV [77]	100	ND	ANMRR, AVR, MRR, NMRR	Espaços HSV e CIELuv
DCDI [116]	2100	COREL	ARR, ANMRR	GLA
CW-HSV [107]	2997	HETWEB	ANMRR	IT
CSIFT [1]	ND	ALOI (Amsterdam)	número de características CSIFT repetidas dividido pelo número de características SIFT	SIFT
LeeSC [60]	5000	HET	ANMRR, Precision, Recall	GCH, Hybrid Graph Representation, Correlograma de Cor
SCD-MPEG7 [72]	5000	CCD MPEG-7	ANMRR	IT
CSD-MPEG7 [72]	5000	CCD MPEG-7	ANMRR	IT
DCD-MPEG7 [72]	5000	CCD MPEG-7	ANMRR	IT
CLD-MPEG7 [72]	5000	CCD MPEG-7	ANMRR	IT
CGCH [99]	3000	HET	Rank	GCH
JAC [112]	24000	Berkeley Digital Library	SCOPE × Recall	GCH, Histograma Conjunto, Correlograma de Cor
Chaira Fuzzy [16]	(1) 100, (2) 120	(1) Bandeiras de países, (2) texturas	Precision, Recall, Precision × Recall	3 funções de distância diferentes: divergência fuzzy, medida de similaridade fuzzy, modelo GTI
WC [75]	1000	COREL	Precision, Rank	*Simplicity*, WBIIS, Correlograma de Cor, Correlograma de Bordas
CCH [28]	20000	COREL	Precision × Recall, θ_{abs}, θ_{rel}	GCH, CCV, LCH
Color Bitmap [69]	(1) 800, (2) 470, (3) 10000	(1) animações, (2) e (3) coloridas	*Retrieval accuracy*	GCH, Momentos de Cor, Método de Chang e Liu
CDE [100]	8000	HETWEB	ANMRR, Recall, Precision	SCH, Geostat
Nayar MRH [39]	(1) 108, (2) 91, (3) 8046	(1) Synthetic, (2) Brodatz, (3) CureT	% de casamentos de texturas (*matches*)	Espectro Fourier, Características de Color, Pacotes Wavelet, Matriz de Co-ocorrência, Campos randômicos de Markov
Coesão [22]	7000	HET	Precision, Recall, Rank1, Rank˜ , P_{20}, P_{50}, P_{Nr}, $R_{P(0.5)}$, R_{100}	GCH, CCV
Color Blocks [63]	ND	ND	ND	GCH
CM [83]	2266	VisTex (suavização, ruído, etc)	Precision, Recall	Intersecção de Histogramas
BIC [29]	20000	COREL	Precision × Recall, theta × Recall, P_R, P_{30}, R_{30}, P_{100}, R_{100}, 3P-Precision, 11P-Precision	GCH, CCV, CBC, LCH
CBC [27]	20000	COREL	Precision × Recall, NavgR	CCV, CMM, GCH, LCH, CSH
CCV [84]	(1)11667, (2)1440, (3)1005, (4)93, (5)349	(1)Chabot, (2)QBIC, (3)COREL, (4)PhotoCD, (5)Cenas de Videos	Rank	GCH
GCH [101]	ND	Objetos em close	Porcentagem média de casamento	Intersecção de Histogramas, Intersecção Incremental
LCH [101]	ND	Objetos em close	Porcentagem média de casamento	Intersecção de Histogramas, Intersecção Incremental
ACC [45]	14554	HET	r-measure, avg-r-measure, p1-measure, avg-p1-measure; Recall × Scope	GCH, CCV, CCV/C

Tabela 3.3: Comparação entre descritores de textura: complexidade da extração de características, complexidade da função de distância e requisitos de espaço. Legenda: n=número de pixels em uma imagem, vs=tamanho do vetor de características, NC=não está claro, ND=não menciona, Estat=estatísticos, Geo=geométricos, Mod=modelos, Proc=processamento de sinais/wavelets.

Descritor	**Extração**	**Distância**	**Espaço**	**Classe taxonômica**
2DWAH [53]	NC	O(vs)	2*LS^2 valores, LS×LS=tamanho da janela de visão	Estat
Newsam SIFT [80]	O(SIFT) + O(k-means)	O(vs)	K valores que representam o número de ocorrências de SIFT features com rótulo i (i=0..K-1)	Proc e Estat
IFLT [48]	O(n)	O(vs)	S*Q valores, S=qtd de escalas, Q=qtd de valores no espaço quantizado	Proc e Estat
HanGabor [40]	O(nlogn)	O(vs)	2*S ou 2*K valores, S=qtd de escalas, K=qtd de orientações (invariância à escala ou à orientação)	Proc
HTDI [127]	O(nlogn)	NC	(2*S*K)+2 valores, S=qtd de escalas, K=qtd de orientações	Proc
CSG [46]	O(n)	O(vs)	1440/k valores, k=quantização dos ângulos	Proc
TBDI [61]	O(nlogn)	O(vs)	13 bits	Proc
Takala LBP [102]	O(n)	O(VQ*VB), VQ=tamanho vetor da imagem de consulta, VB=tamanho vetor da imagem da base	O(N*QLBP), N=qtd de blocos, QLBP=depende do LBP usado	Mod e Estat
QCCH [44]	O(n)	O(vs)	T valores, T=qtd de valores no espaço quantizado	Estat
BSpline [109]	O(n)	O(vs)	20*D valores, D=qtd de decomposições wavelet	Proc
KokareCW [55]	O(n)	O(vs)	8*D valores, D=qtd de decomposições wavelet	Proc
SimZernike [97]	O(nlogn) + O(DMA), DMA=cálculo dos momentos de Zernike modificados	O(vs)	Nz+2 valores, Nz=qtd de momentos	Proc
SASI [130, 131]	O(S*L*n), S*L=tamanho das janelas em pixels	O(vs)	2*QO*$\sum(\lceil(tamDaJanela[i]/4)\rceil+1)$ valores, QO=qtd de orientações, tamDaJanela[i]=largura de cada janela	Estat
SOH [128]	p/ S pequeno=O(n); p/ S grande=O(n^2), S=qtd de escalas	O(t*θ*vs), t=qtd de translações, θ=qtd de rotações	S*K valores, K=qtd de orientações	Estat
MERF [117]	O(nlogn)	O(vs)	S*QA*2 + m1*m2*2 valores, S=qtd de escalas, QA=qtd de ângulos de rotação, m1*m2=qtd de filtros de Gabor	Proc
Sim DCT [96]	O(n) + O(DCT), DCT=complexidade da transformada discreta cosseno	O(vs)	52 valores	Proc
HTD [71,114]	O(nlogn)	O(vs)	2*S*K valores, S=qtd de escalas, K=qtd de orientações	Proc
HTDR [86]	O(nlogn)	O(vs)	2*S*K valores, S=qtd de escalas, K=qtd de orientações	Proc
LAS [103]	O(n)	O(vs)	Q valores, Q=quantização dos valores de gradiente	Estat
Rubner EMD [87]	O(nlogn) + O(C), C=*clustering* com *kd-trees*	O(VQ*VB), VQ=tamanho vetor da imagem de consulta, VB=tamanho vetor da imagem da base	QC*(K*S+1), QC=qtd de clusters, S=qtd de escalas, K=qtd de orientações	Proc
CCOM [57]	O(n)	O(vs)	pior caso Q^2*D*2, Q=qtd de cores no espaço quantizado, D=qtd de distâncias usadas	Estat
TBD [72,114]	O(nlogn)	O(vs)	12 bits	Proc
EHD [72]	O(n)	O(vs)	3*QB*QE bits, QB=qtd de blocos, QE=quantização das bordas (usa 3bits/bin)	Estat
GCOM [41]	O(n) [para calcular matrizes] e O(QG^3) [para extrair informações das matrizes no pior caso], QG=qtd de níveis de cinza no espaço quantizado	ND	O(QG^2*QD*QA), QD=qtd de distâncias usadas, QA=qtd de direções usadas [espaço das matrizes] ou QA*14 [espaço das informações extraídas das matrizes]	Estat
LBP [82]	O(n)	O(vs)	P+2 valores, P=qtd de vizinhos considerados	Mod e Estat
Unser [106]	O(n) [para calcular histogramas] e O(QG) [para extrair informações dos histogramas], QG=qtd de níveis de cinza no espaço quantizado	ND	QA*(QG*2)*2 valores, QA=qtd de direções usadas	Estat
SBP [119]	O(nlogn)	O(vs)	256*S*K (NU-SBP) ou 59*S*K (U-SBP) valores, S=qtd de escalas, K=qtd de orientações	Proc, Mod e Estat
SID [121]	O(nlogn)	O(vs)	2*S*K valores, S=qtd de escalas, K=qtd de orientações	Proc

Tabela 3.4: Comparação entre descritores de textura: ambiente de testes e validação. Legenda: HET=heterogênea, HETWEB=heterogênea da Web, NU=não usou, NC=não comparou com outros descritores, IT=comparou com variações do descritor proposto, ND=não menciona.

Descritor	**Tamanho da base**	**Características da base**	**Medidas de avaliação**	**Comparações**
2DWAH [53]	1760	Brodatz	NMRR, ANMRR	EHD MPEG-7, EPNH, ARP; Gabor, GLCM, ORDC
Newsam SIFT [80]	86154	Imagens de satélite IKONOS	Precision, ANMRR	Gabor
IFLT [48]	(1) 672, (2) 3840	(1) Brodatz; (2) OUTEX_TC_00010	Precisão da classificação	LBP, IT
HanGabor [40]	(1) 1792, (2)ND	Brodatz, MPEG-7	Precision, Recall	Gabor
HTDI [127]	(1)91, (2)1000	(1)Brodatz, (2)UIUC-Tex	Precision	HTD
CSG [46]	2400	Brodatz	nT = precision ou recall	Vetores Gradiente, Wavelet Energy Signature
TBDI [61]	(1)896, (2)1896	(1)Brodatz, (2)Corel	Taxa de Classificação, Tempo de extração (segundos), Mostrou resultados de consultas	HTD MPEG-7 Original
Takala LBP [102]	(1)1350, (2)900	(1)Corel, (2)Selos alemães	Precision, Recall	Correlograma de Cor, EHD MPEG-7
QCCH [44]	800	HETWEB (categorizada)	Precision, P_{20}, P_{50}, P_{80}, Tempo de extração (segundos)	SCH, Gabor
BSpline [109]	1792	Brodatz	Taxa de recuperação média dos 15 top resultados	Gabor, DWT
KokareCW [55]	(1)1728, (2)112, (3)16	(1)Brodatz, (2)USC, (3)1 textura artificial	Taxa de recuperação média dos 16 top resultados	Daubechies wavelets, Gabor wavelets
SimZernike [97]	(1)1856, (2)832, (3)880, (4)115, (5)2400, (6)8000, (7)4900	(1)Brodatz+USC, (2)ICU, (3)Rotacionada (Brodatz+ICU), (4)Escalonada (Brodatz), (5)Corel, (6)LANSAT, (7) Rotacionada/Escalonada = (Brodatz+Corel)	Recall, Tamanho vetor (bytes), Tempo de extração (segundos), Complexidade da distância (qtd de adições e subtrações), Entropia	Gabor, Radon, Wavelet
SASI [130, 131]	(1)1792, (2)976, (3)480, (4)2672, (5)5920	(1)Brodatz, (2)CUReT, (3)PhoTex, (4)VisTex, (5)Todas juntas	Taxa de recuperação média dos 15 top resultados	Gabor
SOH [128]	600	Brodatz	Taxa de recuperação média	Wold, MRSAR
MERF [117]	1744	Brodatz	Taxa de recuperação média dos top 15 resultados	Gabor, MRSAR
Sim DCT [96]	mais que 3000	Brodatz, ICU	Recall, Tamanho vetor (bits), Tempo de extração (segundos), Complexidade da distância (qtd de adições e subtrações)	Gabor, Radon, Wavelet
HTD [71,114]	1856	Brodatz	Taxa de recuperação média dos 15 top resultados, Qtd de elementos no vetor, Tempo de extração (segundos), Tempo de busca e ordenação (segundos)	PWT, TWT, MRSAR
HTDR [86]	880	MPEG-7 (Brodatz=30+ICU=25)	Taxa de recuperação média dos 15 top resultados, Qtd de cálculos de distância, Tempo de distância (segundos)	Distância original do HTD, função completa
LAS [103]	Classificação: 1920. CBIR: 100	Brodatz e imagens de satélite; mais de uma textura por imagem	Classificação: porcentagem de classificações corretas. CBIR: mostrou resultados de consultas	Classificação: Comparou LAS com gradient indexing [usando 4 operadores de gradiente diferentes: Sobel, Prewitt, Roberts e Laplaciano], Campos Randômicos de Markov Gaussianos (GMRF) e Local Linear Transform. CBIR: NC
Rubner EMD [87]	(1)1792 + outras, (2)500	(1)Brodatz, (2)Corel	Mostrou resultados de consultas	NC
CCOM [57]	20000	HET	Mediu a porcentagem de sucesso variando ruído e variando o tamanho da consulta	NC
TBD [72,114]	448	Brodatz	Foram avaliados subjetivamente por 5 pessoas	NC
EHD [72]	11000	MPEG-7	ANMRR	IT
GCOM [41]	(1)243, (2)170, (3)624)	(1)Fotomicrografias de arenitos, (2)Fotos aéreas, (3)Imagens de satélite sobre a costa da Califórnia	Precisão total da classificação	NC
LBP [82]	(1)7744, (2)3840 and 4320	(1)Brodatz, (2)OUTex	Precisão da Classificação	IT, Wavelet invariante à rotação
Unser [106]	(1)3072, (2)768 e (3)192	Brodatz	Porcentagem de classificados corretamente	GCOM
SBP [119]	640	VisTex	Taxa de recuperação média dos N top resultados, com N=[16..64]	Steerable Pyramid Decomposition original, GGD&KLD
SID [121]	(1)208, (2)208, (3)208	(1)Brodatz, (2)Brodatz rotacionada, (3)Brodatz escalonada	Porcentagem de imagens relevantes entre as top N retornadas. Mostrou resultados de 4 consultas.	Steerable Pyramid Decomposition original, Gabor Wavelets

Tabela 3.5: Comparação entre descritores de forma: complexidade da extração de características, complexidade da função de distância e requisitos de espaço. Legenda: SD=dependente de segmentação, n=número de pixels em uma imagem, vs=tamanho do vetor de características, NC=não está claro, ND=não menciona, NSeg=independente de segmentação.

Descritor	**Extração**	**Distância**	**Espaço**	**Classe taxonômica**
Chen RPD [18]	SD	O(vs)	3*m valores, m=qtd de coeficientes de Fourier	Contorno e Região
2DWAH [54]	NC	O(vs)	2*LS^2 valores, LS×LS=tamanho da janela de visão	NSeg
Kong SW [56]	SD	O(vs)	Depende da base: K valores, K=maior qtd de pontos do contorno da base	Contorno
Jang CO [47]	NC	O(vs)	QA+Q+1 valores, QA=qtd de ângulos no espaço quantizado, Q=qtd de cores no espaço quantizado	NSeg
Adnan GS [2]	SD	ND	ND	Contorno
2GFD [92]	SD	O(vs)	2*R*T valores, R=qtd de raios, T=qtd de ângulos	Contorno e Região
CS [24]	SD	O(max(k1^2*k2,k1*k2^2)), k1=tamanho vetor da imagem de consulta, k2=tamanho vetor da imagem da base, $max(x, y) = x$ se $x \geq y$ ou y, caso contrário	2*k valores, k=qtd de pontos de saliência	Contorno
SS [24]	SD	O(vs^2)	S valores, S=qtd de segmentos pré-definidos	Contorno
TAR [5]	SD	O($\|V1\|$. $\|V2\|$ .N^2), $\|V1\|$=qtd de objetos na imagem de consulta, $\|V2\|$=qtd de objetos na imagem da base, N=qtd de pontos do contorno dos objetos (fixo)	O($\|V\| * N^2$)	Contorno
MGFD [64]	O(nRT), R=qtd de raios e T=qtd de ângulos	O(vs)	R*T valores, R=qtd de raios, T=qtd de ângulos	NSeg
TSD [74]	O(nd), d=diagonal da imagem	O(vs^2)	QA valores, QA=qtd de ângulos no espaço quantizado	NSeg
SHD [91]	SD	O(vs)	NC	Região
Naskar RH [78]	SD	O(vs)	QD valores, QD=qtd de valores de distância no espaço quantizado	Contorno
HBA [94]	NC	O(vs)	QD+QA+QC valores, QD=qtd de distâncias no espaço quantizado, QA=qtd de ângulos no espaço quantizado, QC=qtd de cores no espaço quantizado	NSeg
GFDC [90]	SD	O(vs)	9*R*T valores, R=qtd de raios, T=qtd de ângulos	Região
Traina AS [104]	O(n) + O(MRF), MRF=complexidade dos campos randômicos de Markov	O(vs)	6*QC valores, QC=qtd de classes	NSeg
MCS [65]	SD	O(vs)	N valores, N=qtd de pontos do contorno	Contorno
MSF [58]	SD	O(vs)	L*R valores, L=qtd de coeficientes Fourier, R=qtd de escalas	Contorno
Wang CCD [110]	SD	O(QC*QS) + O(QA), QC=qtd de pontos do contorno do objeto da base, QS=qtd de rotações (depende do objeto de consulta), QA=qtd de ângulos no espaço quantizado	QA+1+QP valores, QP=qtd de pontos do contorno	Contorno
EOAC [70]	O(n)	O(vs)	QA*QD valores, QA=qtd de ângulos no espaço quantizado, QD=qtd de distâncias do auto-correlograma	NSeg
GFD [123]	SD	O(vs)	(R*T)+1 valores, R=qtd de raios, T=qtd de ângulos	Região
Spytec [59]	O(n)	O(vs)	F valores, F=qtd de coeficientes wavelet	NSeg
MLEV [51]	SD	O(vs)	$(\frac{4^{QD+1}-1}{3} - 1) * 4 + 2$ valores, QD=qtd de divisões	Região
ZMD [52]	SD	O(vs)	QM valores, QM=qtd de momentos	Região
TSDIZ [7]	SD	O(vs^2)	NS valores, NS=qtd de segmentos do contorno	Contorno e Região
TSCS [7]	SD	O(max(k1^2*k2,k1*k2^2)), k1=tamanho vetor da imagem de consulta, k2=tamanho vetor da imagem da base, $max(x, y) = x$ se $x \geq y$ ou y, caso contrário	2*k valores, k=qtd de pontos de saliência	Contorno
Wang Hash [108]	SD	O(m^2*k), m=qtd máxima de pontos de interesse, k=qtd de entradas em um registro da hash table	Para cada objeto da imagem: (QP-2)*QH valores, pior caso: QH=QP, QP=qtd de pontos do contorno do objeto, QH=basis pairs	Contorno
OR [13]	O(nlogn)	O(vs)	QF*QO valores, QF=qtd de coeficientes de Fourier, QO=qtd de orientações	NSeg
IMI [17]	SD	ND	QM valores, QM=qtd de momentos	Contorno
CSS [76]	SD	O(vs)	2*QP valores, QP=qtd de picos do objeto	Contorno
FD [122, 124]	SD	O(vs)	QF valores, QF=qtd de coeficientes de Fourier	Contorno
MI [43]	SD	O(vs)	QM valores, QM=qtd de momentos	Região

Tabela 3.6: Comparação entre descritores de forma: ambiente de testes e validação. Legenda: HET=heterogênea, HETWEB=heterogênea da Web, NU=não usou, NC=não comparou com outros descritores, IT=comparou com variações do descritor proposto, ND=não menciona.

Descritor	**Tamanho da base**	**Características da base**	**Medidas de avaliação**	**Comparações**
Chen RPD [18]	(1)2970, (2)750	(1)Imagens binárias de objetos, (2)Logotipos e ícones	Precision × Recall, Taxas médias de reconhecimento	Moment Invariants, Shape Signature, Fourier
2DWAH [54]	(1)1000, (2)10000, (3)20000, (4)1400, (5)1760	(1),(2),(3)Corel, (4)Imagens binárias, (5)Brodatz	NMRR, ANMRR	EHD MPEG-7, EPNH, ARP; Gabor, GLCM, ORDC
Kong SW [56]	1100	SQUID	Precision	Moment Invariants, modelos Centroid-radii
Jang CO [47]	300	Imagens naturais	Precision, Recall	GCH, um outro método só de forma
Adnan GS [2]	ND	ND	ND	ND
2GFD [92]	3621	S8 MPEG-7 Still Images Content Set (logotipos), tansformações geométricas e de perspectiva	Precision × Recall	GFD tradicional
CS [24]	11000	Peixes	Separabilidade, Compactabilidade	CSS, BAS, CCS
SS [24]	11000	Peixes	Separabilidade, Compactabilidade	CSS, BAS, CCS
TAR [5]	(1)1400, (2)1500, (3)246	(1)MPEG-7 CE-shape-1 part B, (2) e (3)logotipos	Precision, Medida F. Na base (2) usou 5 pessoas pra avaliar os top 5 resultados de 24 consultas.	(1)CSS, MCC, IDSC. (2)Momentos de Zernike (ZMD), outro baseado em objeto
MGFD [64]	1000	Corel	Taxa de recuperação, mostrou resultados de consultas.	Fourier (FD), IT
TSD [74]	11000	Formas de contornos de peixes	Precision × Recall, θ × Recall	CCS, CSS, BAS
SHD [91]	3621	S8 MPEG-7 Still Images Content Set (logotipos)	Precision × Recall	GFD, GFD+conectividade
Naskar RH [78]	ND	ND	ND	NC
HBA [94]	(1)200 images, 5340 objetos, (2)1490, (3)16, (4) 3621	(1)HET, (2)Corel, (3)TV clips, (4) MPEG-7 CE-2	Precision × Recall, Acurácia da Recuperação, Tempo de extração (milissegundos)	GCH, GCH com pesos, GCH-distância-ponderada; Turning Angle (TA); GFD
GFDC [90]	3621	S8 MPEG-7 Still Image Content Set (logotipos)	Precision × Recall	GFD. Usou também uma versão periférica, que considerou apenas conectividades iguais a 0 e 8.
Traina AS [104]	220	Imagens Médicas	Precision × Recall	Histograma de brilho
MCS [65]	1100	Contornos de criaturas marinhas	Taxas médias de consulta	Método de Nishida, Método de Stein
MSF [58]	(1)1204, (2)1943	(1)papel, (2)metal (imagens com defeitos de processos industriais)	Precision × Recall	CSS, Fourier do contorno
Wang CCD [110]	1400	Folhas de plantas chinesas	Recall	Centroid-Contour Distance (CCD), CSS, Descritor de Fourier modificado (MFD)
EOAC [70]	10000	HET	Precision (modificado), Recall	IT, GCH, Auto-correlogram de cor (ACC), 12 Shape Factors (SF12), Histograma de direção de bordas (EDH), Histograma de direção de bordas similares (DHSE)
GFD [123]	3621	MPEG-7 CE-2	Precision × Recall, Bull's eye	Momentos de Zernike (ZMD)
Spytec [59]	100	Pinturas	AVRR (Rank médio de todas as relevantes), IAVRR (Rank média ideal), AVRR/IAVRR	QBIC
MLEV [51]	(1)3400, (2)2810	MPEG-7 (1)CE-1, (2)CE-2	*Retrieval Accuracy*	Sete Moment Invariants
ZMD [52]	(1)1400+1300, (2)3000	MPEG-7 (1)CE-1, (2)CE-2	Recuperação média correta nas top N, top 2N, top 200 imagens	Hyundai, HHI, CSS
TSDIZ [7]	1400	MPEG-7 CE-shape-1 partB	Precision × Recall, $Recall_{40}$, Separabilidade multiescala	BAS, Dimensão Fractal Multiescala, Moment Invariants, Fourier, TSD, Saliências dos segmentos (SS)
TSCS [7]	(1)1400, (2)11000	(1)MPEG-7 CE-shape-1 partB, (2)formas de peixes	Separabilidade multi escala	Saliências do Contorno (CS)
Wang Hash [108]	450	Animais marinhos	Precision, Recall	NC
OR [13]	500	Ornamentos	Precision médio do top 1 resultado	Reddi's moments
IMI [17]	(1)6, (2)4	(1)Mapas de países, (2)Ferramentas industriais	Tempo de extração (segundos)	Moment Invariants
CSS [76]	450	Peixes	Mostrou resultados de consultas	NC
FD [122, 124]	3340	MPEG-7 A, B, C	Precision × Recall, Tempo de extração (milissegundos), Tempo de distância (milissegundos)	CSS, outros descritores de forma baseados em Fourier
MI [43]	26	Caracteres	NU	NC

Capítulo 4

Ferramenta para automatização de testes comparativos

A realização de testes práticos comparativos de descritores tende a ser um processo trabalhoso e cansativo. Devem ser executados diversos passos e rotinas repetitivas para se obter os resultados. Isto está sujeito a erros humanos e falta de padronização entre testes diferentes. Neste cenário, reconhece-se a necessidade de se padronizar o processo de realização de testes comparativos entre descritores.

Com esta finalidade, uma das contribuições deste trabalho consistiu na especificação e na implementação de uma ferramenta para a realização automatizada de testes comparativos de descritores. Esta ferramenta integra todas as etapas necessárias para a realização de experimentos práticos de forma organizada e padronizada.

A arquitetura da ferramenta é apresentada na Figura 4.1. Seus módulos e aspectos de implementação são explicados a seguir.

4.1 Gerenciamento de descritores

O módulo de gerenciamento de descritores possibilita adicionar e remover descritores ao ambiente da ferramenta. A arquitetura permite que novos descritores sejam adicionados enquanto outros módulos estiverem em funcionamento.

Um descritor de imagens possui duas funções: uma função responsável por extrair o vetor de características de uma imagem e outra responsável por calcular a distância entre dois vetores de características.

Para se adicionar um novo descritor, a implementação do descritor precisa ser feita seguindo instruções pré-definidas. Foi convencionado que a linguagem de programação a ser usada no desenvolvimento do descritor é a linguagem C.

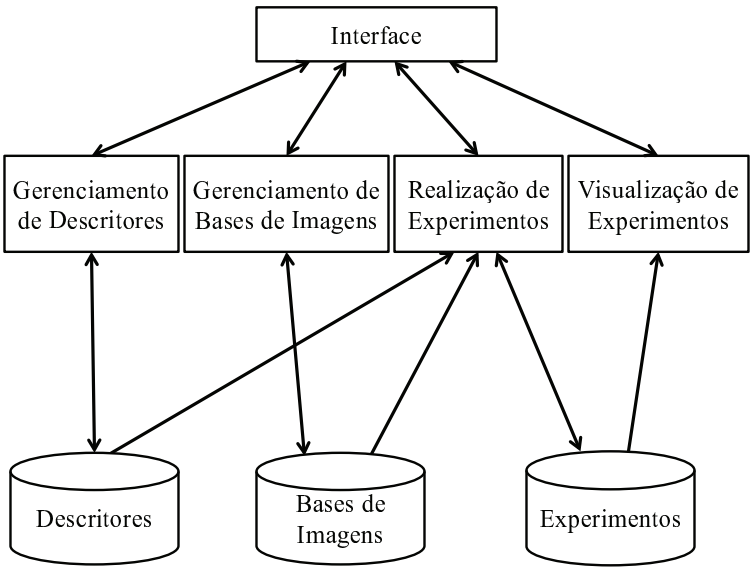


Figura 4.1: Arquitetura da ferramenta para comparação de descritores.

A implementação de um descritor é chamada de *plugin*. Este plugin deve ter obrigatoriamente duas funções com os seguintes cabeçalhos:

- void Extraction(char *img_path, char *fv_path);
- double Distance(char *fv1_path, char *fv2_path);

A função *Extraction* é a responsável por extrair as características de uma imagem. O caminho dessa imagem no sistema de arquivos é o primeiro parâmetro da função. Este caminho inclui o nome do arquivo da imagem. O segundo parâmetro é o caminho no sistema de arquivos do arquivo que será gerado pela função. Este arquivo é o vetor de características da imagem de entrada e irá conter as características extraídas da imagem. Segundo estas especificações, fica a cargo da função *Extraction* a abertura e leitura da imagem de entrada. A ferramenta sempre fornece para a função *Extraction* o caminho de uma imagem em formato PPM.

A função *Distance* é a responsável por comparar dois vetores de características. Ela recebe como parâmetro os caminhos no sistema de arquivos dos dois arquivos de características a serem comparados. A função retorna um valor do tipo *double*, que é a distância calculada entre os dois vetores.

Em seguida, o descritor implementado deve ser compilado de maneira a gerar o plugin que será cadastrado na ferramenta. O plugin é um arquivo com extensão ".so". Depois

de gerado, o plugin gerado deve então ser cadastrado na ferramenta por meio de um formulário de submissão. Assim que é feito o *upload* do plugin, este é verificado por uma rotina para detectar a existência das funções *Extraction* e *Distance*. A rotina usa duas imagens de exemplo como teste, extrai suas características e compara-as. Caso o plugin não esteja de acordo com as especificações, o cadastro é interrompido. Se estiver correto, são exibidas ao usuário as duas imagens com seus respectivos vetores de características e a distância entre eles. O usuário pode então continuar o cadastro do descritor inserindo outras informações sobre o descritor. O usuário deve informar o nome do descritor, o autor ou autores do artigo que o propõe e o tipo do descritor (cor, textura ou forma). O nome do arquivo .so é usado automaticamente como identificador do descritor na ferramenta.

Após o procedimento de cadastro, o descritor já pode ser usado em experimentos.

Na tela de gerenciamento de descritores (ver Figura 4.2), é exibida uma lista com os descritores já cadastrados na ferramenta. Esta lista mostra o identificador, o nome, o autor e o tipo de cada descritor, ordenados por tipo. A exclusão de um descritor só é possível se o descritor não estiver presente em nenhum experimento já realizado. Na exclusão são removidos o registro do descritor no banco de dados e o arquivo de plugin.

Gerenciamento de Descritores

Descritores já cadastrados

id	Nome	Autor	Tipo	Excluir
acc	Autocorrelograma de cor	Jing Huang, S Ravi Kumar, Mandar Mitra, Wei-Jing Zhu, Ramin Zabih	COR	X
bic	BIC	Renato O Stehling, Mario A Nascimento, Alexandre X Falcao	COR	X
cbc	CBC	Renato O Stehling, Mario A Nascimento, Alexandre X Falcao	COR	X
cch	Cell/Color Histogram	Renato O. Stehling, Mario A. Nascimento, Alexandre X. Falcão	COR	X
ccv	CCV	Greg Pass, Ramin Zabih, Justin Miller	COR	X
cgch	Cumulative Color Histogram	Markus Stricker and Markus Orengo	COR	X
cm	Chromaticity Moments	George Paschos, Ivan Radev, Nagarajan Prabakar	COR	X
cm_norm	Chromaticity Moments (normalizado)	George Paschos, Ivan Radev, Nagarajan Prabakar	COR	X
cm_norm_bin	Chromaticity Moments (normalizado, FV binario)	George Paschos, Ivan Radev, Nagarajan Prabakar	COR	X
csd	MPEG-7 Color Structure Descriptor	B. S. Manjunath; J.-R. Ohm; V. V. Vasudevan; A. Yamada	COR	X

Figura 4.2: Tela de gerenciamento de descritores.

4.2 Gerenciamento de bases de imagens

O gerenciamento de bases de imagens permite que sejam adicionadas e removidas bases de imagens ao ambiente da ferramenta.

Para o cadastro de uma nova base de imagens, basta preencher os campos correspondentes aos dados da base e submeter o formulário pela interface da ferramenta. É importante ressaltar que o diretório contendo as imagens deve estar no mesmo servidor em que a ferramenta está instalada.

A exclusão de uma base de imagens é feita pela tela de gerenciamento de bases de imagens que exibe uma lista com as bases já cadastradas (ver Figura 4.3). Ao lado da linha de cada uma das bases existe um botão que exclui a base. A exclusão apenas elimina o registro no banco de dados, sem apagar nenhum arquivo no sistema de arquivos.

Gerenciamento de Bases de Imagens

Bases já cadastradas

id	Nome	Path	Descrição	Classificada?	Excluir
18	Eth-cropped-close-ppm	/exp/otavio/img_databases/eth80-cropped-close128-ppm/	Base Eth com todas as imagens (32B0) - coloridas e em branco e preto	SIM	X
15	imgs_temp	/exp/otavio/img_databases/IMGS_TEMP/	img temp	NÃO	X
21	Problem Images	/exp/otavio/img_databases/problem_images/	Imagens que travaram algum experimento	NÃO	X
22	imgs_temp_classes	/exp/otavio/img_databases/IMGS_TEMP_CLASSES/	Base para testar script de precisionxrecall	SIM	X
20	Yahoo Full	/exp/otavio/img_databases/yahoo_images/home/lika/public_html/	Base completa do Yahoo com mais de 250 mil imagens	NÃO	X
17	yahoo_2000	/exp/otavio/img_databases/yahoo_2000/	2000 imagens da base do yahoo	NÃO	X
26	unicampOne-M	/exp/otavio/img_databases/unicampOne-M/	Pequena parte da base de 1 milhao de imagens do Flickr	SIM	X
27	Brodatz	/exp/otavio/img_databases/brodatz/	Base de textura com 1776 imagens classificadas em 111 classes	SIM	X

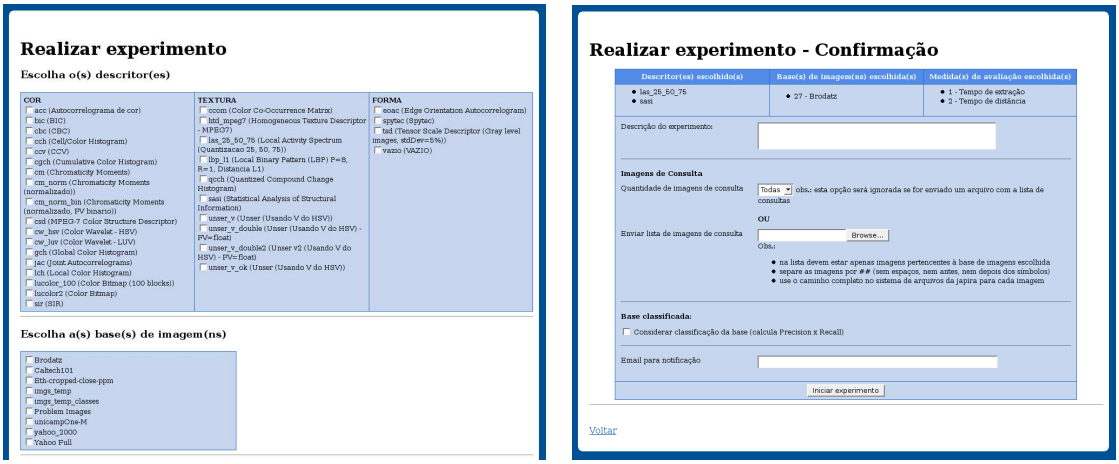
Figura 4.3: Tela de gerenciamento de bases de imagens.

4.3 Realização de experimentos

A realização de experimentos é a função central da ferramenta.

Para iniciar um experimento o usuário deve selecionar quais descritores e quais bases de imagens serão usados (ver Figura 4.4(a)). Em seguida ele insere uma descrição textual sobre o experimento e escolhe quantas ou quais serão as imagens de consulta utilizadas para os cálculos de distância (ver Figura 4.4(b)). O usuário pode usar todas as imagens da base como imagens de consulta, usar um número pré-definido de imagens ou enviar

um arquivo contendo uma lista de imagens de consulta. Esta lista deve conter obrigatoriamente imagens que pertençam à mesma base de imagens usada no experimento. Caso esteja sendo usada uma base de imagens classificada, a ferramenta exibe uma opção para que o usuário escolha se deseja que os valores de Precision × Recall sejam calculados ou não. As medidas de Precision × Recall são detalhadas na Seção 5.1.3.



(a) (b)

Figura 4.4: Telas de configuração do experimento a ser realizado.

A realização do experimento se divide basicamente em duas etapas: extração de características e cálculos de distância.

Assim que o usuário submete o formulário pela interface, um arquivo de configuração é gerado, um diretório responsável por armazenar resultados do experimento é criado no sistema de arquivos e então o experimento se inicia.

Existem duas rotinas principais na ferramenta que são responsáveis pelas duas etapas de um experimento. Quando o experimento se inicia, o código identificador do experimento é passado para a rotina responsável pela extração de características. A rotina obtém as configurações do experimento a partir do arquivo de configurações gerado previamente. Estas configurações incluem os descritores e as bases de imagens a serem usados, o caminho da lista de imagens de consulta, se for o caso, e a informação sobre a geração ou não de curvas de Precision × Recall. A rotina primeiramente cria uma lista em memória contendo os caminhos no sistema de arquivos de todas as imagens das bases escolhidas. São indexadas apenas imagens nos formatos JPG, GIF, PNG e PPM. A partir de então inicia-se o processo iterativo sobre as imagens. Para cada imagem, é verificado se o seu tipo de arquivo é PPM. Caso não seja, é gerada uma cópia da imagem em formato PPM e esta é usada pelos descritores. Para cada descritor escolhido, é criado um subdiretório

dentro do diretório principal do experimento, que será responsável por armazenar os vetores de características do descritor. Após isso, é feita a chamada à função *Extraction* do descritor. A função é executada três vezes e o tempo destas três execuções é medido. Uma variável guarda a média aritmética de tempo destas três execuções. Este valor é também guardado para o cálculo da média final e do desvio padrão dos tempos de extração do descritor. Assim que o descritor atual termina de extrair o vetor de características da imagem atual, passa-se para o próximo descritor, até que todos os descritores do experimento tenham feito o mesmo. Após isso, a cópia da imagem no formato PPM é excluída, caso tenha sido necessária, e passa-se para a próxima imagem.

Assim que todas as imagens tiverem sido processadas por todos os descritores, a rotina calcula a média de tempo de extração de cada descritor, assim como o desvio padrão. Estes valores são inseridos no banco de dados. Além disso, a rotina mantém um arquivo de log que guarda cada passo de execução. Este arquivo de log também fica dentro do diretório do experimento no sistema de arquivos.

Ao se terminar a extração de características, a rotina responsável pelos cálculos de distância é iniciada. Seu parâmetro de entrada, assim como o da rotina de extração, é o código identificador do experimento. A rotina faz a leitura do arquivo de configuração e obtém os dados do experimento. Inicia-se o processo iterativo. Para cada descritor do experimento, cria-se uma lista em memória com todos os vetores de características gerados pelo descritor atual durante a rotina de extração. Se o usuário tiver escolhido um número X de imagens de consulta, é criada uma lista de imagens de consulta que contém apenas as X primeiras entradas da lista de vetores de características criada no início. Caso contrário, a mesma lista completa de vetores de características será usada como a lista de consultas. A rotina então percorre cada imagem da lista de imagens de consulta e a compara com cada imagem da lista completa de vetores. A comparação é feita pela função *Distance* do plugin. Esta função é chamada três vezes para cada comparação e o tempo destas três execuções é calculado. Uma variável guarda a média aritmética destas três execuções. Ao final dos cálculos de distância, a média e desvio padrão dos tempos de cada cálculo de distância são computados e inseridos no banco de dados. Todos os passos do processo são registrados no arquivo de log.

Cada distância calculada é armazenada no banco de dados. A entidade *Distance* mostrada no diagrama Entidade-Relacionamento da Figura 4.8 é usada para representar as distâncias de cada experimento. As informações consideradas pela entidade *Distance* também são usadas para a exibição dos resultados de consultas dos descritores. Mais detalhes do banco de dados da ferramenta na Seção 4.5.

A ferramenta sempre calcula os tempos de extração e os tempos de distância de todos os experimentos realizados.

4.4 Visualização de experimentos

A visualização de experimentos permite que sejam vistos todos os experimentos já realizados e também que se acompanhe o progresso de experimentos em andamento (ver Figura 4.5). Além disso, é possível analisar os resultados de cada experimento. Os resultados que estão disponíveis para todos os experimentos são: tempos de extração, tempos de distância e exibição de resultados de consultas. A possibilidade de realizar uma avaliação de eficácia com usuários só é possível quando o experimento utiliza uma lista pré-definida de imagens de consulta.

Experimentos Realizados

Voltar

Detalhes dos experimentos já realizados

id	Descrição	E-mail responsável	Descritores usados	Bases usadas	Medidas extraidas	Progresso	Resultados
582	TEXTURA: testando descritores de textura na base dividida	otavio@lis.ic.unicamp.br	• lbp_l1	• Brodatz	• Tempo de extração • Tempo de distância tabelas	E=100% D=100%	- LOG - IMGS PR=100%
575	FORMA: Testando descritores de forma na base classificada	otavio@lis.ic.unicamp.br	• eoac • spytec	• Caltech101	• Tempo de extração • Tempo de distância tabelas	E=100% D=100%	- LOG - IMGS - AVAL PR=100%
574	Testando unicampOneM - 1156 imagens.	teste@teste.com	• eoac	• unicampOne-M	• Tempo de extração • Tempo de distância tabelas	E=100% D=100%	- LOG - IMGS PR=100%
567	TEXTURA: Comparando descritores de textura na Brodatz	otavio@lis.ic.unicamp.br	• sasi	• Brodatz	• Tempo de extração • Tempo de distância tabelas	E=100% D=100%	- LOG - IMGS PR=100%
565	TEXTURA: Comparando descritores de textura na Brodatz	otavio@lis.ic.unicamp.br	• ccom • unser_v_ok	• Brodatz	• Tempo de extração • Tempo de distância tabelas	E=100% D=100%	- LOG - IMGS PR=100%
564	COR - ETH: Testando descritores de cor na base classificada. (LUCOLOR_100)	otavio@lis.ic.unicamp.br	• lucolor_100	• Eth-cropped-close-ppm	• Tempo de extração • Tempo de distância tabelas	E=100% D=100%	- LOG - IMGS PR=100%

Figura 4.5: Tela de exibição dos experimentos realizados e experimentos em andamento.

Abaixo são detalhados como são disponibilizados os resultados dos experimentos.

4.4.1 Tempos e progresso do experimento

Os tempos calculados pela ferramenta podem ser visualizados por uma tela específica da interface. Esta tela exibe, além dos tempos calculados, algumas informações para identificação do experimento e o progresso das rotinas de extração e de distância. Os tempos calculados são exibidos em duas tabelas contendo o identificador do descritor, o

tempo médio e o desvio padrão. Uma tabela exibe os tempos de extração e a outra exibe os tempos de distância.

4.4.2 Diretório de resultados

A interface disponibiliza um link para o diretório de resultados do experimento. Desta forma, é possível acessar todos os arquivos gerados durante a realização do experimento, como o arquivo de log. Além disso, caso tenha sido solicitada a geração de curvas de Precision $\times$ Recall, os arquivos contendo os valores destas curvas também poderão ser acessados.

4.4.3 Resultados de consultas

Para cada experimento, é possível ver as imagens retornadas em cada uma das consultas realizadas. A tela exibe em destaque a imagem de consulta e em seguida uma grade contendo os resultados ordenados em ordem crescente de distância (ver Figura 4.6).

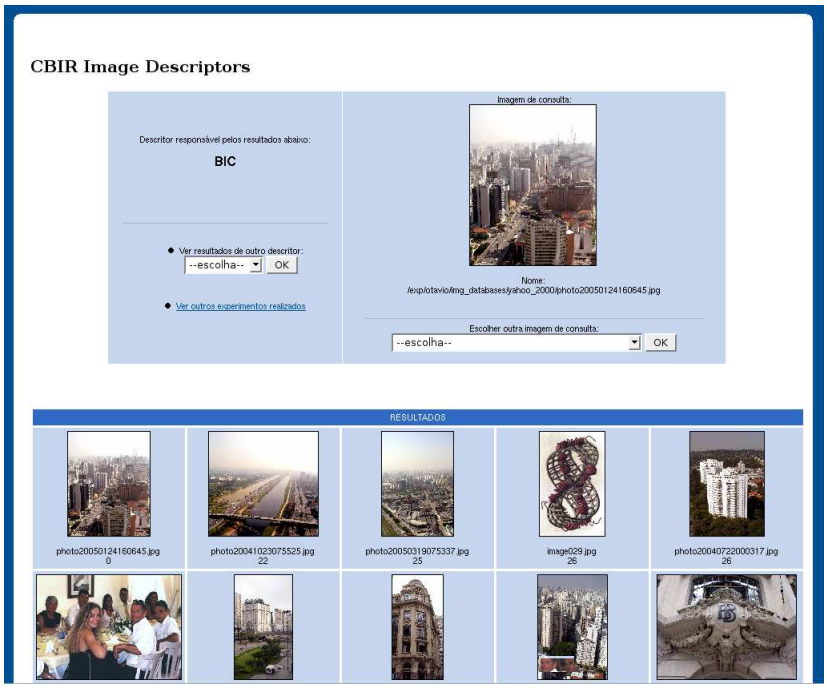


Figura 4.6: Tela de exibição de resultados de consultas.

Os resultados exibidos correspondem aos resultados de um único descritor. O usuário pode escolher, dentre os descritores usados no experimento, o descritor para o qual ele

desejar ver os resultados.

Se o experimento calculou distâncias entre todas as imagens da base usada, cada imagem exibida nos resultados possui um link. Se este link for acionado, a imagem selecionada passa a ser a imagem de consulta. Caso o experimento tenha usado uma lista pré-definida de consultas, não existem links nas imagens de resultado. A troca da imagem de consulta é feita por uma lista que exibe apenas as imagens de consulta utilizadas.

Sempre que um descritor diferente é selecionado, a imagem de consulta atual se mantém. E, da mesma forma, sempre que uma nova imagem de consulta é selecionada, o descritor atual se mantém.

4.4.4 Avaliação de eficácia com usuários

Quando um experimento é executado usando-se uma lista pré-definida de imagens de consulta, a ferramenta fornece uma interface para realizar uma avaliação de eficácia com usuários reais (ver Figura 4.7).

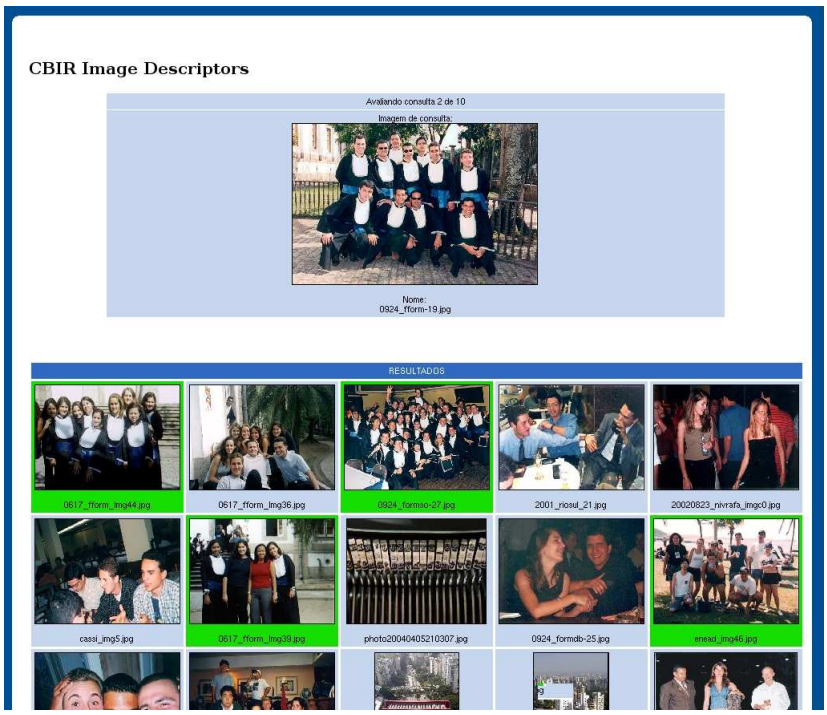


Figura 4.7: Tela para avaliação de eficácia com usuários.

O objetivo da interface para avaliação de eficácia com usuários reais é permitir que um usuário indique na lista de imagens retornadas quais imagens ele considera relevantes.

Baseando-se nas imagens selecionadas pelo usuário, são calculados valores que indicam a eficácia dos descritores avaliados.

Os procedimentos para se configurar a avaliação de eficácia são semelhantes aos realizados em [37]. Para cada descritor do experimento a ser avaliado, foram recuperadas as 30 imagens mais relevantes para cada consulta. Portanto, para cada consulta, existe uma lista com 30 imagens de cada descritor. Estas listas foram unidas em uma única lista, removendo-se as possíveis imagens duplicadas e mantendo-se a referência para o descritor que recuperou cada imagem e a referência da posição de cada imagem na lista do descritor. A lista final foi embaralhada antes de ser exibida ao usuário.

A avaliação é feita com uma imagem de consulta por vez. A exibição das imagens para o usuário é feita numa tela bastante semelhante à usada para a visualização de resultados de consultas. A imagem de consulta é exibida em destaque e abaixo dela são exibidas as imagens da lista embaralhada. O usuário não tem a informação de quais descritores foram usados para recuperar aquelas imagens. Ele deve apenas marcar quais imagens ele considera relevantes em relação à imagem de consulta. Assim que ele terminar a avaliação da consulta atual, ele clica em um botão no final da tela para avançar para a próxima imagem de consulta.

Assim que o usuário clica no botão, os valores de eficácia são calculados baseados nas imagens selecionadas como relevantes. A ferramenta calcula as medidas de P_{10}, P_{20} e P_{30} (detalhadas na Seção 5.1.3).

O processo de avaliação se repete até que todas as imagens de consulta tenham sido avaliadas. As medidas de eficácia são calculadas para cada usuário avaliador em cada imagem de consulta e para cada descritor. Portanto, é possível saber a eficácia de cada descritor em cada imagem de consulta.

4.5 Aspectos de implementação

A ferramenta utiliza um banco de dados para o armazenamento das principais informações por ela gerenciadas. O banco de dados foi criado seguindo o modelo apresentado na Figura 4.8.

A entidade *Descriptor* é usada para representar os descritores cadastrados na ferramenta e possui os atributos: identificador e nome do descritor, autor ou autores do artigo que propõe o descritor e tipo do descritor (cor, textura ou forma).

A entidade *Image Database* contém as informações das bases de imagens cadastradas na ferramenta, como: identificador, nome, caminho no sistema de arquivos, descrição e um valor verdadeiro ou falso indicando se a base é classificada ou não.

A entidade *Evaluation Measure* representa as medidas de avaliação que a ferramenta pode calcular em seus experimentos. Esta entidade contém dois atributos: identificador

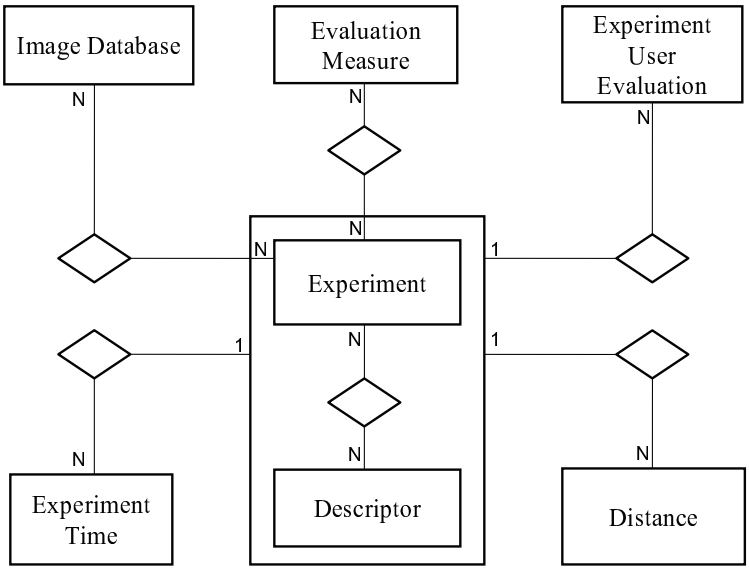


Figura 4.8: Modelo Entidade-Relacionamento do banco de dados da ferramenta para comparação de descritores.

e nome da medida de avaliação.

A entidade *Experiment* representa um experimento executado ou em execução na ferramenta. Seus atributos são: identificador e descrição do experimento. Como os relacionamentos da entidade *Experiment* com as entidades *Descriptor*, *Image Database* e *Evaluation Measure* possuem cardinalidade $N : N$, existem mais três entidades que guardam informações dos experimentos. O relacionamento *Experiment-Descriptor* guarda os identificadores dos descritores usados em cada experimento. O relacionamento *Experiment-Image Database* guarda os identificadores das bases de imagens usadas em cada experimento. O relacionamento *Experiment-Evaluation Measure* armazena os identificadores das medidas de avaliação usadas em cada experimento. Desta maneira, o conjunto de entidades e relacionamentos relacionados com a entidade *Experiment* possui todas as informações relevantes sobre os experimentos executados na ferramenta.

As entidades *Experiment Time*, *Distance* e *Experiment User Evaluation* representam resultados parciais ou finais dos experimentos. A entidade *Experiment Time* representa os tempos calculados pela ferramenta, como tempos de extração e tempos de distância. A entidade *Distance*, por sua vez, representa todas as distâncias calculadas durante a rotina de cálculos de distância. Cada entidade de distância possui os identificadores do experimento e do descritor usado, os caminhos das duas imagens comparadas e a distância

calculada entre elas. A entidade *Experiment User Evaluation* representa os resultados das avaliações de eficácia com usuários reais. Seus atributos são os identificadores do experimento e do descritor usado, o e-mail do usuário que avaliou, a imagem de consulta utilizada, os valores de P_{10}, P_{20} e P_{30} calculados.

O banco de dados da ferramenta foi implementado utilizando-se o SGBD PostgreSQL.

A ferramenta possui uma interface Web que permite o acesso a todas as suas funcionalidades (ver Figura 4.9). A implementação da interface foi feita em linguagem PHP. As rotinas de execução de experimentos foram codificadas em linguagem Python.

Figura 4.9: Tela inicial da ferramenta.

A estrutura de diretórios da ferramenta é a seguinte:

- / (raiz)
  - /codes
  - /descriptors
  - /img_databases
  - /results

Na raiz encontram-se os arquivos PHP responsáveis pela interface da ferramenta. No diretório *codes* estão as rotinas para extração de características, cálculos de distância, visualização de imagens, geração de curvas de Precision × Recall e a interface para avaliação de eficácia com usuários. No diretório *descriptors* ficam armazenados os plugins dos descritores cadastrados na ferramenta. O diretório *img_databases* contém as bases de

imagens. O diretório *results* guarda todos os arquivos gerados durante a realização de cada experimento. Dentro dele existe um diretório para cada experimento.

Capítulo 5

Experimentos

Este capítulo apresenta os resultados da comparação experimental entre os descritores estudados. Também são apresentados detalhes do projeto dos experimentos, como as medidas de avaliação e as bases de imagens usadas.

Os experimentos realizados comparam os descritores do estudo teórico que apresentam qualidades importantes para um sistema de CBIR na Web, como explicado no Capítulo 3.

O objetivo dos experimentos práticos é avaliar a eficácia de cada descritor em uma base de imagens heterogêneas, assim como a eficiência para a geração do vetor de características, a eficiência no cálculo de distância e a eficiência no armazenamento de vetores de características.

Foram realizados diversos experimentos prévios ao experimento na base heterogênea. Os primeiros experimentos comparam cada tipo de descritor separadamente em bases de imagens específicas para o tipo de descritor usado. Em outras palavras, descritores de cor, textura e forma foram comparados entre si em bases de imagens específicas para cor, textura e forma, respectivamente.

A seguir são apresentadas as medidas de avaliação utilizadas para comparar os descritores nos experimentos práticos (Seção 5.1), as bases de imagens utilizadas (Seção 5.2) e os resultados obtidos (Seção 5.3).

5.1 Medidas de avaliação

Para a realização de um estudo comparativo de descritores é necessário o uso de medidas de avaliação que permitam avaliar os descritores segundo critérios pré-estabelecidos. Os critérios usados nesta dissertação são apresentados na Seção 2.4. Para avaliar os descritores em experimentos práticos, foram utilizadas medidas de tempo, armazenamento e eficácia.

5.1.1 Eficiência na extração de vetores de características e no cálculo de distância

Os tempos de extração e de cálculo de distância de cada descritor foram medidos em *segundos*, *milissegundos* ou *microssegundos*, dependendo do caso. As medições de tempo foram feitas pela ferramenta conforme explicado na Seção 4.3. A partir dos valores absolutos de tempo obtidos para cada descritor, foi feita uma análise em termos relativos, considerando um descritor como referência. Portanto, além de apresentar os tempos absolutos de cada descritor, os resultados de tempo informam também o quanto mais rápido ou o quanto mais lento um determinado descritor foi em relação ao descritor de referência.

Foi feita uma análise estatística entre os tempos médios dos descritores para verificar se a diferença de tempo medida é significativa em relação ao descritor padrão. Foi considerada significativa uma diferença de tempo maior do que o dobro do máximo desvio padrão entre os dois tempos comparados.

As medições de tempo foram feitas em um computador com 2 processadores Intel Xeon QuadCore de 2.0 GHz, 4 GB de memória RAM, 3 HDs de 500 GB ligados em RAID-5 por hardware e sistema operacional Linux. Apesar da capacidade de paralelização do computador usado, as implementações não foram ajustadas para serem paralelizadas.

5.1.2 Eficiência no armazenamento de vetores de características

As medições de armazenamento foram feitas em *bytes*, *bits* ou múltiplos deles, dependendo do caso. Baseando-se na quantidade de valores presentes no vetor de características gerado pelo descritor, foi calculado o tamanho do vetor de características considerando que o valor está em formato inteiro ou real (*float*) e que o tamanho de um valor inteiro ou real é de 4 bytes. As quantizações usadas para determinar a quantidade de valores do vetor foi baseada nos parâmetros sugeridos pelo artigo que propõe cada descritor. Assim como as medições de tempo, as medições de armazenamento foram feitas, primeiramente, em termos absolutos e, posteriormente, em termos relativos. Desta forma, os resultados de armazenamento dos descritores informam também o quanto mais compacto ou o quanto menos compacto é o vetor gerado por um determinado descritor em relação ao vetor gerado pelo descritor de referência.

5.1.3 Eficácia

A avaliação de eficácia de um descritor tem o objetivo de mensurar a capacidade do descritor de recuperar imagens relevantes. A interpretação do que é uma imagem relevante pode variar dependendo da intenção do usuário que realizou a busca. Por isso, o processo de avaliação de eficácia é bastante complicado em alguns casos. Existem basicamente duas

formas de avaliar eficácia. Uma delas é feita sobre uma base de imagens classificadas. Neste caso, a avaliação parte do pressuposto de que a base de imagens é classificada em classes de imagens relevantes entre si. Desta forma, imagens relevantes são aquelas que pertencem à mesma classe da imagem de consulta. A avaliação de eficácia neste caso pode ser feita automaticamente. Nos experimentos realizados com os descritores nas bases específicas a avaliação de eficácia foi feita desta maneira.

Na segunda forma de avaliação, não existe uma classificação da base de imagens e, portanto, não é possível a obtenção automática de medidas quantitativas de eficácia. Uma maneira de avaliar a eficácia neste caso é por meio de usuários reais. Este tipo de avaliação tem a vantagem de conseguir medições que refletem a opinião de usuários potenciais do sistema.

Uma avaliação de eficácia com usuários reais foi utilizada no experimento realizado na base de imagens heterogêneas e foi aplicada segundo as especificações apresentadas na Seção 4.4.4.

A avaliação de eficácia foi mensurada por curvas de Precision $\times$ Recall e por valores de P_{10}, P_{20} e P_{30}.

A medida Precision mede a fração de imagens relevantes retornadas em uma determinada consulta em relação ao total de imagens retornadas. Se o valor de Precision obtido para uma certa consulta for de 80%, por exemplo, sabe-se que 80% das imagens retornadas são relevantes e que 20% das imagens retornadas são irrelevantes.

A medida Recall indica a fração de imagens relevantes retornadas em uma determinada consulta em relação ao total de imagens relevantes existentes na base. Se o valor de Recall obtido para uma certa consulta for de 50%, por exemplo, sabe-se que 50% das imagens relevantes da base foram retornadas naquele conjunto de resultados. Nota-se que a medida Recall precisa conhecer previamente a quantidade de imagens relevantes existentes na base de imagens.

Uma curva de Precision $\times$ Recall indica a variação dos valores de Precision para diferentes valores de Recall. De uma maneira informal, pode-se entender que uma curva deste tipo indica o quão misturadas estão as imagens relevantes e irrelevantes nos resultados. Em um gráfico com curvas de Precision $\times$ Recall, quanto mais alta uma curva estiver, mais eficaz é um descritor.

As medidas P_{10}, P_{20} e P_{30} indicam a fração de imagens relevantes retornadas sobre o total de imagens retornadas quando este total é igual a 10, 20 e 30, respectivamente.

Considerando o cenário Web, é importante ressaltar que, pela enorme quantidade de imagens existente, apenas uma parcela ínfima será exibida aos usuários durante uma consulta. Portanto é importante que imagens relevantes estejam nas primeiras posições dos resultados. De certa maneira, considerando uma curva de Precision $\times$ Recall, isto seria equivalente a dizer que um descritor deve apresentar altos valores de Precision para

valores pequenos de Recall.

5.2 Bases de imagens

Para a realização dos experimentos práticos dos descritores, foram usadas diferentes bases de imagens. Entre elas estão três bases classificadas e uma base não classificada. As bases classificadas usadas são específicas para cada tipo de descritor e são detalhadas nas Seções 5.2.1, 5.2.2 e 5.2.3. A base heterogênea com conteúdo da Web e não classificada é detalhada na Seção 5.2.4.

5.2.1 ETH

A base *ETH* é uma base livre disponível online[1] que contém 3280 imagens coloridas divididas em 8 classes com 410 imagens cada classe. Todas as imagens possuem um único objeto em foco sobre um fundo de cor homogênea. Os objetos aparecem em diferentes posições e pontos de vista. As classes são as seguintes: maçãs, pêras, tomates, vacas, cachorros, cavalos, xícaras, carros.

A Figura 5.1 ilustra as imagens existentes em cada categoria da base ETH.

Figura 5.1: Exemplos de imagens da base ETH.

Esta base foi usada para avaliar os descritores de cor.

[1] http://www.mis.informatik.tu-darmstadt.de/Research/Projects/categorization/eth80-db.html

5.2.2 Brodatz

A base *Brodatz* [14] é uma das bases de imagens mais populares para a avaliação de descritores de textura. Como visto na Seção 3.2.2, ela é a base mais usada dentre os descritores de textura avaliados nesta dissertação, e é usada também nos experimentos realizados.

A base Brodatz é composta de 111 imagens de texturas diferentes. Como é feito na grande maioria dos artigos que usam esta base, cada imagem de textura é subdividida em células de tamanho fixo. Nos experimentos realizados, cada textura foi dividida em 16 blocos, totalizando 1776 imagens. Dessa maneira, tem-se uma base com 111 classes contendo 16 imagens cada classe.

A Figura 5.2 ilustra algumas das texturas da base Brodatz.

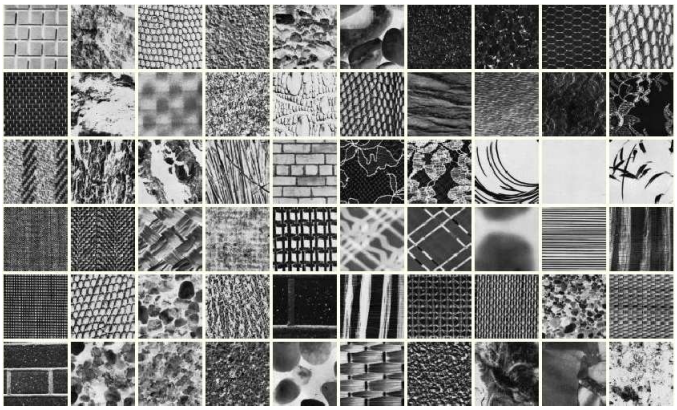


Figura 5.2: Exemplos de imagens da base Brodatz.

5.2.3 Caltech101

A base de imagens *Caltech101* [33][2] contém 8677 imagens coloridas de diversos tipos de objetos agrupadas em 101 classes. A quantidade de imagens por classe varia entre 40 e 800 imagens, tendo a maioria das classes 50 imagens aproximadamente. Cada imagem possui um objeto principal em foco sobre um fundo que pode ser homogêneo ou bastante colorido. A classificação da base é feita de acordo com o objeto principal em foco. Entre as classes de imagens da base estão: aviões, formiga, cérebro, câmera fotográfica, cadeira, elefante, panda, pizza, saxofone, girassol, etc.

A Figura 5.3 ilustra algumas das imagens da base Caltech101.

[2] http://www.vision.caltech.edu/Image_Datasets/Caltech101/Caltech101.html

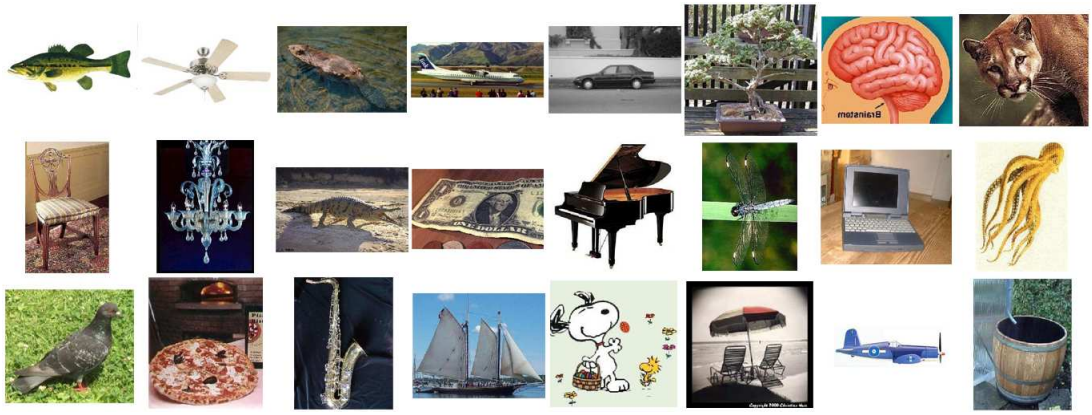


Figura 5.3: Exemplos de imagens da base Caltech101.

Como a divisão da base é baseada no objeto principal da imagem, ela considera, de uma certa maneira, a forma deste objeto para a classificação. Devido a isto, esta base de imagens foi usada para avaliar os descritores de forma.

5.2.4 Yahoo

A base de imagens *Yahoo* foi coletada por pesquisadores da Universidade Federal do Amazonas (UFAM) com o objetivo de criar uma coleção de dados representativos da Web. A coleta dos dados foi feita de maneira recursiva a partir de endereços de sites encontrados no diretório do Yahoo. Neste processo foram coletadas 254.908 imagens (excluídos ícones e banners) e 1 milhão e 200 mil documentos HTML, mantendo-se a referência entre cada HTML e cada imagem. Dessa maneira, a base de imagens gerada possui um conteúdo bastante heterogêneo e representativo do conteúdo da Web e não possui nenhum tipo de classificação.

Para a realização dos experimentos nesta base, uma rotina verificou cada arquivo de imagem nela presente e, após a eliminação de arquivos corrompidos, restaram 234.828 imagens válidas.

A Figura 5.4 ilustra algumas das imagens da base Yahoo.

Esta base foi usada para avaliar o conjunto final de descritores e usou uma avaliação de eficácia com usuários reais.

Figura 5.4: Exemplos de imagens da base Yahoo.

5.3 Resultados

A seguir são apresentados os resultados dos experimentos realizados com cada tipo de descritor assim como os resultados do conjunto final de descritores na base Yahoo.

5.3.1 Cor

Os descritores de cor implementados (Seção 3.1.3) foram testados na base ETH (Seção 5.2.1). A seguir são apresentados os resultados das medições dos tempos de extração, tempo de distância, requisitos de armazenamento e eficácia dos descritores de cor.

Eficiência na extração de vetores de características

Após aplicar cada um dos descritores de cor na base de imagens ETH, foram obtidos os tempos de extração de características de cada descritor. Como descritor de referência foi escolhido o descritor GCH, por ser o mais utilizado em comparações na literatura. A Tabela 5.1 apresenta o tempo médio de extração de 1 vetor de características de cada descritor de cor implementado, assim como o desvio padrão. É apresentado também o tempo relativo de cada descritor em relação ao descritor GCH. Os tempos relativos foram obtidos considerando o tempo médio de extração. A Tabela 5.1 está ordenada em ordem crescente de tempo.

Pelos tempos calculados, observa-se que os descritores CGCH, GCH e LCH são os mais rápidos para extrair o vetor de características, levando, em média, menos de 1 milissegundo para a extração. Os descritores Color Bitmap, BIC, CW-HSV e CM também são rápidos para extrair vetores de características, levando, em média, menos de 3 milissegundos

Tabela 5.1: Comparação entre descritores de cor: tempos de extração de 1 vetor de características absolutos e relativos ao descritor GCH. Unidade: milissegundos

Descritor	Média	Desvio padrão	Relativo ao GCH (caso médio)
CGCH	0,3	0,9	0,81
GCH	0,4	1,1	1,00
LCH	0,6	0,9	1,45
Color Bitmap	1,1	1,3	2,69
BIC	2,3	3,0	5,58
CW-HSV	2,4	0,9	6,05
CM	2,6	1,3	6,31
CCV	3,3	0,9	8,18
CSD	15,0	4,3	37,13
ACC	16,6	8,8	40,89
CW-LUV	18,7	1,4	46,21
JAC	31,4	4,1	77,58
CBC	51,0	8,2	126,01

em uma extração. Considerando os tempos relativos ao descritor GCH, nota-se que os descritores Color Bitmap, BIC, CW-HSV e CM são mais de 2 vezes mais lentos, no caso médio.

Observa-se também que os descritores CSD, ACC e CW-LUV extraem o vetor de características em tempos próximos a 15 milissegundos, sendo em torno de 40 vezes mais lentos do que o descritor GCH, na média.

O descritor JAC foi bem mais lento que o descritor GCH para extrair o vetor, levando aproximadamente 30 milissegundos na média e sendo, portanto, em torno de 77 vezes mais lento que o descritor GCH.

O descritor CBC, que usa uma abordagem baseada em regiões para a extração do vetor de características, foi o mais lento. Ele levou em torno de 50 milissegundos para extrair um vetor, o que significa ser, em média, mais de 100 vezes mais lento que o descritor GCH.

Estatisticamente, são mais lentos que o descritor GCH, apenas os descritores CCV, CW-LUV, CSD, JAC e CBC. Nenhum descritor é estatisticamente mais rápido que o descritor GCH.

Os tempos calculados apresentados na Tabela 5.1 ilustram o quanto será acrescentado ao tempo de resposta durante a realização de uma consulta na base ETH (Seção 2.4). Pode-se dizer então que o descritor CBC acrescentaria aproximadamente 50 milissegundos no tempo de resposta a uma consulta, e que a maioria dos outros descritores acrescentaria um tempo muito menor.

Eficiência no cálculo de distância

A utilização dos descritores de cor na base ETH permitiu a medição dos tempos de realização de 1 cálculo de distância de cada descritor. Os tempos médios são exibidos na Tabela 5.2, assim como o desvio padrão e o tempo relativo ao descritor GCH no caso médio. A Tabela 5.2 apresenta os descritores em ordem crescente de tempo.

Tabela 5.2: Comparação entre descritores de cor: tempos de 1 cálculo de distância absolutos e relativos ao descritor GCH. Unidade: microssegundos

Descritor	Média	Desvio padrão	Relativo ao GCH (caso médio)
CM	28,6	7,5	0,65
CW-HSV	29,9	1,4	0,68
CW-LUV	32,0	1,4	0,73
Color Bitmap	36,4	2,9	0,83
CGCH	43,2	2,0	0,98
GCH	43,8	1,7	1,00
BIC	51,4	1,6	1,17
CCV	52,1	1,8	1,19
CSD	60,7	2,0	1,38
ACC	71,8	7,1	1,64
LCH	179,4	17,7	4,09
JAC	788,8	140,0	17,99
CBC	1154,7	1470,9	26,34

Pelos tempos calculados, observa-se que os descritores CM, CW-HSV, CW-LUV e Color Bitmap são os mais rápidos no cálculo de distâncias, sendo mais rápidos que o descritor GCH inclusive, no caso médio. Os descritores BIC e CCV apresentam tempos semelhantes: em torno de 50 microssegundos para realizar 1 cálculo de distância. São, no caso médio, aproximadamente 20% mais lentos que o descritor GCH. O descritor CSD foi quase 40% mais lento que o descritor GCH e o descritor ACC realiza 1 cálculo de distância em um tempo quase 65% maior do que o descritor GCH, ambos no caso médio. Já o descritor LCH realiza 1 cálculo de distância em tempo bem maior do que o descritor GCH, sendo, em média, aproximadamente 4 vezes mais lento. Os descritores JAC e CBC foram os mais lentos para calcular distância. O descritor JAC leva quase 1 milissegundo para 1 cálculo de distância, o que significa ser, em média, quase 18 vezes mais lento que o descritor GCH. O descritor CBC leva, em média, mais de 1 milissegundo para comparar dois vetores de caraceterísticas, o que representa, em média, ser mais de 25 vezes mais lento que o descritor GCH.

Pelo estudo teórico (Capítulo 3), observa-se que todos os descritores implementados possuem função de distância com complexidade linear no tamanho do vetor de carac-

terísticas, exceto pelo descritor CBC. Os tempos obtidos ilustram, portanto, a influência do tamanho do vetor de características no tempo do cálculo de distâncias. Os descritores mais rápidos são os que geram vetores mais compactos, como os descritores CW-HSV e CW-LUV.

Estatisticamente, são mais lentos no cálculo de distâncias do que o descritor GCH os descritores BIC, CCV, CSD, ACC, LCH e JAC. Os descritores CM, CW-HSV, CW-LUV e Color Bitmap são mais rápidos estatisticamente do que o descritor GCH.

Como mencionado na Seção 2.4, o tempo do cálculo de distância apresenta grande influência no tempo de resposta a uma consulta. Pode-se estimar o tempo para os cálculos de distância em uma consulta realizada numa base qualquer contendo 10 mil imagens considerando-se os tempos médios dos descritores. Assumindo uma busca seqüencial, seriam realizados 10 mil cálculos de distância. Enquanto o descritor CM levaria 0,29 segundos para realizar os cálculos de distância, o descritor JAC levaria 7,89 segundos. O descritor GCH levaria 0,44 segundos para realizar os cálculos de distância.

Eficiência no armazenamento de vetores de características

Levando-se em conta o vetor de características gerado por cada descritor foi possível fazer uma análise comparativa entre os seus tamanhos. Considerando a configuração informada na Seção 3.1.3 para cada descritor, foram projetados a quantidade de valores existentes no vetor de características e estes foram convertidos em bytes para a análise comparativa. Os resultados apresentados na Tabela 5.3 estão ordenados em ordem crescente de tamanho do vetor.

Como os vetores gerados pelo descritor CBC possuem tamanho variável, foi feita uma média dos tamanhos dos vetores gerados na base ETH.

A análise dos tamanhos dos vetores permite projetar os impactos do custo com dispositivos de armazenamento em um sistema de CBIR. Os descritor CW-HSV e CW-LUV geram vetores bastante compactos, tendo apenas 63 e 127 bits, respectivamente. Os vetores gerados pelos descritores CM e Color Bitmap também são bastante compactos, possuindo um tamanho de apenas 19% e 24%, respectivamente, do tamanho do vetor gerado pelo descritor GCH.

Já os vetores gerados pelos descritores BIC e CCV, necessitam de um espaço 2 vezes maior do que os gerados pelo descritor GCH. O descritor CSD gera vetores 2,88 vezes maiores do que o do descritor GCH, enquanto que o descritor CBC gera vetores 3,38 vezes maiores do que os do descritor GCH na maioria das vezes.

O descritor LCH calcula um histograma de cor para cada célula, o que o faz gerar vetores maiores que os vetores gerados pelo descritor GCH uma quantidade proporcional ao número de células usadas. O descritor JAC, por calcular um auto-correlograma conjunto de mais de uma propriedade da imagem, gera vetores pouco compactos, sendo 500 vezes

Tabela 5.3: Comparação entre descritores de cor: tamanho absoluto em bytes de 1 vetor de características e relativo ao descritor GCH.

Descritor	Em bytes	Relativo ao GCH
CW-HSV	7,875	0,03
CW-LUV	15,875	0,06
CM	48	0,19
Color Bitmap	61,5	0,24
GCH	256	1,00
CGCH	256	1,00
BIC	512	2,00
CCV	512	2,00
CSD	736	2,88
CBC	864	3,38
ACC	1024	4,00
LCH	4096	16,00
JAC	128000	500,00

maiores que os gerados pelo descritor GCH.

Considerando um cenário Web, no qual se tem uma quantidade muito grande de imagens, pode-se estimar a quantidade de espaço necessário para o armazenamento dos vetores de características dos descritores. Assumindo uma base de 1 bilhão de imagens, o descritor CW-HSV, por exemplo, necessitaria de aproximadamente 7 GB de espaço para seus vetores de características e o descritor Color Bitmap precisaria de 57 GB. O descritor GCH necessitaria de 238 GB aproximadamente e o descritor CSD precisaria de 685 GB de espaço.

Eficácia

A avaliação de eficácia foi feita por meio de curvas de Precision $\times$ Recall, considerando a média dos valores de Precision de cada consulta. Todas as imagens da base ETH foram utilizadas como imagens de consulta.

As curvas são mostradas na Figura 5.5.

Pelas curvas, observa-se que os descritores com melhor eficácia para valores de 10% de Recall são os descritores BIC, CSD, JAC, ACC e Color Bitmap, nesta ordem. Eles apresentam, neste caso, valores de Precision acima de 70%. Nota-se também que os descritores CM e CW-LUV apresentam os piores valores de Precision para 10% de Recall, apresentando Precision abaixo de 60%.

Para valores de Recall de até 60% os descritores BIC, CSD, JAC, ACC e Color Bitmap se mantêm como os de melhor eficácia. Aos 60% de Recall, o descritor Color Bitmap

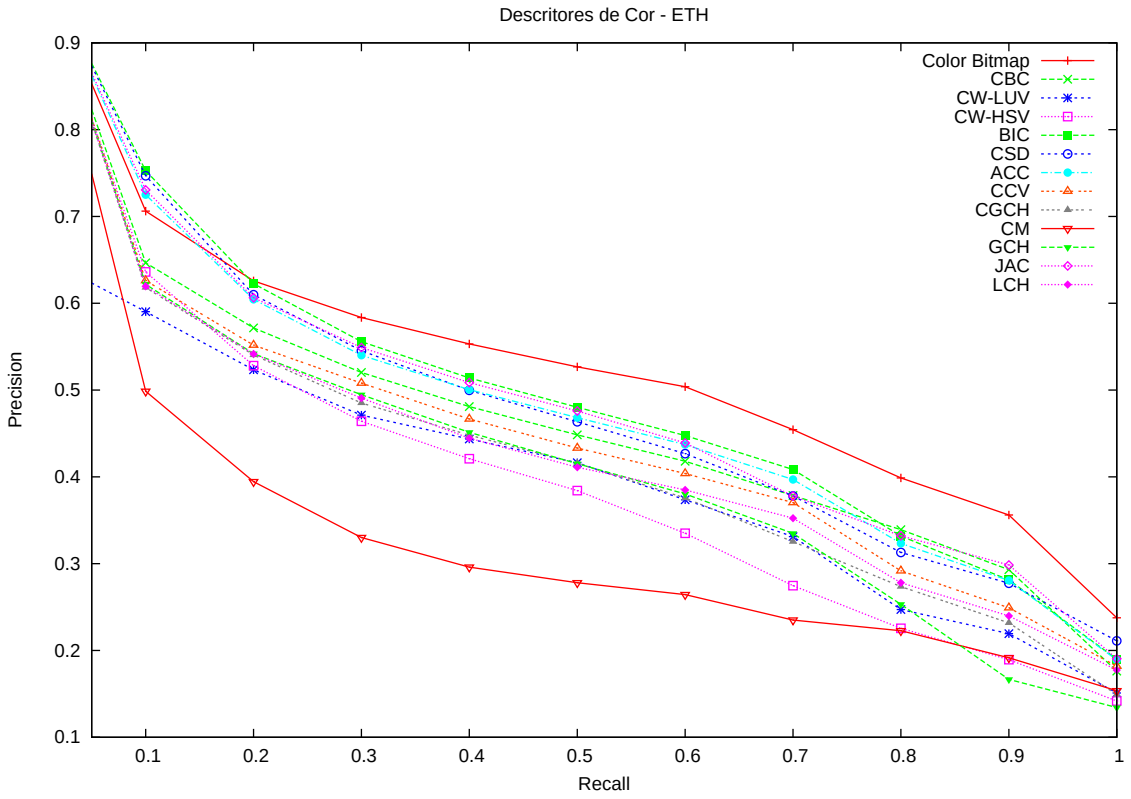


Figura 5.5: Curvas de Precision × Recall para os descritores de cor.

apresenta melhor eficácia seguido pelos descritores BIC, JAC, ACC e CSD. O descritor Color Bitmap apresenta melhor eficácia para valores de Recall a partir de 20%.

Para Recall igual a 100%, os dois melhores descritores são os descritores Color Bitmap e CSD, nesta ordem, apresentando Precision abaixo de 25%.

Conclusão

Para a realização dos experimentos na base de imagens heterogêneas, foram escolhidos 5 descritores de cor. A escolha ponderou eficácia e eficiência.

Pelo gráfico da Figura 5.5, foram escolhidos os 5 descritores com melhor eficácia: BIC, CSD, JAC, ACC e Color Bitmap. No entanto, o descritor JAC apresentou resultados ruins de eficiência, sendo lento tanto para extrair o vetor de características quanto para realizar cálculos de distância. Além disso, os vetores gerados são pouco compactos. Por essas razões, ele foi substituído pelo histograma global de cor (GCH) que, apesar de ter apresentado eficácia um pouco abaixo dos outros 5 descritores mencionados, é bastante

rápido para extrair o vetor e para calcular distâncias. Além disso, o GCH é o descritor mais popular da literatura. Os outros quatro descritores, BIC, CSD, ACC e Color Bitmap apresentaram resultados aceitáveis em termos de eficiência e do tamanho do vetor de características.

Portanto, os 5 descritores escolhidos para comparação na base Yahoo foram: BIC, CSD, ACC, Color Bitmap e GCH.

5.3.2 Textura

Os descritores de textura implementados (Seção 3.2.3) foram testados na base Brodatz (Seção 5.2.2). São apresentados a seguir os resultados das medições dos tempos de extração e de distância, das medições dos requisitos de armazenamento e da avaliação de eficácia dos descritores de textura.

Eficiência na extração de vetores de características

Os experimentos com cada um dos descritores de textura na base de imagens Brodatz possibilitaram a obtenção dos tempos médios de extração de 1 vetor de características para cada descritor. Os valores obtidos são apresentados na Tabela 5.4. Os descritores estão ordenados do mais rápido para o mais lento. Para o cálculo do tempo relativo, foi escolhido o descritor LBP como o descritor de referência, por ser um descritor bastante difundido na literatura e por ser bastante simples. O cálculo do tempo relativo foi baseado no tempo médio de cada descritor.

Tabela 5.4: Comparação entre descritores de textura: tempos de extração de 1 vetor de características absolutos e relativos ao descritor LBP. Unidade: milissegundos

Descritor	**Média**	**Desvio padrão**	**Relativo ao LBP (caso médio)**
LBP	1,7	0,4	1,00
CCOM	3,1	0,6	1,84
Unser	3,3	0,5	2,02
LAS	5,1	1,5	3,09
QCCH	21,8	1,0	13,15
SID	218,2	0,7	131,54
SASI	430,3	0,9	259,39
HTD	4892,6	49,6	2949,22

Pelos valores apresentados na Tabela 5.4, nota-se que o descritor LBP é o mais rápido entre os descritores comparados. Este descritor extrai o vetor de características em média em menos de 2 milissegundos. Os descritores CCOM e Unser extraem o vetor de características em aproximadamente 3 milissegundos, o que representa, em média, um tempo

em torno de 90% maior do que o tempo do descritor LBP. O descritor LAS também é rápido, levando em torno de 5 milissegundos para extrair um vetor de características. Isto significa ser próximo de 3 vezes mais lento que o descritor LBP no caso médio.

Já os descritores QCCH, SID, SASI e HTD foram consideravelmente mais lentos. O descritor QCCH leva aproximadamente 21 milissegundos para realizar 1 extração, sendo mais de 13 vezes mais lento que o descritor LBP no caso médio. O descritor SID e o descritor SASI levam um tempo alto para realizar a extração de 1 vetor de características, sendo, respectivamente, mais de 100 e 200 vezes mais lentos que o descritor LBP em média. O descritor HTD foi o mais lento. Seu algoritmo de extração de características leva quase 5 segundos para extrair 1 vetor de características, o que significa ser quase 3 mil vezes mais lento que o descritor LBP.

Estatisticamente, todos os descritores são mais lentos que o descritor LBP para realizar a extração de 1 vetor de características.

Estes tempos de extração indicam o quanto de acréscimo haverá ao tempo de resposta a uma consulta na base Brodatz, considerando esta consulta feita por meio de uma imagem de exemplo (ver Seção 2.4). No caso dos descritores LBP, CCOM, Unser, LAS e QCCH, este tempo é pequeno, estando na casa dos milissegundos. No entanto, no caso dos descritores SID, SASI e HTD, o tempo de resposta é muito alto. A extração do descritor SID acrescentaria aproximadamente 0,2 segundos ao tempo de resposta, a do descritor SASI acrescentaria quase meio segundo ao tempo de resposta e a extração do HTD acrescentaria quase 5 segundos.

Eficiência no cálculo de distância

Os resultados das medições dos tempos de distância são apresentados na Tabela 5.5. São exibidos o tempo médio, o desvio padrão e o tempo relativo ao descritor LBP considerando o tempo médio de 1 cálculo de distância. A Tabela 5.5 apresenta os descritores ordenados em ordem crescente de tempo.

Pela análise dos tempos de distância, pode-se notar que os descritores apresentam tempos muito próximos uns dos outros. Exceto pelo descritor CCOM, todos os outros descritores realizam 1 cálculo de distância em tempo médio aproximado de 30 microssegundos. O descritor CCOM tem tempo médio 22% maior que o tempo do descritor LBP.

Existe diferença estatisticamente significativa em relação ao descritor LBP, apenas para os descritores Unser e CCOM. O descritor Unser é mais rápido que o descritor LBP e o descritor CCOM é mais lento.

Pelo estudo teórico (Capítulo 3), observa-se que todos os descritores de textura implementados possuem função de distância com complexidade linear no tamanho do vetor de características. Dessa maneira, a diferença existente entre os tempos de distância é

Tabela 5.5: Comparação entre descritores de textura: tempos de 1 cálculo de distância absolutos e relativos ao descritor LBP. Unidade: microssegundos

Descritor	Média	Desvio padrão	Relativo ao LBP (caso médio)
Unser	31,3	1,3	0,90
SID	31,9	1,4	0,92
HTD	31,9	1,5	0,92
QCCH	32,2	1,6	0,93
LAS	32,8	1,3	0,95
SASI	33,9	1,0	0,98
LBP	34,6	1,3	1,00
CCOM	42,2	1,9	1,22

principalmente resultado do tamanho do vetor de características.

A Seção 2.4 define a importância do tempo do cálculo de distância no tempo de resposta a uma consulta. Simulando-se uma consulta de maneira seqüencial em uma base com 10 mil imagens, seriam 10 mil cálculos de distância. O descritor Unser levaria, na média, 0,31 segundos para calcular as distâncias enquanto que o descritor CCOM levaria aproximadamente 0,42 segundos. O descritor LBP levaria em média 0,35 segundos.

Eficiência no armazenamento de vetores de características

A análise comparativa entre os tamanhos dos vetores de características foi feita considerando a configuração informada na Seção 3.2.3. Foram projetados a quantidade de valores existentes no vetor de características e estes foram convertidos em bytes para a análise comparativa. Os resultados são apresentados na Tabela 5.6, ordenados em ordem crescente de tamanho do vetor.

Tabela 5.6: Comparação entre descritores de textura: tamanho absoluto em bytes de 1 vetor de características e relativo ao descritor LBP.

Descritor	Em bytes	Relativo ao LBP
LBP	40	1,00
SID	64	1,60
Unser	128	3,20
QCCH	160	4,00
HTD	192	4,80
CCOM	248	6,20
SASI	256	6,40
LAS	1024	25,60

Os vetores gerados pelo descritor CCOM possuem tamanho variável. Os valores exi-

bidos na Tabela 5.6 ilustram a média destes tamanhos detectados na base Brodatz.

Os dados da Tabela 5.6 mostram que o descritor LBP gera os vetores de características mais compactos. O descritor SID gera vetores 60% maiores do que os gerados pelo LBP. O descritor Unser gera um vetor com mais de 3 vezes o tamanho do vetor gerado pelo descritor LBP, enquanto que o descritor QCCH gera um vetor que ocupa 4 vezes mais espaço. Existe uma grande diferença entre o tamanho do vetor gerado pelo descritor LAS e os tamanhos dos vetores gerados pelos outros descritores. O LAS gera um vetor que ocupa mais de 25 vezes o espaço ocupado por um vetor gerado pelo descritor LBP.

Se for considerado um cenário Web, no qual uma quantidade muito grande de imagens existe, as diferenças de tamanho dos vetores ficam mais explícitas. Assumindo uma base de 1 billhão de imagens, o descritor LBP, por exemplo, necessitaria de aproximadamente 37 GB de espaço para seus vetores de características e o descritor HTD precisaria de 179 GB. O descritor SASI necessitaria de 238 GB aproximadamente e o descritor LAS precisaria de 954 GB de espaço.

Eficácia

Para a avaliação de eficácia, todas as imagens da base Brodatz foram usadas como imagens de consulta. As curvas de Precision $\times$ Recall apresentadas na Figura 5.6 representam os valores médios para todas as consultas realizadas.

As curvas indicam que os descritores LAS e SASI são bastante superiores em eficácia em relação aos outros descritores. Para Recall igual a 10%, o descritor LAS tem diferença favorável em relação ao descritor CCOM de aproximadamente 7% no valor de Precision. Para Recall até 30%, a melhor eficácia é a do descritor LAS, seguido pelo descritor SASI e pelo descritor CCOM. O descritor SASI tem a melhor eficácia para Recall acima de 40%.

Os descritores LAS e SASI recuperam todas as imagens relevantes da base (Recall igual a 100%) mantendo uma precisão acima de 45%. Enquanto isso, os outros descritores o fazem com precisão abaixo de 30%.

Conclusão

Pelos resultados apresentados para os tempos de extração e distância, armazenamento e eficácia de cada descritor de textura, foi possível escolher alguns deles para testes na base Yahoo.

O critério de escolha ponderou eficácia e eficiência.

Baseando-se nas curvas de Precision $\times$ Recall, os descritores com melhor eficácia seriam os melhores candidatos para testes na base Yahoo. As melhores eficácias foram as dos descritores LAS, SASI e CCOM. O descritor LAS é rápido para extrair características e para calcular distâncias, mas gera vetores de características maiores do que os dos outros

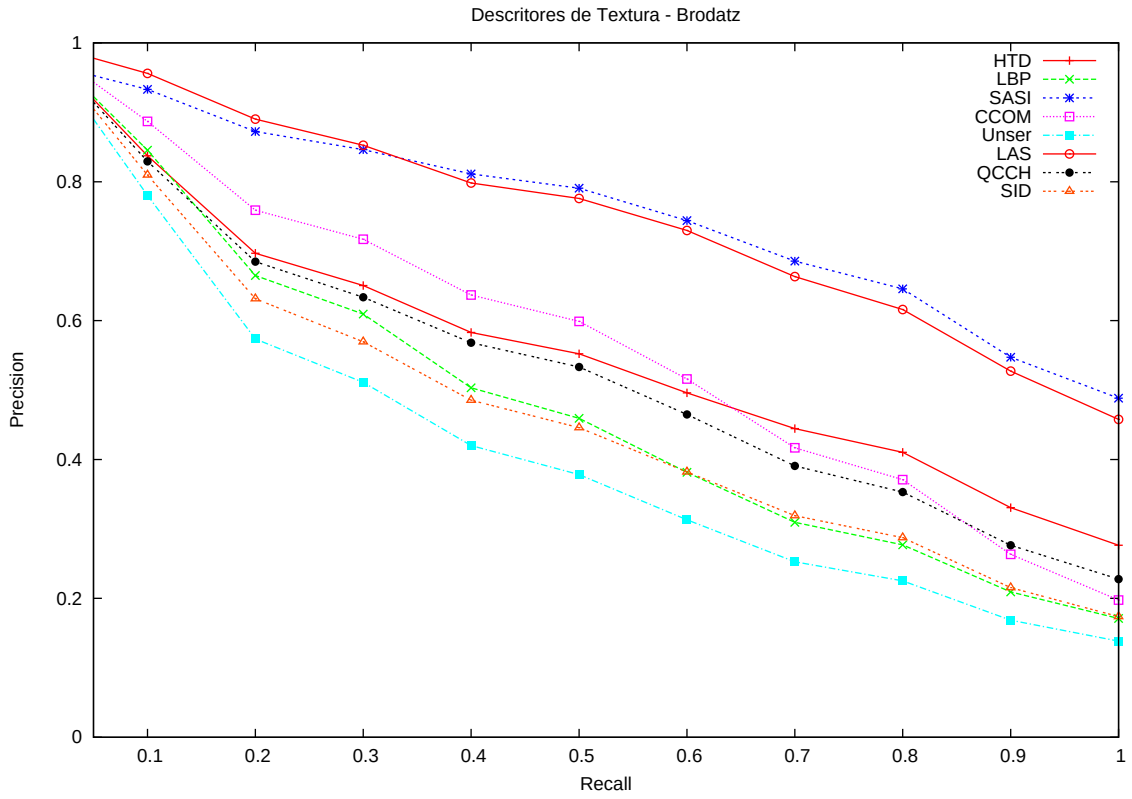


Figura 5.6: Curvas de Precision × Recall para os descritores de textura.

descritores. O descritor SASI é rápido nos cálculos de distância e gera vetores um pouco menos compactos que a maioria dos outros descritores. No entanto, é muito lento para extrair as características de uma imagem. Sua extração adiciona quase meio segundo no tempo de resposta a uma consulta. Por esse motivo, ele não foi usado na base Yahoo. O descritor CCOM apresenta boa eficiência para extração de características e cálculos de distância e não gerou vetores muito grandes na maioria dos casos na base Brodatz.

Para substituir o descritor SASI foi escolhido o descritor LBP. Entre as razões, estão sua boa eficácia para Recall até 10% e sua simplicidade na extração de características e cálculos de distância.

Portanto, os 3 descritores de textura escolhidos para testes na base Yahoo são: LAS, CCOM e LBP

5.3.3 Forma

Os descritores de forma apresentados na Seção 3.3.3 foram testados na base Caltech101 (Seção 5.2.3). A seguir são exibidos os resultados das medições dos tempos de extração e de distância, das medições dos requisitos de armazenamento e da avaliação de eficácia dos descritores de forma.

Eficiência na extração de vetores de características

Foi calculado o tempo médio de extração de 1 vetor de características para cada descritor quando aplicado na base Caltech101. Os resultados estão na Tabela 5.7 que também mostra o desvio padrão e o tempo relativo entre os descritores implementados, considerando o descritor EOAC como referência.

Tabela 5.7: Comparação entre descritores de forma: tempos de extração de 1 vetor de características absolutos e relativos ao descritor EOAC. Unidade: milissegundos

Descritor	**Média**	**Desvio padrão**	**Relativo ao EOAC (caso médio)**
Spytec	39,6	18,5	0,88
EOAC	45,2	20,4	1,00

Os tempos calculados ilustram uma pequena diferença de tempo médio entre os descritores EOAC e Spytec. Estatisticamente um não pode ser considerado mais rápido do que o outro.

Conforme explicado na Seção 2.4, o tempo de extração contribui para um aumento no tempo de resposta a uma consulta, quando esta é feita por meio de uma imagem de exemplo. Considerando este cenário e o tempo médio de cada descritor, o acréscimo ao tempo de resposta seria de aproximadamente 40 milissegundos utilizando-se os descritores EOAC e Spytec na base Caltech101.

Eficiência no cálculo de distância

A Tabela 5.8 apresenta os resultados das medições dos tempos dos cálculos de distância de cada descritor. São mostrados o tempo médio de 1 cálculo de distância, o desvio padrão e o tempo relativo entre os descritores considerando o tempo médio e o descritor EOAC como padrão.

Os tempos de distância dos descritores de forma indicam valores muito próximos. Estatisticamente, um não pode ser considerado mais rápido que o outro. Os tempos calculados mostram que os dois descritores são bastante rápidos para a realização de 1 cálculo de distância, fazendo-o em tempo aproximado de 30 microssegundos.

Tabela 5.8: Comparação entre descritores de forma: tempos de 1 cálculo de distância absolutos e relativos ao descritor EOAC. Unidade: microssegundos

Descritor	Média	Desvio padrão	Relativo ao EOAC (caso médio)
Spytec	30,4	1,8	0,91
EOAC	33,5	2,1	1,00

Como mencionado na Seção 2.4, os cálculos de distância são um dos principais responsáveis pelo tempo de reposta a uma consulta. O tempo estimado para a realização dos cálculos de distância em uma base com 10 mil imagens seria de 0,3 segundos para o descritor Spytec e de 0,34 segundos para o descritor EOAC, considerando uma busca seqüencial.

Eficiência no armazenamento de vetores de características

Considerando os parâmetros sugeridos na implementação de cada descritor (Seção 3.3.3), foi projetada a quantidade de valores que cada vetor de características possui e assim foi calculado o tamanho de 1 vetor em bytes. Os resultados estão na Tabela 5.9, que também apresenta o tamanho relativo ao vetor gerado pelo descritor EOAC.

Tabela 5.9: Comparação entre descritores de forma: tamanho absoluto em bytes de 1 vetor de características e relativo ao descritor EOAC.

Descritor	Em bytes	Relativo ao EOAC
Spytec	64	0,06
EOAC	1152	1,00

Os valores calculados indicam um vetor bem mais compacto gerado pelo descritor Spytec. Com 64 bytes, o vetor tem um tamanho de apenas 6% o tamanho do vetor gerado pelo descritor EOAC.

Pode-se estimar o espaço necessário para armazenamento de vetores de características considerando um cenário Web. Se for considerada uma base com 1 bilhão de imagens, os vetores de características gerados pelo descritor Spytec usariam aproximadamente 60 GB de espaço, enquanto que os vetores gerados pelo descritor EOAC usariam pouco mais que 1 TB de espaço.

Eficácia

Para a avaliação de eficácia dos descritores de forma, foram feitas 1010 consultas na base Caltech101. Como a base é dividida em 101 classes, foram escolhidas 10 imagens de cada classe para serem consultas.

As curvas de Precision × Recall obtidas são apresentadas na Figura 5.7 e foram obtidas pelos valores médios entre todas as consultas realizadas.

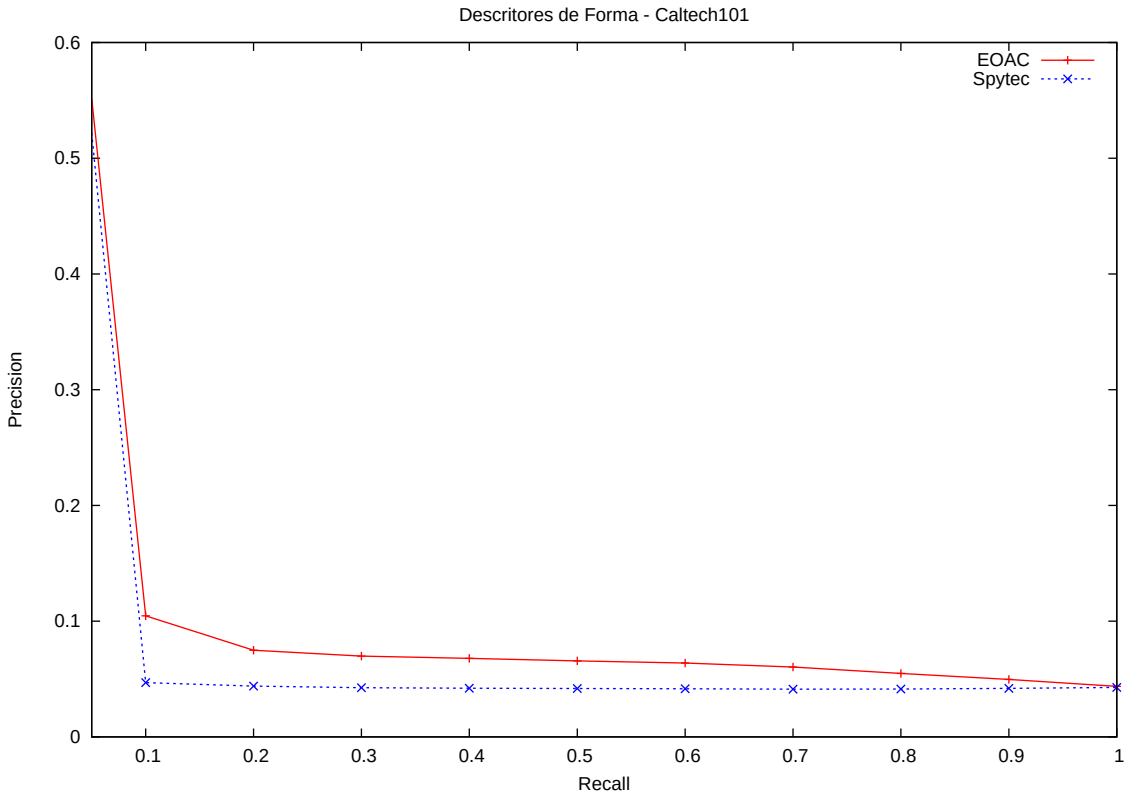


Figura 5.7: Curvas de Precision × Recall para os descritores de forma.

As curvas indicam uma eficácia muito baixa para os dois descritores. Para Recall 10%, o descritor EOAC teve Precision de 10,5% e o descritor Spytec teve Precision de 4,7%.

O descritor EOAC teve eficácia um pouco mais alta do que o descritor Spytec, mesmo assim o Precision ficou abaixo de 10% para todos os valores de Recall acima de 10%.

Devido aos resultados ruins com os descritores de forma, foram avaliados empiricamente alguns resultados de consultas. As Figuras 5.8, 5.9, 5.10 e 5.11 ilustram as 20 primeiras imagens retornadas para 4 consultas diferentes usando os descritores EOAC e Spytec.

Pelos resultados de consultas, observa-se que para algumas imagens de consulta os resultados parecem ser quase aleatórios, como nas Figuras 5.8 e 5.9. Entretanto, no casos de consultas como "motorbike1" e "leopard8" (Figuras 5.10 e 5.11), os descritores tiveram uma eficácia melhor. Em especial, a precisão do descritor EOAC para a consulta "motorbike1" foi praticamente 100% (Figura 5.10 (a)).

(a)EOAC (b)Spytec

Figura 5.8: 20 primeiras imagens retornadas para a consulta “elephant5” usando os descritores de forma.

Conclusões

Pelos resultados observados nos experimentos dos descritores de forma EOAC e Spytec na base Caltech101, observa-se que os tempos de extração e os tempos de distância são factíveis para um sistema de CBIR na Web.

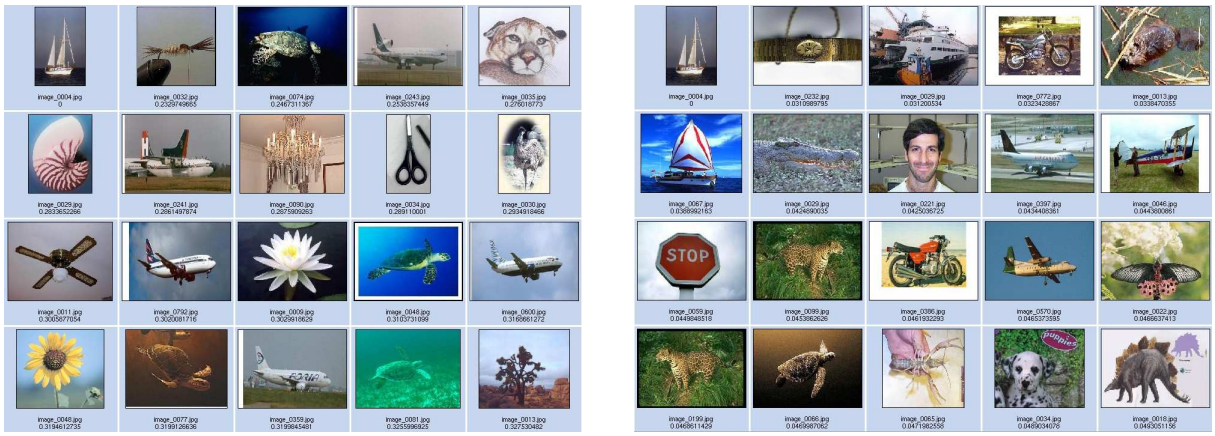
Os vetores de características gerados diferem bastante de um descritor para outro. O descritor Spytec gera vetores bem mais compactos que os gerados pelo descritor EOAC. O tamanho do vetor do descritor EOAC é 18 vezes maior do que o do descritor Spytec.

A análise de eficácia mostrou que os descritores comparados apresentam deficiências para recuperar imagens relevantes na base Caltech101. Entre as possíveis razões estão a influência do fundo da imagem sobre o objeto de interesse.

Como num ambiente Web as imagens são bastante heterogêneas e os objetos podem estar muitas vezes bem misturados com o fundo, assim como acontece em muitas imagens da base Caltech101, a eficácia dos descritores de forma analisados pode ser muito baixa. Por estes motivos, nenhum dos descritores de forma avaliados foi usado nos experimentos na base Yahoo.

5.3.4 Avaliação na base Yahoo

Nesta seção são apresentados os resultados da avaliação dos descritores na base de imagens heterogêneas Yahoo. Após os experimentos com cada tipo de descritor em bases específicas, foi possível selecionar um conjunto de descritores que obtiveram bom desempenho para serem avaliados na base de imagens heterogêneas. Os descritores selecionados



(a)*EOAC* (b)*Spytec*

Figura 5.9: 20 primeiras imagens retornadas para a consulta "ketch4" usando os descritores de forma.

são: BIC, CSD, ACC, Color Bitmap, GCH, LAS, CCOM e LBP. Destes, os 5 primeiros são descritores de cor e os 3 últimos são descritores de textura.

Os experimentos na base Yahoo possibilitaram medir os tempos de extração do vetor de características, os tempos para cálculos de distância, os requisitos de armazenamento e a eficácia de cada descritor.

Eficiência na extração de vetores de características

A Tabela 5.10 apresenta os tempos médios de extração de 1 vetor de características, o desvio padrão e o tempo relativo ao descritor GCH no caso médio.

Tabela 5.10: Comparação entre descritores na base Yahoo: tempos de extração de 1 vetor de características absolutos e relativos ao descritor GCH. Unidade: milissegundos

Descritor	Média	Desvio padrão	Relativo ao GCH (caso médio)
GCH	5,6	28,3	1,00
Color Bitmap	8,9	33,1	1,58
LBP	18,6	45,9	3,31
BIC	20,6	48,6	3,67
CCOM	30,9	60,4	5,51
LAS	55,2	93,3	9,82
CSD	82,3	72,9	14,65
ACC	108,0	165,4	19,23

(a)EOAC (b)Spytec

Figura 5.10: 20 primeiras imagens retornadas para a consulta “motorbike1” usando os descritores de forma.

Pode-se observar que os tempos medidos são maiores que os tempos medidos nas bases individuais. O motivo é que a base Yahoo possui imagens de variados tamanhos enquanto que as bases específicas possuem apenas imagens de tamanhos pequenos e quase sempre normalizados entre todas as imagens da base. Essa variação de tamanho também contribuiu para um maior desvio padrão.

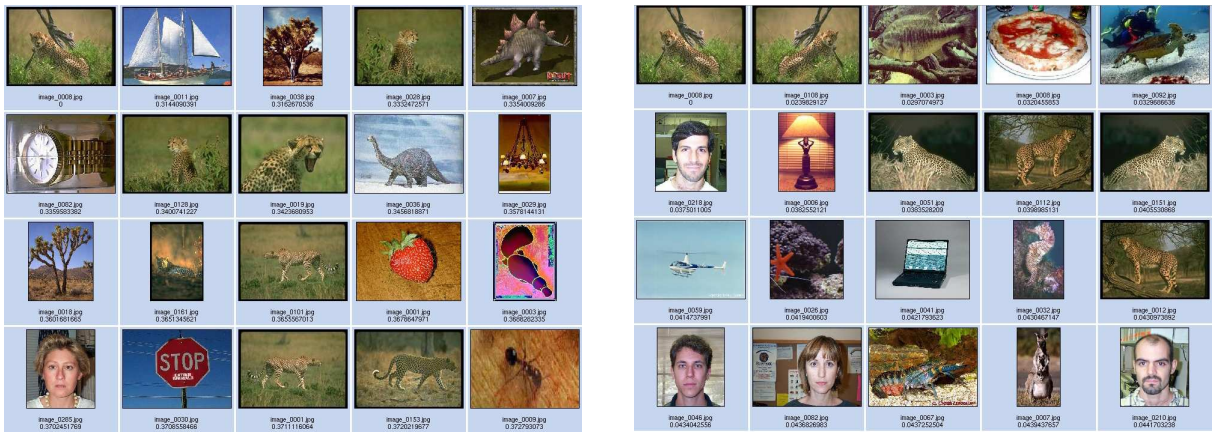
Apesar de estatisticamente não existir um descritor mais rápido ou mais lento do que outro na Tabela 5.10, existem diferenças entre os tempos médios.

O descritor Color Bitmap é o mais rápido para extrair o vetor de características depois do descritor GCH. Na média, ele é aproximadamente 58% mais lento que o descritor GCH. Os descritores LBP e BIC extraem o vetor de características gastando mais de 3 vezes o tempo gasto pelo descritor GCH, no caso médio. Além destes, os descritores CCOM, LAS e CSD também extraem o vetor em tempo médio abaixo de 100 milissegundos. O único descritor com tempo acima de 100 milissegundos é o descritor ACC.

Eficiência no cálculo de distância

A Tabela 5.11 apresenta os tempos médios de 1 cálculo de distância de cada descritor, assim como o desvio padrão e o tempo relativo ao descritor GCH no caso médio. Os descritores estão ordenados em ordem crescente de tempo.

Pelos valores obtidos na base Yahoo, vê-se que, na média, os descritores LAS, LBP e Color Bitmap são os mais rápidos, sendo no caso médio mais rápidos que o descritor GCH. No caso médio, o descritor BIC é 20% mais lento que o descritor GCH, enquanto



(a)*EOAC* (b)*Spytec*

Figura 5.11: 20 primeiras imagens retornadas para a consulta “leopard8” usando os descritores de forma.

que o descritor CSD é 42% mais lento. O descritor ACC foi 59% mais lento que o descritor GCH no caso médio.

Entre todos os descritores, o descritor CCOM foi o mais lento. Em comparação com o descritor LBP, o descritor CCOM havia sido 22% mais lento no caso médio nos experimentos na base Brodatz. No entanto, na base Yahoo, a diferença aumentou para 14 vezes aproximadamente. Enquanto o tempo do descritor LBP permaneceu estatisticamente igual nas duas bases, o tempo do descritor CCOM aumentou em mais de 12 vezes. Um dos motivos é que os vetores gerados pelo descritor CCOM podem variar de tamanho dependendo da imagem de entrada. Portanto, na base Yahoo foram gerados vetores maiores (ver comparação de armazenamento a seguir) o que impactou no tempo de distância

Tabela 5.11: Comparação entre descritores na base Yahoo: tempos de 1 cálculo de distância absolutos e relativos ao descritor GCH. Unidade: microssegundos

Descritor	**Média**	**Desvio padrão**	**Relativo ao GCH (caso médio)**
LAS	34,9	5,7	0,77
LBP	37,7	5,5	0,84
Color Bitmap	38,8	4,3	0,86
GCH	45,1	4,7	1,00
BIC	54,2	33,3	1,20
CSD	63,8	4,8	1,42
ACC	71,9	13,6	1,59
CCOM	539,9	246,3	11,97

do descritor. Além disso, as imagens da base Brodatz não são coloridas ao contrário das imagens da base Yahoo. Isto influenciou o tamanho do vetor gerado, já que o descritor CCOM considera as cores para a extração do vetor.

Exceto pelo descritor CCOM, todos os outros descritores realizam 1 cálculo de distância em menos de 100 microssegundos.

As diferenças de tempos apresentadas nas Tabelas 5.11 indicam que, estatisticamente, nenhum dos descritores pode ser considerado mais rápido do que o descritor GCH. Estatisticamente, apenas os descritores CSD e CCOM podem ser considerados mais lentos do que o descritor GCH.

Eficiência no armazenamento de vetores de características

A Tabela 5.12 mostra os tamanhos dos vetores de características gerados por cada descritor. Os tamanhos considerados seguem os mesmos padrões estabelecidos nos experimentos dos descritores em suas bases específicas (Seções 5.3.1 e 5.3.2). Os tamanho são exibidos em bytes e em valores relativos ao tamanho do vetor gerado pelo descritor GCH. Os vetores gerados pelo descritor CCOM possuem tamanho variável dependente da imagem de entrada, portanto, o valor exibido na Tabela 5.12 é a média dos tamanhos dos vetores gerados na base Yahoo.

Tabela 5.12: Comparação entre descritores na base Yahoo: tamanho absoluto em bytes de 1 vetor de características e relativo ao descritor GCH.

Descritor	Em bytes	Relativo ao GCH
LBP	40	0,16
Color Bitmap	61,5	0,24
GCH	256	1,00
BIC	512	2,00
CSD	736	2,88
ACC	1024	4,00
LAS	1024	4,00
CCOM	11740	45,86

Os descritores LBP e Color Bitmap são os únicos que geram vetores mais compactos que os vetores gerados pelo descritor GCH. O tamanho do vetor gerado pelo descritor LBP equivale a apenas 16% do tamanho do vetor gerado pelo descritor GCH, enquanto que o do descritor Color Bitmap equivale a 24%.

O vetor do descritor BIC tem o dobro do tamanho do vetor do descritor GCH e o tamanho do vetor gerado pelo descritor CSD é 2,88 vezes maior que o gerado pelo GCH. Os descritores ACC e LAS geram vetores 4 vezes maiores que o gerado pelo descritor GCH. O descritor CCOM gera vetores de tamanhos variados, mas considerando o tamanho médio

dos vetores gerados para a base Yahoo, observa-se que eles são aproximadamente 46 vezes maiores que os gerados pelos descritor GCH.

Eficácia

A avaliação de eficácia dos descritores na base Yahoo foi feita com usuários reais. O principal motivo para a realização deste tipo de avaliação é devido à base Yahoo não ser classificada. Além disso, uma avaliação de eficácia com usuários reais consegue captar a eficácia dos descritores em um cenário real de uso, no qual são avaliados por potenciais usuários do sistema.

Para a avaliação, foram escolhidas 16 imagens de consulta. A escolha destas imagens foi feita procurando-se considerar categorias diferentes de imagens. Procurou-se também considerar imagens com uma semântica bem definida. As imagens de consulta escolhidas são exibidas na Figura 5.12.

Figura 5.12: Imagens de consulta escolhidas para a avaliação de eficácia com usuários reais na base Yahoo.

A avaliação com usuários foi configurada conforme explicado na Seção 4.4.4.

A avaliação foi feita por 15 usuários. Para o cálculo das medidas, foram consideradas apenas avaliações efetuadas por completo, ou seja, foram desconsiderados usuários que não avaliaram todas as 16 imagens de consulta. As medidas de P_{10}, P_{20} e P_{30} foram calculadas para cada imagem de consulta e cada descritor.

A Tabela 5.13 apresenta os valores de P_{10}, P_{20} e P_{30} médios entre todas as imagens de consulta para cada descritor. Os valores indicam que o descritor de melhor eficácia geral é o descritor BIC. Ele apresenta os maiores valores de P_{10}, P_{20} e P_{30}. O descritor ACC obteve eficácia bastante próxima à do descritor BIC, ficando abaixo 4% no valor de P_{10}, 3% no valor de P_{20} e 2% no valor de P_{30} em relação ao descritor BIC.

Pelos resultados gerais, o descritor CCOM apresentou a pior eficácia em quase todas as consultas. O descritor LBP também teve eficácia abaixo da maioria dos outros descritores em muitas das consultas realizadas.

Os resultados obtidos indicam que a eficácia geral dos descritores é baixa na base heterogênea Yahoo. O descritor com melhor resultado geral consegue recuperar, em média, apenas 3 imagens relevantes num conjunto de 10 resultados.

Apesar do descritor BIC ter sido o melhor em eficácia média geral, ele não apresentou a melhor eficácia em todas as consultas realizadas. As Tabelas 5.14 a 5.29 apresentam os valores de P_{10}, P_{20} e P_{30} de cada imagem de consulta para cada descritor.

Pelos valores exibidos nas tabelas, nota-se que, na grande maioria dos casos, há uma queda na eficácia dos descritores quando o conjunto de imagens retornadas passa de 10 para 20 e de 20 para 30 imagens. Isto significa que, ao se recuperar mais imagens para os resultados, são recuperadas mais imagens irrelevantes do que imagens relevantes.

Para algumas consultas certos descritores apresentaram alta eficácia. Nas consultas 9, 10, 11 e 14 alguns descritores (BIC, ACC, GCH e LAS) obtiveram P_{10} acima de 50%. Para a consulta 10, os descritores ACC e BIC obtiveram P_{10} acima de 70%.

Algumas peculiaridades dos descritores podem ser observadas de acordo com as características das imagens de consulta. As consultas 7 e 9 possuem um cachorro com fundo heterogêneo e um cachorro com fundo homogêneo, respectivamente. A eficácia da maioria dos descritores foi maior para a consulta com fundo homogêneo. Isto ilustra a influência do fundo da imagem no desempenho dos descritores. No caso da consulta 10, na qual os descritores tiveram melhor eficácia, o objeto principal da imagem está bem destacado do fundo.

Observa-se uma grande disparidade de eficácia nos resultados para a consulta 14, que representa a imagem com o pôr-do-sol. O descritor LAS foi destacadamente superior a todos os outros descritores para os três valores de precisão. No caso de P_{10}, a diferença dele para o segundo melhor descritor, o descritor Color Bitmap, foi de 44%. A diferença se manteve próxima de 40% para os valores de P_{20} e P_{30}.

Na consulta 6, que contém o carro vermelho, o descritor LAS obteve uma diferença favorável de 10% de P_{20} e de 9% de P_{30} em relação ao segundo melhor descritor na mesma consulta.

Na consulta 11, que exibe o carro de corrida, também houve uma diferença grande de eficácia em favor do descritor BIC em relação aos outros descritores. Os valores de P_{10} e de P_{30} do descritor BIC ficaram 16% acima dos mesmos valores do segundo melhor descritor na consulta. No caso do valor de P_{20}, a diferença foi de 23% em relação ao descritor ACC. O valor de P_{10} do descritor BIC na consulta 3 foi bem mais alto em relação aos outros descritores. A diferença foi igual a 18% em relação ao segundo melhor descritor nesta consulta.

No geral, os descritores de cor apresentaram melhor desempenho do que os descritores de textura. Em consultas nas quais a informação de cor é fundamental para a caracterização da semântica da imagem, os descritores de cor foram superiores. Por exemplo, nas

consultas 9 e 12, que possuem informação de cor fundamental para caracterização do objeto em foco, os descritores de textura tiveram eficácia abaixo da maioria dos descritores de cor.

Tabela 5.13: Médias gerais dos valores de P_{10}, P_{20} e P_{30} para todas as consultas na base Yahoo.

Descritor	$\mathbf{P_{10}}$
BIC	0,31
ACC	0,27
GCH	0,25
LAS	0,23
CSD	0,18
Color Bitmap	0,17
LBP	0,14
CCOM	0,10

Descritor	$\mathbf{P_{20}}$
BIC	0,21
ACC	0,18
GCH	0,17
LAS	0,16
Color Bitmap	0,12
CSD	0,12
LBP	0,08
CCOM	0,05

Descritor	$\mathbf{P_{30}}$
BIC	0,17
ACC	0,15
LAS	0,13
GCH	0,13
Color Bitmap	0,10
CSD	0,09
LBP	0,06
CCOM	0,03

Tabela 5.14: Médias dos valores de P_{10}, P_{20} e P_{30} para a consulta 1.

Descritor	$\mathbf{P_{10}}$
BIC	0,39
ACC	0,35
GCH	0,23
CSD	0,18
LAS	0,17
Color Bitmap	0,14
LBP	0,13
CCOM	0,09

Descritor	$\mathbf{P_{20}}$
BIC	0,22
ACC	0,19
GCH	0,14
Color Bitmap	0,14
LAS	0,13
LBP	0,11
CSD	0,10
CCOM	0,05

Descritor	$\mathbf{P_{30}}$
BIC	0,17
ACC	0,16
Color Bitmap	0,12
GCH	0,12
LAS	0,11
CSD	0,09
LBP	0,08
CCOM	0,03

Os resultados das consultas realizadas na base Yahoo com cada descritor podem ser vistos no endereço *http://www.lis.ic.unicamp.br/~otavio/cbir_eval/view503.html*.

A interface de avaliação com usuários pode ser acessada no endereço *http://www.lis.ic.unicamp.br/~otavio/cbir_eval/experiment503.html*.

Conclusão

As medidas de eficiência indicaram pequena variação em relação aos experimentos específicos. Os tempos de extração de vetores de características aumentaram em relação aos tempos calculados nos experimentos específicos. Uma razão é a existência de imagens de dimensões maiores na base Yahoo. Os tempos de distância permaneceram semelhantes aos tempos medidos nos experimentos específicos, devido a maioria dos descritores

Tabela 5.15: Médias dos valores de P_{10}, P_{20} e P_{30} para a consulta 2.

Descritor	$\mathbf{P_{10}}$	**Descritor**	$\mathbf{P_{20}}$	**Descritor**	$\mathbf{P_{30}}$
BIC	0,15	Color Bitmap	0,09	Color Bitmap	0,07
ACC	0,15	BIC	0,09	ACC	0,07
Color Bitmap	0,15	CSD	0,08	BIC	0,07
GCH	0,11	ACC	0,08	CSD	0,06
CSD	0,10	GCH	0,05	GCH	0,04
LBP	0,09	LBP	0,05	LBP	0,03
LAS	0,09	CCOM	0,05	CCOM	0,03
CCOM	0,09	LAS	0,04	LAS	0,03

Tabela 5.16: Médias dos valores de P_{10}, P_{20} e P_{30} para a consulta 3.

Descritor	$\mathbf{P_{10}}$	**Descritor**	$\mathbf{P_{20}}$	**Descritor**	$\mathbf{P_{30}}$
BIC	0,45	BIC	0,31	BIC	0,24
LAS	0,27	GCH	0,25	GCH	0,21
GCH	0,24	Color Bitmap	0,23	Color Bitmap	0,19
CSD	0,19	LAS	0,14	LAS	0,11
ACC	0,13	ACC	0,13	ACC	0,09
Color Bitmap	0,13	CSD	0,10	CSD	0,07
LBP	0,12	LBP	0,08	LBP	0,06
CCOM	0,10	CCOM	0,05	CCOM	0,03

gerar vetores de características de tamanhos constantes e independentes da imagem de entrada, exceto pelo descritor CCOM. No caso dele, a variação no tamanho do vetor de características fez com que seus tempos de distância aumentassem consideravelmente.

A eficácia dos descritores na base Yahoo foi pior do que quando os descritores foram testados em suas bases específicas. A heterogeneidade prejudicou o desempenho dos descritores. O descritor Color Bitmap, por exemplo, obteve uma das melhores eficácias na base ETH. No entanto, sua eficácia caiu bastante na base Yahoo. Isto indica que possivelmente a heterogeneidade das imagens prejudicou seu desempenho. O descritor de textura CCOM também foi bastante afetado pelo mesmo motivo. Na base Brodatz sua precisão ficou acima de 50% em muitos casos, entretanto, na base Yahoo, a precisão ficou abaixo de 10% para a maioria das consultas.

A existência de imagens com cores semelhantes e semântica diferente fez com que o desempenho dos descritores de cor fosse degradado. A existência de diversas texturas presentes numa mesma imagem ou de texturas indefinidas prejudicou o desempenho dos descritores de textura.

Além disso, notou-se a influência do fundo das imagens no desempenho dos descritores. Um fundo heterogêneo com cores e formas que se misturam com o objeto principal

Tabela 5.17: Médias dos valores de P_{10}, P_{20} e P_{30} para a consulta 4.

Descritor	$\mathbf{P_{10}}$	**Descritor**	$\mathbf{P_{20}}$	**Descritor**	$\mathbf{P_{30}}$
BIC	0,40	BIC	0,37	BIC	0,35
ACC	0,32	CSD	0,29	CSD	0,22
LAS	0,30	LAS	0,28	GCH	0,20
Color Bitmap	0,29	GCH	0,21	LAS	0,20
GCH	0,23	Color Bitmap	0,20	Color Bitmap	0,18
CSD	0,22	ACC	0,18	ACC	0,18
LBP	0,09	LBP	0,06	LBP	0,04
CCOM	0,07	CCOM	0,04	CCOM	0,04

Tabela 5.18: Médias dos valores de P_{10}, P_{20} e P_{30} para a consulta 5.

Descritor	$\mathbf{P_{10}}$	**Descritor**	$\mathbf{P_{20}}$	**Descritor**	$\mathbf{P_{30}}$
GCH	0,27	CSD	0,22	CSD	0,18
CSD	0,25	BIC	0,18	ACC	0,16
BIC	0,19	GCH	0,14	GCH	0,14
ACC	0,13	ACC	0,13	BIC	0,14
LAS	0,11	LAS	0,07	LAS	0,06
LBP	0,10	LBP	0,06	LBP	0,04
CCOM	0,09	Color Bitmap	0,05	Color Bitmap	0,04
Color Bitmap	0,09	CCOM	0,04	CCOM	0,03

da imagem de consulta influenciou na geração dos valores contidos nos vetores de características das imagens. Enquanto que a visão humana praticamente ignora o fundo das imagens e se concentra na semântica do objeto principal, os algoritmos de extração de vetores de características não diferenciam, na maioria dos casos, o que está em evidência no conteúdo de uma imagem. A avaliação de eficácia com usuários reais possibilitou observar este fenômeno. No caso de fundos homogêneos, os descritores geraram vetores de características contendo propriedades mais informativas sobre os objetos em destaque na imagem. Isto se mostrou verdade devido à melhor eficácia dos descritores em imagens de consulta com objetos sobrepostos a fundos homogêneos.

Outro possível motivo para as variações dos valores de precisão entre as diferentes imagens de consulta foi a ambigüidade na interpretação da imagem de consulta pelo usuário. Algumas imagens de consulta podem ter sido interpretadas de maneiras diferentes por diferentes usuários. Por exemplo, a consulta 1 pode significar tanto um grupo de pessoas quanto um grupo de alunos formandos. A consulta 6, por exemplo, pode significar um carro vermelho ou simplesmente um carro.

Os resultados experimentais na base Yahoo indicam que o descritor BIC é o que apresenta melhor eficácia geral. Entre as possíveis razões para seu bom desempenho está

Tabela 5.19: Médias dos valores de P_{10}, P_{20} e P_{30} para a consulta 6.

Descritor	$\mathbf{P_{10}}$
LAS	0,23
LBP	0,19
GCH	0,19
CSD	0,12
CCOM	0,11
BIC	0,11
ACC	0,09
Color Bitmap	0,09

Descritor	$\mathbf{P_{20}}$
LAS	0,20
LBP	0,10
GCH	0,10
Color Bitmap	0,07
ACC	0,07
CSD	0,06
CCOM	0,05
BIC	0,05

Descritor	$\mathbf{P_{30}}$
LAS	0,19
GCH	0,10
ACC	0,09
BIC	0,07
LBP	0,07
CSD	0,05
Color Bitmap	0,05
CCOM	0,04

Tabela 5.20: Médias dos valores de P_{10}, P_{20} e P_{30} para a consulta 7.

Descritor	$\mathbf{P_{10}}$
BIC	0,29
ACC	0,21
LAS	0,14
CSD	0,11
LBP	0,10
CCOM	0,09
GCH	0,09
Color Bitmap	0,09

Descritor	$\mathbf{P_{20}}$
BIC	0,19
ACC	0,14
LAS	0,07
CSD	0,06
LBP	0,05
CCOM	0,05
GCH	0,05
Color Bitmap	0,05

Descritor	$\mathbf{P_{30}}$
BIC	0,15
ACC	0,13
Color Bitmap	0,06
LAS	0,05
LBP	0,04
CSD	0,04
CCOM	0,03
GCH	0,03

o seu algoritmo de extração de características. A classificação dos pixels da imagem em pixels de borda ou de interior realiza uma segmentação simples entre regiões homogêneas e regiões texturizadas. De certa maneira, isto codifica propriedades de textura ou forma da imagem em conjunto com propriedades de cor. Outro descritor de bom desempenho geral foi o descritor ACC. Suas desvantagens principais em relação ao descritor BIC são o tempo de extração de características e o tamanho do vetor gerado.

Dentre os descritores de textura, o descritor LAS foi o que obteve melhor desempenho geral. Ele possui algoritmos simples para extrair características e calcular distâncias. Sua principal desvantagem está no tamanho do vetor de características gerado, o qual é maior do que a maioria dos vetores gerados pelos descritores de textura comparados. Sua eficácia foi uma das mais altas na base Brodatz e foi a mais alta na base Yahoo dentre os descritores de textura.

A avaliação de eficácia com usuários reais possibilitou medir a capacidade de recuperação de imagens relevantes dos descritores de acordo com potenciais usuários de um sistema de CBIR.

Tabela 5.21: Médias dos valores de P_{10}, P_{20} e P_{30} para a consulta 8.

Descritor	$\mathbf{P_{10}}$
ACC	0,15
GCH	0,10
BIC	0,10
LAS	0,09
Color Bitmap	0,08
CSD	0,07
LBP	0,07
CCOM	0,07

Descritor	$\mathbf{P_{20}}$
ACC	0,16
LAS	0,08
BIC	0,07
GCH	0,06
LBP	0,05
CSD	0,04
Color Bitmap	0,04
CCOM	0,04

Descritor	$\mathbf{P_{30}}$
ACC	0,10
BIC	0,07
LAS	0,06
GCH	0,04
CSD	0,04
LBP	0,03
Color Bitmap	0,03
CCOM	0,02

Tabela 5.22: Médias dos valores de P_{10}, P_{20} e P_{30} para a consulta 9.

Descritor	$\mathbf{P_{10}}$
ACC	0,67
BIC	0,57
GCH	0,57
Color Bitmap	0,28
LBP	0,18
LAS	0,18
CSD	0,15
CCOM	0,09

Descritor	$\mathbf{P_{20}}$
ACC	0,44
BIC	0,34
GCH	0,29
Color Bitmap	0,14
LBP	0,09
LAS	0,09
CSD	0,07
CCOM	0,04

Descritor	$\mathbf{P_{30}}$
BIC	0,31
ACC	0,29
GCH	0,19
Color Bitmap	0,13
LBP	0,09
LAS	0,06
CSD	0,05
CCOM	0,03

Tabela 5.23: Médias dos valores de P_{10}, P_{20} e P_{30} para a consulta 10.

Descritor	$\mathbf{P_{10}}$
ACC	0,77
BIC	0,76
LAS	0,65
GCH	0,59
LBP	0,44
CSD	0,40
Color Bitmap	0,40
CCOM	0,20

Descritor	$\mathbf{P_{20}}$
ACC	0,44
BIC	0,38
LAS	0,37
GCH	0,35
Color Bitmap	0,25
CSD	0,25
LBP	0,22
CCOM	0,10

Descritor	$\mathbf{P_{30}}$
LAS	0,30
ACC	0,29
BIC	0,25
GCH	0,24
LBP	0,18
Color Bitmap	0,17
CSD	0,17
CCOM	0,07

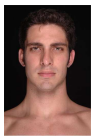
Tabela 5.24: Médias dos valores de P_{10}, P_{20} e P_{30} para a consulta 11.

Descritor	$\mathbf{P_{10}}$
BIC	0,57
GCH	0,41
ACC	0,39
LAS	0,21
CSD	0,13
LBP	0,13
CCOM	0,09
Color Bitmap	0,09

Descritor	$\mathbf{P_{20}}$
BIC	0,46
ACC	0,23
GCH	0,22
LBP	0,14
LAS	0,11
CSD	0,07
Color Bitmap	0,06
CCOM	0,05

Descritor	$\mathbf{P_{30}}$
BIC	0,36
ACC	0,20
GCH	0,18
LAS	0,15
LBP	0,09
Color Bitmap	0,06
CSD	0,06
CCOM	0,03

Tabela 5.25: Médias dos valores de P_{10}, P_{20} e P_{30} para a consulta 12.



Descritor	$\mathbf{P_{10}}$
ACC	0,31
GCH	0,27
BIC	0,23
CSD	0,20
Color Bitmap	0,17
LAS	0,13
LBP	0,11
CCOM	0,09

Descritor	$\mathbf{P_{20}}$
ACC	0,24
GCH	0,23
BIC	0,19
Color Bitmap	0,15
CSD	0,15
LBP	0,09
LAS	0,08
CCOM	0,04

Descritor	$\mathbf{P_{30}}$
ACC	0,22
GCH	0,18
BIC	0,18
Color Bitmap	0,12
CSD	0,12
LBP	0,09
LAS	0,07
CCOM	0,03

Tabela 5.26: Médias dos valores de P_{10}, P_{20} e P_{30} para a consulta 13.

Descritor	$\mathbf{P_{10}}$
LAS	0,22
CSD	0,19
BIC	0,13
LBP	0,13
CCOM	0,13
ACC	0,13
GCH	0,13
Color Bitmap	0,13

Descritor	$\mathbf{P_{20}}$
LAS	0,13
CSD	0,09
BIC	0,08
LBP	0,07
ACC	0,07
CCOM	0,06
GCH	0,06
Color Bitmap	0,06

Descritor	$\mathbf{P_{30}}$
LAS	0,10
CSD	0,10
BIC	0,08
LBP	0,06
ACC	0,04
CCOM	0,04
GCH	0,04
Color Bitmap	0,04

Tabela 5.27: Médias dos valores de P_{10}, P_{20} e P_{30} para a consulta 14.

Descritor	$\mathbf{P_{10}}$
LAS	0,55
Color Bitmap	0,11
ACC	0,09
BIC	0,07
GCH	0,07
CSD	0,06
LBP	0,05
CCOM	0,05

Descritor	$\mathbf{P_{20}}$
LAS	0,45
ACC	0,06
Color Bitmap	0,05
BIC	0,05
GCH	0,04
CSD	0,03
LBP	0,03
CCOM	0,03

Descritor	$\mathbf{P_{30}}$
LAS	0,40
CSD	0,07
ACC	0,04
BIC	0,04
Color Bitmap	0,04
GCH	0,03
LBP	0,02
CCOM	0,02

Tabela 5.28: Médias dos valores de P_{10}, P_{20} e P_{30} para a consulta 15.

Descritor	$\mathbf{P_{10}}$
GCH	0,30
BIC	0,29
ACC	0,17
Color Bitmap	0,13
CSD	0,10
LBP	0,09
LAS	0,09
CCOM	0,09

Descritor	$\mathbf{P_{20}}$
GCH	0,28
BIC	0,22
ACC	0,19
Color Bitmap	0,12
LAS	0,07
CSD	0,05
LBP	0,04
CCOM	0,04

Descritor	$\mathbf{P_{30}}$
GCH	0,22
BIC	0,18
ACC	0,16
Color Bitmap	0,10
LAS	0,07
CSD	0,03
LBP	0,03
CCOM	0,03

Tabela 5.29: Médias dos valores de P_{10}, P_{20} e P_{30} para a consulta 16.

Descritor	$\mathbf{P_{10}}$
Color Bitmap	0,41
CSD	0,39
GCH	0,29
ACC	0,28
LAS	0,24
BIC	0,22
LBP	0,19
CCOM	0,17

Descritor	$\mathbf{P_{20}}$
Color Bitmap	0,23
CSD	0,22
BIC	0,20
ACC	0,19
GCH	0,16
LAS	0,16
LBP	0,10
CCOM	0,08

Descritor	$\mathbf{P_{30}}$
Color Bitmap	0,17
LAS	0,16
BIC	0,16
CSD	0,14
ACC	0,12
GCH	0,11
LBP	0,08
CCOM	0,06

Capítulo 6

Conclusões

Devido à crescente quantidade de imagens disponíveis atualmente, verifica-se a necessidade de sistemas de busca eficientes e eficazes.

Sistemas de busca de imagens por conteúdo podem ser usados para atender a esta necessidade. A principal virtude destes sistemas relaciona-se à representação de imagens por suas propriedades visuais, como cor, textura e forma de objetos.

O componente principal de sistemas de CBIR é o descritor de imagens. Descritores de imagens são os responsáveis por caracterizar as propriedades visuais de uma imagem e conseguir compará-las.

Esta dissertação apresenta um estudo comparativo entre descritores de imagens para a recuperação de imagens por conteúdo na Web. A Web apresenta um cenário extremamente heterogêneo e de grandes dimensões.

Esta dissertação serve como ferramental para desenvolvedores e pesquisadores que procuram descritores para um sistema de busca de imagens por conteúdo em um ambiente heterogêneo.

O estudo comparativo realizado compara os descritores segundo aspectos teóricos e práticos. São comparados descritores de cor, textura e forma em termos de eficiência e eficácia. Além disso, são apresentadas as principais abordagens para descritores de cor, textura e forma.

A comparação teórica considera os seguintes critérios: complexidade do algoritmo de extração do vetor de características, complexidade da função de distância, requisitos de armazenamento e o ambiente de validação dos descritores. A comparação teórica apresenta tabelas comparativas com 88 descritores, dentre os quais existem descritores importantes da literatura e descritores recentemente propostos.

Pelo estudo teórico realizado, observa-se que a maioria dos descritores comparados possui complexidade $O(n)$ para extrair o vetor de características de uma imagem. Apesar disso, existem muitos descritores com algoritmos de extração mais complexos. Em relação

à complexidade das funções de distância dos descritores, nota-se a predominância de funções com complexidade linear no tamanho do vetor de características.

Alguns dos descritores analisados teoricamente foram implementados e testados em experimentos práticos. Os experimentos práticos possibilitaram mensurar questões de eficiência do algoritmo de extração do vetor de características, eficiência no cálculo de distância, eficiência no armazenamento do vetor de características e eficácia.

Os diferentes tipos de descritores foram comparados independentemente em experimentos específicos. Foram usadas bases de imagens que refletem as propriedades visuais codificadas por cada tipo de descritor. Os resultados dos experimentos específicos possibilitaram realizar a escolha de um conjunto de descritores para experimentos em uma base de imagens heterogênea e de grandes dimensões. Nesta base foi possível medir o comportamento dos descritores em um ambiente Web.

A eficácia dos descritores na base de imagens heterogênea foi medida por meio de experimentos com usuários reais.

Os resultados indicam que muitos descritores sofrem degradação de desempenho, principalmente em termos de eficácia, quando aplicados a um ambiente heterogêneo e de grandes dimensões. Descritores de cor tiveram precisão acima de 70% na base ETH enquanto que na base Yahoo a precisão média não chegou aos 40%. No caso dos descritores de textura, a precisão na base Brodatz chegou a passar os 80% para os descritores comparados. No entanto, na base Yahoo a precisão média não chegou a 30%. De maneira geral, quando aplicados na base Yahoo, os descritores de cor tiveram degradação menor de eficácia se comparados aos descritores de textura.

Os resultados experimentais na base Yahoo indicam que o descritor BIC é o que apresenta melhor eficácia geral. Além disso, o descritor BIC possui algoritmos simples para extrair características e para calcular distâncias. Dentre os descritores de textura, o descritor LAS é o que apresenta melhor desempenho geral.

Em relação aos descritores de forma, a eficácia deles foi muito baixa na base Caltech101. Apesar disso, o desempenho foi aceitável nos critérios de eficiência. Como o conteúdo da base Caltech101 é relativamente heterogêneo nenhum dos descritores de forma implementados foi usado nos experimentos na base Yahoo. O uso de descritores de forma independentes de segmentação, como os descritores de forma implementados, não pode ser considerado inviável em ambientes heterogêneos, apesar da baixa eficácia na base Caltech101. Experimentos com mais descritores precisam ser realizados para que esta afirmação possa ser usada com mais certeza.

Um dos principais responsáveis pela queda de eficácia dos descritores na base Yahoo foi a informação semântica das imagens. Para os usuários, encontrar imagens semelhantes consiste em encontrar imagens com semântica semelhante e não necessariamente com conteúdo visual semelhante. Por exemplo, no caso da imagem de consulta contendo o pôr-

do-sol, muitas das imagens retornadas pelos descritores de cor possuíam características visuais semelhantes a um pôr-do-sol. Entretanto, eram imagens sobre assuntos diferentes. Nas consultas com os carros, era mais importante que os resultados exibissem simplesmente fotos de carros do que fotos com carros na mesma posição e com cores iguais.

No entanto, o conteúdo visual pode ser muito importante para a melhoria dos resultados em sistemas de busca de imagens de propósito geral. A combinação de descrições textuais com propriedades visuais pode ajudar a atingir este objetivo.

Esta dissertação também apresenta uma ferramenta para a realização de testes comparativos automatizados entre descritores de imagens. Sua utilização possibilita a execução de testes comparativos de forma organizada, evitando possíveis erros humanos e erros de padronização.

6.1 Contribuições

Em resumo, as principais contribuições desta dissertação são:

- comparação teórica de descritores de imagens em termos de complexidade assintótica de algoritmos;
- comparação teórica considerando 88 descritores de imagens;
- comparação de uma grande quantidade de descritores em experimentos práticos;
- comparação de descritores em uma base de imagens heterogênea com mais de 200 mil imagens;
- avaliação de eficácia dos descritores por meio de experimentos com usuários reais;
- especificação e implementação de uma ferramenta para a realização de testes comparativos automatizados entre descritores de imagens.

6.2 Trabalhos futuros

- Realização de experimentos com novos descritores:

  Uma primeira extensão do trabalho consiste em adicionar novos descritores ao estudo teórico. Entre eles, estão descritores de relacionamento espacial entre objetos. Espera-se também adicionar novos descritores aos experimentos práticos, tanto nas bases específicas quanto na base de imagens heterogêneas. Em especial, outros descritores de forma independentes de segmentação podem ser adicionados ao estudo.

- Incorporação de novas funcionalidades à ferramenta de comparação de descritores:

  Possíveis extensões à ferramenta dizem respeito à realização de experimentos considerando outras medidas de avaliação (por exemplo, ANMRR) e a realização de testes estatísticos para a comparação de resultados. Espera-se também completar pequenas questões de desenvolvimento de maneira que a ferramenta possa ser disponibilizada e compartilhada entre diferentes pesquisadores.

- Comparação de medidas de eficácia:

  Verificar as vantagens e desvantagens no uso de diferentes medidas para a avaliação de eficácia dos descritores.

- Implementação de protótipo de sistema de recuperação de imagens na Web:

  Uma outra possível extensão consiste na implementação de um protótipo de sistema de recuperação de imagens por conteúdo na Web aproveitando os módulos implementados da ferramenta de comparação de descritores desenvolvida.

- Pesquisa em técnicas para combinação de descritores:

  Pelos resultados obtidos observou-se que não houve um descritor com bom desempenho em todas as consultas. Dessa forma, cabe investigar técnicas que viabilizem a combinação dos melhores descritores de maneira que se possa melhorar a eficácia. Uma das técnicas que poderiam ser usadas seria a Programação Genética [25].

- Pesquisa em técnicas para combinação de descritores textuais e descritores de imagens:

  Apesar dos descritores de imagens não terem apresentado alta eficácia na base de imagens heterogêneas, a combinação deles com descritores textuais pode ser promissora para aumentar a eficácia tanto de sistemas de CBIR quanto de sistemas baseados em descrições textuais.

Referências Bibliográficas

[1] A. E. Abdel-Hakim and A. A. Farag. Csift: A sift descriptor with color invariant characteristics. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1978–1983, Washington, DC, USA, 2006.

[2] A. Adnan, S. Gul, M. Ali, and A. H. Dar. Content based image retrieval using geometrical-shape of objects in image. *International Conference on Emerging Technologies*, pages 222–225, November 2007.

[3] R. Agrawal, W. Grosky, F. Fotouhi, and C. Wu. Application of diffusion kernel in multimodal image retrieval. In *ISMW '07: Proceedings of the Ninth IEEE International Symposium on Multimedia Workshops*, pages 271–276, Washington, DC, USA, 2007.

[4] U. A. Ahmad, K. Kidiyo, and R. Joseph. Texture features based on fourier transform and gabor filters: an empirical comparison. *International Conference on Machine Vision*, pages 67–72, December 2007.

[5] N. Alajlan, M. S. Kamel, and G. Freeman. Multi-object image retrieval based on shape and topology. *Signal Processing: Image Communication*, 21(10):904 – 918, 2006.

[6] L. Amsaleg, P. Gros, and S. Berrani. Robust object recognition in images and the related database problems. *Multimedia Tools and Applications*, 23(3):221–235, 2004.

[7] F. A. Andaló. Descritores de forma baseados em tensor scale. Master's thesis, Instituto de Computação - Unicamp, March 2007.

[8] J. Annesley, J. Orwell, and J.-P. Renno. Evaluation of mpeg7 color descriptors for visual surveillance retrieval. *2nd Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pages 105–112, 15-16 October 2005.

[9] A. Antani, R. Kasturi, and R. Jain. A Survey on the Use of Pattern Recognition Methods for Abstraction, Indexing and Retrieval of Images and Video. *Pattern Recognition*, 35(4):945–965, April 2002.

[10] Y. A. Aslandogan and C. T. Yu. Diogenes: a web search agent for person images. In *Proceedings of the eighth ACM international conference on Multimedia*, pages 481–482, New York, NY, USA, 2000.

[11] Y. Alp Aslandogan and C. T. Yu. Techniques and systems for image and video retrieval. *IEEE Transactions on Knowledge and Data Engineering*, 11(1):56–63, 1999.

[12] M. F. Bears, B. W. Connors, and M. A. Paradiso. *Neurociências: desvendando o sistema nervoso*. Artmed, second edition, 2002.

[13] J. Bigun, S. K. Bhattacharjee, and S. Michel. Orientation radiograms for image retrieval: An alternative to segmentation. *International Conference on Pattern Recognition*, 3:346, 1996.

[14] P. Brodatz. *Textures: A Photographic Album for Artists and Designers*. Dover, 1966.

[15] Jr. C. Traina, A. Traina, C. Faloutsos, and B. Seeger. Fast indexing and visualization of metric data sets using slim-trees. *IEEE Transactions on Knowledge and Data Engineering*, 14(2):244–260, 2002.

[16] T. Chaira and A. K. Ray. Fuzzy measures for color image retrieval. *Fuzzy Sets and Systems*, 150(3):545–560, 2005.

[17] C. Chen. Improved moment invariants for shape discrimination. *Pattern Recognition*, 26(5):683 – 686, 1993.

[18] Y. W. Chen and C. L. Xu. Rolling penetrate descriptor for shape-based image retrieval and object recognition. *Pattern Recognition Letters*, In Press, Corrected Proof:–, 2008.

[19] P. Ciaccia, M. Patella, and P. Zezula. M-tree: An Efficient Access Method for Similarity Search in Metric Spaces. In *Proceedings of 23rd International Conference on Very Large Data Bases*, pages 426–435, Athens, Greece, 1997.

[20] T. A. S. Coelho, P. P. Calado, L. V. Souza, B. A. Ribeiro-Neto, and R. Muntz. Image retrieval using multiple evidence ranking. *IEEE Transactions on Knowledge and Data Engineering*, 16(4):408–417, 2004.

[21] T. A. S. Coelho, L. Vieira Souza, and B. A. Ribeiro-Neto. Recuperação de imagens na web baseada em múltiplas evidências textuais. In *Brazilian Symposium on Databases*, pages 125–139, 2001.

[22] D. da S. Andrade and N. J. Leite. Testes de significância estatísticos e avaliação de um modelo de recuperação de imagens por conteúdo. Master's thesis, Unicamp, 2004.

[23] R. da S. Torres and A. X. Falcão. Content-Based Image Retrieval: Theory and Applications. *Revista de Informática Teórica e Aplicada*, 13(2):161–185, 2006.

[24] R. da S. Torres and A. X. Falcão. Contour salience descriptors for effective image retrieval and analysis. *Image and Vision Computing*, 25(1):3 – 13, 2007. SIBGRAPI.

[25] R. da S. Torres, A. X. Falcão, M. A. Gonçalves, J. P. Papa, B. Zhang, W. Fan, and E. A. Fox. A genetic programming framework for content-based image retrieval. *Pattern Recognition*, 42(2):283 – 292, 2009. Learning Semantics from Multimedia Content.

[26] R. da S. Torres, J. A. M. Zegarra, J. A. dos Santos, C. D. Ferreira, O. A. B. Penatti, F. A. Andaló, and J. Almeida. Recuperação de imagens: Desafios e novos rumos. In *Seminário Integrado de Software e Hardware (SEMISH)*, July 2008.

[27] R. de O. Stehling, M. A. Nascimento, and A X. Falcao. An adaptive and efficient clustering-based approach for content-based image retrieval in image databases. In *Proceedings of the International Database Engineering & Applications Symposium*, pages 356–365, Washington, DC, USA, 2001.

[28] R. de O. Stehling, M. A. Nascimento, and A. X. Falcão. Cell histograms versus color histograms for image representation and retrieval. *Knowledge and Information Systems*, 5(3):315–336, 2003.

[29] R. de O. Stehling, M. A. Nascimento, and A. X. Falcão. A compact and efficient image retrieval approach based on border/interior pixel classification. In *Proceedings of the eleventh international conference on Information and knowledge management*, pages 102–109, New York, NY, USA, 2002.

[30] A. del Bimbo. *Visual Information Retrieval*. Morgan Kaufmann Publishers, San Francisco, CA, USA, 1999.

[31] L. M. del Val Cura and N. J. Leite. Um modelo para recuperação por conteudo de imagens de sensoriamento remoto. Master's thesis, Unicamp, 2002.

[32] Y. Deng, B. S. Manjunath, C. Kenney, M. S. Moore, and H. Shin. An efficient color representation for image retrieval. *IEEE Transactions on Image Processing*, 10(1):140–147, January 2001.

[33] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *Computer Vision and Pattern Recognition Workshop*, 12:178, 2004.

[34] C. D. Ferreira. Image retrieval with relevance feedback based on genetic programming. Master's thesis, Instituto de Computação - Unicamp, July 2007.

[35] C. D. Ferreira, R. da S. Torres, M. A. Goncalves, and W. Fan. Image Retrieval with Relevance Feedback based on Genetic Programming. In *Brazilian Symposium on Databases*, Campinas, SP, October 2008.

[36] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1992.

[37] M. A. Gonçalves, E. A. Fox, A. Krowne, P. Calado, A. H. F. Laender, A. S. da Silva, and B. A. Ribeiro-Neto. The effectiveness of automatically structured queries in digital libraries. *Joint Conference on Digital Libraries*, 0:98–107, 2004.

[38] S. Goodall, P. H. Lewis, K. Martinez, P. A. S. Sinclair, F. Giorgini, M. Addis, M. J. Boniface, C. Lahanier, and J. Stevenson. Sculpteur: Multimedia retrieval for museums. In *International Conference on Image and Video Retrieval*, pages 638–646, 2004.

[39] E. Hadjidemetriou, M. D. Grossberg, and S. K. Nayar. Multiresolution histograms and their use for recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(7):831–847, 2004.

[40] J. Han and K. Ma. Rotation-invariant and scale-invariant gabor features for texture image retrieval. *Image and Vision Computing*, 25(9):1474–1481, 2007.

[41] R. M. Haralick, K. Shanmugam, and I. Dinstein. Textural features for image classification. *IEEE Transactions on Systems, Man and Cybernetics*, 3(6):610–621, November 1973.

[42] K. Hirata and T. Kato. Query by visual example - content based image retrieval. In *Proceedings of the 3rd International Conference on Extending Database Technology*, pages 56–71, London, UK, 1992. Springer-Verlag.

[43] M. Hu. Visual pattern recognition by moment invariants. *IRE Transactions on Information Theory*, 8(2):179–187, February 1962.

[44] C. Huang and Q. Liu. An orientation independent texture descriptor for image retrieval. *International Conference on Communications, Circuits and Systems*, pages 772–776, July 2007.

[45] J. Huang, S. R. Kumar, M. Mitra, W. Zhu, and R. Zabih. Image indexing using color correlograms. In *Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition*, page 762, Washington, DC, USA, 1997.

[46] P. W. Huang, S. K. Dai, and P. L. Lin. Texture image retrieval and image segmentation using composite sub-band gradient vectors. *Journal of Visual Communication and Image Representation*, 17(5):947 – 957, 2006.

[47] S. Jang and H. Kwak. Content-based image retrieval using shape information of central object. *The 9th International Conference on Advanced Communication Technology*, 1:502–505, February 2007.

[48] P. Janney and Z. Yu. Invariant features of local textures - a rotation invariant local texture descriptor. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–7, June 2007.

[49] M. L. Kherfi, D. Ziou, and A. Bernardi. Atlas wise: A web-based image retrieval engine. In *Proceedings of the International Conference on Image and Signal Processing*, pages 69–77, 2003.

[50] M. L. Kherfi, D. Ziou, and A. Bernardi. Image retrieval from the world wide web: Issues, techniques, and systems. *ACM Computing Surveys*, 36(1):35–67, 2004.

[51] H. Kim and J. Kim. Region-based shape descriptor invariant to rotation, scale and translation. *Signal Processing: Image Communication*, 16(1-2):87 – 93, 2000.

[52] W. Kim and Y. Kim. A region-based shape descriptor using zernike moments. *Signal Processing: Image Communication*, 16(1-2):95 – 102, 2000.

[53] S. Kiranyaz, M. Ferreira, and M. Gabbouj. A directional texture descriptor via 2d walking ant histogram. *International Workshop on Image Analysis for Multimedia Interactive Services*, 0:9, 2007.

[54] S. Kiranyaz, M. Ferreira, and M. Gabbouj. A generic shape/texture descriptor over multiscale edge field: 2-d walking ant histogram. *IEEE Transactions on Image Processing*, 17(3):377–391, March 2008.

[55] M. Kokare, B. N. Chatterji, and P. K. Biswas. Cosine-modulated wavelet based texture features for content-based image retrieval. *Pattern Recognition Letters*, 25(4):391–398, 2004.

[56] X. Kong, Q. Luo, G. Zeng, and M. H. Lee. A new shape descriptor based on centroid-radii model and wavelet transform. *Optics Communications*, 273(2):362 – 366, 2007.

[57] V. Kovalev and S. Volmer. Color co-occurence descriptors for querying-by-example. *MultiMedia Modeling*, 0:32–38, October 1998.

[58] I. Kunttu, L. Lepistö, J. Rauhamaa, and A. Visa. Multiscale fourier descriptor for shape-based image retrieval. *International Conference on Pattern Recognition*, 2:765–768, 2004.

[59] D. Lee and H. Kim. A fast content-based indexing and retrieval technique by the shape information in large image database. *Journal of Systems and Software*, 56(2):165 – 182, 2001.

[60] H. Y. Lee, H. K. Lee, and Y. Ho Ha. Spatial color descriptor for image retrieval and video segmentation. *IEEE Transactions on Multimedia*, 5(3):358–367, September 2003.

[61] K. Lee and L. Chen. An efficient computation method for the texture browsing descriptor of mpeg-7. *Image and Vision Computing*, 23(5):479 – 489, 2005.

[62] M. S. Lew. Next-generation web searches for visual content. *Computer*, 33(11):46–53, 2000.

[63] X. Li. Image retrieval based on perceptive weighted color blocks. *Pattern Recognition Letters*, 24(12):1935–1941, 2003.

[64] Y. Li, M. J. Kyan, and L. Guan. Improving shape-based cbir for natural image content using a modified gfd. In *Second International Conference on Image Analysis and Recognition*, pages 593–600, 2005.

[65] H. Lin, Y. Kao, S. Yen, and C. Wang. A study of shape-based image retrieval. *International Conference on Distributed Computing Systems Workshops*, 1:118–123, 2004.

[66] W. Lin, R. Jin, and A. Hauptmann. Web image retrieval re-ranking with relevance model. In *WI '03: Proceedings of the 2003 IEEE/WIC International Conference on Web Intelligence*, page 242, Washington, DC, USA, 2003.

[67] S. Loncaric. A Survey of Shape Analysis Techniques. *Pattern Recognition*, 31(8):983–1190, August 1998.

[68] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[69] T. Lu and C. Chang. Color image retrieval technique based on color features and image bitmap. *Information Processing and Management*, 43(2):461–472, 2007.

[70] F. Mahmoudi, J. Shanbehzadeh, A. Eftekhari-Moghadam, and H. Soltanian-Zadeh. Image retrieval based on shape similarity by edge orientation autocorrelogram. *Pattern Recognition*, 36(8):1725 – 1736, 2003.

[71] B. S. Manjunath and W. Y. Ma. Texture features for browsing and retrieval of image data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(8):837–842, August 1996.

[72] B. S. Manjunath, J.-R. Ohm, V. V. Vasudevan, and A. Yamada. Color and texture descriptors. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(6):703–715, June 2001.

[73] T. A. A. Matos, I. R. da Silva, and C. A. Z. Barcelos. Uma proposta de construção de índice invertido para recuperação de imagens baseada em conteúdo. Master's thesis, UFU, 2009.

[74] P. A. V. Miranda, R. da S. Torres, and A. X. Falcão. Tsd: A shape descriptor based on a distribution of tensor scale local orientation. In *XVIII Brazilian Symposium on Computer Graphics and Image Processing*, page 139, Washington, DC, USA, 2005. IEEE Computer Society.

[75] H. A. Moghaddam, T. T. Khajoie, A. H. Rouhi, and M. S. Tarzjan. Wavelet correlogram: A new approach for image indexing and retrieval. *Pattern Recognition*, 38(12):2506–2518, 2005.

[76] F. Mokhtarian, S. Abbasi, and J. Kittler. Efficient and robust retrieval by shape content through curvature scale space. In *International Workshop on Image Databases and Multi-Media Search*, pages 35–42, 1996.

[77] K. Nallaperumal, M. S. Banu, and C. C. Christiyana. Content based image indexing and retrieval using color descriptor in wavelet domain. *International Conference on Computational Intelligence and Multimedia Applications*, 3:185–189, 13-15 December 2007.

[78] S. Naskar and R. Parekh. A content based shape retrieval system. *48th Midwest Symposium on Circuits and Systems*, pages 1745–1748 Vol. 2, August 2005.

[79] R. Nevatia. *Machine Perception*. Prentice-Hall, 1982.

[80] S. Newsam and Y. Yang. Comparing global and interest point descriptors for similarity retrieval in remote sensed imagery. In *Proceedings of the 15th annual ACM international symposium on Advances in geographic information systems*, pages 1–8, New York, NY, USA, 2007. ACM.

[81] T. Ojala, M. Aittola, and E. Matinmikko. Empirical evaluation of mpeg-7 xm color descriptors in content-based retrieval of semantic image categories. *International Conference on Pattern Recognition*, 2:21021, 2002.

[82] T. Ojala, M. Pietikäinen, and T. Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987, 2002.

[83] G. Paschos, I. Radev, and N. Prabakar. Image content-based retrieval using chromaticity moments. *IEEE Transactions on Knowledge and Data Engineering*, 15(5):1069–1072, 2003.

[84] G. Pass, R. Zabih, and J. Miller. Comparing images using color coherence vectors. In *Proceedings of the fourth ACM international conference on Multimedia*, pages 65–73, New York, NY, USA, 1996.

[85] O. A. B. Penatti and R. da S. Torres. Color descriptors for web image retrieval: A comparative study. *XXI Brazilian Symposium on Computer Graphics and Image Processing*, pages 163–170, October 2008.

[86] Y. M. Ro and H. K. Kang. Hierarchical rotational invariant similarity measurement for mpeg-7 homogeneous texture descriptor. *Electronics Letters*, 36(15):1268–1270, July 2000.

[87] Y. Rubner and C. Tomasi. Texture-based image retrieval without segmentation. *IEEE International Conference on Computer Vision*, 2:1018, 1999.

[88] Y. Rui, T. Huang, and S. Chang. Image retrieval: current techniques, promising directions and open issues. *Journal of Visual Communication and Image Representation*, 10(4):39–62, April 1999.

[89] M. Safar, C. Shahabi, and X. Sun. Image retrieval by shape: A comparative study. In *IEEE International Conference on Multimedia and Expo*, pages 141–144, 2000.

[90] A. Sajanhar, G. Lu, and D. Zhang. Discriminating shape descriptors based on connectivity. In *IEEE International Conference on Multimedia and Expo*, pages 2075–2078, 2004.

[91] A. Sajjanhar, G. Lu, and D. Zhang. Spherical harmonics descriptor for 2d-image retrieval. *IEEE International Conference on Multimedia and Expo*, pages 105–108, July 2005.

[92] A. Sajjanhar, G. Lu, D. Zhang, and W. Zhou. A composite descriptor for shape retrieval. *ACIS International Conference on Computer and Information Science*, 0:795–800, 2007.

[93] J. A. dos Santos, C. D. Ferreira, and R. da S. Torres. A genetic programming approach for relevance feedback in region-based image retrieval systems. In *XXI Brazilian Symposium on Computer Graphics and Image Processing*, pages 155–162, Washington, DC, USA, 2008. IEEE Computer Society.

[94] E. Saykol, U. Güdükbay, and Ö. Ulusoy. A histogram-based approach for object-based query-by-shape-and-color in image and video databases. *Image and Vision Computing*, 23(13):1170 – 1180, 2005.

[95] S. Sclaroff, L. Taycher, and M. La Cascia. Imagerover: A content-based image browser for the world wide web. In *Proceedings of the 1997 Workshop on Content-Based Access of Image and Video Libraries*, page 2, Washington, DC, USA, 1997.

[96] D. Sim, H. Kim, and R. Park. Fast texture description and retrieval of dct-based compressed images. *Electronics Letters*, 37(1):18–19, January 2001.

[97] D. Sim, H. Kim, and R. Park. Invariant texture retrieval using modified zernike moments. *Image and Vision Computing*, 22(4):331 – 342, 2004.

[98] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1349–1380, 2000.

[99] M. A. Stricker and M. Orengo. Similarity of color images. In W. Niblack and R. C. Jain, editors, *Proc. SPIE Storage and Retrieval for Image and Video Databases III*, volume 2420, pages 381–392, March 1995.

[100] J. Sun, X. Zhang, J. Cui, and L. Zhou. Image retrieval based on color distribution entropy. *Pattern Recognition Letters*, 27(10):1122–1126, 2006.

[101] M. J. Swain and D. H. Ballard. Color indexing. *International Journal of Computer Vision*, 7(1):11–32, 1991.

[102] V. Takala, T. Ahonen, and M. Pietikäinen. Block-based methods for image retrieval using local binary patterns. In *Scandinavian Conference on Image Analysis*, pages 882–891, 2005.

[103] B. Tao and B. W. Dickinson. Texture recognition and image retrieval using gradient indexing. *Journal of Visual Communication and Image Representation*, 11(3):327 – 342, 2000.

[104] A. J. M. Traina, A. G. R. Balan, L. M. Bortolotti, and C. Traina Jr. Content-based image retrieval using approximate shape of objects. *IEEE Symposium on Computer-Based Medical Systems*, 0:91, 2004.

[105] M. Tuceryan and A. K. Jain. Texture analysis. pages 235–276, 1993.

[106] M. Unser. Sum and difference histograms for texture classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(1):118–125, 1986.

[107] A. Utenpattanant, O. Chitsobhuk, and A. Khawne. Color descriptor for image retrieval in wavelet domain. *Eighth International Conference on Advanced Communication Technology*, 1:818–821, 20-22 February 2006.

[108] J. Wang, W. Chang, and R. Acharya. Efficient and effective similar shape retrieval. *International Conference on Multimedia Computing and Systems*, 1:9875, 1999.

[109] Q. Wang, D. D. Feng, and Z. Chi. B-spline over-complete wavelet based fractal signature analysis for texture image retrieval. *Proceedings of 2004 International Symposium on Intelligent Multimedia, Video and Speech Processing*, pages 462–466, October 2004.

[110] Z. Wang, Z. Chi, and D. Feng. Shape based leaf image retrieval. *IEE Proceedings - Vision, Image and Signal Processing*, 150(1):34–43, February 2003.

[111] R. Weber, H. Schek, and S. Blott. A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. In *Proceedings of the 24rd International Conference on Very Large Data Bases*, pages 194–205, San Francisco, CA, USA, 1998.

[112] A. Williams and P. Yoon. Content-based image retrieval using joint correlograms. *Multimedia Tools and Applications*, 34(2):239–248, 2007.

[113] K. Wong, L. Po, and K. Cheung. A compact and efficient color descriptor for image retrieval. *IEEE International Conference on Multimedia and Expo*, pages 611–614, 2-5 July 2007.

[114] P. Wu, B. S. Manjunath, S. Newsam, and H. D. Shin. A texture descriptor for browsing and similarity retrieval. *Signal Processing: Image Communication*, 16(1-2):33 – 43, 2000.

[115] F. Xu and Y. Zhang. Evaluation and comparison of texture descriptors proposed in mpeg-7. *Journal of Visual Communication and Image Representation*, 17(4):701 – 716, 2006.

[116] N. Yang, W. Chang, C. Kuo, and T. Li. A fast mpeg-7 dominant color extraction with new similarity measure for image retrieval. *Journal of Visual Communication and Image Representation*, 19(2):92–105, 2008.

[117] X. Yang and J. Liu. Maximum entropy random fields for texture analysis. *Pattern Recognition Letters*, 23(1-3):93 – 101, 2002.

[118] A. Yoshitaka and T. Ichikawa. A survey on content-based retrieval for multimedia databases. *IEEE Transactions on Knowledge and Data Engineering*, 11(1):81–93, 1999.

[119] J. A. M. Zegarra, J. Beeck, N. J. Leite, R. da S. Torres, and A. X. Falcão. Combining global with local texture information for image retrieval applications. *International Symposium on Multimedia*, 0:148–153, 2008.

[120] J. A. M. Zegarra, N. J. Leite, and R. da S. Torres. Wavelet-based Feature Extraction for Fingerprint Image Retrieval. *Journal of Computational and Applied Mathematics*, 2008. Accepted.

[121] J. A. M. Zegarra, N. J. Leite, and R. da S. Torres. Rotation-invariant and scale-invariant steerable pyramid decomposition for texture image retrieval. In *XX Brazilian Symposium on Computer Graphics and Image Processing*, pages 121–128, Washington, DC, USA, 2007. IEEE Computer Society.

[122] D. Zhang and G. Lu. A comparative study of fourier descriptors for shape representation and retrieval. In *Proceedings of the 5th Asian Conference on Computer Vision*, pages 646–651, 2002.

[123] D. Zhang and G. Lu. Generic fourier descriptor for shape-based image retrieval. *IEEE International Conference on Multimedia and Expo*, 1:425–428 vol.1, 2002.

[124] D. Zhang and G. Lu. Evaluation of mpeg-7 shape descriptors against other shape descriptors. *Multimedia Systems*, 9(1):15–30, 2003.

[125] D. Zhang and G. Lu. Review of Shape Representation and Description. *Pattern Recognition*, 37(1):1–19, January 2004.

[126] D. Zhang, W. Wang, W. Gao, and S. Jiang. An effective local invariant descriptor combining luminance and color information. *IEEE International Conference on Multimedia and Expo*, pages 1507–1510, 2-5 July 2007.

[127] L. Zhang, J. Ma, X. Xu, and B. Yuan. Rotation invariant image classification based on mpeg-7 homogeneous texture descriptor. *Proceedings of the Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing*, 3:798–803, 2007.

[128] J. Zhou, L. Xin, and D. Zhang. Scale-orientation histogram for texture image retrieval. *Pattern Recognition*, 36(4):1061 – 1063, 2003.

[129] X. S. Zhou and T. S. Huang. Relevance Feedback in Content-based Image Retrieval: Some Recent Advances. *Information Sciences*, 148(1-4):129–137, Dec. 2002.

[130] A. Çarkacioglu and F. Yarman-Vural. Sasi: a new texture descriptor for content based image retrieval. *International Conference on Image Processing*, 2:137–140 vol.2, October 2001.

[131] A. Çarkacioglu and F. Yarman-Vural. Sasi: a generic texture descriptor for image retrieval. *Pattern Recognition*, 36(11):2615 – 2633, 2003.