Recuperação de Imagens por Cor utilizando Análise de Distribuição Discreta de Características

Este exemplar corresponde à redação final da Dissertação devidamente corrigida e defendida por Jurandy Gomes de Almeida Junior e aprovada pela Banca Examinadora.

Campinas, 10 de outubro de 2007.

Prof. Dr. Siome Klein Goldenstein
(Orientador)

Prof. Dr. Ricardo da Silva Torres
(Co-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 Júlia Milani Rodrigues – CRB8a / 2116

Almeida Junior, Jurandy Gomes de
AL64r Recuperação de imagens por cor utilizando análise de distribuição discreta de características / Jurandy Gomes de Almeida Junior -- Campinas, [S.P. :s.n.], 2007.

Orientador : Siome Klein Goldenstein ; Ricardo da Silva Torres
Dissertação (mestrado) - Universidade Estadual de Campinas, Instituto de Computação.

1. Processamento de imagens. 2. Banco de dados. 3. Recuperação da informação. 4. Análise de aglomerados. I. Goldenstein, Siome Klein. II. Torres, Ricardo da Silva. III. Universidade Estadual de Campinas. Instituto de Computação. IV. Título.

Título em inglês: Color-based image retrieval using discrete distribution features analysis.

Palavras-chave em inglês (Keywords): 1. Image processing. 2. Database. 3. Information retrieval. 4. Clustering analysis.

Área de concentração: Sistemas de Informação

Titulação: Mestre em Ciência da Computação

Banca examinadora: Prof. Dr. Siome Klein Goldenstein (IC-UNICAMP)
Profa. Dra. Agma Juci Machado Traina (ICMC-USP)
Prof. Dr. Jorge Stolfi (IC-UNICAMP)
Prof. Dr. Alexandre Xavier Falcão (IC-UNICAMP)

Data da defesa: 08/08/2007

Programa de Pós-Graduação: Mestrado em Ciência da Computação

TERMO DE APROVAÇÃO

Dissertação Defendida e Aprovada em 08 de agosto de 2007, pela Banca examinadora composta pelos Professores Doutores:

Profª. Drª. Agma Juci Machado Traina
ICMC - USP.

Prof. Dr. Jorge Stolfi
IC - UNICAMP.

Prof. Dr. Siome Klein Goldenstein
IC - UNICAMP.

Instituto de Computação
Universidade Estadual de Campinas

Recuperação de Imagens por Cor utilizando Análise de Distribuição Discreta de Características

Jurandy Gomes de Almeida Junior[1]

13 de julho de 2007

Banca Examinadora:

- Prof. Dr. Siome Klein Goldenstein
  (Orientador)
- Profa. Dra. Agma Juci Machado Traina
  ICMC – USP
- Prof. Dr. Jorge Stolfi
  IC – UNICAMP
- Prof. Dr. Alexandre Xavier Falcão
  IC – UNICAMP (Suplente)

[1]Financiado pela Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), processo 05/52959-3, no período de Agosto/2005 a Julho/2007.

Resumo

A evolução das tecnologias de aquisição, transmissão e armazenamento de imagens tem permitido a construção de bancos de imagens cada vez maiores. À medida em que cresce o volume de imagens nessas coleções, cresce também o interesse por sistemas capazes de recuperar essas imagens.

Essa tarefa tem sido endereçada pelos sistemas de recuperação de imagens por conteúdo. Nesses sistemas, o conteúdo de uma imagem é descrito a partir de suas características visuais de baixo nível, tais como cor, forma e textura.

Um sistema de recuperação de imagens por conteúdo ideal deve ser eficaz e eficiente. A eficácia é resultado de representações abstratas das imagens. Em geral, os métodos que realizam esse processo normalmente falham na presença de diferentes condições de iluminação, oclusão e foco. A eficiência, por outro lado, é resultado da organização dada à essas representações. Em geral, os métodos de agrupamento constituem uma das técnicas mais úteis para diminuir o espaço de busca e acelerar o processamento de uma consulta.

Para endereçar a eficácia, este trabalho apresenta o SIFT-Texton, um método capaz de incorporar informações sobre iluminação, oclusão e foco nas características visuais de baixo nível. Esse método baseia-se na distribuição discreta de características invariantes locais e em propriedades de baixo nível das imagens.

Em relação às questões de eficiência, este trabalho apresenta o DAH-Cluster, um novo paradigma de agrupamento aplicado à recuperação de imagens por conteúdo. Esse método combina características dos paradigmas hierárquicos divisivo e aglomerativo. Além disso, o DAH-Cluster introduz um novo conceito, chamado fator de reagrupamento, que permite agrupar elementos similares que seriam separados pelos paradigmas tradicionais.

Experimentos mostram que a combinação dessas técnicas permite a criação de um mecanismo robusto de recuperação de imagens por conteúdo, atingindo resultados mais eficazes e mais eficientes que as abordagens tradicionais descritas na literatura.

As principais contribuições deste trabalho são: (1) um novo método para recuperação de imagens capaz de incorporar informações sobre iluminação, oclusão e foco nas características visuais de baixo nível; e (2) um novo paradigma de agrupamento de dados que pode ser aplicado à recuperação de informação.

Abstract

Advances in data storage, data transmission, and image acquisition have enabled the creation of large images datasets. This has spurred great interest for systems that are able to efficiently retrieve images from these collections.

This task has been addressed by the so-called *Content-Based Image Retrieval* (CBIR) systems. In these systems, image content is represented by their low-level features, such as color, shape, and texture.

An ideal CBIR system should be effective and efficient. Effectiveness is achieved from image's abstract representations. In general, traditional approaches for this process often fail in presence of different illumination, occlusion, and viewpoint conditions. Efficiency, on the other hand, is achieved from the organization given for these representations. In general, data clustering approaches are one of the most useful techniques to reduce search space and speed up query processing.

To address effectiveness issues, this work presents SIFT-Texton, a new method to incorporate illumination, occlusion, and viewpoint conditions into low-level features. This approach is based on discrete distributions of local invariant features and low-level image properties.

With regard to efficiency issues, this work presents DAH-Cluster, a new clustering paradigm applied to CBIR. This approach combines features from both divisive and agglomerative hierarchical clustering paradigms. In addition, DAH-Cluster introduces a new concept, called factor of reclustering, that allows grouping similar elements that would be separated by traditional clustering paradigms.

Experiments show that the combination of these techniques allows the creation of a robust CBIR mechanism, achieving more effective and efficient results than traditional approaches in literature.

The main contributions of this work are: (1) a new method for image retrieval that incorporates illumination, occlusion, and viewpoint conditions into low-level features; and (2) a new data clustering paradigm that can be applied to information retrieval tasks.

Agradecimentos

Agradeço...

A Deus, por todas as oportunidades que Ele me oferece.

Aos meus pais Jurandy e Sônia, por me ensinarem a dar valor às coisas mais preciosas que a vida nos oferece.

A toda minha minha família pelo incentivo constante, pelo apoio irrestrito e por seus valiosos conselhos.

Ao meu orientador e co-orientador, Prof. Dr. Siome Klein Goldenstein e Prof. Dr. Ricardo da Silva Torres, pela oportunidade e pela orientação, cujos conselhos e sugestões foram muito valiosos tanto para este trabalho quanto para minha vida.

Ao meu companheiro Anderson, pelas sugestões que contribuíram para este trabalho.

Ao meu irmão Tiago, pelas correções que foram fundamentais para melhorar o texto.

Aos demais colegas de pós-graduação, pelas críticas e sugestões.

A todos os amigos que direta ou indiretamente contribuíram para este trabalho.

Aos professores e funcionários do Instituto de Computação, pelos serviços prestados.

A Unicamp, pela infra-estrutura.

A Fapesp, pelo apoio financeiro (processo 05/52959-3).

Sumário

Lista de Figuras

Lista de Abreviaturas

$11P$	Média da precisão em onze pontos
$3P$	Média da precisão em três pontos
BDI	Banco de Dados de Imagens
BIC	*Border/Interior pixel Classification*
CBC	*Color-Based Clustering*
CBIR	*Content-Based Image Retrieval*
CCV	*Color Coherence Vector*
CIE	*Commission Internationale de L'Éclairage*
DAHC	Modelo hierárquico divisivo e aglomerativo de agrupamento
EMD	*Earth Mover's Distance*
GCH	*Global Color Histogram*
DHC	Modelo hierárquico divisivo de agrupamento
HSV	*Hue, Saturation, and Value*
IRM	*Intergrated Region Matching*
JD	Divergência de Jeffrey
KL	Divergência de Kullback-Leibler
LCH	*Local Color Histogram*
MMM	*Markov Models Mediators*
MSHP	*Most Similar Highest Priority*

$P \times R$	Precisão vs. Revocação
$p(100)$	Precisão após 100 imagens recuperadas
$p(30)$	Precisão após 30 imagens recuperadas
$p(R)$	Precisão no primeiro ponto no qual a recuperação é igual 100%
PC	Modelo particional de agrupamento
PCA	*Principal Components Analisys*
QBE	*Query-By-Example*
$r(100)$	Revocação após 100 imagens recuperadas
$r(30)$	Revocação após 30 imagens recuperadas
RGB	*Red, Green, and Blue*
SIFT	*Scale Invariant Features Transform*
SGBD	Sistema de Gerenciamento de Bancos de Dados
WWW	*World Wide Web*

Capítulo 1

Introdução

Uma imagem vale mais do que mil palavras. Existe algo sobre a obra *"The Scream"* de Munch que nenhuma palavra pode expressar. Isso também ocorre quando se admiram as maravilhas da floresta amazônica, da divisão celular ou do nascimento de uma nova vida.

Bancos de dados de imagens (BDIs) têm se tornado cada vez mais freqüentes nos mais variados domínios de aplicações, tais como bibliotecas digitais [27, 43, 60], bancos de dados geográficos [64, 78] e bancos de dados médicos [33, 38]. A evolução das tecnologias de aquisição, transmissão e armazenamento de imagens têm permitido a construção de BDIs cada vez maiores. À medida em que cresce o volume de imagens nessas coleções, cresce também o interesse por sistemas capazes de recuperar essas imagens.

Os sistemas de recuperação baseiam-se em descrições compactas das imagens [72]. Essas descrições são normalmente distintas para imagens de domínios diferentes, mudando gradualmente quando o foco varia de um domínio específico (*narrow domain*) para um domínio geral (*broad domain*) [63]. Um domínio específico é composto por imagens que apresentam uma variabilidade limitada e previsível em todos os aspectos relevantes de sua aparência, por exemplo, em sistemas de sensoriamento remoto [17]. Um domínio geral, por outro lado, é formado por imagens que apresentam uma variabilidade ilimitada e imprevisível em seu conteúdo, tal como, no conjunto de imagens heterogêneas que compõem a *World Wide Web* (WWW) [65].

É possível descrever imagens utilizando-se atributos que são independentes de seu conteúdo visual, tais como o seu formato gráfico, o seu tamanho físico e a sua resolução. Esses atributos podem ser manipulados por sistemas gerenciadores de bancos de dados (SGBDs) [18, 56]. Entretanto, as consultas nesses sistemas ficam restritas aos atributos utilizados, dificultando o processo de recuperação, uma vez que não descrevem o conteúdo armazenado.

Uma alternativa a esses sistemas consiste em utilizar palavras-chave para descrever o conteúdo das imagens [44, 53]. Essa técnica requer uma anotação prévia de cada imagem,

uma tarefa que consome muito tempo. Além disso, esse processo normalmente é pouco eficaz, uma vez que a interpretação do conteúdo visual de uma imagem varia de acordo com o conhecimento, o objetivo, a experiência e a percepção de cada usuário [20].

Essas dificuldades têm sido endereçadas pelos sistemas de recuperação de imagens por conteúdo (CBIR – *Content-Based Image Retrieval*) [76]. Nesses sistemas, o conteúdo de uma imagem é descrito a partir de suas características visuais de baixo nível, tais como cor [54, 67, 70], forma [1, 3, 73] e textura [57, 75].

Um sistema de recuperação de imagens por conteúdo ideal deve ser eficaz e eficiente. A eficácia é resultado de representações abstratas das imagens, enquanto a eficiência vem da organização dessas representações. Assim, o maior desafio desses sistemas é minimizar o tempo de processamento de uma consulta e manter o resultado o mais eficaz possível.

A eficácia desses sistemas é resultado da tradução das percepções de alto nível de um usuário em características visuais de baixo nível de uma imagem. Esse processo requer o uso de descritores. Um descritor pode ser visto como um par, composto por (1) um algoritmo capaz de codificar as propriedades de baixo nível das imagens em vetores de características; e (2) uma medida de similaridade para comparar duas imagens a partir de seus vetores [72].

Dentre as características visuais de baixo nível que podem ser utilizadas na recuperação de imagens por conteúdo, a informação de cor é uma das mais amplamente utilizadas [9]. Essa preferência pela informação de cor se deve a alguns fatores [68]: (1) a cor é uma característica visual que é imediatamente percebida quando se olha uma imagem; (2) os conceitos envolvidos são simples de serem entendidos e implementados; (3) a informação de cor está presente na ampla maioria dos domínios de imagens; (4) os resultados obtidos utilizando a informação de cor são satisfatórios em geral e (5) a informação de cor pode ser processada de forma automática.

Este trabalho tem como foco a informação de cor. Em geral, os métodos de recuperação de imagens por conteúdo que manipulam essa informação normalmente falham na presença de diferentes condições de iluminação, oclusão e foco em imagens de domínio geral [42].

Essas informações são obtidas a partir de abstrações extraídas das propriedades de baixo nível. Essas abstrações constituem as chamadas informações de médio nível [22]. Muitas técnicas em processamento de imagens e em visão computacional são capazes de descrever informações de médio nível, como iluminação, oclusão e foco [51]. Entre essas técnicas, o SIFT (*Scale Invariant Features Transform*) [48] destaca-se em diversas aplicações, como por exemplo, no reconhecimento de objetos [47], no reconhecimento de panoramas [11] e na reconstrução tridimensional [10]. Esse método permite a extração características invariantes locais (*local invariant features*) de uma cena ou de um objeto.

Este trabalho apresenta o SIFT-Texton, um novo método para recuperação de imagens capaz de incorporar informações de médio nível nas características visuais de baixo nível.

Esse método baseia-se na distribuição discreta de características invariantes locais e em propriedades de baixo nível. Esse método visa a aumentar a eficácia dos sistemas de recuperação de imagens por conteúdo.

A eficiência de um sistema de recuperação por conteúdo, por outro lado, é resultado da organização dada aos vetores de características. Uma organização eficiente deve encontrar e recuperar imagens relevantes antes de imagens não relevantes, reduzindo o tempo de processamento de uma consulta.

O algoritmo mais simples para realizar uma consulta é uma busca sequencial. Nesse método, todas as imagens do banco são analisadas sequencialmente. Embora simples, esse método é inviável para grandes coleções, uma vez que o tempo gasto para processar uma consulta é proporcional ao tamanho do banco [68].

Existem extensivos estudos em técnicas indexação e estruturas de dados para acelerar o processo de consulta de uma imagem de forma que imagens relevantes possam ser encontradas rapidamente [15, 21]. Entretanto, na maioria dessas técnicas, a eficiência é comprometida pela sobreposição dos dados, característica comum na recuperação de imagens por conteúdo [31, 32].

Em contraste, agrupamento (*clustering*) é uma das técnicas de descoberta de conhecimento mais úteis para identificar correlações em grandes conjuntos de dados. Nesse método, os vetores de características associados às imagens são organizados em grupos, aos quais é associado um elemento representativo. Assim, uma consulta é realizada utilizando-se apenas um pequeno número de elementos representativos, reduzindo o espaço de busca [62].

Em geral, os algoritmos de agrupamento podem ser hierárquicos ou particionais. Os algoritmos hierárquicos encontram grupos sucessivos utilizando grupos pré-estabelecidos, enquanto os algoritmos particionais encontram todos os grupos de uma única vez [7].

As estratégias de agrupamento hierárquico podem ser divididas em dois paradigmas básicos: aglomerativos e divisivos. Uma estratégia aglomerativa começa com cada elemento como um grupo independente e os une, sucessivamente, em grupos cada vez maiores. Por outro lado, uma estratégia divisiva começa com um grande grupo e o divide, sucessivamente, em grupos menores [7].

Em relação às questões de eficiência, este trabalho apresenta o DAH-Cluster, um método hierárquico divisivo e aglomerativo. Esse método combina características dos modelos hierárquicos divisivo e aglomerativo de agrupamento, reduzindo os erros obtidos por cada um desses paradigmas e melhorando a qualidade das tarefas de agrupamento. Além disso, o DAH-Cluster introduz um novo conceito, chamado fator de reagrupamento, que permite agrupar elementos similares que seriam separados pelos paradigmas tradicionais.

Experimentos realizados utilizando diferentes coleções de imagens e diversos métodos de extração de características visuais, mostram que a utilização das técnicas propostas

neste trabalho na recuperação de imagens por cor garantem melhores resultados que os métodos descritos na literatura.

Assim, as principais contribuições deste trabalho são:

1. Especificação e implementação de um novo método para recuperar de imagens capaz de codificar informações de médio nível. Esse método é uma técnica genérica que pode ser aplicada a qualquer sistema de recuperação de imagens por conteúdo baseado em propriedades de baixo nível para incluir informações sobre iluminação, oclusão e foco.

2. Especificação e implementação de um novo paradigma de agrupamento de dados. Esse modelo é um método genérico que pode ser aplicado a qualquer sistema de recuperação de informação na redução do tempo de processamento de uma consulta.

O restante deste trabalho está organizado como se segue. O Capítulo 2 introduz a base teórica envolvida neste trabalho, incluindo conceitos sobre recuperação de imagens por conteúdo, métodos para recuperação de imagens por cor e paradigmas de agrupamento de dados. O Capítulo 3 apresenta o SIFT-Texton, a metodologia de validação utilizada e os resultados obtidos na aplicação desse método na recuperação de imagens por cor. O Capítulo 4 descreve o DAH-Cluster, incluindo a metodologia de validação utilizada e a descrição da sua aplicação na recuperação de imagens por cor. Por fim, o Capítulo 5 discute as conclusões e as possíveis extensões deste trabalho.

Capítulo 2

Conceitos básicos

Este capítulo apresenta a base teórica necessária para a compreensão deste trabalho. A Seção 2.1 introduz os conceitos básicos da recuperação de imagens por conteúdo. A Seção 2.2 discute técnicas de recuperação de imagens por cor. Por fim, a Seção 2.3 revisa métodos de agrupamento utilizados na recuperação de informação.

2.1 Recuperação de imagens por conteúdo

A recuperação de imagens por conteúdo é uma área multidisciplinar e envolve, principalmente, técnicas de banco de dados, processamento de imagens, recuperação de informação, reconhecimento de padrões e interfaces humano-máquina [63].

Esta seção introduz alguns conceitos fundamentais dessa área.

2.1.1 Imagem digital

Uma imagem (monocromática) é uma função bidimensional $I(x, y)$, onde x e y são coordenadas espaciais e o valor de I em qualquer ponto (x, y) é proporcional ao brilho (ou nível de cinza) da imagem nesse ponto [22]. Uma imagem digital nada mais é que uma imagem $I(x, y)$ que teve tanto as suas coordenadas espaciais quanto o seu brilho discretizados (digitalizados). Dessa forma, uma imagem digital pode ser interpretada como uma matriz na qual cada elemento é identificado pelos índices da linha e da coluna às quais pertence, cujo valor corresponde ao seu brilho ou nível de cinza. Os elementos dessa matriz são conhecidos como pixels (*picture elements*) [22].

A digitalização das coordenadas espaciais é conhecida como amostragem (*image sampling*) e a digitalização do brilho é conhecida como quantização do nível de cinza (*gray-level quantization*) [22]. A resolução de uma imagem (o grau de detalhes perceptíveis) é fortemente dependente desses dois parâmetros. Quanto mais finas a amostragem e a

quantização, melhor a imagem digitalizada se aproxima do conteúdo da imagem original. No entanto, os custos de armazenamento e de processamento da imagem digital crescem rapidamente com o aumento da resolução [22].

No caso de imagens digitais coloridas, cada pixel é descrito não apenas pelo seu brilho, mas também por outras propriedades como matiz e saturação. Em geral, a cor de cada pixel é representada como um ponto em um sistema de coordenadas 3D conhecido como espaço de cor. Um exemplo de espaço de cor é o espaço RGB (*Red, Green, and Blue*), no qual cada cor é representada como uma combinação de três cores primárias (vermelho, verde e azul) [22].

Um pixel p com coordenadas espaciais (x, y) possui quatro vizinhos no espaço (horizontais e verticais) cujas coordenadas são [22]:

$$N_4(p) = \{(x+1, y), (x-1, y), (x, y+1), (x, y-1)\} \tag{2.1}$$

Adicionalmente, é possível definir outros quatros vizinhos (diagonais) [22]:

$$N_D(p) = \{(x+1, y+1), (x+1, y-1), (x-1, y+1), (x-1, y-1)\} \tag{2.2}$$

A união dos conjuntos anteriores determina um total de oito vizinhos [22]:

$$N_8(p) = N_4(p) \cup N_D(p) \tag{2.3}$$

Um caminho entre dois pixels p e q cujas coordenadas espaciais são (x, y) e (r, s) é uma seqüência de pixels distintos com coordenadas:

$$(x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n) \tag{2.4}$$

onde $(x_0, y_0) = (x, y)$ e $(x_n, y_n) = (r, s)$, (x_i, y_i) é adjacente a (x_{i-1}, y_{i-1}) de acordo com algum critério de adjacência (por exemplo, considerando-se quatro vizinhos por pixel), $0 < i \leq n$, e n é o tamanho do caminho [22].

Se p e q são pixels que pertencem a um subconjunto S de pixels da imagem, então p é conexo a q em S se existe um caminho entre p e q formado apenas por pixels que pertencem a S. Para qualquer pixel $p \in S$, o conjunto de todos os pontos que são conexos a p em S é conhecido como uma componente conexa de S. Assim, dois pixels quaisquer de uma mesma componente conexa são conexos entre si e duas componentes conexas diferentes são disjuntas [22].

2.1.2 Arquitetura típica de um sistema CBIR

A Figura 2.1 mostra a arquitetura típica de um sistema de recuperação de imagens por conteúdo [72]. Duas funcionalidades principais devem ser suportadas por essa arquitetura: a inserção de dados e o processamento de consultas.

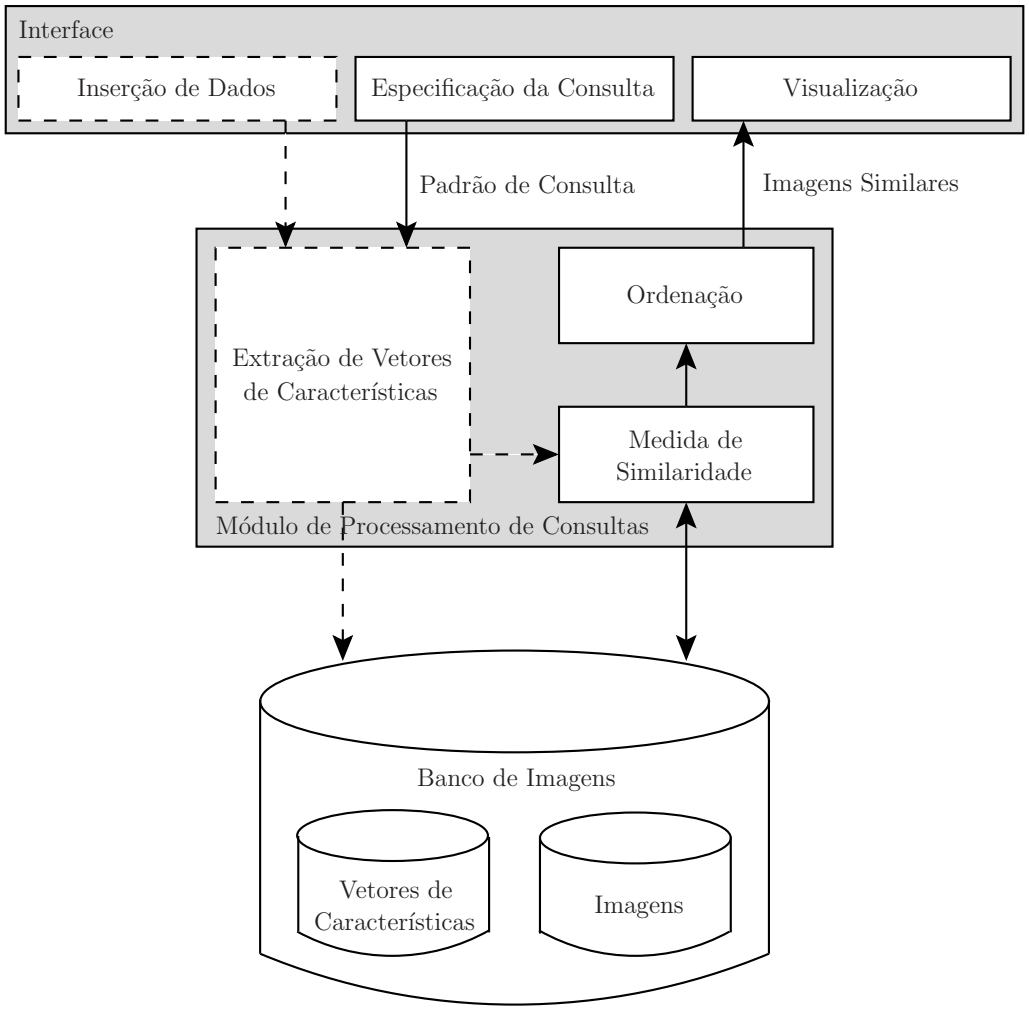


Figura 2.1: Arquitetura típica de um sistema de recuperação de imagens por conteúdo [72].

A inserção de dados é responsável por extrair características visuais apropriadas das imagens e armazená-las em um banco de imagens (módulos e setas tracejadas). Esse processo é normalmente realizado *offline* [72].

O processamento de consultas, por outro lado, é organizado como segue: a interface permite um usuário especificar um padrão de consulta e visualizar as imagens similares recuperadas. O módulo de processamento de consultas extrai um vetor de características desse padrão de consulta e aplica uma função de distância para avaliar a sua similaridade em relação às imagens do banco. A seguir, ordenam-se essas imagens em ordem decrescente de similaridade em relação ao padrão de consulta especificado e enviam-se as imagens mais similares para o módulo de interface [72].

2.1.3 Descritores de imagens

Uma solução típica de um sistema de recuperação de imagens por conteúdo requer a construção de um descritor. Um descritor pode ser visto como um par, composto por: (1) um algoritmo de extração capaz de codificar as propriedades visuais das imagens em vetores de características; e (2) uma medida de similaridade para comparar duas imagens a partir desses vetores [72].

Formalmente, um vetor de características $\vec{v}_I$ de uma imagem I pode ser visto como um ponto em um espaço $\mathbb{R}^n$: $\vec{v}_I = (v_1, v_2, \ldots, v_n)$, onde n é a dimensão do vetor. Esses vetores codificam características visuais das imagens, tais como cor, forma e textura. Note que diferentes tipos de vetores de características podem exigir diferentes medidas de similaridade [72].

Um descritor D é definido como uma tupla $(\varepsilon_D, \delta_D)$, onde [72]:

- $\varepsilon_D : I \rightarrow \mathbb{R}^n$ é uma função que extrai um vetor de características $\vec{v}_I$ da imagem I.
- $\delta_D : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$ é uma função que computa a similaridade entre duas imagens.

A Figura 2.2 ilustra o uso de um descritor simples D para computar a similaridade entre duas imagens I_A e I_B. Primeiro, o algoritmo de extração ε_D é utilizado para computar os vetores de características $\vec{v}_{I_A}$ e $\vec{v}_{I_B}$ associados às imagens. A seguir, a função de similaridade δ_D é utilizada para determinar a similaridade s entre as imagens I_A e I_B. Eventualmente, diversos descritores podem ser combinados em um descritor complexo, que pode codificar diversas propriedades das imagens ao mesmo tempo [74].

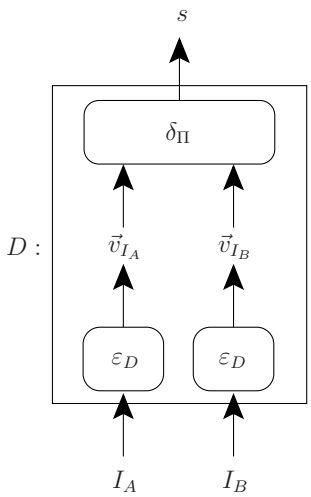


Figura 2.2: Uso de um descritor simples para comparar duas imagens [72].

2.1.4 Funções de distância

Uma busca em um sistema de recuperação imagens por conteúdo, ao contrário de uma busca exata em bancos de dados tradicionais, visa a encontrar imagens próximas ou similares a uma dada imagem de consulta. Esse processo requer uma medida de similaridade para comparar e ordenar imagens a partir de seus vetores de características.

Uma medida de similaridade é uma função de comparação que fornece o grau de similaridade entre duas imagens quaisquer. Essa função é normalmente definida como a inversa da função de distância utilizada para comparar duas imagens, na qual, quanto maior o valor de distância, menos similares são as imagens [72].

Quanto melhor uma função de distância simula as percepções humanas de similaridade utilizando as características visuais disponíveis, mais eficaz é o sistema para recuperar imagens relevantes de acordo com as necessidades do usuário [68].

Uma função de distância d utilizada para comparar imagens é uma métrica se, para quaisquer imagens A, B e C, as seguintes propriedades são satisfeitas [9]:

- Positividade – $d(A, B) \geq 0$
- Simetria – $d(A, B) = d(B, A)$
- Identidade – $d(A, A) = 0$
- Desigualdade triangular – $d(A, C) \leq d(A, B) + d(B, C)$

A desigualdade triangular é a propriedade mais importante utilizada pelos métodos de indexação para reduzir o espaço de busca de uma consulta. Dessa forma, a função de distância utilizada para comparar duas imagens afeta diretamente o tempo gasto para processar uma consulta visual (eficiência) e a qualidade do processo de recuperação (eficácia) [68].

Esta seção introduz algumas funções de distância utilizadas em sistemas de recuperação de imagens por conteúdo. Nas seções seguintes, $d(A, B)$ denota a função de distância entre uma imagem de consulta A e uma imagem do banco B; e $\vec{v}_A = \{a_1, a_2, \ldots, a_m\}$ e $\vec{v}_B = \{b_1, b_2, \ldots, b_n\}$ representam vetores de características associados as imagens A e B, respectivamente.

Distância de Minkowski

Se as dimensões dos vetores de características $\vec{v}_A$ e $\vec{v}_B$ forem iguais, independentes entre si e de igual relevância, então as funções de distância de Minkowski L_p são apropriadas para comparar as características visuais dessas imagens [46].

Uma função de distância da família L_p é definida como [9]:

$$d(A,B) = \left(\sum_i |a_i - b_i|^p \right)^{\frac{1}{p}} \tag{2.5}$$

Alguns membros conhecidos da família L_p são as seguintes funções de distância [9]:

- L_1 (Manhattan): $d(A,B) = \sum_i |a_i - b_i|$
- L_2 (Euclidiana): $d(A,B) = \sqrt{\sum_i |a_i - b_i|^2}$
- L_∞ (Chebyshev): $d(A,B) = \max_i |a_i - b_i|$

Distância quadrática

As funções de distância de Minkowski tratam todas as entradas dos vetores de características de maneira independente, não levando em conta o fato que alguns pares dessas entradas correspondem a características mais similares que outros [46].

Esse problema pode ser resolvido por meio de uma função de distância quadrática, que é definida como [52]:

$$d(A,B) = \sqrt{(\vec{v}_A - \vec{v}_B)^T S (\vec{v}_A - \vec{v}_B)} \tag{2.6}$$

onde $S = [s_{ij}]$ é uma matriz de similaridade, simétrica e positiva definida, e s_{ij} denota a similaridade entre as entradas i e j do conjunto de vetores de características. A distância Euclidiana L_2 é um caso particular dessa função de distância no qual $S = I$.

Distância de Mahalanobis

A função de distância de Mahalanobis é apropriada quando as dimensões dos vetores de características são iguais, dependentes entre si e com diferentes graus de relevância. Essa função é definida como [46]:

$$d(A,B) = \sqrt{(\vec{v}_A - \vec{v}_B)^T C^{-1} (\vec{v}_A - \vec{v}_B)} \tag{2.7}$$

onde C é a matriz de covariância do conjunto de vetores de características.

Distância de Kolmogorov-Smirnov

Em estatística, a distância de Kolmogorov-Smirnov é uma medida utilizada para avaliar se duas distribuições diferem uma da outra. Essa função de distância é definida como [58]:

$$d(A,B) = \max_i \left| \sum_{j=0}^{i} (a_i - b_i) \right| \tag{2.8}$$

Histogram Intersection

Em estatística, um histograma é uma representação gráfica de uma distribuição, que apresenta agrupamentos de um conjunto de dados em células ou *bins*. Se os vetores de características $\vec{v}_A$ e $\vec{v}_B$ constituem dois histogramas, eles podem ser comparados utilizando a função de distância proposta por Swain e Ballard [70], denominada *Histogram Intersection*. Essa função de distância é definida como:

$$d(A,B) = 1 - \frac{\sum_i \min(a_i, b_i)}{\sum_i b_i} \tag{2.9}$$

Divergências de Kullback-Leibler e de Jeffrey

A divergência de Kullback-Leibler (KL) [39] é uma função de distância que mede o grau de ineficiência médio obtido ao codificar um histograma utilizando outro como referência. Essa função de distância é definida como:

$$d(A,B) = \sum_i a_i \log \frac{a_i}{b_i} \tag{2.10}$$

A divergência de Jeffrey (JD) é uma modificação da divergência KL que é numericamente estável, simétrica e robusta em relação a ruídos [55]. Essa função de distância é definida como:

$$d(A,B) = \sum_i \left(a_i \log \frac{a_i}{m_i} + b_i \log \frac{b_i}{m_i} \right) \tag{2.11}$$

onde $m_i = \frac{a_i + b_i}{2}$.

Integrated Region Matching

Li et al. [41] propuseram o *Integrated Region Matching* (IRM), uma função de distância que mede a dissimilaridade entre dois subconjuntos do $\mathbb{R}^n$ por meio da soma ponderada das distâncias entre pares de elementos desses conjuntos.

Esse processo é realizado como segue. Inicialmente, o método obtém a distância entre todos os pares de elementos desses conjuntos. Iterativamente, cada par é associado a um peso que indica a sua importância em relação à dissimilaridade total. Esse peso é estabelecido de acordo com um critério predefinido, que limita o peso total que pode ser associado ao sistema. A escolha do par analisado em cada iteração segue o princício MSHP (*Most Similar Highest Priority*), que prioriza pares de elementos com menor distância. Esse processo termina quando o peso total é atingido.

A dissimilaridade entre as imagens A e B é dada por [41]:

$$d(A,B) = \sum_{a \in \vec{v}_A} \sum_{b \in \vec{v}_B} w_{a,b} d_{a,b} \tag{2.12}$$

onde $w_{a,b}$ é peso associado ao par (a, b) e $d_{a,b}$ denota a distância entre a entrada a de $\vec{v}_A$ e a entrada b de $\vec{v}_B$.

Earth Mover's Distance

Earth Mover's Distance (EMD) [58] é uma função de distância que mede o custo mínimo que deve ser pago para transformar uma distribuição em outra. Intuitivamente, dadas duas distribuições, uma pode ser vista como uma massa de terra espalhada no espaço e a outra como uma coleção de buracos nesse mesmo espaço. Então, EMD mede o esforço mínimo necessário para preencher os buracos com terra [58].

Esse processo é modelado como um problema de transporte [6]. Suponha que diversos fornecedores, cada um com uma quantidade pré-estabelecida de produtos, devem atender a diversos consumidores, cada um com uma capacidade limitada de consumo. Para cada par fornecedor-consumidor existe um custo para transportar cada unidade de produto. O problema de transporte consiste em encontrar o fluxo de produtos dos fornecedores aos consumidores que minimize os custos de transporte e satisfaça a demanda [58].

A comparação entre as imagens A e B pode ser tratada como um problema de transporte, no qual $\vec{v}_A$ representa os fornecedores e $\vec{v}_B$, os consumidores; e o custo de cada par fornecedor-consumidor é igual à distância entre duas entradas desses vetores. Dessa forma, essa função de distância mede o esforço mínimo (normalizado) π exigido para transformar A em B [58]:

$$d(A, B) = \min_{\pi:\vec{v}_A \to \vec{v}_B} \sum_{a \in \vec{v}_A} \mathcal{D}(a, \pi(a)) \tag{2.13}$$

onde $\mathcal{D}$ é a função que mede a distância entre uma entrada de $\vec{v}_A$ e uma entrada de $\vec{v}_B$.

2.1.5 Medidas de avaliação de desempenho

O tempo de resposta e o espaço de armazenamento são as medidas normalmente adotadas para avaliar um sistema de recuperação. No domínio de recuperação de informação, entretanto, é necessário avaliar a relevância da informação recuperada (eficácia) [68].

Em sistemas de recuperação de documentos, existem diversas coleções de referência disponíveis (por exemplo, CACM, ADI, INSPEC) e mesmo uma conferência (TREC) dedicada ao assunto [4]. Dessa forma, há uma série de experimentos, procedimentos e pesquisadores interessados em comparar seus resultados utilizando uma estrutura comum. Infelizmente, no domínio dos sistemas de recuperação de imagens por conteúdo a situação é diferente. A comparação entre diversos sistemas é difícil de ser realizada, uma vez que grupos distintos conduzem seus experimentos focando em aspectos distintos do processo de recuperação [68].

A eficácia de um sistema de recuperação é uma medida relacionada à satisfação do usuário com o resultado obtido. Para estabelecer uma medida de eficácia, a primeira decisão a ser tomada é definir quais julgamentos são permitidos ao usuário em sua avaliação [37]. A escolha básica é entre uma medida binária e uma medida de múltipla escolha. A medida binária é a mais simples de ser implementada e utilizada, na qual cada imagem pode ser aceita ou rejeitada. Essas condições estão normalmente ligadas à relevância de uma imagem para um usuário. Entretanto, essas condições apresentam um alto grau de subjetividade, uma vez que diferentes usuários, ou os mesmos usuários em diferentes circunstâncias, podem perceber o conteúdo visual de uma imagem de maneiras diferentes. Um sistema é eficaz se as medidas de avaliação fornecem resultados satisfatórios utilizando um critério de relevância externo [4].

As medidas de avaliação de desempenho utilizadas no sistemas de recuperação de imagens por conteúdo, em geral, foram originalmente desenvolvidas para avaliar sistemas de recuperação de documentos. A utilização dessas medidas é aceitável, uma vez que o propósito principal em ambos sistemas é avaliar a informação recuperada de acordo com um julgamento externo de relevância [68].

Precisão e revocação

Entre as diversas medidas de avaliação de desempenho existentes, a curva *Precisão vs. Revocação* ($P \times R$) [4, 9, 37] é a medida mais conhecida e utilizada na prática. Quando uma escala binária é utilizada tanto para a relevância quanto para a abrangência, uma tabela de contingência (Tabela 2.1) pode ser estabelecida, mostrando como as imagens do banco estão divididas de acordo com essas duas classificações [37].

	Recuperadas	Não recuperadas
Relevantes	A	B
Não relevantes	C	D

Tabela 2.1: Tabela de contingência para avaliar a eficácia de recuperação.

De acordo com essa tabela de contingência, a precisão P_r é a fração das r imagens recuperadas que são relevantes para a consulta [37]:

$$P_r = \frac{A}{r} = \frac{A}{A + C} \tag{2.14}$$

A revocação R_r é a proporção do número total de imagens relevantes que foram recuperadas entre as r imagens retornadas [37]:

$$R_r = \frac{A}{A + B} \tag{2.15}$$

Se 50 imagens são recuperadas como resposta a uma consulta e 35 delas são relevantes, a precisão para $r = 50$ é $P_{50} = 70\%$. Se, nessa mesma consulta, há 70 imagens dentro dessa coleção que são relevantes, a revocação para $r = 50$ é $R_{50} = 50\%$, uma vez que 35 das 70 imagens relevantes foram selecionadas dentro das primeiras 50 imagens recuperadas (*top-50*).

Medidas de valor único

Alguns pesquisadores acreditam que uma medida de eficácia de recuperação deve ser expressa por meio de único número, capaz de representar valores relativos e absolutos [4]. Essa medida pode ser obtida utilizando um único ponto das curvas $P \times R$ para avaliar a eficácia de um descritor, por exemplo [4]: (1) a precisão no ponto mínimo no qual a revocação é igual a 100% (*R-value*), (2) a precisão após a primeira imagem relevante ser recuperada, (3) a precisão em um ponto fixo de revocação ou (4) a precisão após r imagens serem recuperadas (*top-r*).

É também comum a análise de um único número que caracteriza a eficácia de todas as consultas, por exemplo, a precisão média de um número fixo de pontos de revocação. Essas medidas alternativas, garantem uma informação específica sobre os resultados e, portanto, apresentam um limitado contexto de aplicação. Entretanto, a utilização de diversas medidas além de um gráfico $P \times R$ permite caracterizar um processo de recuperação sob diferentes pontos de vista [68].

2.2 Recuperação de imagens por cor

A Figura 2.3 mostra a representação esquemática de uma imagem sendo armazenada em um sistema que faz uso da informação de cor para descrever, representar, comparar e recuperar imagens. Após uma imagem ser fornecida como entrada, o seu conteúdo visual é analisado e resumido em um espaço de cores pré-estabelecido. Em seguida, são extraídas características visuais a partir das informações de cor. Por fim, representações compactas são escolhidas para as informações analisadas durante a etapa anterior. Essas representações determinam vetores de características que são armazenados e indexados em um banco de imagens.

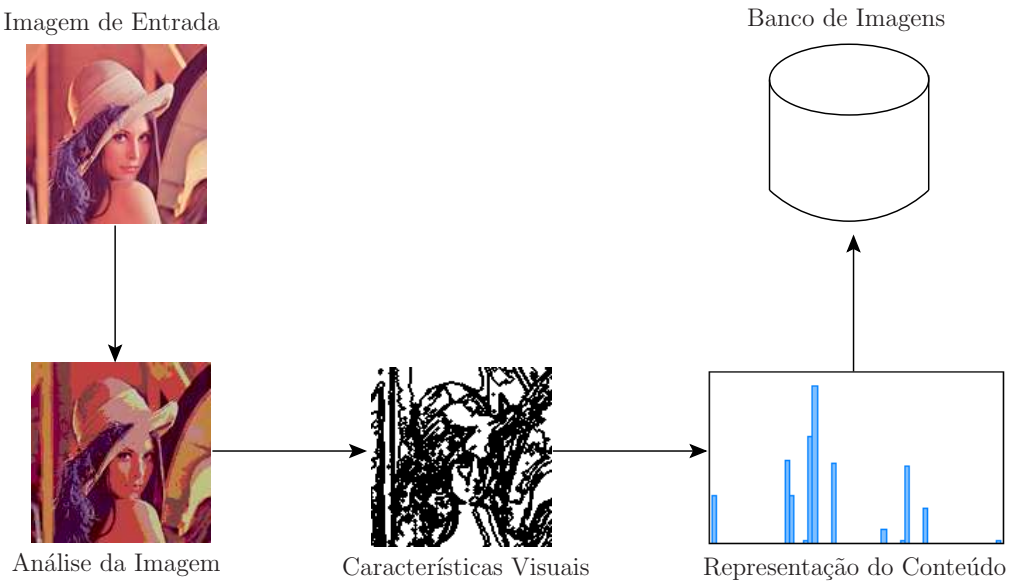


Figura 2.3: Representação esquemática de uma imagem sendo armazenada em um BDI.

De acordo com o esquema descrito acima, pode-se considerar a existência de três tópicos chave que precisam ser explorados para que se realize um processo automático de recuperação de imagens por cor: (1) qual espaço de cor deve ser utilizado para descrever, analisar e comparar imagens (Seção 2.2.1); (2) como descrever imagens por meio da sua distribuição de cores (Seção 2.2.2); e (3) como representar o conteúdo de uma imagem (características visuais) em um banco de imagens (Seção 2.2.3). As seções seguintes discutem cada um desses tópicos.

2.2.1 Espaços de cor

A cor de um pixel é representada por três valores, um para cada canal de um determinado espaço de cor. Um espaço de cor é uma especificação de um sistema de coordenadas 3D e um subespaço dentro desse sistema, no qual cada cor é representado por um único ponto [22]. A escolha de um espaço de cor no qual as imagens serão representadas, analisadas e comparadas é o primeiro passo em qualquer sistema de recuperação de imagens por cor. Os espaços de cor existentes podem ser classificados em três categorias principais [22]: (1) orientado ao *hardware*, (2) orientado ao usuário e (3) espaços de cor uniformes.

Um modelo orientado ao *hardware* é definido de acordo com as propriedades dos dispositivos utilizados para reproduzir as cores (monitores, impressoras coloridas, etc) [22]. O espaço de cor mais conhecido e mais utilizado é um modelo orientado ao *hardware* chamado RGB (*Red, Green, and Blue*) [9]. O espaço de cor RGB é dependente do dispositivo utilizado, isto é, a cor exibida depende não somente dos valores RGB, mas também

das especificações do dispositivo que o exibe. A percepção do espaço de cor RGB não é uniforme, pois as diferenças entre as cores RGB não refletem as diferenças percebidas pelos humanos. O espaço de cor RGB é um cubo, como mostrado na Figura 2.4, no qual a diagonal principal representa valores de cinza do preto (S) ao branco (W) e qualquer ponto (cor) nesse cubo é representado pela soma ponderada de vermelho (R), verde (V) e azul (B).

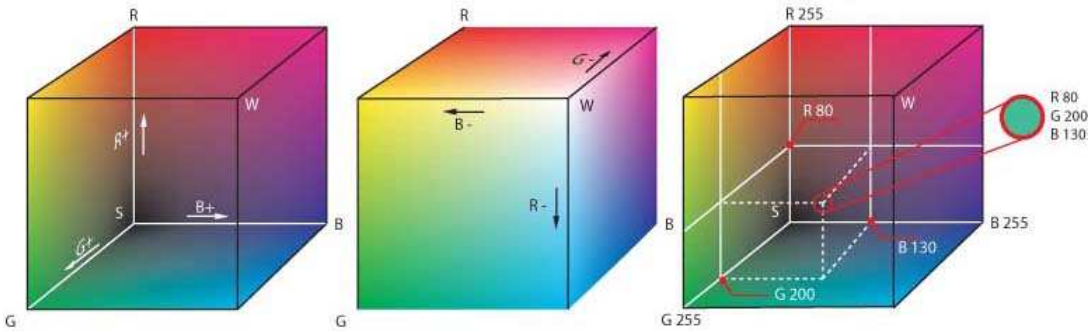


Figura 2.4: O espaço de cor RGB.

Os espaços de cor orientados ao usuário baseiam-se na percepção humana das cores [22]. Eles exploram as características que são utilizadas por humanos para distinguir uma cor de outra, como matiz (o comprimento de onda dominante que produz a sensação visual de vermelho, amarelo, verde, azul, ou a combinação de duas dessas cores), saturação (o grau de pureza da cor, o qual está relacionado ao desvio padrão em relação ao comprimento de onda dominante) e a intensidade (o brilho da cor, que está relacionado à quantidade de branco que ela apresenta) [22]. O espaço de cor HSV (*Hue, Saturation, and Value*) [9] é um exemplo de espaço de cor orientado ao usuário. O espaço de cor HSV é representado por um cone (Figura 2.5). O eixo vertical desse cone representa os valores de intensidade (I), o ângulo formado por esse eixo define a matiz (H) e a distância desse eixo fornece a saturação (S).

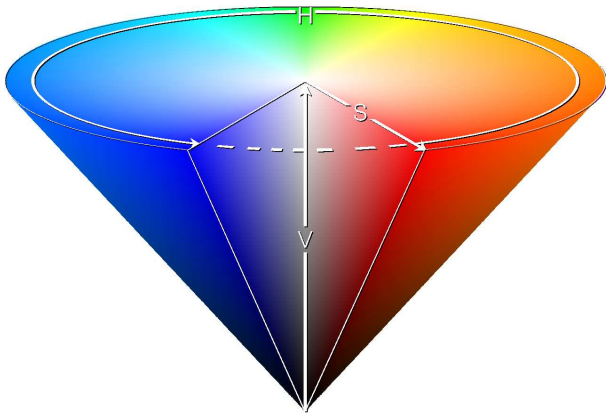


Figura 2.5: O espaço de cor HSV.

Os espaços de cor uniformes são espaços nos quais as diferenças numéricas entre as cores refletem as diferenças percebidas pelos humanos [22]. O espaço de cor Lab da CIE[1] (*Commission Internationale de L'Éclairage*) [9] é um exemplo de espaço de cor uniforme. O modelo CIE Lab representa as diferenças de três pares elementares: vermelho-verde, amarelo-azul e branco-preto. Dessa forma, como mostra a Figura 2.6, o eixo a do espaço de cor CIE Lab se estende do verde $(-a)$ ao vermelho $(+a)$ e o eixo b do azul $(-b)$ ao amarelo $(+b)$. O eixo L varia o brilho do preto $(-L)$ ao branco $(+L)$.

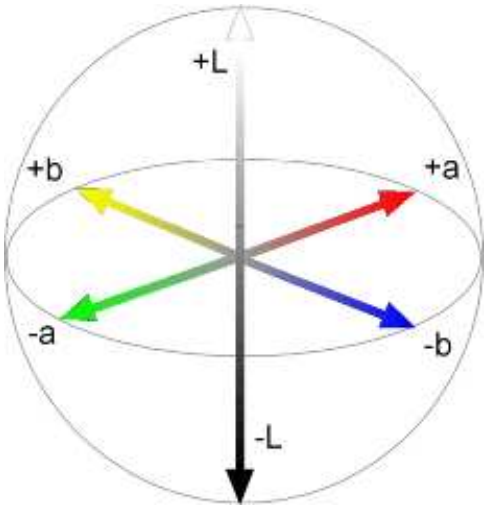


Figura 2.6: O espaço de cor CIE Lab.

[1] *Commission Internationale de L'Éclairage* (CIE) – `http://www.cie.co.at/cie/`

2.2.2 Redução do espaço de cor

A informação de cor em imagens digitais poderia ser representada no nível dos pixels, mas isso tornaria os sistemas de recuperação de imagens por cor inviáveis. Suponha, por exemplo, que cada canal de cor do espaço RGB seja representado utilizando 8 *bits*, dessa forma, é possível representar $2^8 = 256$ níveis distintos para cada componente de cor, resultando em $256 \times 256 \times 256 = 16.777.216$ cores distintas. Além disso, considere uma imagem com dimensões espaciais 300×300, isso significa que 90.000 pontos devem ser considerados em uma análise comparativa pixel a pixel de duas imagens. Esses valores (quantidade de cores e dimensão espacial) são grandes o suficiente para impedir a comparação de imagens no nível dos pixels [68].

Dessa forma, um sistema de recuperação de imagens por cor requer uma representação compacta da distribuição da cor e do espaço. A distribuição das cores indica a porcentagem de cada cor em uma imagem, enquanto a distribuição do espaço indica em quais regiões da imagem uma dada cor aparece. Essa representações podem ser reduzidas por meio de métodos estáticos ou dinâmicos. Os métodos estáticos utilizam um esquema fixo para todas as imagens, enquanto os métodos dinâmicos exploram o conteúdo visual da imagem para produzir representações mais compactas [68].

Métodos estáticos

O esquema mais simples para reduzir o número de cores presentes em uma imagem é utilizar uma quantização uniforme de cada canal de cor [68]. Por exemplo, ao invés de utilizar 8 *bits* para representar cada canal de cor, podem-se usar os dois *bits* mais significativos, quantizando uniformemente cada canal em $2^2 = 4$ níveis distintos, obtendo um total de $4 \times 4 \times 4 = 64$ cores distintas. Isso facilita o processo de comparação, pois o número de cores é igual para todas as imagens do banco.

Entretanto, a quantização uniforme de um espaço de cor apresenta algumas desvantagens bem conhecidas. Primeiro, as cores presentes em uma imagem não estão uniformemente distribuídas em um espaço de cor. Segundo, é difícil obter uma granularidade adequada para a quantização. Ela deve ser fina o suficiente para não agrupar cores distintas, mas grossa o bastante para distinguir as cores presentes em uma imagem. Por fim, a quantização uniforme não é apropriada para espaços de cor não uniformes, como RGB e HSV, uma vez que cores similares podem ser separadas e cores não similares podem ser agrupadas [68].

Uma alternativa para evitar uma etapa de quantização estática é reduzir a informação de cor por meio de estatísticas sobre a distribuição cores, por exemplo, a cor média. Esses métodos têm a vantagem de ser computacionalmente simples, resultar em descritores bastante compactos e garantir uma forma eficiente para comparar imagens. Porém,

sua eficácia é normalmente baixa porque imagens compostas por cores completamente diferentes podem resultar em estatísticas idênticas [68].

Esquemas de quantização estática também podem ser utilizados para reduzir a distribuição espacial das cores. Isso corresponde a reduzir a resolução da imagem por meio de uma amostragem dos pixels ou impondo uma grade de células sobre a imagem, de forma que a distribuição de cor em cada célula seja processada individualmente [68].

Métodos dinâmicos

Os métodos dinâmicos exploram o conteúdo visual das imagens para reduzir, simultaneamente, a distribuição da cor e do espaço. Esses métodos utilizam técnicas de segmentação de imagens para agrupar uma vizinhança de pixels com cor similar. Cada grupo representa uma região da imagem cuja cor é a média das cores do pixels que a compõe. Dessa forma, o número de cores distintas presentes na imagem original é reduzida. Simultaneamente, a imagem é segmentada em regiões com um alto grau de similaridade de cor e bem definidas em localização, tamanho e forma. Essas características são mais compactas e significativas que a posição de cada pixel da imagem isoladamente [68].

Em geral, os métodos dinâmicos utilizam uma das seguintes técnicas de segmentação de imagens [68]: detecção de bordas [22], crescimento de regiões [22], divisão de regiões [22], estimação de densidade [13] e agrupamento hierárquico [66].

A detecção de bordas assume que a transição entre duas regiões pode ser determinada por meio de propriedades visuais de descontinuidade. O crescimento de regiões começa com um conjunto de pontos sementes e, a partir desses pontos, cresce regiões através do agrupamento de pixels vizinhos com cor similar. A divisão de regiões subdivide uma imagem em um conjunto arbitrário de regiões disjuntas e, então, agrupa ou separa essas regiões de acordo com sua cor. A estimação de densidade baseia-se na suposição de que a densidade dos dados é uma mistura de densidades gaussianas. A média e a covariância dessas gaussianas são utilizadas para particionar os dados em regiões. Por fim, o agrupamento hierárquico utiliza métodos de agrupamento para agrupar uma vizinhança de pixels com cor similar.

2.2.3 Extração de características visuais de cor

Um vez escolhida uma representação compacta para a informação de cor presente em uma imagem, o próximo passo em um sistema de recuperação de imagens por cor consiste em representar essa informação em um banco de imagens. As abordagens existentes para esse processo podem ser classificadas em [68]: (1) globais [54, 67, 70], (2) baseadas em particionamento [69] e (3) regionais [66]. Essa classificação baseia-se no tipo de representação adotada para as características visuais extraídas das imagens.

Cada uma dessas três categorias oferece vantagens de desvantagens distintas relacionadas à complexidade dos algoritmos de análise das imagens, à utilização de espaço em disco para representar suas características visuais, à complexidade da função de distância utilizada para comparar as características visuais extraídas e, finalmente, à eficácia do processo de recuperação das imagens. É importante observar que cada categoria possui características desejáveis e também limitações bem conhecidas [68].

Abordagens globais

As abordagens globais [54, 67, 70] descrevem a distribuição de cores das imagens como um todo, desprezando a sua distribuição espacial. Em geral, essas abordagens são as mais eficientes (menor tempo de processamento e espaço de armazenamento) em termos de extração, representação e comparação das características visuais [68].

Abordagens baseadas em particionamento

As abordagens baseadas em particionamento [69] decompõem espacialmente as imagens utilizando uma estratégia de particionamento simples e comum a toda imagem, sem levar em consideração o seu conteúdo visual. Por exemplo, cada imagem é dividida em regiões retangulares de mesmo tamanho. A distribuição de cor de cada partição é descrita individualmente. O objetivo do particionamento espacial é adicionar informação de como as cores estão espacialmente distribuídas dentro da imagem. Assim como em abordagens globais, a extração das características visuais é bastante eficiente. No entanto, a representação e a comparação das imagens ficam computacionalmente mais caras, uma vez que o conteúdo de cada partição é representado e comparado individualmente [68].

Abordagens regionais

As abordagens regionais [66] utilizam técnicas automáticas de segmentação para decompor as imagens de acordo com o seu conteúdo visual. O número de regiões obtidas, assim como seu tamanho, forma e localização variam para cada imagem. Nesse contexto, o objetivo da segmentação não é, necessariamente, segmentar de maneira precisa todos os objetos presentes, mas decompor uma imagem em regiões cujos pixels possuem um alto grau de similaridade de acordo com alguma propriedade visual pré-estabelecida. As abordagens regionais utilizam algoritmos complexos (computacionalmente caros) para segmentar e comparar imagens, limitando o seu potencial de aplicação [68].

2.2.4 Descritores de cor

Esta seção apresenta algumas técnicas conhecidas para recuperação de imagens por cor. Esses métodos foram utilizados como referência nos experimentos realizados neste trabalho.

Global Color Histogram

O método mais simples para codificar a informação de cor presente em uma imagem é o *Global Color Histogram* (GCH) [70]. Um GCH é um conjunto de valores ordenados, um para cada cor distinta, que representa a probabilidade de um pixel ser de uma determinada cor. Esse método utiliza técnicas de quantização uniforme e de normalização para reduzir o número de cores distintas e evitar problemas relacionados à escala.

A implementação mais comum do GCH utiliza um espaço de cor RGB uniformemente quantizado em 64 cores distintas e utiliza a função de distância L_1 para analisar a dissimilaridade entre histogramas. Os histogramas são eficazes para recuperação, uma vez que capturam o padrão de cor presente em uma imagem. A Figura 2.7 mostra um exemplo de uma imagem descrita por meio dessa representação.

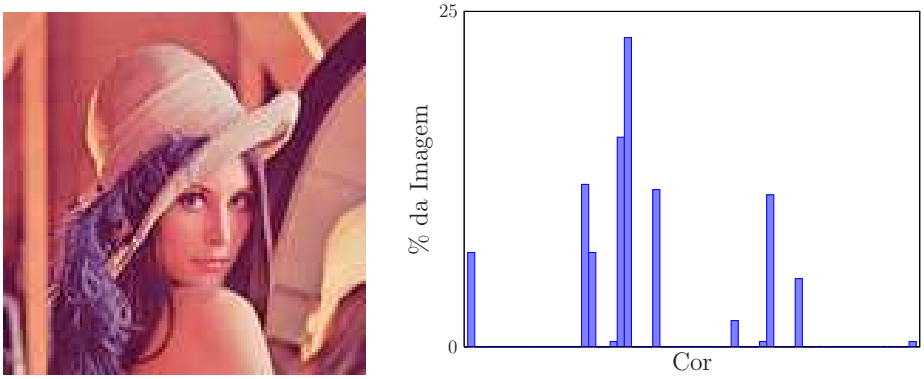


Figura 2.7: Exemplo de uma imagem e seu histograma global.

Local Color Histogram

Um *Local Color Histogram* (LCH) [49] é, provavelmente, o método baseado em particionamento mais simples. Ele decompõe uma imagem em células de tamanhos iguais e, o contéudo de cada célula é descrito individualmente através de um histograma local de cor. A Figura 2.8 mostra um exemplo de uma imagem dividida em quatro células e seus respectivos histogramas locais.

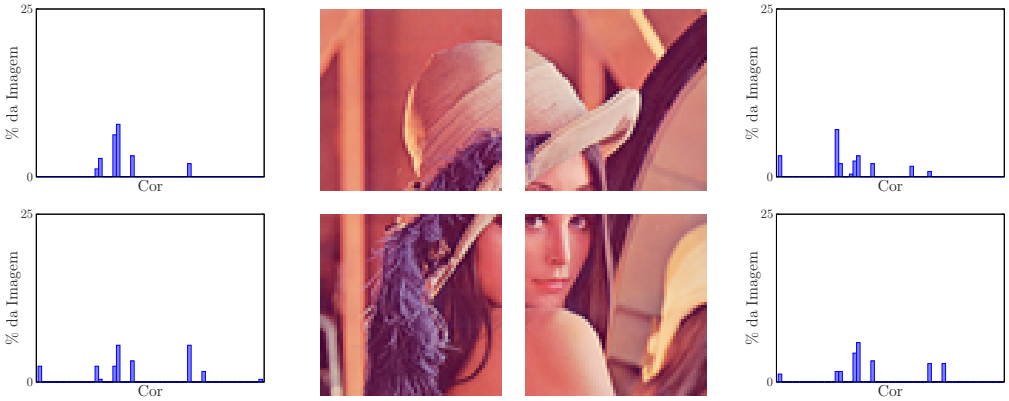


Figura 2.8: Exemplo de uma imagem dividida em quatro células e seus respectivos histogramas locais.

Formalmente, o conteúdo de uma imagem é representado por uma matriz de histogramas locais de cor, um para cada célula, de acordo com a seguinte equação:

$$h[i][j][k] = \frac{a[i][j][k]}{n} \tag{2.16}$$

onde n é o número de pixels da imagem, $[i][j]$ representa a célula pertencente a linha i e a coluna j da grade disposta na imagem, k é a k-ésima cor do espaço quantizado de cores e $a[i][j][k]$ é o número de pixels pertencentes a célula $[i][j]$ que são da cor k. A distância entre duas representações LCH é dada pela média das distâncias L_1 entre os histogramas locais de cor de células correspondentes.

Color Coherence Vectors

Pass et al. [54] propuseram um método para comparar imagens utilizando vetores de coerência de cor (CCVs – *Color Coherence Vectors*). Eles definem a coerência de uma cor como o grau no qual pixels dessa cor são pertencentes a grandes regiões com uma cor homogênea . Essas regiões são denominadas de regiões coerentes.

Inicialmente, o método quantiza o espaço de cor da imagem para eliminar pequenas variações entre pixels vizinhos. A seguir, o método encontra as componentes conexas da imagem, classificando os pixels de uma dada cor como coerentes (pertencentes a grandes componentes conexas de cor homogênea) ou incoerentes (membros de pequenas componentes conexas de cor homogênea). Por fim, dois histogramas de cor são extraídos: um para os pixels classificados como coerentes e outro para os pixels classificados como incoerentes.

A Figura 2.9 mostra um exemplo de uma imagem analisada em termos da classificação binária dos pixels em coerentes ou incoerentes. A imagem original está disposta à esquerda. A imagem binária ao seu lado mostra os pixels classificados como coerentes em preto e os pixels classificados como incoerentes em branco.

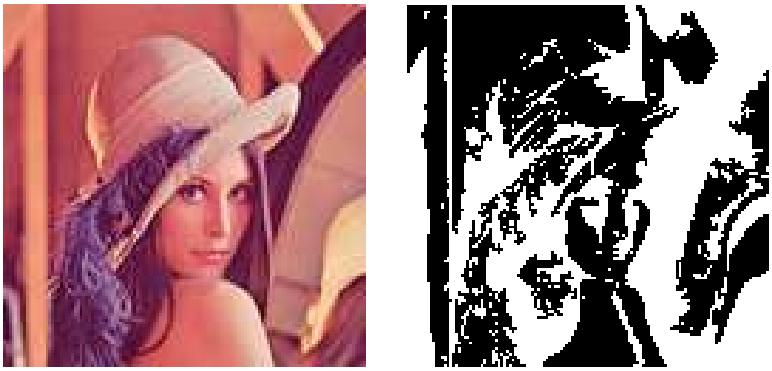


Figura 2.9: Exemplo da classificação binária dos pixels de uma imagem em pixels coerentes (preto) e pixels incoerentes (branco).

A classificação binária de um CCV baseia-se em uma propriedade visual não binária das imagens (o tamanho das componentes conexas). Dessa forma, é necessário estabelecer um tamanho limite para distinguir pequenas e grandes componentes conexas. Em geral, encontrar um limiar ótimo é um problema difícil. Os autores sugerem um tamanho de corte igual a 1% do número de pixels da imagem analisada.

Color-Based Clustering

Color-Based Clustering (CBC) [66] é um método regional baseado em um algoritmo de agrupamento que executa em tempo $O(n \ \log \ n)$, onde n é o tamanho da imagem analisada (número de pixels).

Esse método decompõe uma imagem em um conjunto de regiões conexas disjuntas. Cada região é maior (em número de pixels) que um tamanho mínimo $s_{\min}$. Além disso, todos os pixels de uma região apresentam um grau pré-estabelecido de similaridade de cor, de acordo com uma dissimilaridade máxima $d_{\max}$.

Dessa forma, pode-se denotar esse método como $CBC(d_{\max}, s_{\min})$, onde os limitantes $d_{\max}$ e $s_{\min}$ são parâmetros definidos pelo usuário. Os autores sugerem uma configuração igual $CBC(3, 0.1)$.

Cada região obtida é caracterizada por um vetor de características (L, a, b, h, v, s), onde L, a e b representam a cor média Lab da região, h e v são as coordenadas normalizadas horizontal e vertical do seu centro geométrico e s é o seu tamanho (número de pixels) normalizado pelo tamanho da imagem.

A Figura 2.10 mostra um exemplo de uma imagem segmentada com o algoritmo $CBC(3, 0.1)$.

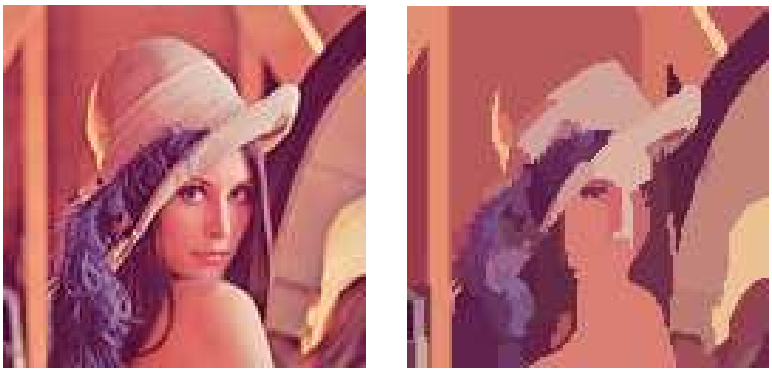


Figura 2.10: Exemplo de uma imagem segmentada com o algoritmo CBC.

A distância entre duas regiões a_i e b_j pertencentes as imagens A e B, respectivamente, é dado por:

$$\mathcal{D}(a_i, b_j) = \alpha \times L_2(a_i.cor, b_j.cor) + (1 - \alpha) \times L_2(a_i.centro, b_j.centro) \tag{2.17}$$

onde $a_i.cor$ e $b_j.cor$ representam, respectivamente, os valores L, a e b dos vetores de características das imagens A e B; $a_i.centro$ e $b_j.centro$ representam, respectivamente, os valores h e v dos vetores de características das imagens A e B; $L_2(\cdot, \cdot)$ representa a distância Euclidiana entre seus argumentos e α define o peso utilizado para combinar a distância entre a cor e a distância entre a posição de cada região.

Por fim, a distância $d(A, B)$ entre duas imagens A e B é dada pela composição ponderada das distâncias entre as regiões que constituem cada imagem, $\mathcal{D}(a_i, b_j)$. As funções de distância IRM [41] e EMD [58], descritas na Seção 2.1.4, são as medidas de dissimilaridade mais utilizadas para compor as distâncias $\mathcal{D}(a_i, b_j)$ no processamento de $d(A, B)$.

Border/Interior pixel Classification

Stehling et al. [67] propuseram a classificação de pixels em borda ou interior (BIC – *Border/Interior pixel Classification*), um método compacto para descrever imagens. Esse método utiliza um espaço de cor RGB uniformemente quantizado em $4 \times 4 \times 4 = 64$ cores.

Após essa quantização, os pixels da imagem são classificados como borda ou interior. Um pixel é classificado como borda se ele pertence à borda da imagem ou se pelo menos um de seus quatro vizinhos (superior, inferior, direito e esquerdo) apresenta uma cor diferente. Um pixel é classificado como interior se seus quatro vizinhos apresentam uma mesma cor.

A seguir, dois histogramas de cor são extraídos: um para os pixels classificados como borda e outro para os pixels classificados como interior. Esses dois histogramas são armazenados como um único histograma de 128 *bins*.

A Figura 2.11 mostra um exemplo de uma imagem analisada em termos da classificação binária dos pixels em borda ou interior. A imagem original está disposta à esquerda. A imagem binária ao seu lado mostra os pixels classificados como borda em preto e os pixels classificados como interior em branco.

Figura 2.11: Exemplo da classificação binária dos pixels de uma imagem em pixels de borda (preto) e pixels de interior (branco).

A comparação entre os histogramas BIC é realizada utilizando a função de distância *dLog* [67]. Essa função é definida como:

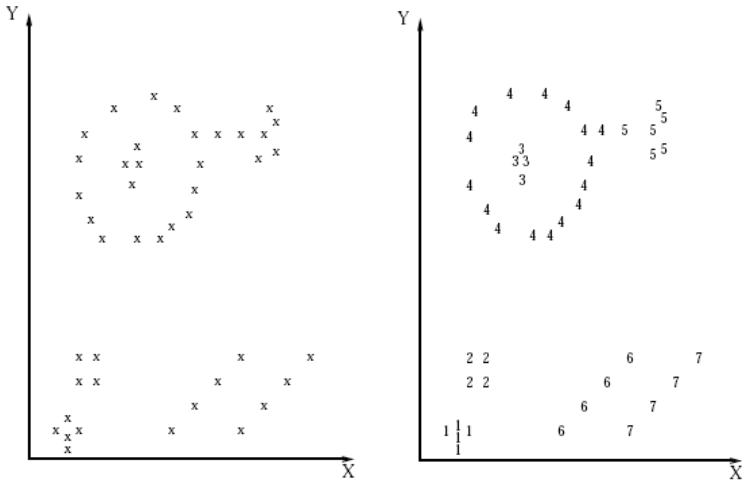
$$d(A, B) = \sum_i |f(a_i) - f(b_i)| \tag{2.18}$$

$$f(x) = \begin{cases} 0, & \text{se } x = 0 \\ 1, & \text{se } 0 < x \leq 1 \\ \lceil \log_2 x \rceil + 1, & \text{caso contrário} \end{cases} \tag{2.19}$$

2.3 Agrupamento de dados

Agrupamento é uma classificação não supervisionada de padrões (observações, itens de dados ou vetores de características) em grupos. Intuitivamente, cada grupo é composto por padrões que são similares entre si e dissimilares em relação aos padrões de outros grupos [29].

Um exemplo de agrupamento é descrito na Figura 2.12. O padrão de entrada é mostrado na Figura 2.12(a) e os grupos são mostrados na Figura 2.12(b). Pontos dentro de um mesmo grupo apresentam um mesmo rótulo.



(a) Padrão de Entrada. (b) Grupos.

Figura 2.12: Exemplo de agrupamento de dados [29].

2.3.1 Etapas de uma tarefa de agrupamento

A Figura 2.13 descreve uma sequência típica das atividades envolvidas em uma tarefa de agrupamento, incluindo um caminho de retorno pelo qual a saída do processo de agrupamento pode afetar as tarefas seguintes [29].

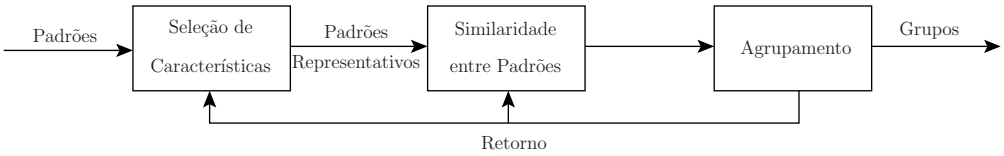


Figura 2.13: Estágios de um processo de agrupamento.

Os padrões representativos se referem ao número de classes, ao número de padrões avaliados e ao número, tipo e escalas de características disponíves no processo de agrupamento. A seleção de características é o processo de identificar o subconjunto mais eficaz de características para serem utilizadas no agrupamento. A similaridade entre padrões é normalmente medida por meio de um função de distância definida entre pares de padrões. Por fim, a etapa de agrupamento consiste na técnica utilizada para formar grupos a partir desses valores de similaridade [29].

2.3.2 Paradigmas de agrupamento

Em geral, as técnicas de agrupamento podem ser hierárquicas ou particionais. Os algoritmos hierárquicos encontram grupos sucessivos utilizando grupos pré-estabelecidos, enquanto os algoritmos particionais encontram todos os grupos de uma única vez [29].

Modelo hierárquico

Um modelo hierárquico de agrupamento constrói uma hierarquia de grupos ou, em outras palavras, uma árvore de grupos, conhecida como *dendrograma*. Dessa forma, esse método permite explorar dados sob diferentes níveis de granularidade [7].

As estratégias de agrupamento hierárquico podem ser divididas em dois paradigmas básicos: aglomerativos e divisivos. Uma estratégia aglomerativa começa com cada elemento como um grupo independente e os une sucessivamente em grupos cada vez maiores. Por outro lado, uma estratégia divisiva começa com um grande grupo e o divide sucessivamente em grupos menores [7].

Modelo particional

Um modelo particional de agrupamento obtém uma única partição dos dados ao invés de um estrutura de grupos, como o dendrograma produzido pelos métodos hierárquicos [29]. K-means e K-medoids são os dois métodos particionais de agrupamento mais conhecidos [25]. K-means é voltado para aplicações em que todas as variáveis são quantitativas e as dissimilaridades entre elas podem ser medidas em um espaço Euclidiano. K-medoids é uma generalização do K-means no sentido que outras funções de distâncias, além da distância Euclidiana, podem ser utilizadas para medir as dissimilaridades entre os elementos [7].

2.3.3 Aplicações de técnicas de agrupamento

Agrupamento é útil em diversos problemas, tais como [29]: na recuperação de documentos [19, 59, 77], na recuperação de imagens [8, 12, 35, 62], na segmentação de imagens [66, 71] e no reconhecimento de padrões [2, 28, 42].

Muitos pesquisadores têm utilizado métodos de agrupamento na recuperação de documentos [59, 77]. Ferragina e Gulli [19] propuseram o *SnakeT*, um método hierárquico de agrupamento para organizar os resultados de diversos mecanismos de busca sob demanda. Eles utilizaram a hierarquia obtida para complementar a análise sequencial dos resultados retornados pelos mecanismos de busca disponíveis.

Um processo de agrupamento pode ser aplicado em diferentes contextos. Por exemplo, alguns métodos utilizam agrupamento no seu próprio espaço de características [5]. Por ou-

tro lado, etapas de agrupamento podem ser utilizadas para encontrar grupos significativos de baixa dimensão e reduzir o impacto dos dados de alta dimensionalidade [61].

Cooper et al. [16] propuseram um método de agrupamento baseado na similaridade entre sequências de fotos. Os autores mostraram que fotografias de um mesmo evento e muito próximas no tempo podem reduzir 33% do tempo de processamento de uma consulta nesse sistema. Kim e Chung [35] utilizaram uma técnica adaptativa de agrupamento para filtrar os resultados de um sistema de recuperação de imagens por conteúdo. Shyu et al. [62] introduziram uma estrutura unificada para facilitar o agrupamento em bancos de atributos e recuperar imagens utilizando modelos mediadores de Markov (MMM – *Markov Models Mediators*).

Heller e Ghahramani [26] desenvolveram um algoritmo hierárquico aglomerativo de agrupamento baseado em probabilidades. Entretanto, esse método é inviável para recuperação de imagens por conteúdo, uma vez que é difícil encontrar um modelo probabilístico apropriado para representar imagens.

Antani et al. [2] desenvolveram uma técnica de agrupamento para recuperação de imagens médicas em sistemas híbridos, baseados em texto e imagem. Thies et al. [71] propuseram um método para superar os problemas dos algoritmos de crescimento de regiões, como a seleção dos pontos iniciais e a ordem de seu processamento. Nesse método, pixels vizinhos são agrupados para criar grupos representativos. Stehling et al. [66], por sua vez, propuseram um algoritmo aglomerativo adaptativo de agrupamento para segmentar imagens em regiões de alta similaridade por meio de componentes conexas e similaridade de cor.

Bhatia [8] introduziu uma técnica hierárquica de agrupamento de imagens. Nesse método, os modelos armazenados são representados hierarquicamente em um banco ao invés de utilizar uma estrututra sequencial. Entretanto, essa técnica apresenta o requisito indesejável de exigir a mudança da maneira física como as imagens estão armazenadas, destruindo a independência lógica e física desses bancos.

Kinoshenko et al. [36] propuseram uma técnica para particionar uma imagem em subconjuntos disjuntos. Nesse método, o sistema divide cada consulta em subclasses representativas e encontra a subclasse mais similar armazenada para cada parte da consulta. Dessa forma, a consulta é hierarquicamente classificada em partes. Entretanto, o método escolhe essas classes de maneira fortemente conexa, isto é, as classes das imagens precisam representar uma estrutura hierárquica, por exemplo, o relacionamento presente na imagem de um carro e suas partes. Em geral, é difícil encontrar essas hierarquias em imagens de domínio geral.

Capítulo 3

SIFT-Texton

Em um sistema de recuperação de imagens por cor, o conteúdo de uma imagem é descrito a partir de suas características visuais de baixo nível relacionadas à cor. Esses métodos normalmente falham na presença de diferentes condições de iluminação, oclusão e foco em imagens de domínio geral.

Muitas técnicas em processamento de imagens e em visão computacional são capazes de descrever esse tipo de informação [51]. Entre essas técnicas, o SIFT (*Scale Invariant Features Transform*) [48] destaca-se em diversas aplicações, como por exemplo, no reconhecimento de objetos [47], no reconhecimento de panoramas [11] e na reconstrução tridimensional [10]. Esse método permite a extração de características invariantes locais (*local invariant features*) de uma cena ou de um objeto.

Este capítulo apresenta o SIFT-Texton, um novo método para recuperação de imagens capaz de incorporar informações de médio nível nas características visuais de baixo nível. Esse método baseia-se na distribuição discreta de características invariantes locais e em propriedades de baixo nível.

O restante deste capítulo está organizado como se segue. A Seção 3.1 introduz o SIFT. A Seção 3.2 apresenta o SIFT-Texton. A Seção 3.3 descreve a função de distância utilizada para comparar duas imagens a partir da distribuição discreta de suas características invariantes locais. Por fim, a Seção 3.4 discute os resultados experimentais obtidos na aplicação desse método na recuperação de imagens por cor.

3.1 SIFT e suas variantes

O *Scale Invariant Features Transform* (SIFT[1]) [48] é um método de extração de pontos característicos de uma cena ou de um objeto, utilizado em diversas aplicações em visão

[1]O autor disponibiliza uma versão demonstrativa desse método em `http://www.cs.ubc.ca/~lowe/keypoints/`.

computacional. Esse método consiste em quatro etapas [48]:

1. **Detecção dos extremos do espaço escala:** o primeiro estágio consiste na busca de pontos de interesse em diferentes posicionamentos e escalas. Isso é realizado utilizando uma função baseada na diferença de gaussianas para identificar pontos de interesse invariantes à escala e à orientação.

2. **Localização dos pontos característicos:** para cada ponto de interesse é associado um modelo detalhado para ajustar seu posicionamento e sua escala. Os pontos característicos são selecionados de acordo com a medida de sua estabilidade.

3. **Atribuição de uma orientação:** uma ou mais orientações são associadas a cada ponto característico de acordo com as direções de seus gradientes locais. Todas as operações seguintes são realizadas em relação à posição, à escala e à orientação associada, garantindo invariância a essas transformações.

4. **Descrição dos pontos característicos:** para cada ponto característico é obtido, na sua escala, uma medida que associa os gradientes locais em seu redor. Eles são transformados em uma representação que permite níveis significativos de variações em iluminação, oclusão e foco.

As seções seguintes detalham cada uma dessas etapas.

3.1.1 Detecção dos extremos do espaço escala

Essa etapa consiste em identificar, sob diferentes posicionamentos e escalas, pontos de interesse que podem ser associados repetidamente sobre diferentes visões de um mesmo objeto. Isso é obtido utilizando uma função contínua da escala, conhecida como espaço escala [48].

O espaço escala de uma imagem é definido como uma função $L(x, y, \sigma)$, que é obtida por meio da convolução de uma função gaussiana $G(x, y, \sigma)$ com a imagem analisada $I(x, y)$ [48]:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y), \tag{3.1}$$

onde $*$ é a operação de convolução em x e em y, e

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \tag{3.2}$$

Para detectar pontos característicos estáveis, Lowe [48] propõe a utilização de um espaço escala extremo, baseado na convolução de uma função da diferença de gaussianas,

$D(x, y, \sigma)$, que pode ser calculado a partir da diferença de duas escalas próximas separadas por uma constante k [48]:

$$\begin{aligned} D(x, y, \sigma) &= (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma) \end{aligned} \tag{3.3}$$

Existem inúmeras razões para escolher essa função. Primeiro, é uma função fácil de ser calculada, uma vez que as imagens borradas L precisam ser computadas na construção do espaço escala e D pode ser obtida por uma simples subtração de imagens [48].

Além disso, a função da diferença de gaussianas garante uma aproximação do laplaciano de uma gaussiana $\sigma^2\nabla^2 G$, estudado por Lindeberg [45]. Lindeberg mostrou que a normalização do laplaciano por um fator σ^2 é necessário para garantir invarância em relação à escala. Em experimentos, Mikolajczyk [50] encontrou que o mínimo e o máximo de $\sigma^2\nabla^2 G$ produz as características mais estáveis de uma imagem.

O relacionamento entre D e $\sigma^2\nabla^2 G$ pode ser obtido por meio de uma equação de difusão de calor (parametrizada em termos de σ ao invés de $t = \sigma^2$) [48]:

$$\frac{\partial G}{\partial \sigma} = \sigma\nabla^2 G \tag{3.4}$$

A partir dessa equação, pode-se notar que $\nabla^2 G$ pode ser computada como uma aproximação da diferença finita de $\partial G/\partial\sigma$, utilizando a diferença de escalas próximas [48]:

$$\sigma\nabla^2 G = \frac{\partial G}{\partial \sigma} \approx \frac{G(x, y, k\sigma) - G(x, y, \sigma)}{k\sigma - \sigma} \tag{3.5}$$

e assim,

$$G(x, y, k\sigma) - G(x, y, \sigma) \approx (k - 1)\sigma^2\nabla^2 G \tag{3.6}$$

Isso mostra que, quando a função de diferença de gaussianas apresenta escalas que diferem por um fator constante, essa função já incorpora o fator de normalização σ^2 necessário para se obter invariância em relação à escala em um laplaciano. O fator $(k-1)$ na equação é uma constante em todas as escalas e, dessa forma, não influencia a detecção dos extremos [48].

A Figura 3.1 ilustra um método eficiente para a construção de $D(x, y, \sigma)$. A imagem de entrada é recusivamente convolvida com uma função gaussiana para produzir imagens separadas por uma constante k. Imagens em escalas adjacentes são subtraídas, produzindo uma diferença de gaussianas. Esse processo é repetido, recursivamente, reduzindo-se a escala da imagem por um fator de 2.

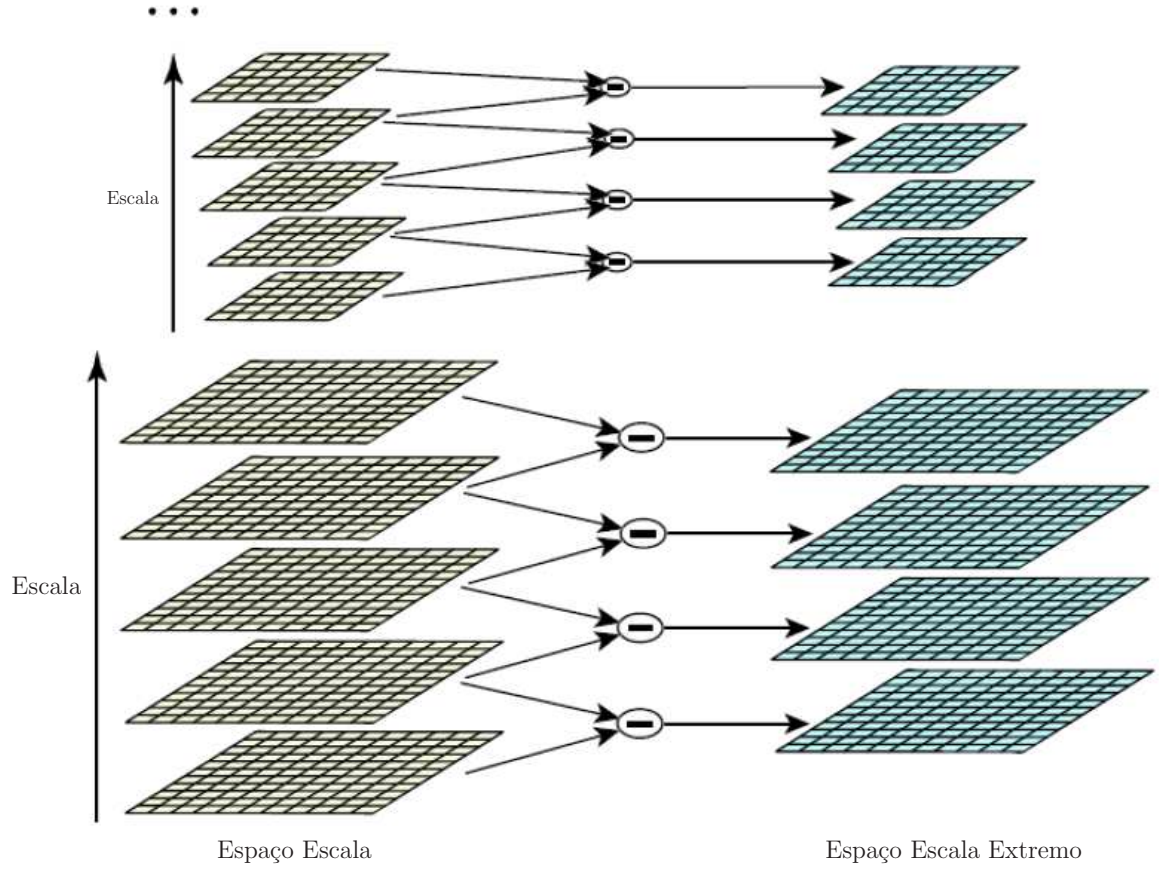


Figura 3.1: Construção de um espaço escala baseado na diferença de gaussianas [48].

3.1.2 Localização dos pontos característicos

Esse estágio consiste em detectar os máximos e mínimos locais de $D(x, y, \sigma)$. Para isso, cada ponto é comparado com seus vizinhos nas diferentes escalas (Figura 3.2), sendo selecionados os pontos que apresentarem um valor maior (ou menor) que todos os seus vizinhos nas diferentes escalas [48].

Uma vez que os pontos de interesse foram encontrados, o próximo passo consiste em ajustar os seus posicionamentos. Isso é realizado por meio da expansão de Taylor de cada ponto, $D(x, y, \sigma)$, em torno da origem [48]:

$$D(X) = D + \frac{\partial D^T}{\partial X} X + \frac{1}{2} X^T \frac{\partial^2 D}{\partial X^2} X, \tag{3.7}$$

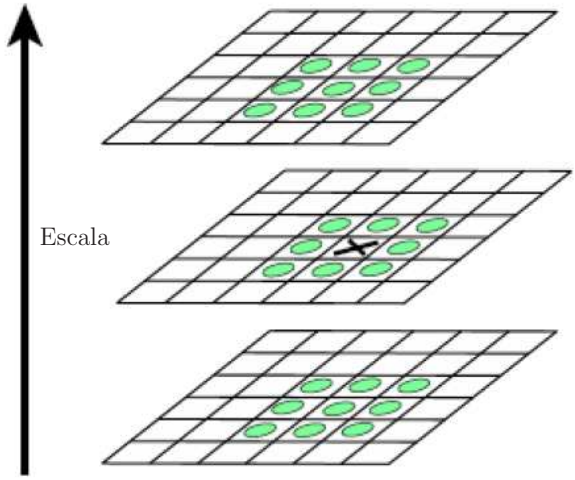


Figura 3.2: Localização dos pontos extremos em diferentes escalas [48].

onde D e suas derivadas são obtidas ao redor do ponto em análise; e $X = (x, y, \sigma)^T$ é o deslocamento desse ponto. Os pontos extremos $\hat{X}$ são obtidos quando sua derivada em relação a X é igual a zero,

$$\hat{X} = \frac{\partial^2 D^{-1}}{\partial X^2} \frac{\partial D}{\partial X} \tag{3.8}$$

Assim, os pontos de baixo contraste podem ser eliminados selecionando-se os pontos, tal que [48]:

$$D(\hat{X}) = D + \frac{1}{2} \frac{\partial D}{\partial X} \hat{X} \tag{3.9}$$

A função diferença de gaussianas possui uma forte influência ao longo dos pontos de borda, os quais são instáveis mediante a presença de pequenos ruídos. Esses pontos definem picos nessa função, apresentando um baixo valor principal de curvatura na direção ao longo das bordas e um alto valor principal de curvatura na direção perpendicular [48].

Os valores principais de curvatura podem ser calculados por meio de uma matriz hessiana H, obtida a partir da diferença de cada ponto em D com seus vizinhos,

$$H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix} \tag{3.10}$$

Os autovalores de H são proporcionais aos valores principais de curvatura de D. Tomando α como sendo o autovalor de maior magnitude e β como sendo o autovalor de menor magnitude, têm-se [48]:

$$Tr(H) = D_{xx} + D_{yy} = \alpha + \beta \tag{3.11}$$

$$Det(H) = D_{xx} D_{yy} - (D_{xy})^2 = \alpha\beta \tag{3.12}$$

Tomando r como a proporção entre os autovalores, $\alpha = r\beta$,

$$\frac{Tr(H)^2}{Det(H)} = \frac{(\alpha+\beta)^2}{\alpha\beta} = \frac{(r\beta+\beta)^2}{r\beta^2} = \frac{(r+1)^2}{r} \tag{3.13}$$

Assim, os pontos de bordas podem ser eliminados selecionando-se pontos, tal que,

$$\frac{Tr(H)^2}{Det(H)} < \frac{(r+1)^2}{r} \tag{3.14}$$

3.1.3 Atribuição de uma orientação

Associando-se uma orientação consistente a cada ponto característico, pode-se utilizar um posicionamento relativo para representar esse ponto, garantindo invariância em relação às transformações de rotação [48].

A escala do ponto característico é utilizada para selecionar a imagem gaussiana L com escala mais próxima, de forma que todas as operações seguintes possam ser realizadas de maneira invariante à escala. Para cada imagem dessa escala, pode-se obter a magnitude de seu gradiente $m(x, y)$ e sua orientação $\theta(x, y)$ por meio da diferença entre seus vizinhos [48]:

$$m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2} \tag{3.15}$$

$$\theta(x,y) = \tan^{-1}((L(x,y+1) - L(x,y-1))/(L(x+1,y) - L(x-1,y))) \tag{3.16}$$

A seguir, um histograma de orientações é formado a partir da amostragem da orientação do gradiente dos pontos ao redor do ponto característico. Esse histogram é composto por 36 *bins* cobrindo o espaço de 360 graus das orientações. Cada ponto adicionado ao histograma é ponderado pela magnitude de seu gradiente e por uma janela gaussiana circular com um σ que é $1,5$ vezes a escala do ponto característico [48].

Os picos nesse histograma de orientações correspondem às direções dominante desses gradientes locais. Esses picos são utilizados para construir vetores de características na orientação do seu *bin* correspondente [48].

3.1.4 Descrição dos pontos característicos

A Figura 3.3 ilustra o processo de descrição dos pontos característicos. Primeiro, as magnitudes e as orientações do gradiente são amostradas ao redor do ponto característico, utilizando a sua escala para selecionar a imagem gaussiana correspondente. A fim de garantir invariância à orientação, as coordenadas da região utilizada para codificar os vetores de características e a orientação dos gradientes amostrados são rodados em relação à orientação atribuída na etapa anterior [48].

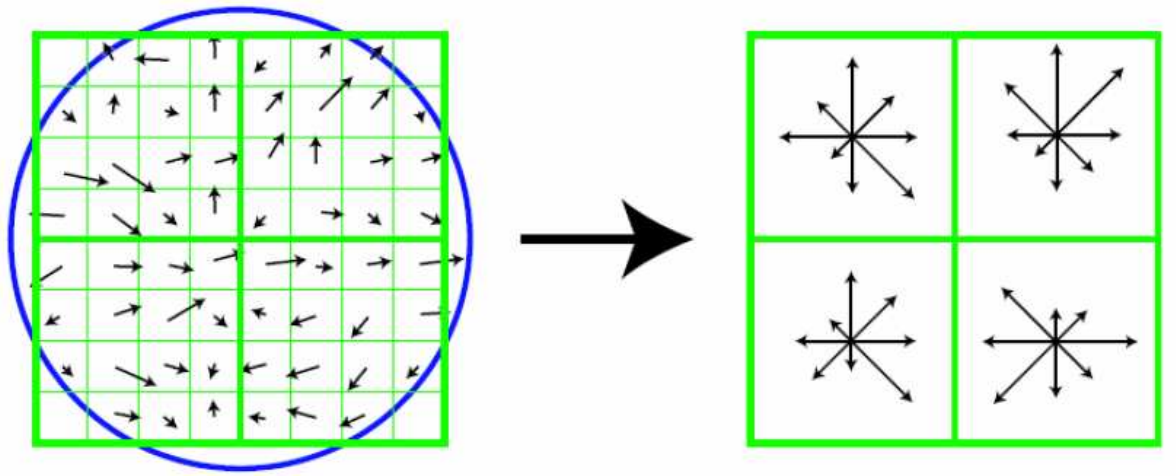


Figura 3.3: Estrutura para descrição dos pontos característicos [48].

Uma função gaussiana com σ igual a metade do tamanho da região analisada é utilizada para ponderar a magnitude do gradiente de cada ponto amostrado, como mostra o lado esquerdo da Figura 3.3. Essa etapa tem o propósito de evitar mudanças abruptas nos vetores de características, dando prioridade aos gradientes mais próximos de seu centro [48].

O vetor de características que descreve cada ponto representativo é mostrado no lado direito da Figura 3.3. Ele é formado pela composição dos histogramas de orientações de uma grade composta por $4 \times 4 = 16$ células. Esses histogramas são obtidos a partir da projeção dos gradientes locais em oito direções predefinidas [48].

Embora a Figura 3.3 mostre um exemplo de uma grade com $2 \times 2 = 4$ células, os autores recomendam a utilização de uma grade de $4 \times 4 = 16$ células. Dessa forma, cada ponto constitui um vetor de características composto por $4 \times 4 \times 8 = 128$ elementos.

Por fim, esse vetor de características é modificado para reduzir os efeitos ocasionados pelas mudanças de iluminação. Primeiro, o vetor é normalizado para uma magninute igual a um. A seguir, um valor máximo é definido para a magnitude de cada elemento e, então, esse vetor é normalizado novamente. Isso reduz a influência da magnitude dos gradientes, que estão diretamente ligados às mudanças de iluminação, dando ênfase na distribuição de suas orientações [48].

A Figura 3.4 ilustra o resultado desse método. As setas brancas indicam a posição, a orientação e a escala dos pontos característicos encontrados.

Figura 3.4: Extração de pontos característicos invariantes locais de uma imagem.

3.1.5 Redução do espaço de características

Cada ponto extraído pelo SIFT constitui um vetor de características composto por $4 \times 4 \times 8 = 128$ elementos (SIFT-128). Uma representação mais compacta pode ser obtida utilizando-se uma grade com $2 \times 2 = 4$ células ao redor de cada ponto, resultando em um vetor de características composto por $2 \times 2 \times 8 = 32$ elementos (SIFT-32). Entretanto, esse método pode reduzir a eficácia das tarefas de recuperação.

Ke e Sukthankar [34] introduziram uma solução para esse problema. Eles aplicaram a técnica de redução de dimensionalidade *Principal Components Analisys* (PCA) [30] no espaço de características formado pelas 128 dimensões do vetor características extraído pelo SIFT, criando uma representação mais compacta e mais robusta, denominada PCA-SIFT[2]. Os experimentos deste trabalho utilizam um vetor de características PCA-SIFT com 8 dimensões (PCA-SIFT-8).

3.2 SIFT-Texton

Na presença de diferentes condições de iluminação, oclusão e foco em imagens de domínio geral, os sistemas de recuperação por conteúdo baseados em propriedades de baixo nível normalmente falham.

Esta seção apresenta o SIFT-Texton, um novo método para recuperação de imagens capaz de incorporar informações de médio nível (iluminação, oclusão e foco) nas características visuais de baixo nível. Este trabalho é resultado de uma série de experimentos

[2]O autor disponibiliza uma implementação desse método em `http://www.cs.cmu.edu/~yke/pcasift/`.

que foram conduzidos a fim de avaliar as características mais relevantes envolvidas ao trabalhar com o SIFT em CBIR, apresentados no Apêndice A.

Esse método baseia-se na distribuição discreta de características invariantes locais e em propriedades de baixo nível. O Algoritmo 1 resume as etapas principais desse processo. As seções seguintes detalham cada uma dessas etapas.

Algoritmo 1 Etapas principais do SIFT-Texton.

Entrada: Imagem I;

1: **Extração de pontos característicos:** encontre pontos característicos invariantes locais K. ▷ Seção 3.2.1

2: **Espaço escala:** construa os espaços escala L_R, L_G e L_B de cada canal do espaço de cor RGB da imagem I. ▷ Seção 3.2.2

3: **Extração de regiões:** ▷ Seção 3.2.3

- **Para cada** ponto característico $k \in K$
  - Extraia uma região escalada e orientada P ao redor de k no espaço escala colorido formado pela composição de L_R, L_G e L_B.

4: **Extração de características visuais:** ▷ Seção 3.2.4

- **Para cada** região $p \in P$
  - Utilize um descritor baseado em propriedades de baixo nível para codificar as características visuais de p.

3.2.1 Extração de pontos característicos

Essa etapa consiste na extração de pontos característicos invariantes locais utilizando o SIFT em um canal do espaço de cor da imagem I. Os experimentos apresentados no Apêndice A mostram que essa técnica apresenta pouca variação em relação aos canais de cor. Isso ocorre porque o SIFT codifica informações de gradiente. Essas informações são preservadas em todos os canais de um espaço de cor. Este trabalho utiliza o canal V do espaço de cor HSV como padrão de informação analisada.

A Figura 3.5 ilustra esse processo. Inicialmente, a imagem (a) é fornecida como entrada. A seguir, o canal V do espaço de cor HSV é processado para essa entrada, resultando na imagem (b). Por fim, os pontos característicos invariantes locais são extraídos

para essa imagem, como mostra a imagem (c).

(a) Imagem. (b) Canal V. (c) Pontos.

Figura 3.5: Extração dos pontos característicos de uma imagem.

3.2.2 Espaço escala

Esta etapa é responsável pela construção dos espaços escala L_R, L_G e L_B de cada canal do espaço de cor RGB da imagem I. Esse processo é realizado utilizando o método descrito na Seção 3.1.1. A Figura 3.6 mostra a pirâmide resultante da composição dos espaços escala L_R, L_G e L_B construídos para a imagem da Figura 3.5(a).

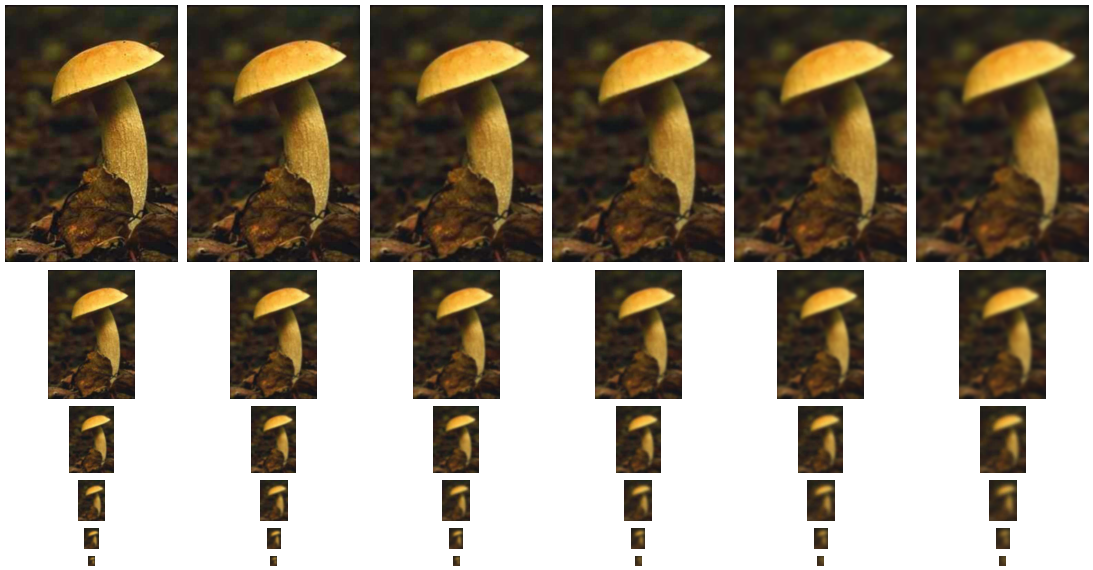


Figura 3.6: Pirâmide resultante da composição dos espaços escala L_R, L_G e L_B.

3.2.3 Extração de regiões

Para cada ponto característico, é extraído, na sua escala, uma região ao seu redor no espaço escala formado pela composição de L_R, L_G e L_B. Nesse processo, os pixels são amostrados ao redor do ponto característico, utilizando a sua escala para selecionar a imagem gaussiana correspondente. A fim de garantir invariância à orientação, as coordenadas dessa região são rotacionadas em relação à orientação desse ponto. Essas regiões capturam informações de médio nível, como iluminação, oclusão e foco.

A Figura 3.7 mostra as regiões extraídas para a imagem da Figura 3.5(a).

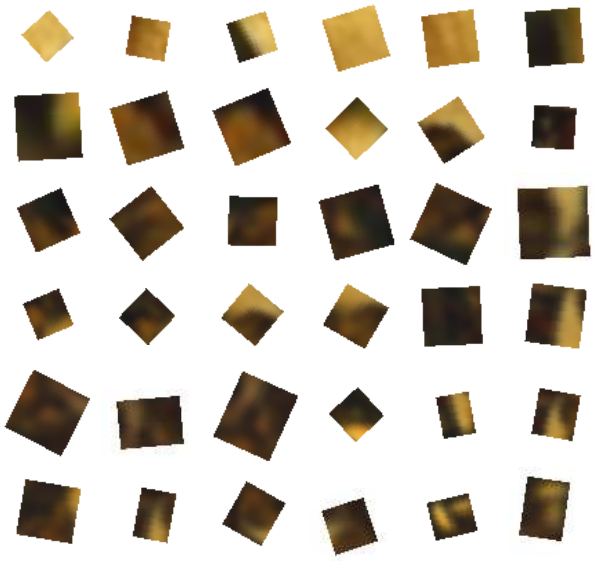


Figura 3.7: Regiões extraídas ao redor dos pontos característicos.

3.2.4 Extração de características visuais

Para cada região extraída, um descritor de propriedades de baixo nível (por exemplo, GCH, LCH, CCV ou BIC) é utilizado para obter características visuais capazes de representar o seu conteúdo.

A Figura 3.8 mostra as características visuais extraídas das regiões obtidas para a imagem da Figura 3.5(a) utilizando o BIC para codificar as propriedades de baixo nível. De acordo com a classificação do BIC, as áreas brancas representam bordas e as áreas pretas representam interior.

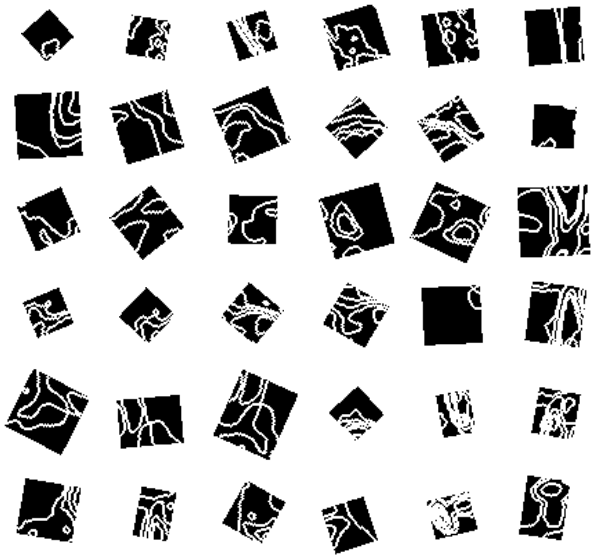


Figura 3.8: Características visuais extraídas ao redor dos pontos característicos.

3.3 Função de distância

Este trabalho utiliza a função de distância EMD [58] (Seção 2.1.4), para avaliar a dissimilaridade entre duas imagens. Essa função garante uma maneira eficaz para comparar imagens com base na distribuição discreta de suas características invariantes locais [24].

Esse processo consiste em estabelecer correspondências entre as características invariantes locais de duas imagens. Entretanto, a complexidade em encontrar uma correspondência ótima entre duas distribuições é cúbica em relação ao número de características por imagem [24].

Charikar [14] propôs um algoritmo rápido de aproximação da EMD que tem viabilizado a utilização dessa função de distância em diversas aplicações [23, 24]. Esse algoritmo utiliza uma função de aproximação h para mapear a função de distância EMD em uma função de distância L_1 com uma baixa distorção, de forma que:

$$\frac{1}{C}EMD(A,B) \leq L_1(h(A),h(B)) \leq EMD(A,B) \tag{3.17}$$

onde C é o fator de distorção.

Esse mapeamento é realizado impondo-se grades G_j, $-1 \leq j \leq \log \Delta$, no espaço $\mathbb{R}^n$, onde G_j é composto por células quadradas cujo lado mede 2^j e Δ é igual ao diâmetro de $A \cup B$. Cada grade G_j é transladada por um vetor escolhido aleatoriamente entre $[0, \Delta]^n$. Para embutir as características invariantes locais de A nessas grades, um vetor $\vec{v}_j \in \mathbb{R}^n$ é criado para cada grade G_j, com uma coordenada por célula, onde cada coordenada conta

o número de características invariantes locais dentro de sua célula correspondente, isto é, cada vetor $\vec{v}_j$ forma um histograma de A.

O mapeamento h é dado pela concatenação dos vetores $\vec{v}_j$ escalados pelo tamanho de sua grade correspondente:

$$h(A) = [\vec{v}_{-1}(A), \vec{v}_0(A), 2\vec{v}_1(A), \ldots, 2^j\vec{v}_j(A), \ldots] \tag{3.18}$$

Dessa forma, a distância entre as imagens A e B é dada pela distância L_1 entre seus respectivos mapeamentos h:

$$d(A, B) = L_1(h(A), h(B)) \tag{3.19}$$

Portanto, uma vez pré-processados os mapeamentos h de todas as imagens, a comparação entre elas se resume à distância L_1 dos vetores esparsos que as representam. O Algoritmo 2 [23] apresenta o pseudocódigo para realizar esse processo.

3.4 Experimentos

Esta seção compara o SIFT-Texton com os descritores de imagens baseados em cor apresentados na Seção 2.2.4. A Seção 3.4.1 descreve a metodologia de validação utilizada na avaliação desses métodos. Os resultados experimentais obtidos são discutidos na Seção 3.4.2. Por fim, a Seção 3.4.3 compara resultados visuais obtidos quando uma consulta é processada.

3.4.1 Metodologia de validação

Nos experimentos realizados, foi adotado o paradigma de consultas por exemplo (QBE – *Query-By-Example*) [46], que tem se mostrado a metodologia mais adequada na avaliação de sistemas de recuperação de imagens por conteúdo.

Nesse paradigma, uma imagem é fornecida como exemplo da informação requisitada. Essa imagem é analisada e suas características visuais são extraídas. Essas características são utilizadas para medir a similaridade entre a imagem de consulta e as imagens armazenadas em um banco de imagens. Essas imagens são recuperadas em ordem decrescente de similaridade em relação à imagem de consulta. O propósito dos experimentos é avaliar a eficácia de diferentes métodos em recuperar imagens relevantes antes de imagens não relevantes.

Dessa forma, para avaliar a eficácia de um descritor, é necessário ter um banco de imagens como referência, um conjunto de imagens de consulta, um conjunto de imagens relevantes para cada consulta e uma medida para avaliar a eficácia de recuperação.

Algoritmo 2 Procedimento para mapear a distância EMD em uma distância L_1 [23]

Entrada: N imagens $\{A_1, \ldots, A_N\}$, onde $A_i = \{(f_1, w_1), \ldots, (f_{m_i}, w_{m_i})\}$ é um conjunto de características invariantes locais composto por m_i d-dimensional vetores de características F^i de pesos escalares $W^i = [w_1, \ldots, w_{m_i}]$, no qual $t(A_i) = \sum_{j=1}^{m_i} w_j$ e Δ é o diâmetro de $\cup_{i=1}^{N} F^i$.

Saída: N vetores esparsos representando os mapeamentos $h(A_1), \ldots, h(A_N)$.

$t_{max} = \max_i t(A_i)$
para todo $i = 1, \ldots, N$ **faça**
    se $t(A_i) < t_{max}$ **então**
        $A_i \leftarrow \{(f_1, w_1), \ldots, (f_{m_i}, w_{m_i}), (d_1, u), \ldots, (d_q, u)\}$, onde u é uma unidade de peso, $q = t_{max} - t(A_i)$ e d_z é escolhido aleatoriamente de $\{f_1, \ldots, f_{m_i}\}$.
    fim se
fim para
$L = \lceil \log \Delta / \log 2 \rceil + 1$
para $1 \leq l \leq L$ **faça**
    $s^l = [s_1^l, \ldots, s_d^l]$, onde s_k^l é escolhido aleatoriamente de $[0, 2^l]$.
fim para
para todo $A_i, 1 \leq i \leq N$ **faça**
    para todo $(f_j = [f_1^j, \ldots, f_d^j], w_j) \in A_i$ **faça**
        para todo $s^l, 1 \leq l \leq L$ **faça**
            $x_l^j = [c(s_1^l, f_1^j), \ldots, c(s_d^l, f_d^j)]$, onde $c(s_\beta^\alpha, f) = trunc((f - s_\beta^\alpha)/2^\alpha)$.
            $v_l^j = w_j \times 2^l$
        fim para
        $p_j^i = [(x_1^j, v_1^j), \ldots, (x_L^j, v_L^j)]$
    fim para
    $h(A_i) = tally(sort([p_1^i, \ldots, p_{m_i}^i]))$, onde cada par (x, v) representa uma entrada de um vetor esparso com índice x e valor v, $sort()$ retorna $[(x_{s_1}, v_{s_1}), \ldots, (x_{s_n}, v_{s_n})]$ tal que $x_{s_i} \leq_{LEX} x_{s_{i+1}}$ (ordem lexicográfica do vetor de elementos concatenados) e $tally()$ soma os valores de entradas do vetor esparso com índices iguais.
fim para

Este trabalho utiliza dois bancos de imagens descritos na literatura. O primeiro, denominado *Relevants*, é um subconjunto do *Corel Photo Gallery*, reportado em [67]. Esse banco é bastante heterogêneo e composto por imagens de diferentes domínios. O segundo, denominado *ETH-80* [40], é um banco livre[3] formado por imagens com um fundo comum e diferentes condições de iluminação, oclusão e foco.

O objetivo deste trabalho é recuperar objetos em um banco de imagens. Dessa forma, as imagens que não representam um objeto foram excluídas. A combinação desses bancos de imagens, utilizada nos experimentos deste trabalho, é composta por 1.320 imagens de domínio geral de 72 classes distintas com tamanhos diferentes, conforme mostra a Figura 3.9. Alguns exemplos do banco de imagens resultante dessa combinação, denominado *ETHRel-72*, são apresentados na Figura 3.10.

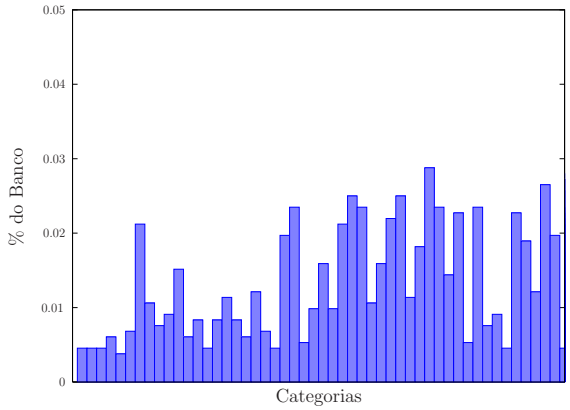


Figura 3.9: Distribuição de imagens por classe para o banco *ETHRel-72*.

Figura 3.10: Exemplos de imagens do banco *ETHRel72*.

[3] *The ETH-80 Image Set* – `http://www.mis.informatik.tu-darmstadt.de/Research/Projects/categorization/eth80-db.html`

Neste trabalho, todas as imagens do banco são utilizadas como imagem de consulta. Portanto, os resultados apresentados nos experimentos visam avaliar a eficácia média obtida no processamento de 1.320 consultas.

Este trabalho utiliza a curva *Precisão vs. Revocação* ($P \times R$) [4, 9, 37] para avaliar a eficácia de recuperação. Em geral, quanto mais próximo do topo do gráfico for essa curva, melhor é o desempenho do método analisado.

Além disso, também foram utilizadas medidas de valor único. Uma dessas medidas corresponde à precisão obtida quando o número de imagens recuperadas é suficiente para incluir todas as imagens relevantes para a consulta. Esse valor é conhecido como *R-value* [4], por isso, a precisão nesse ponto foi chamada $p(R)$.

Além disso, também foram avaliados os valores $p(30)$, $r(30)$, $p(100)$ e $r(100)$. Esses valores são uma estimativa do número de imagens recuperadas que um usuário médio avaliaria em um sistema [4].

Por fim, outras duas medidas de valor único consistem na média de três e de onze pontos de precisão [4]. A média de três pontos ($3P$) é obtida por meio da média da precisão em três pontos predefinidos de revocação, normalmente, 20%, 50% e 80%. A média de onze pontos ($11P$) é calculada por meio da média da precisão em onze pontos predefinidos de revocação: 0%, 10%, ..., 90%, 100%.

3.4.2 Resultados experimentais

Esta seção discute os resultados experimentais relativos à eficácia do SIFT-Texton na recuperação de imagens por cor. Nesses resultados, o método SIFT-Texton é representado por SIFT-Texton$_{BIC}$ e SIFT-Texton$_{GCH}$, que utilizam os descritores BIC e GCH, respectivamente, para codificar características visuais de baixo nível. Nesta seção, esses métodos são comparados com as técnicas apresentadas nas Seções 2.2.4 e 3.1.

A Figura 3.11 apresenta a curva $P \times R$ de cada método analisado. Essa curva representa a precisão média obtida em pontos fixos de revocação (0%, 10%, ..., 90%, 100%) no processamento de 1.320 consultas. A Tabela 3.1 apresenta a média dos resultados obtidos nessas consultas para sete medidas de valor único: $3P$, $11P$, $p(30)$, $r(30)$, $p(100)$, $r(100)$ e $p(R)$.

Os resultados obtidos nas oito medidas de avaliação de eficácia confirmam que incorporar informações de médio nível nas características visuais, aumenta a eficácia de recuperação dos métodos baseados em propriedades de baixo nível (SIFT-Texton$_{BIC}$ vs. BIC e SIFT-Texton$_{GCH}$ vs. GCH).

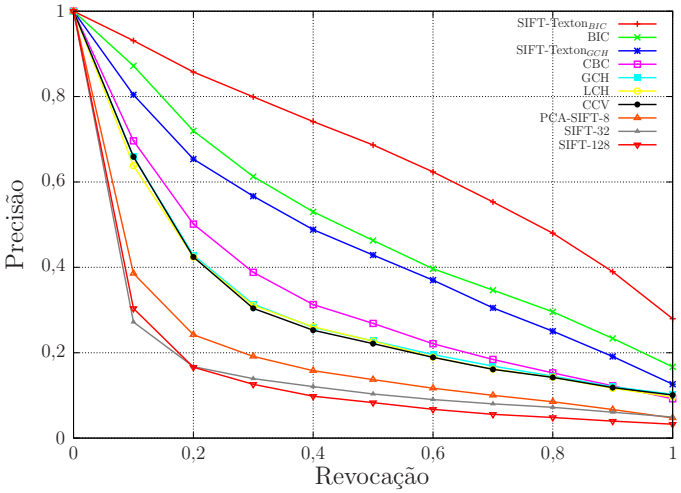


Figura 3.11: SIFT-Texton$_{BIC,\ GCH}$ vs. outros métodos.

Método	$3P$	$11P$	$p(30)$	$r(30)$	$p(100)$	$r(100)$	$p(R)$
SIFT-Texton$_{BIC}$	**0,67**	**0,58**	**0,53**	**0,66**	**0,32**	**0,90**	**0,28**
BIC	0,49	0,42	0,39	0,52	0,23	0,76	0,17
SIFT-Texton$_{GCH}$	**0,44**	**0,38**	**0,37**	**0,49**	**0,21**	**0,70**	**0,13**
CBC	0,31	0,27	0,27	0,38	0,16	0,58	0,09
GCH	0,27	0,24	0,23	0,34	0,14	0,50	0,10
LCH	0,26	0,23	0,23	0,34	0,14	0,51	0,10
CCV	0,26	0,23	0,23	0,34	0,14	0,50	0,10
PCA-SIFT-8	0,15	0,14	0,14	0,22	0,09	0,39	0,05
SIFT-32	0,11	0,10	0,10	0,17	0,06	0,28	0,05
SIFT-128	0,10	0,09	0,10	0,17	0,06	0,29	0,03

Tabela 3.1: Resultados obtidos nas medidas de valor único.

Por outro lado, a ausência das propriedades de cor nos métodos baseados em características invariantes locais (PCA-SIFT-8, SIFT-32 e SIFT-128), ocasiona uma baixa eficácia dessas técnicas na recuperação de imagens por conteúdo, conforme detalhado no Apêndice A.

Por fim, analisando esses resultados, pode-se concluir que o SIFT-Texton$_{BIC}$ é mais eficaz que os demais métodos na recuperação de imagens de domínio geral mediante diferentes condições de iluminação, oclusão e foco.

A Tabela 3.2 apresenta o tempo de execução utilizado por cada método para processar 1.320 consultas. Esses experimentos foram realizados em um processador Athlon64 3200+ 2 GHz com 2 Gb DDR 400 MHz de memória.

Método	Tempo de execução (min)
SIFT-Texton$_{BIC}$	**180,30**
BIC	0,25
SIFT-Texton$_{GCH}$	**215,85**
CBC	25,57
GCH	0,15
LCH	1,85
CCV	0,17
PCA-SIFT-8	49,87
SIFT-32	155,92
SIFT-128	549,78

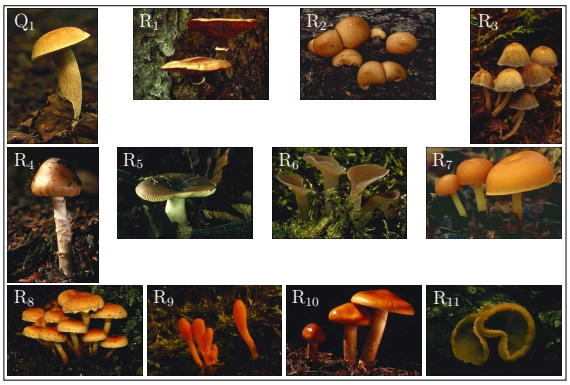
Tabela 3.2: Tempo de execução médio para processar 1.320 consultas.

Analisando essa tabela, pode-se notar um alto tempo de execução dos métodos baseados no SIFT em relação aos demais. Isso ocorre porque a função de distância EMD, utilizada por esses métodos para avaliar a dissimilaridade entre duas imagens, é computacionalmente cara. Nesse contexto, o Capítulo 4 apresenta uma técnica capaz de aumentar a eficiência dos métodos de recuperação de imagens por conteúdo e, portanto, reduzir o tempo de processamento de consultas.

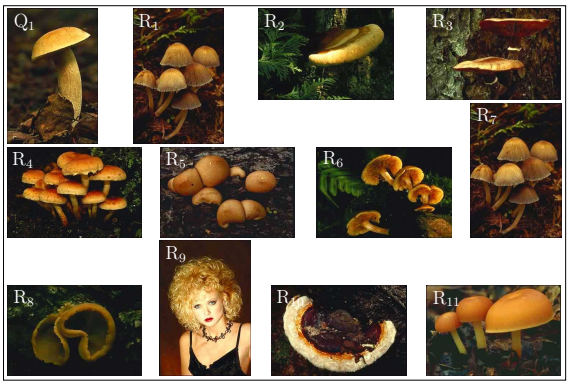
3.4.3 Exemplos visuais

Esta seção apresenta resultados de duas consultas Q_1 e Q_2 para os métodos SIFT-Texton$_{BIC}$ e BIC. Nesses resultados, a imagem de consulta é mostrada no topo à esquerda e as imagens recuperadas apresentam-se ordenadas da esquerda para direita e de cima para baixo.

As Figuras 3.12(a) e 3.12(b) mostram as onze primeiras imagens recuperadas na consulta Q_1 para os métodos SIFT-Texton$_{BIC}$ e BIC, respectivamente. O SIFT-Texton$_{BIC}$ recupera todas as imagens corretamente, enquanto o BIC recupera a imagem não relevante R_9. Analisando essas figuras, pode-se concluir que a análise local das características visuais de baixo nível do SIFT-Texton$_{BIC}$ garante melhores resultados que a análise global do BIC.



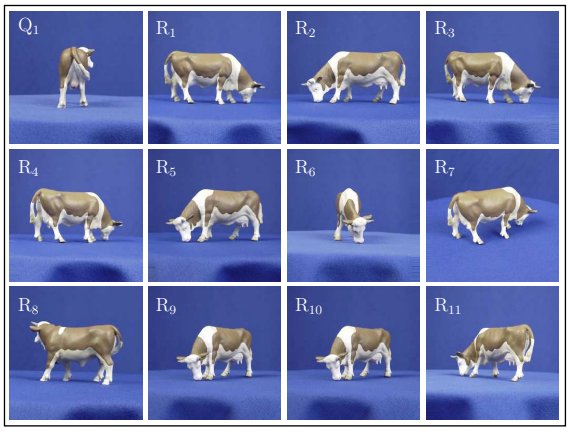
(a) SIFT-Texton$_{BIC}$ recupera todas as imagens corretamente.



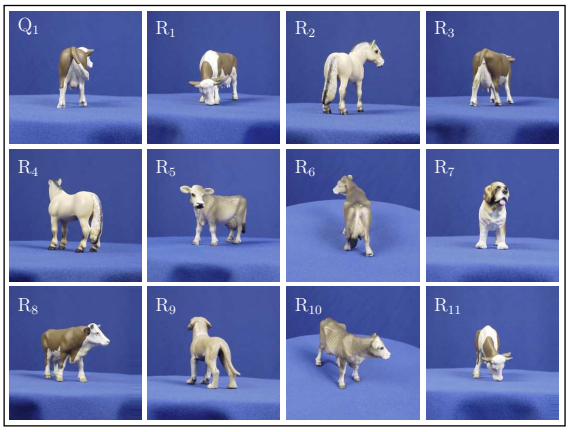
(b) BIC recupera a imagem não relevante R_9.

Figura 3.12: Onze primeiras imagens recuperadas na consulta Q_1.

As Figuras 3.13(a) e 3.13(b) mostram as onze primeiras imagens recuperadas na consulta Q_2 para os métodos SIFT-Texton$_{BIC}$ e BIC, respectivamente. O SIFT-Texton$_{BIC}$ recupera todas as imagens corretamente, enquanto o BIC recupera as imagens não relevantes R_2, R_4, R_5, R_7 e R_9. Analisando essas figuras, pode-se concluir que o SIFT-Texton$_{BIC}$ captura variações em iluminação, oclusão e foco de maneira mais eficaz que o BIC.



(a) SIFT-Texton$_{BIC}$ recupera todas as imagens corretamente.



(b) BIC recupera as imagens não relevantes R_2, R_4, R_5, R_7 e R_9.

Figura 3.13: Onze primeiras imagens recuperadas na consulta Q_2.

Capítulo 4

DAH-Cluster

O algoritmo mais simples para realizar uma consulta em um sistema de recuperação de imagens por conteúdo é uma busca sequencial. Nesse método, todas as imagens do banco são analisadas sequencialmente. Embora simples, esse método é inviável para grandes coleções, uma vez que o tempo gasto para processar uma consulta é proporcional ao tamanho do banco [68].

Existem extensivos estudos em técnicas de indexação e estruturas de dados para acelerar o processo de consulta de uma imagem de forma que imagens relevantes possam ser encontradas rapidamente [15, 21]. Entretanto, na maioria dessas técnicas, a eficiência é comprometida pela sobreposição dos dados, característica comum na recuperação de imagens por conteúdo [31, 32].

Em contraste, agrupamento é uma das técnicas de descoberta de conhecimento mais úteis para identificar correlações em grandes conjuntos de dados. Nesse método, os vetores de características associados às imagens são organizados em grupos, aos quais é associado um elemento representativo. Assim, uma consulta é realizada utilizando-se apenas um pequeno número de elementos representativos, reduzindo o espaço de busca [62].

Este capítulo apresenta o DAH-Cluster (*Divisive-Agglomerative Hierarchical Clustering*), um método hierárquico divisivo e aglomerativo de agrupamento. Esse método combina características dos modelos hierárquicos divisivo e aglomerativo de agrupamento. Além disso, o DAH-Cluster introduz um novo conceito, chamado fator de reagrupamento, que permite agrupar elementos similares que seriam separados pelos paradigmas tradicionais.

A utilização do DAH-Cluster na recuperação de imagens por conteúdo consiste em duas etapas: (1) construir uma estrutura hierárquica *offline* que melhor representa os relacionamentos semânticos entre as imagens; e (2) utilizar essa estrutura para reduzir o tempo de processamento de uma consulta *online*.

O restante deste capítulo é organizado como segue. A Seção 4.1 apresenta o DAH-

Cluster. Por fim, a Seção 4.2 discute os resultados experimentais obtidos na aplicação desse método na recuperação de imagens por cor.

4.1 DAH-Cluster

Em geral, muitos acreditam que os métodos hierárquicos aglomerativos de agrupamento garantem melhores resultados que os métodos hierárquicos divisivos [77]. Esta seção apresenta um método híbrido: hierárquico divisivo e aglomerativo de agrupamento.

Esse método é híbrido porque combina as características dos paradigmas hierárquicos divisivo e aglomerativo. Essa combinação permite reduzir os erros obtidos por essas técnicas, aumentando a qualidade das tarefas de agrupamento.

As seções seguintes apresentam mais detalhes sobre esse método.

4.1.1 Visão geral

Sejam:

- c – um grupo;
- $c.rep$ – o elemento representativo do grupo c;
- $c.elements$ – o conjunto de elementos do grupo c;
- $c.son$ – um apontador para o próximo nível da hierarquia de grupos;
- C – um conjunto de grupos;
- k – o número de grupos em todas as tarefas de agrupamento;
- $f \in [0, 1)$ – um fator de reagrupamento;
- E – o conjunto de elementos analisados;
- D – uma medida para avaliar as dissimilaridades entre os elementos em E.

A Figura 4.1 ilustra uma representação esquemática do DAH-Cluster. Inicialmente, têm-se E elementos para serem agrupados (a). Após a primeira iteração, é construído o nível C_1 da hierarquia, com k grupos $c_1 \dots c_k$ (b). Para cada grupo c_i em C_1 , são selecionados os $\lfloor f \times k \rfloor$ grupos mais próximos, criando um novo conjunto de elementos E_{c_i} (c). Para cada conjunto de elementos E_{c_i} é executado um novo processo de agrupamento, gerando um novo nível na hierarquia, composto pelos grupos $c_{11} \dots c_{1k} \dots c_{k1} \dots c_{kk}$ (d). Esse processo é repetido enquanto $|E_{c_i}| > k$ (e-h).

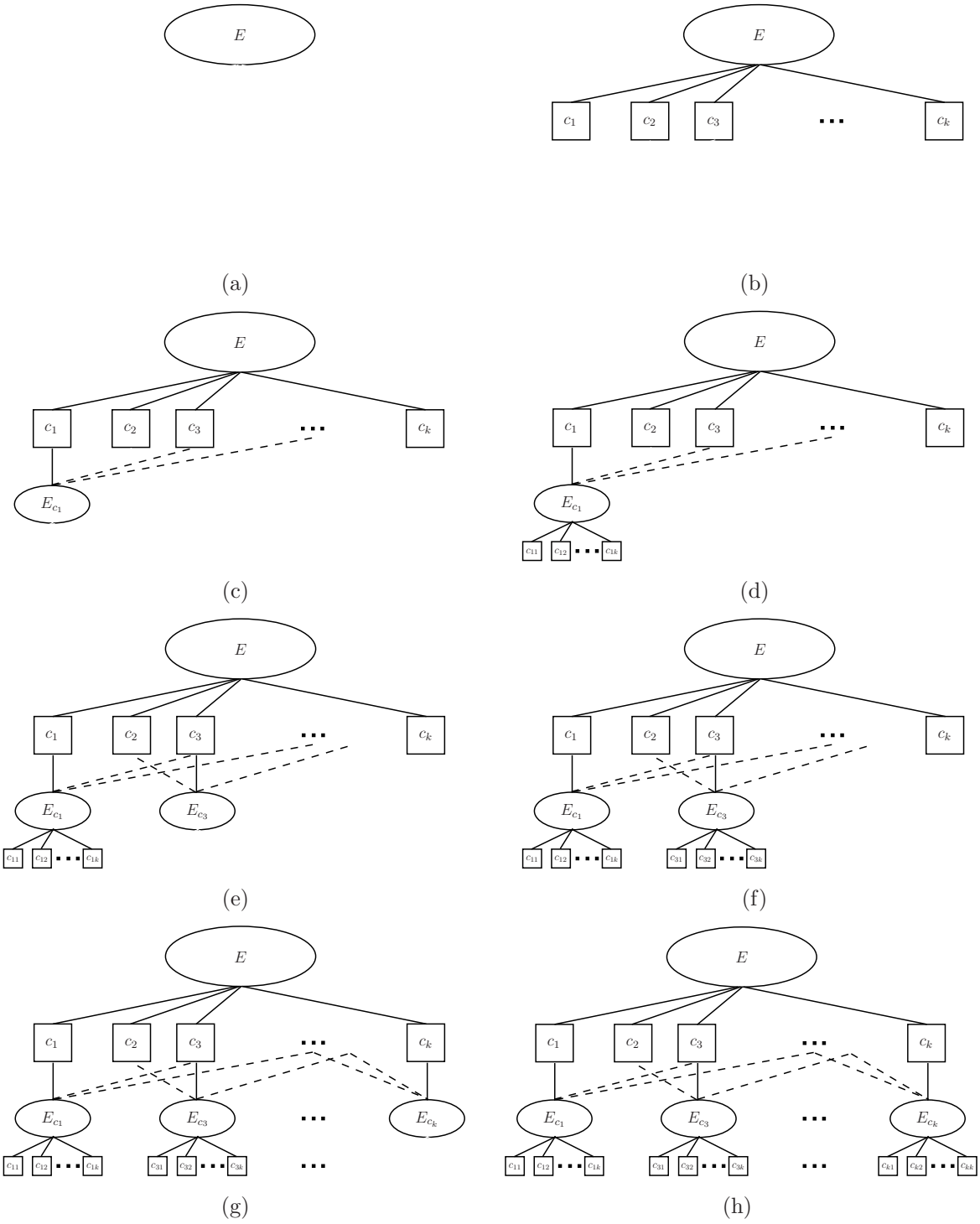


Figura 4.1: Representação esquemática do DAH-Cluster.

O Algoritmo 3 apresenta o DAH-Cluster. Esse método consiste em três etapas: (1) inicialmente, executa-se um estágio de agrupamento no conjunto de elementos E, gerando um conjunto de grupos C (linha 2); (2) para cada grupo $c \in C$, é construído um novo conjunto de elementos E^* agrupando-se os elementos $c.elements$ dos $\lfloor f \times k \rfloor$ grupos mais próximos ao representante $c.rep$ (linhas 4–8); (3) se o número de elementos em E^* for maior que o número de grupos k, então um novo nível $c.son$ é criado na hierarquia de c, reiniciando as etapas (1–3) para o conjunto de elementos E^* (linhas 9–11).

Algoritmo 3 DAH-Cluster

Entrada: O número de grupos k, o fator de reagrupamento $f \in [0, 1)$, o conjunto de elementos E e a função de distância D.

```
1:  procedure DAH-Cluster(k,f,E,D)
2:      C ← Cluster(k, E, D)
3:      enquanto c ∈ C faça
4:          C* ← ⌊f × k⌋ grupos mais próximos de c ∈ C
5:          E* ← {}
6:          enquanto c* ∈ C* faça
7:              E* ← E* ∪ c*.elements
8:          fim enquanto
9:          se |E*| > k então                    ▷ | · | é o tamanho de {·}
10:             c.son ← DAH-Cluster(k, f, E*, D)
11:         fim se
12:     fim enquanto
13: fim procedure
```

Esse método representa um paradigma hierárquico aglomerativo no sentido que começa com cada elemento como um grupo representativo e encontra k grupos. Por outro lado, representa um paradigma hierárquico divisivo no sentido que começa com um conjunto de elementos E e, iterativamente, particiona E em subconjuntos E^*. Esse particionamento é feito por meio de um fator de reagrupamento que permite agrupar elementos similares que seriam separados pelos paradigmas tradicionais.

O procedimento Cluster(k, E, D) pode ser qualquer método particional de agrupamento, como K-means e K-medoids [25]. K-medoids é recomendado quando se deseja independência do espaço utilizado, pois requer apenas uma matriz de dissimilaridade entre os elementos. K-means, por outro lado, exige uma função de distância do espaço Euclidiano para medir as dissimilaridades entre os elementos.

4.1.2 Convergência

Esta seção apresenta uma prova para a convergência do DAH-Cluster.

Teorema 1. *O DAH-Cluster sempre converge no número de grupos (largura) e no número de níveis (profundidade).*

Demonstração. Existem duas possibilidades: (1) a convergência em largura e (2) a convergência em profundidade.

A primeira, é diretamente controlada pelo número de grupos k. Esse método utiliza algoritmos estáveis de agrupamento (como K-means ou K-medoids), que sempre convergem para uma solução, tanto pela estabilidade quanto por um número fixo de iterações. Uma prova da convergência desses algoritmos pode ser encontrada em [25].

A segunda, é controlada pelo fator de reagrupamento f. Analisando a hierarquia, tem-se que

$$|c_i| < |E|, \tag{4.1}$$

isto é, o tamanho de cada grupo é sempre menor que $|E|$. Além disso, pode-se notar que

$$\sum_{i=1}^{k} |c_i| = |E|, \tag{4.2}$$

ou seja, a soma dos elementos de todos os grupos em um nível da hierarquia é sempre igual a $|E|$. O próximo nível da hierarquia para o grupo c_i é formado pelos elementos dos $\lfloor f \times k \rfloor$ grupos mais próximos. Assim,

$$\lfloor f \times k \rfloor < k, \tag{4.3}$$

isto é, o número de grupos selecionados para o próximo nível é sempre menor que k, uma vez que $f \in [0, 1)$. Dessa forma,

$$\sum_{i=1}^{\lfloor f \times k \rfloor} |c_i| < |E|, \tag{4.4}$$

ou seja, a soma dos elementos dos grupos selecionados na etapa de reagrupamento para o nível c_i é sempre menor que $|E|$. Entretanto, o número de elementos para serem agrupados no próximo nível $|E^*|$ é dado por

$$|E^*| = \sum_{i=1}^{\lfloor f \times k \rfloor} |c_i| \tag{4.5}$$

provando que

$$|E^*| < |E| \tag{4.6}$$

□

4.2 Experimentos

Esta seção apresenta os resultados da aplicação do DAH-Cluster na recuperação de imagens por cor. Nos experimentos realizados, as características visuais das imagens foram extraídas utilizando os descritores apresentados na Seção 2.2.4.

A Seção 4.2.1 descreve a metodologia de validação utilizada na avaliação desse método. Na Seção 4.2.2 é apresentado um exemplo visual de um sistema de recuperação de imagens por cor utilizando o DAH-Cluster. Por fim, a Seção 4.2.3 discute os resultados experimentais obtidos na aplicação desse método no processamento de consultas.

4.2.1 Metodologia de validação

Neste trabalho, foi novamente adotado o paradigma de consultas por exemplo (QBE – *Query-By-Example*) [46], descrito na Seção 3.4.1. Foram utilizados dois bancos de imagens conhecidos na literatura. O primeiro, denominado *Relevants*, é uma seleção de 1.624 imagens de 50 classes distintas do *Corel Photo Gallery*, reportado em [67]. Esse banco é bastante heterogêneo e composto por imagens de diferentes domínios. A Figura 4.2 apresenta alguns exemplos das imagens desse banco.

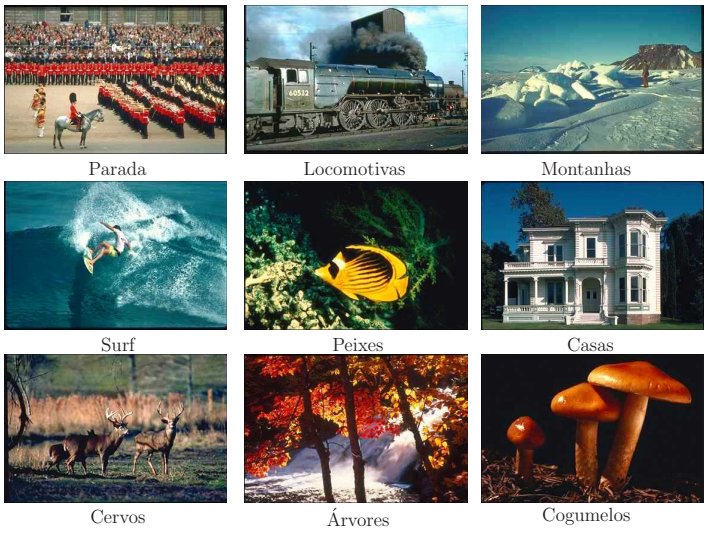


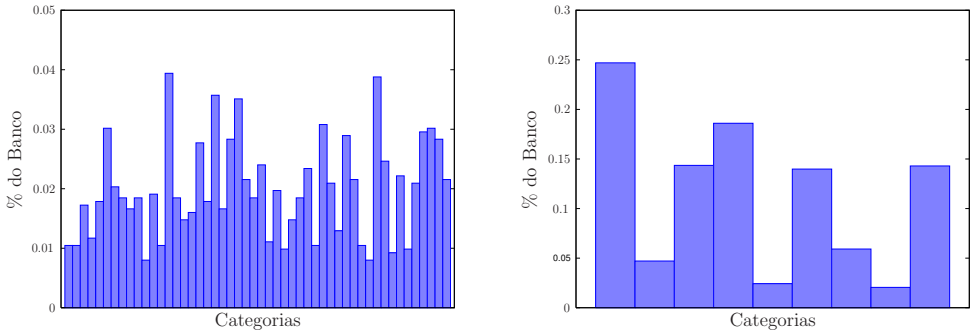
Figura 4.2: Exemplos de imagens do banco *Relevants*.

O segundo, denominado *FreeFoto*, é um banco livre[1], é composto por 3.462 imagens de paisagens naturais divididas em 9 classes. A Figura 4.3 apresenta alguns exemplos das imagens desse banco.

[1] *FreeFoto* – `http://www.freefoto.com/preview.jsp?id=15-19-1/`

Figura 4.3: Exemplos de imagens do banco *FreeFoto*.

As Figuras 4.4(a) e 4.4(b) mostram a distribuição de imagens por classe para esses bancos. Note que o banco *FreeFoto* é desbalanceado. Essa característica, em geral, impõe uma dificuldade maior aos métodos de agrupamento.



(a) Relevants (b) FreeFoto

Figura 4.4: Distribuição de imagens por classe.

Para avaliar a eficiência desses métodos, cada banco de imagens foi dividido em dois conjuntos: treino e teste. Esse processo foi realizado por meio um sistema de validação cruzada de 5 partições, isto é, cada banco de imagens foi aleatoriamente dividido em 5 partes iguais: 4 para compor o conjunto de treino e 1 para compor o conjunto de teste. Esse processo foi repetido 10 vezes e o resultado médio foi obtido.

Os métodos analisados utilizaram o conjunto de treino para construir uma estrutura *offline* que melhor representa os relacionamentos semânticos entre as imagens. A seguir, as imagens do conjunto de teste foram utilizadas como imagem de consulta em relação às imagens do conjunto treino.

Nesses experimentos, foi avaliado o número de comparações necessárias para recuperar 30 imagens (*top-30*). Assim, o valor da precisão após 30 imagens serem recuperadas foi utilizado como referência da eficácia do sistema. Esse valor é uma estimativa do número de imagens recuperadas que um usuário médio avaliaria em um sistema [4].

4.2.2 Exemplos visuais

Esta seção apresenta um exemplo da utilização do DAH-Cluster na recuperação de imagens por cor. Nesse exemplo, foram selecionados 24 imagens de 8 classes (3 imagens por classe) do banco *Relevants*. As características visuais dessas imagens foram extraídas utilizando o BIC [67], descrito na Seção 2.2.4.

A Figura 4.5 apresenta uma imagem consulta e seus três primeiros resultados em uma busca sequencial. Esse método pode recuperar imagens não relevantes, por exemplo, a imagem R_3 que não pertence a mesma classe que a imagem de consulta Q. Isso ocorre devido a uma limitação ao trabalhar-se com descritores de imagens, que consiste na falta de coincidência entre a informação que pode ser extraída das imagens e a interpretação que essa imagem tem para um usuário em uma dada situação, um problema conhecido como descontinuidade semântica (*semantic gap*).

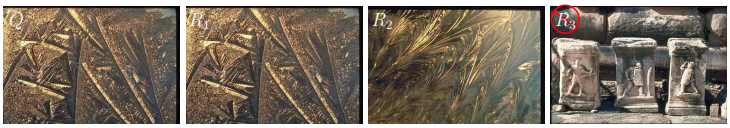


Figura 4.5: Três primeiras imagens recuperadas utilizando uma busca sequencial.

O DAH-Cluster cria uma estrutura hierárquica *offline* que melhor representa grupos de imagens, como mostra a Figura 4.6. Nessa estrutura, cada retângulo representa uma hierarquia de grupos c. A imagem localizada na linha superior de cada grupo representa o seu elemento representativo $c.rep$, enquanto as imagens abaixo constituem o conjunto de elementos $c.elements$. Os retângulos internos a cada grupo formam um novo nível $c.son$ dessa hierarquia.

Utilizando essa estrutura, o processo de recuperação é realizado como se segue. Em cada nível, a imagem de consulta Q é comparada com os elementos representativos ($c_1.rep$, $c_2.rep$, $c_3.rep$, $c_4.rep$ e $c_5.rep$) para encontrar o grupo mais similar (c_5). Iterativamente, esse processo é repetido para o próximo nível da hierarquia desse grupo ($c_5.son$) até que o último nível seja alcançado. No último nível (c_{51}) os grupos são ordenados de acordo com a similaridade de seus elementos representativos ($c_{511}.rep$, $c_{512}.rep$, $c_{513}.rep$, $c_{514}.rep$, $c_{515}.rep$) em relação à imagem de consulta Q. Por fim, seguindo-se essa ordem (c_{512}, c_{515}, c_{514}, c_{513} e c_{511}), os elementos dentro de cada grupo são ordenados e retornados até que seja atingido o número de elementos desejado na consulta.

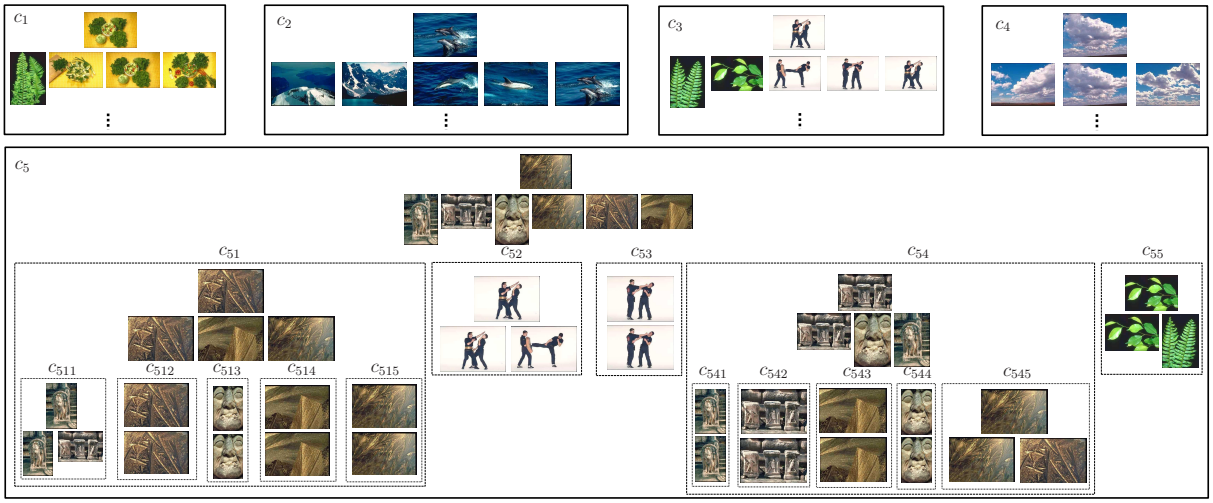


Figura 4.6: Exemplo da estrutura hierárquica criada pelo DAH-Cluster.

Além de garantir um processo de recuperação mais eficiente, o DAH-Cluster pode melhorar a eficácia de recuperação de imagens, como mostra a Figura 4.7. Em geral, isso pode ocorrer porque os elementos não relevantes para uma consulta normalmente se localizam nas bordas de seus grupos. Uma vez que o processo de recuperação é realizado por meio dos centróides de cada grupo, essas imagens tendem a ser eliminadas quando uma consulta é processada. Nesse exemplo, a imagem não relevante R_3 na Figura 4.5 é inserida no grupo c_{511}, cujo elemento representativo $c_{511}.rep$ é mais distante da imagem de consulta Q em relação ao elemento representativo $c_{514}.rep$, definindo uma nova ordenação para as imagens.

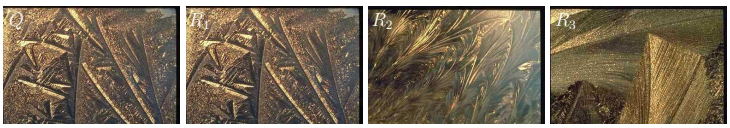


Figura 4.7: Três primeiras imagens recuperadas utilizando o DAH-Cluster.

4.2.3 Resultados experimentais

Nesta seção, são discutidos os resultados do DAH-Cluster na recuperação de imagens por cor. Nesses experimentos, o algoritmo K-medoids [25] foi utilizado em todas as tarefas de agrupamento.

Foram analisados e comparados os resultados de três paradigmas de agrupamento: particional (PC), hierárquico divisivo (DHC) e hierárquivo divisivo e aglomerativo (DAHC-f). Nesses resultados, DAHC-f indica o método DAH-Cluster com um fator de reagrupamento f, onde f é expresso em porcentagem.

Nesses experimentos, foram utilizados valores de k múltiplos de 5. Os valores para f foram escolhidos de forma que o cálculo $f \times k$ resultasse em números inteiros. Dessa forma, os resultados reportados utilizam $f \in \{0,05; 0,1; 0,15; 0,2\}$. Maiores valores para f resultam em um sobrecarga na criação *offline* da estrutura hierárquica de grupos.

O propósito desses experimentos é avaliar a eficiência de um sistema recuperação por cor ao utilizar cada um desses métodos no processamento de uma consulta. A eficácia desses sistemas está diretamente ligada ao descritor utilizado para extrair características visuais. Dessa forma, foram avaliados o comportamento de cada descritor mediante diferentes bancos de imagens.

Apesar de não compor a metodologia utilizada neste trabalho, foram incluídos nesta seção, os resultados obtidos nesses métodos para os bancos e descritores propostos no Capítulo 3.

As Figuras 4.8, 4.9, 4.10, 4.11 e 4.12 apresentam os resultados obtidos para os descritores GCH, LCH, CCV, CBC e BIC, respectivamente. À esquerda, estão dispostos o comportamento da eficácia obtida por esses descritores, enquanto que à direita, são apresentados os resultados obtidos para a sua eficiência. De cima para baixo, estão dispostos resultados para os bancos *Relevants*, *FreeFoto* e *ETHRel-72*, respectivamente.

Em cada gráfico, uma curva representa o comportamento de um dado método em relação ao número de grupos k. Esse comportamento é analisado avaliando-se os resultados médios obtidos ao utilizar 1/5 do banco como imagens de consulta em relação aos 4/5 restantes. A linha tracejada indica os resultados obtidos ao utilizar uma busca sequencial, que é independente do número de grupos k utilizado.

Analisando os resultados, pode-se notar que o DAH-Cluster reduz o número de comparações necessárias para realizar uma consulta independente do banco ou do descritor utilizado. Além disso, os ganhos obtidos na eficiência implicam uma queda de eficácia inferior a 0.05%.

Existe uma combinação entre os parâmetros f e k que garante melhores resultados. Quanto maior o número de grupos k, melhor é a eficácia obtida. Entretanto, quanto maior o número de grupos k, maior é o número de comparações necessárias no processamento de uma consulta. Por outro lado, quanto maior o fator de reagrupamento f, maior é a qualidade das tarefas de agrupamento. Contudo, quanto maior o fator de reagrupamento f, maior é a sobrecarga na criação *offline* da estrutura hierárquica de grupos.

A Figura 4.13 apresenta os resultados obtidos no banco *ETHRel-72* utilizando descritores baseados em características invariantes locais. À esquerda, estão dispostos o comportamento da eficácia obtida por esses descritores, enquanto à direita, são apresentados os resultados obtidos para a sua eficiência. De cima para baixo, estão dispostos resultados para os descritores SIFT-Texton$_{BIC}$, SIFT-Texton$_{GCH}$ e SIFT-128, respectivamente.

Observando os gráficos, pode-se notar que o método DAH-Cluster, além de reduzir o número de comparações necessárias para realizar uma consulta, melhora a eficácia de recuperação do SIFT-Texton$_{BIC}$.

Assim, combinação das técnicas SIFT-Texton$_{BIC}$ e DAH-Cluster, permite a criação de um mecanismo robusto para recuperar imagens por cor, atingindo resultados mais eficazes e mais eficientes que as abordagens tradicionais descritas na literatura.

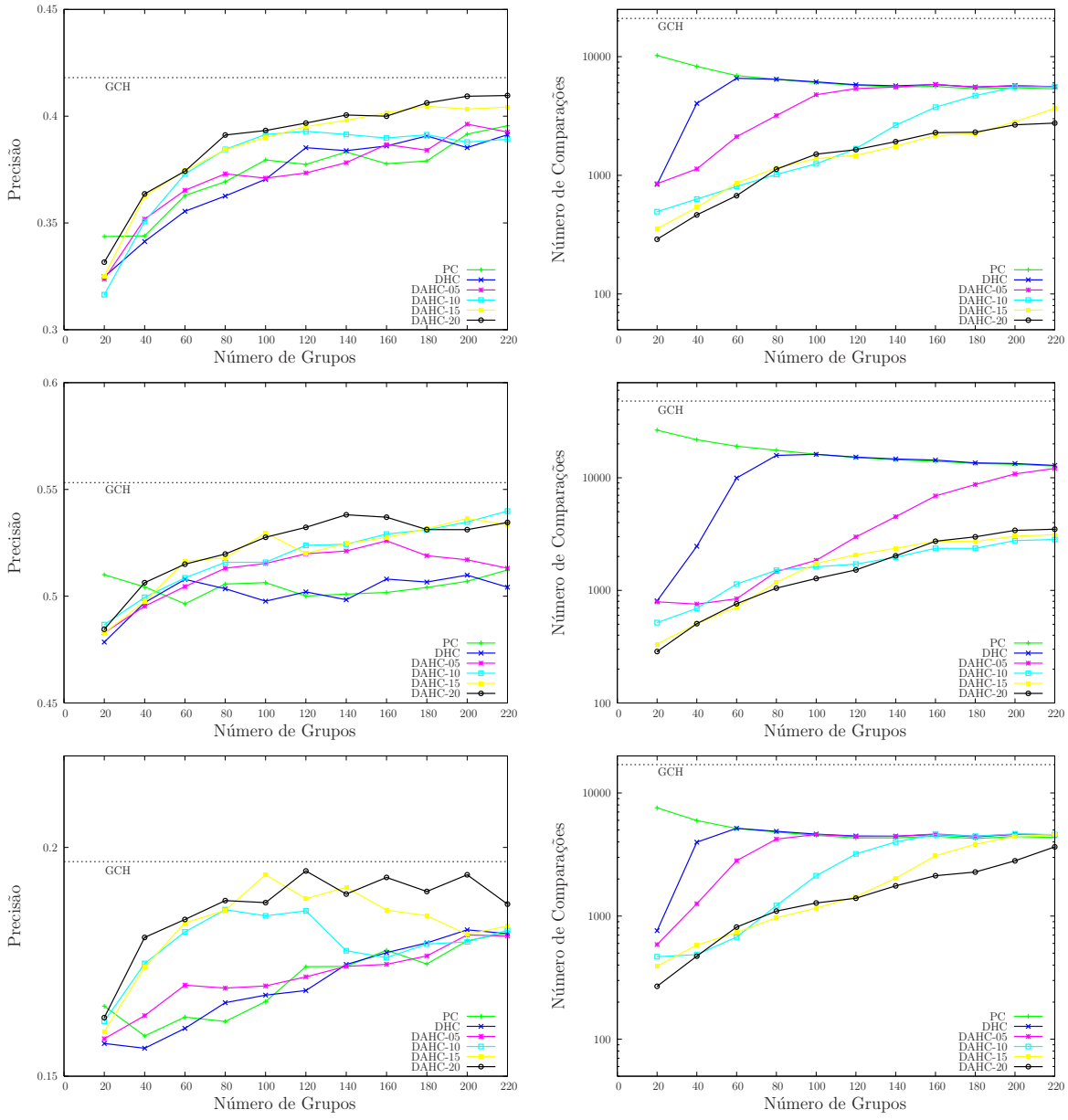


Figura 4.8: Eficácia (esquerda) e eficiência (direita) de recuperação utilizando o descritor GCH. Os resultados para os bancos *Relevants*, *FreeFoto* e *ETHRel-72*, são mostrados, respectivamente, de cima para baixo.

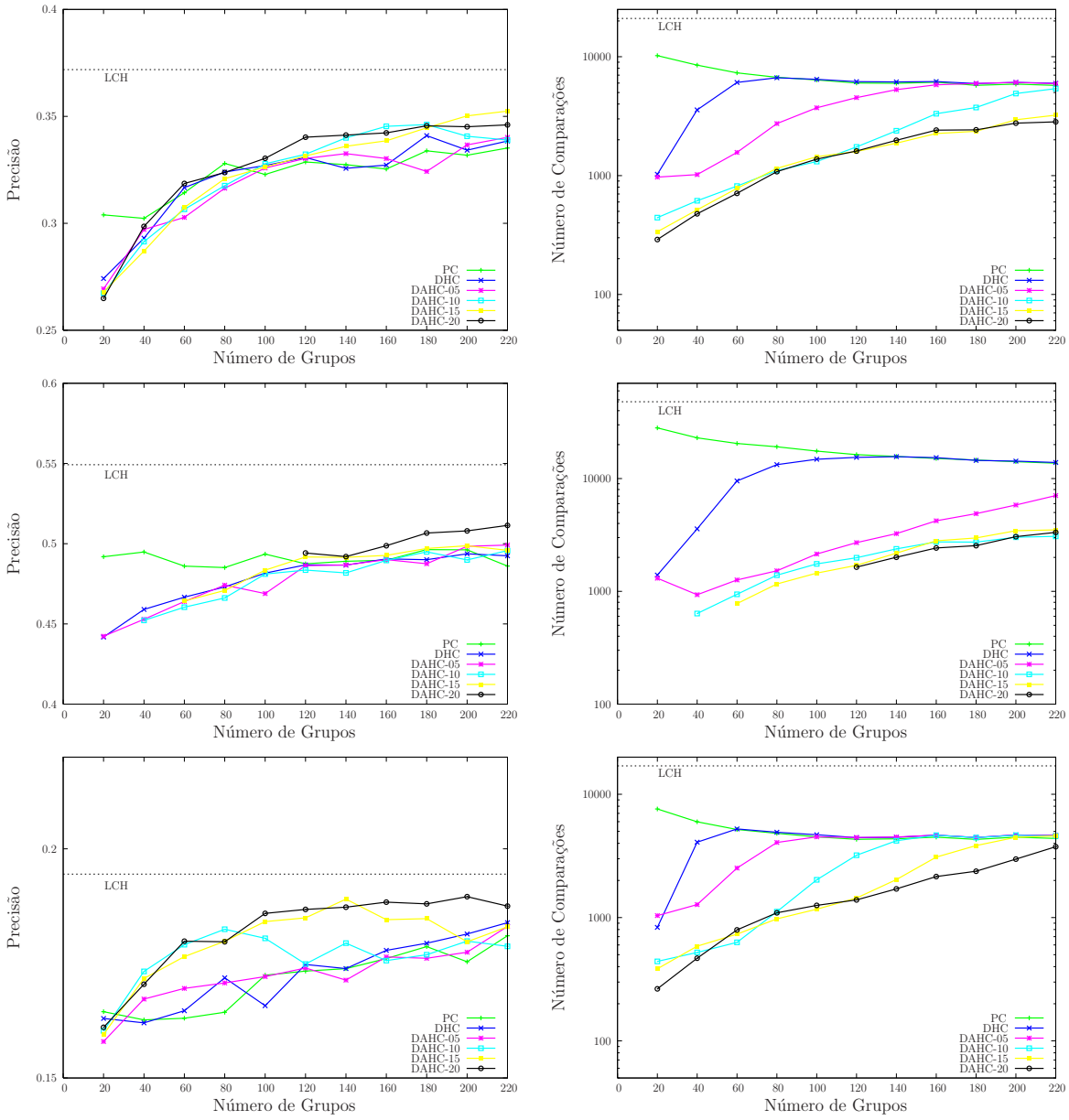


Figura 4.9: Eficácia (esquerda) e eficiência (direita) de recuperação utilizando o descritor LCH. Os resultados para os bancos *Relevants*, *FreeFoto* e *ETHRel-72*, são mostrados, respectivamente, de cima para baixo.

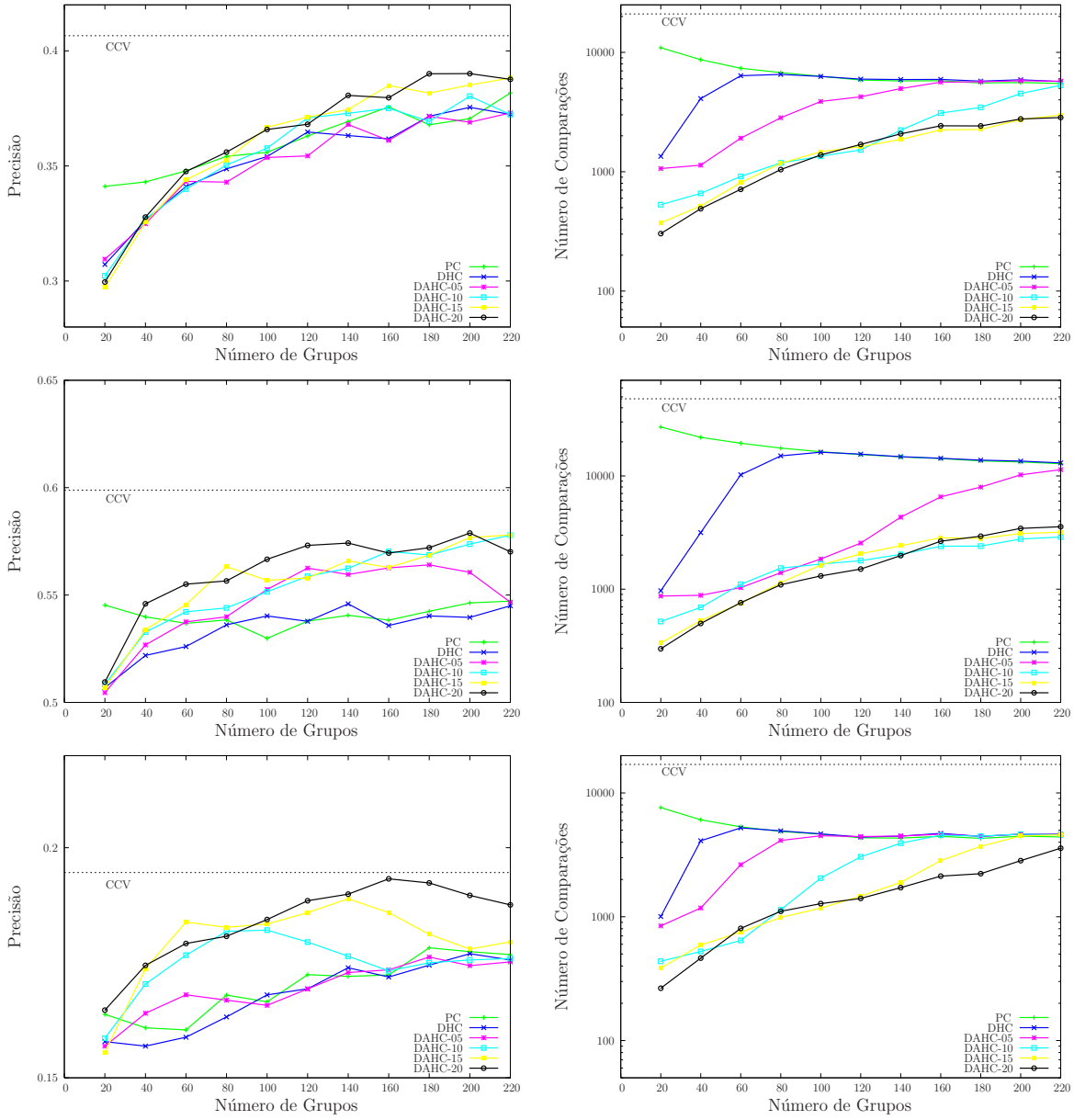


Figura 4.10: Eficácia (esquerda) e eficiência (direita) de recuperação utilizando o descritor CCV. Os resultados para os bancos *Relevants*, *FreeFoto* e *ETHRel-72*, são mostrados, respectivamente, de cima para baixo.

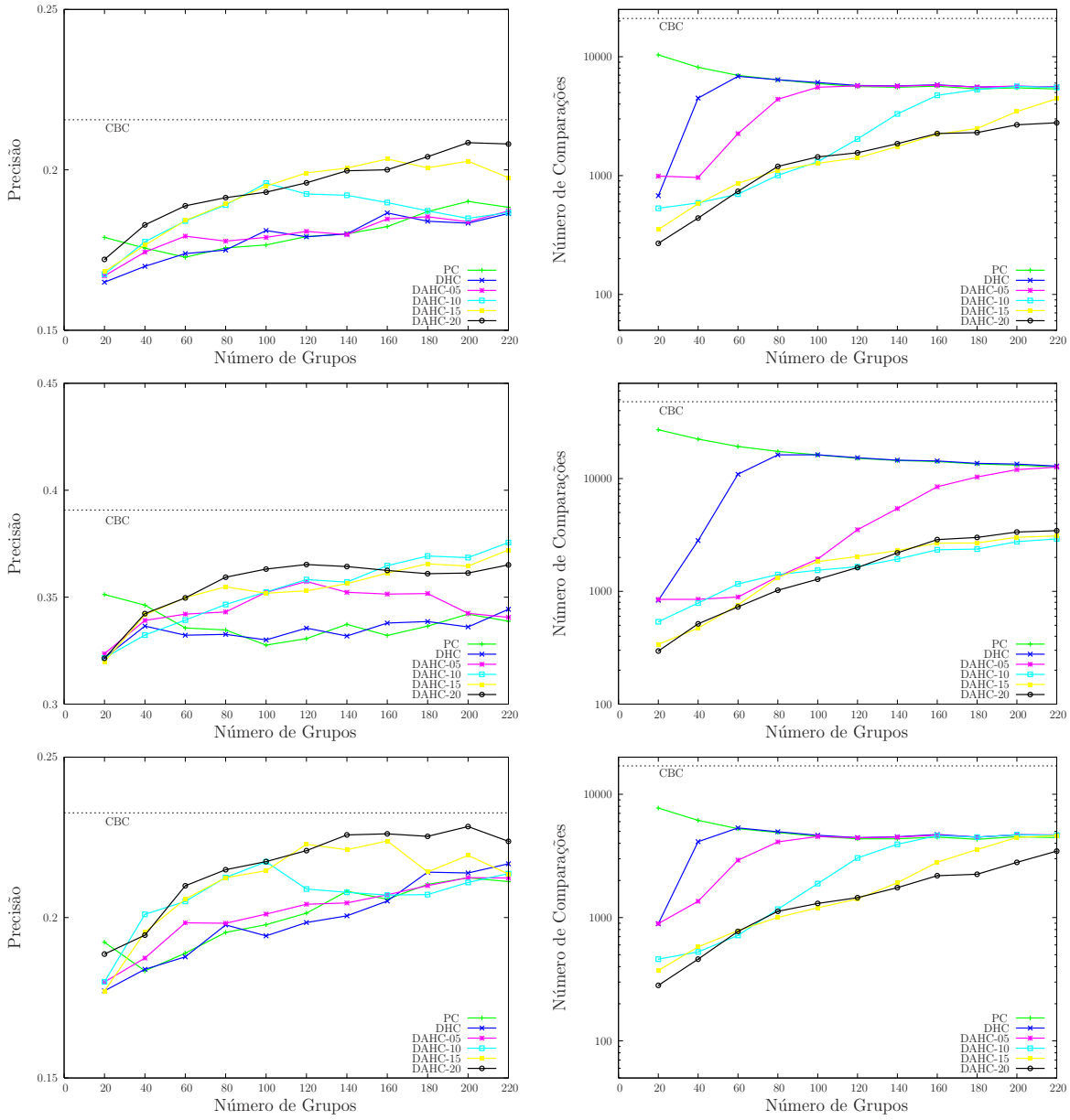


Figura 4.11: Eficácia (esquerda) e eficiência (direita) de recuperação utilizando o descritor CBC. Os resultados para os bancos *Relevants*, *FreeFoto* e *ETHRel-72*, são mostrados, respectivamente, de cima para baixo.

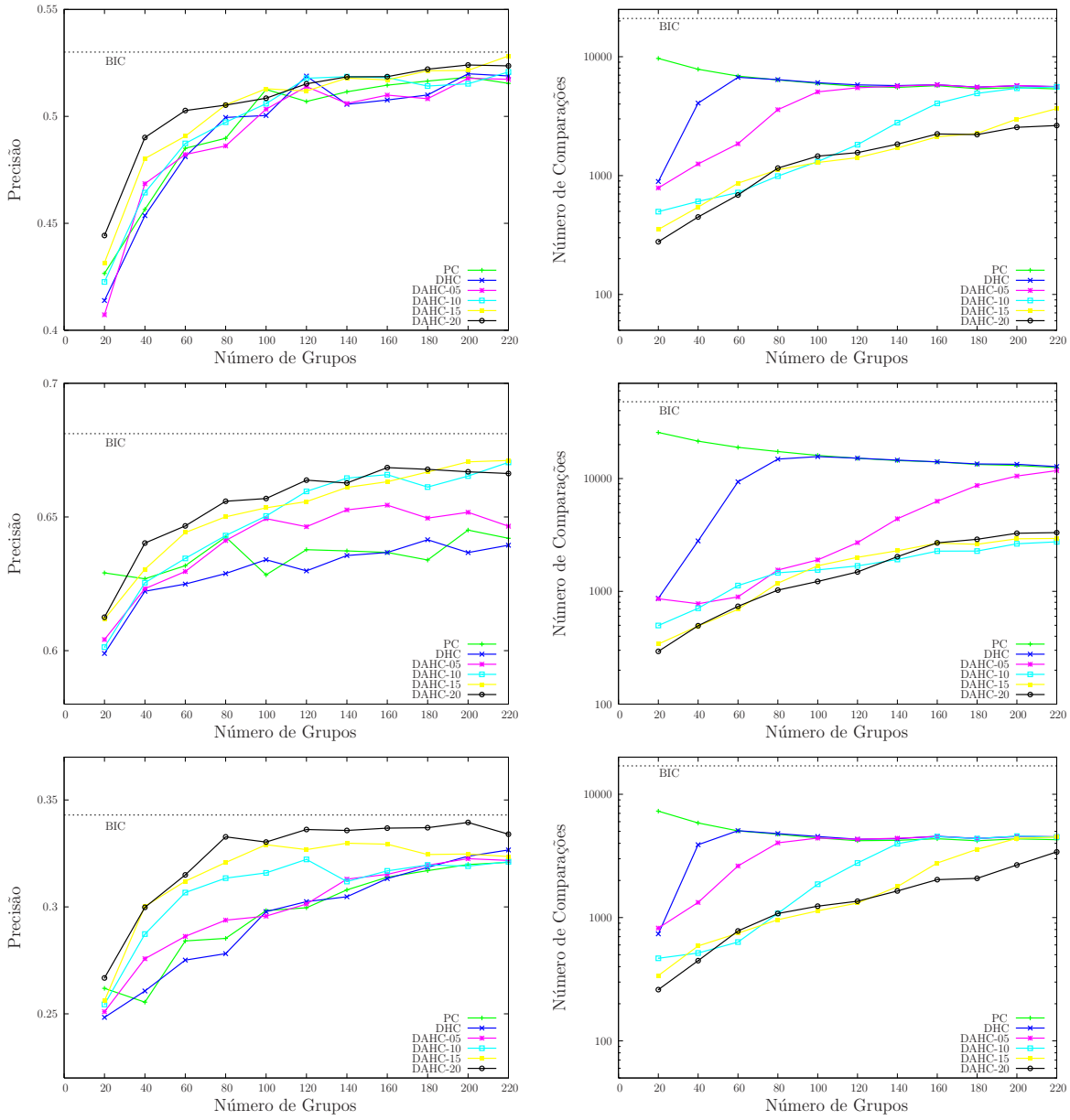


Figura 4.12: Eficácia (esquerda) e eficiência (direita) de recuperação utilizando o descritor BIC. Os resultados para os bancos *Relevants*, *FreeFoto* e *ETHRel-72*, são mostrados, respectivamente, de cima para baixo.

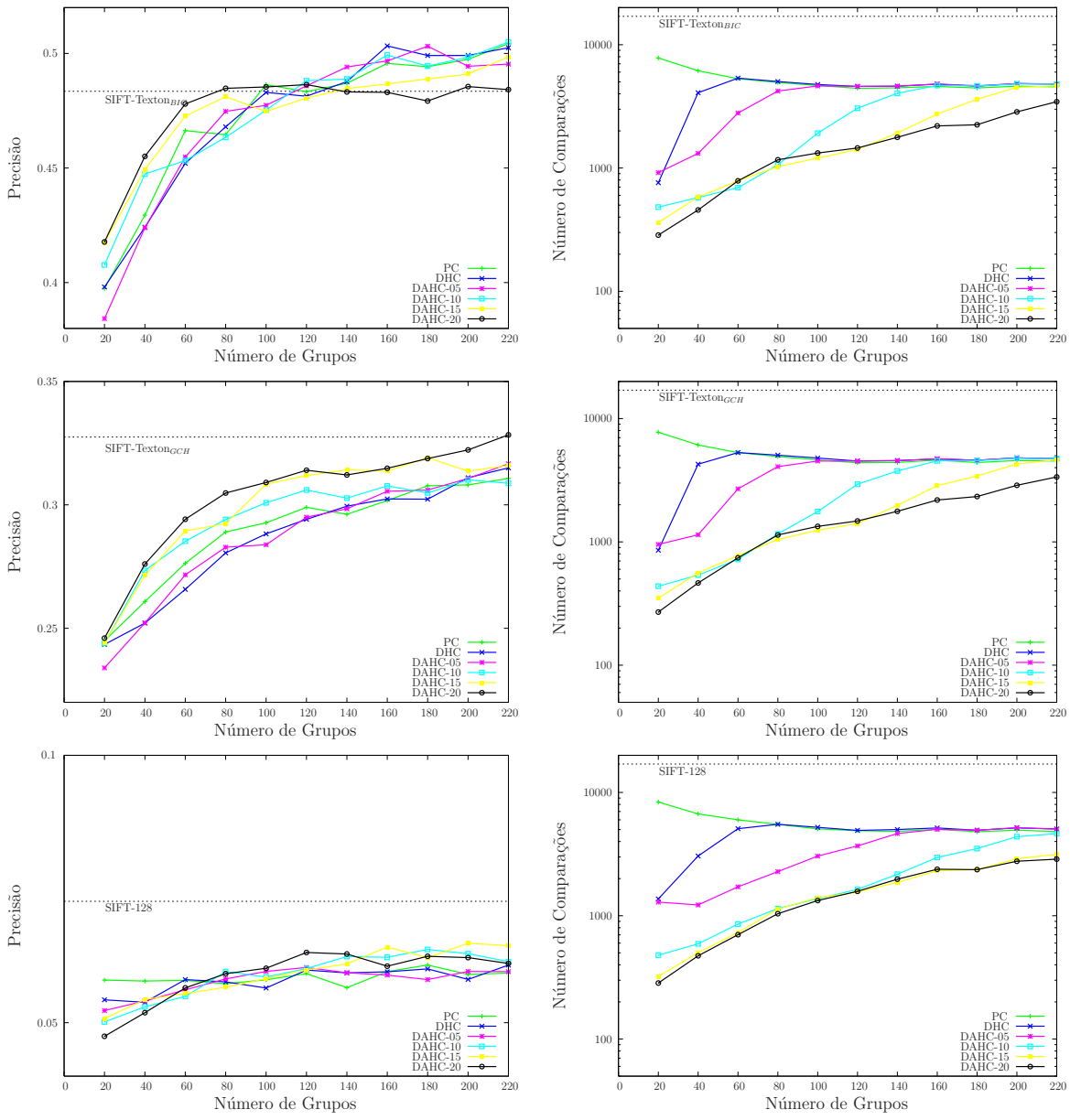


Figura 4.13: Eficácia (esquerda) e eficiência (direita) de recuperação para o banco de imagens *ETHRel-72*. Os resultados para os descritores SIFT-Texton$_{BIC}$, SIFT-Texton$_{GCH}$ e SIFT-128, são mostrados, respectivamente, de cima para baixo.

Capítulo 5

Conclusões

Este capítulo apresenta algumas considerações finais deste trabalho. A Seção 5.1 discute as principais contribuições do trabalho desenvolvido. Por fim, a Seção 5.2 apresenta possíveis extensões e trabalhos futuros.

5.1 Contribuições

Um sistema de recuperação de imagens por conteúdo ideal deve ser eficaz e eficiente. A eficácia é resultado de representações abstratas das imagens, enquanto a eficiência vem da organização dessas representações. Este trabalho apresenta técnicas que buscam melhorar a eficácia e a eficiência dos sistemas de recuperação de imagens por conteúdo.

Quanto à eficácia desses sistemas, este trabalho apresenta o SIFT-Texton, um método capaz de incorporar informações de médio nível nas características visuais de baixo nível. Esse método baseia-se na distribuição discreta de características invariantes locais e em propriedades de baixo nível das imagens.

Este trabalho mostra que o SIFT-Texton é apropriado para tarefas de recuperação de imagens por cor. Experimentos realizados neste trabalho mostram que esse método captura variações em iluminação, oclusão e foco de maneira mais eficaz que as técnicas tradicionais, aumentando a eficácia de recuperação dos métodos baseados em propriedades de baixo nível descritos na literatura.

Em relação às questões de eficiência, este trabalho apresenta o DAH-Cluster, um novo paradigma de agrupamento aplicado à recuperação de imagens por conteúdo. Esse método combina características dos paradigmas hierárquicos divisivo e aglomerativo de agrupamento, reduzindo os erros obtidos nesses paradigmas e melhorando a qualidade das tarefas de agrupamento. Além disso, o DAH-Cluster introduz um novo conceito, chamado fator de reagrupamento, que permite agrupar elementos similares que seriam separados pelos paradigmas tradicionais.

Experimentos mostram que essa técnica, além de reduzir o número de comparações necessárias para efetuar uma consulta em um banco de imagens, também pode melhorar a eficácia de recuperação desse processo.

O DAH-Cluster está ligado à escolha de dois parâmetros: o número de grupos k e o fator de reagrupamento f. Existe uma combinação de f e k que garante melhores resultados. Em geral, se k aumenta, a eficácia sobe. Entretanto, altos valores para k implicam um maior número de comparações necessárias no processamento de uma consulta. Por outro lado, se f aumenta, melhora a qualidade das tarefas de agrupamento. Contudo, altos valores para f implicam uma sobrecarga na criação *offline* da estrutura hierárquica de grupos.

A função de distância EMD, utilizada pelo SIFT-Texton para avaliar a dissimilaridade entre duas imagens, é computacionalmente cara. A combinação entre as técnicas SIFT-Texton e DAH-Cluster elimina a necessidade de comparar uma imagem de consulta com todo o banco e permite a criação de um mecanismo robusto de recuperação de imagens por conteúdo, atingindo resultados mais eficazes e mais eficientes que as abordagens tradicionais descritas na literatura.

Assim, as principais contribuições deste trabalho são:

1. Especificação e implementação de um novo método para recuperar imagens capaz de codificar informações de médio nível. Esse método é uma técnica genérica que pode ser aplicada a qualquer sistema de recuperação de imagens por conteúdo baseado em propriedades de baixo nível para incluir informações sobre iluminação, oclusão e foco.

2. Especificação e implementação de um novo paradigma de agrupamento de dados. Esse modelo é um método genérico que pode ser aplicado a qualquer sistema de recuperação de informação na redução do tempo de processamento de uma consulta.

5.2 Extensões e trabalhos futuros

Várias extensões a este trabalho, tanto do ponto de vista teórico quanto prático, podem ser alvo de pesquisas futuras. Algumas dessas extensões são apresentadas a seguir:

- **Uso de outros métodos de extração de pontos característicos.** O SIFT-Texton utiliza o SIFT para extrair pontos característicos. Entretanto, existem diversas outras abordagens que podem ser aplicadas nessa tarefa. Algumas dessas técnicas podem ser encontradas em [51].

- **Extensão para outras propriedades de baixo nível.** Este trabalho tem como foco a informação de cor. Porém, o SIFT-Texton é uma técnica genérica que pode ser aplicada a qualquer sistema de recuperação de imagens por conteúdo baseado em propriedades de baixo nível. Assim, um trabalho futuro seria testar essa técnica utilizando outras propriedades de baixo nível das imagens.

- **Combinação de descritores.** Na recuperação de imagens por conteúdo, uma imagem pode ser descrita a partir de suas propriedades de baixo nível, tais como cor, forma e textura. Um trabalho futuro nesse sentido consiste em investigar a utilização do SIFT-Texton combinado com mais de uma propriedade de baixo nível.

- **Extensão para outros domínios.** O DAH-Cluster é uma técnica genérica que pode ser aplicado a qualquer sistema de recuperação de informação na redução do tempo de processamento de uma consulta. Dessa forma, outra extensão possível seria testar essa técnica para outros tipos de sistemas de recuperação de informação – por exemplo, na recuperação de textos e documentos.

- **Atualização da estrutura hierárquica de grupos.** A utilização do DAH-Cluster na recuperação de informação consiste em duas etapas: (1) construção de uma estrutura hierárquica *offline* que melhor respresenta os relacionamentos semânticos entre os dados; e (2) utilização dessa estrutura para reduzir o tempo de processamento de uma consulta *online*. Este trabalho não abordou aspectos relacionados à atualização dessa estrutura. Assim, a definição e implementação de regras que permitam atualizar essa estrutura precisam ser investigadas.

- **Comparação do DAH-Cluster com estruturas de indexação.** As estruturas de indexação [31, 32] são as técnicas mais utilizadas na recuperação de informação para reduzir o tempo de processamento de uma consulta. Entretanto, na maioria dessas técnicas, a eficiência é comprometida pela sobreposição dos dados, característica comum na recuperação de imagens por conteúdo, uma vez que uma imagem pode ser vista sob diferentes interpretações. Nesse contexto, um trabalho futuro consiste em comparar essas técnicas tanto sob o ponto de vista quantitativo quanto qualitativo.

Apêndice A

SIFT em CBIR

Este apêndice discute resultados obtidos em experimentos realizados para avaliar o comportamento do SIFT na recuperação de imagens por conteúdo. Esses experimentos foram conduzidos a fim de avaliar as características mais relevantes envolvidas ao trabalhar-se com esse método, tais como: a área analisada ao redor de cada ponto característico, a dimensão dos vetores de características e a relevância das características visuais analisadas. Todos os experimentos realizados neste capítulo utilizam a metodologia descrita na Seção 3.4.1, tendo como referência o banco *Relevants*.

A.1 Canais de cor

O SIFT não codifica informações de cor. Dessa forma, os primeiros experimentos buscaram analisar quais canais de cor seriam mais relevantes ao utilizar esse método para recuperar imagens por conteúdo. Inicialmente, dois experimentos foram propostos: o primeiro, utilizando o canal Y do espaço de cor YCbCr; e o segundo, utilizando o canal V do espaço de cor HSV. A Figura A.1 apresenta os resultados obtidos nesses experimentos.

Analisando o gráfico obtido, pode-se notar que os canais Y e V apresentaram resultados muito próximos. Isso ocorreu porque o SIFT apresenta pouca variação em relação aos canais de cor. Visando validar essa hipótese, um novo experimento foi realizado agrupando-se os pontos característicos obtidos em cada canal do espaço de cor RGB e o canal Y do espaço de cor YCbCr. A Figura A.2 apresenta os resultados obtidos nesses experimentos.

Observando o gráfico, novamente, pode-se notar uma pequena variação nos resultados. Isso ocorre porque o SIFT codifica informações de gradiente. As informações de gradiente são preservadas em diferentes canais de cor. Nos experimentos seguintes, o canal V do espaço de cor HSV foi escolhido como padrão de informação analisada.

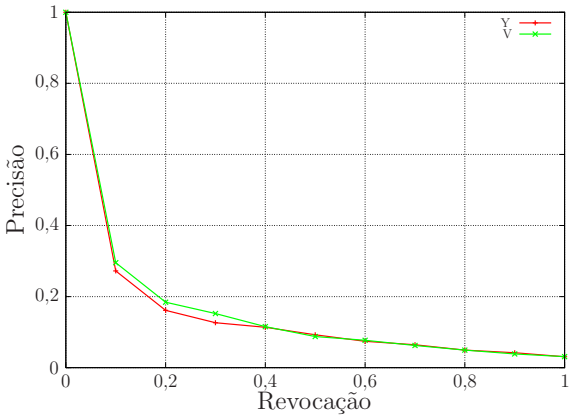


Figura A.1: Eficácia de recuperação utilizando os canais de cor Y e V.

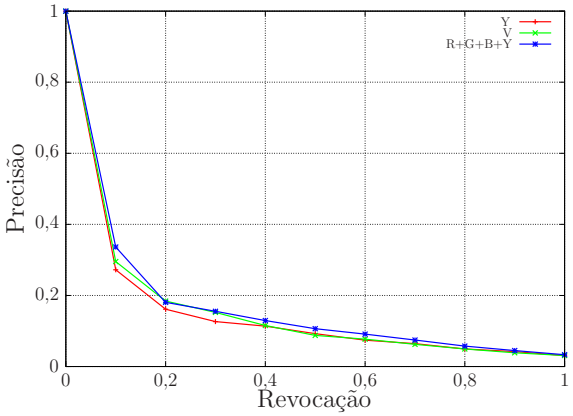


Figura A.2: Eficácia de recuperação utilizando os canais de cor R+G+B+Y.

A.2 Área analisada ao redor de cada ponto

Cada vetor de características obtido pelo SIFT é composto por um histograma de orientações das informações de gradiente da vizinhança ao redor do ponto característico ao qual esse vetor foi associado. Esse histograma acumula os valores de gradiente ao longo de 8 orientações. Para cada ponto é analisado uma janela de 16×16 pixels, a partir da qual são obtidos 4×4 histogramas.

Dessa forma, cada vetor de características é composto por $4 \times 4 \times 8 = 128$ dimensões. Essa dimensionalidade é controlada por meio de dois parâmetros: o primeiro, que controla a quantidade de histogramas e o tamanho da vizinhança analisada; e o segundo, que representa o número de orientações armazenadas por histograma.

A fim de analisar o comportamento desse método, três experimentos foram propostos variando-se o tamanho da vizinhança analisada ao redor de cada ponto característico: o

primeiro, considerando 4×4 histogramas e, portanto, uma vizinhança de 16×16 pixels; o segundo, considerando 2×2 histogramas e uma vizinhança de 8×8 pixels; e, por fim, o terceiro, considerando 1×1 histograma e, assim, uma vizinhança de 4×4 pixels.

A Figura A.3 apresenta os resultados obtidos nesses experimentos. Observando o gráfico, nota-se uma melhoria significativa ao reduzir a área de 16×16 para 8×8. Entretanto, os resultados pioram ao reduzir a área de 8×8 para 4×4. Isso ocorre devido à queda da capacidade descritiva do método ao analisar uma vizinhança muito pequena. Por outro lado, os resultados mais eficazes obtidos ao reduzir a área de 16×16 para 8×8 ocorrem devido ao excesso de informações irrelevantes codificadas ao utilizar uma vizinhança muito grande.

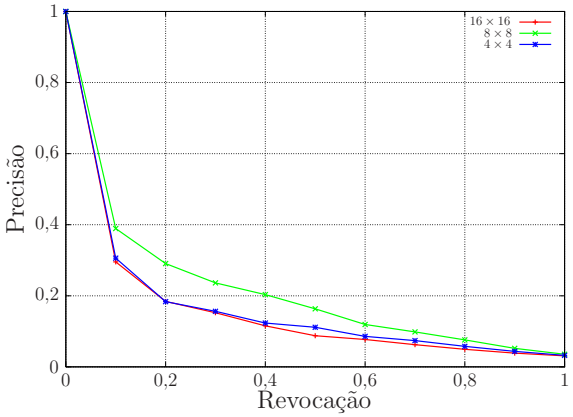


Figura A.3: Eficácia de recuperação variando-se a área analisada ao redor de cada ponto.

A.3 Dimensão dos vetores de características

Cada vetor de características do SIFT está associado a um ponto extremo - mínimo ou máximo - do gradiente da imagem analisado. Assim, as imagens que apresentam um baixo contraste de cor e uma textura suave, normalmente apresentam um número menor de pontos característicos. Dessa forma, a área total analisada por seus vetores de características é muito pequena em relação ao tamanho da imagem. Reduzir o tamanho da vizinhaça analisada ao redor de cada ponto característico de 16×16 para 8×8 pixels, conforme os resultados do experimento anterior, causa um aumento na eficácia de recuperação.

Por isso, novos experimentos foram propostos a fim de avaliar a relevância das características codificadas ao utilizar uma área de 16×16 pixels. Esses experimentos foram executados visando analisar o comportamento desse método variando-se a dimensão dos

vetores de características obtidos, mas mantendo-se a maior capacidade descritiva possível. Isso pode ser realizado utilizando uma estatística conhecida, denominada *Principal Components Analysis* (PCA) [30]. Nesses experimentos, foi utilizado PCA para reduzir a dimensão dos vetors de características obtidos, de 128 para 96, 64, 32, 16, 8 e 4.

A Figura A.4 apresenta os resultados obtidos nesses experimentos. Analisando o gráfico, pode-se observar o excesso de informações irrelevantes ao utilizar 128 dimensões, pois, os resultados melhoram até atingir somente 8 dimensões. Todavia, uma dimensionalidade reduzida ocasiona uma elevada queda na capacidade descritiva do método, justificando os resultados obtidos ao reduzir de 8 para 4 dimensões.

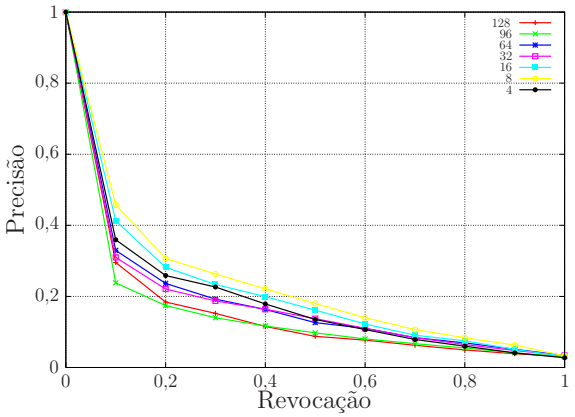


Figura A.4: Eficácia de recuperação reduzindo-se a dimensão do vetor de características.

A.4 Relevância das características visuais analisadas

As Figuras A.5 e A.6 apresentam as oito primeiras imagens recuperadas nas consultas de melhor e de pior resultados, respectivamente, utilizando o SIFT com um vetor de características de 128 dimensões. As setas brancas indicam a posição, a orientação e a escala associada a cada ponto característico. A primeira imagem em cada figura (canto superior esquerdo) é a imagem de consulta utilizada como referência nas buscas.

Analisando essas figuras, pode-se notar que as imagens que compõem a Figura A.5 contém um único objeto à frente de um mesmo fundo. Observando a Figura A.6, verifica-se que as oscilações da água na imagem de consulta, são confundidas com as reentrâncias das estalactites das imagens de cavernas. Isso ocorre porque o gradiente de um modelo apresenta uma alta influência ao longo das bordas dos objetos que o constituem, característica eficaz na recuperação de objetos. Na recuperação imagens de domínio geral, objeto e fundo podem se confundir.

Figura A.5: Oito primeiras imagens recuperadas na consulta de melhor resultado.

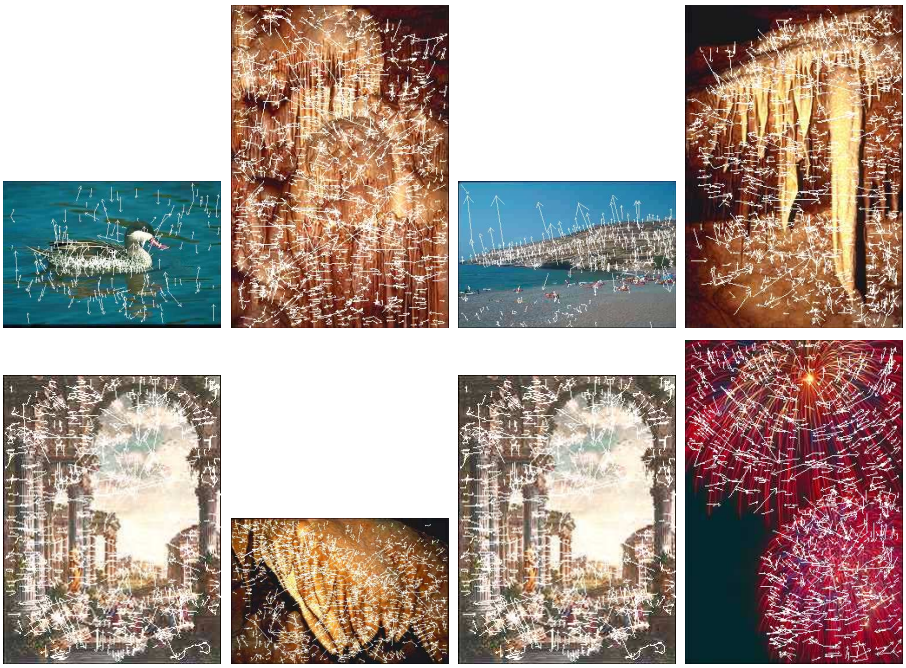


Figura A.6: Oito primeiras imagens recuperadas na consulta de pior resultado.

Uma solução para esse problema é substituir as informações de gradiente por informações de cor. A fim de validar essa hipótese, foi proposta uma variação do SIFT capaz de codificar informações de cor. Nessa proposta, os histogramas de gradiente que compõem os vetores de características do SIFT foram substituídos por histogramas de cor. Assim, um novo experimento foi realizado para comparar os resultados do SIFT utilizando essas informações. A Figura A.7 apresenta os resultados obtidos nesse experimento. Observando o gráfico, pode-se notar que as informações de cor garantem resultados melhores que as informações de gradiente na recuperação de imagens.

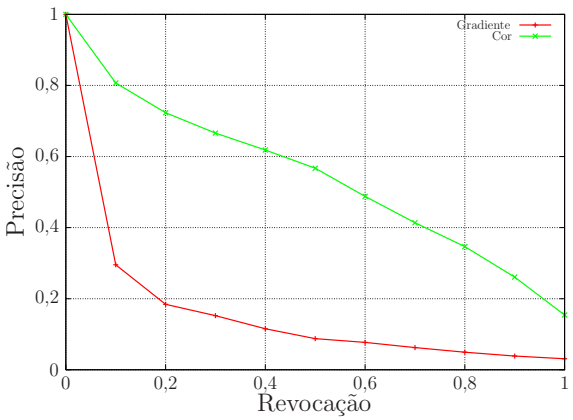


Figura A.7: Eficácia de recuperação substituindo-se gradiente por cor.

Um histograma de cor constitui um GCH da área analisada ao redor de cada ponto característico. Por isso, essa proposta foi chamada SIFT-Texton$_{GCH}$. Dessa forma, uma nova proposta desse método foi obtida substituindo-se GCH pelo BIC, dando origem ao SIFT-Texton$_{BIC}$. A Figura A.8 compara os resultados obtidos por essas propostas. Analisando o gráfico, pode-se notar que SIFT-Texton$_{BIC}$ apresenta resultados mais eficazes que o SIFT-Texton$_{GCH}$.

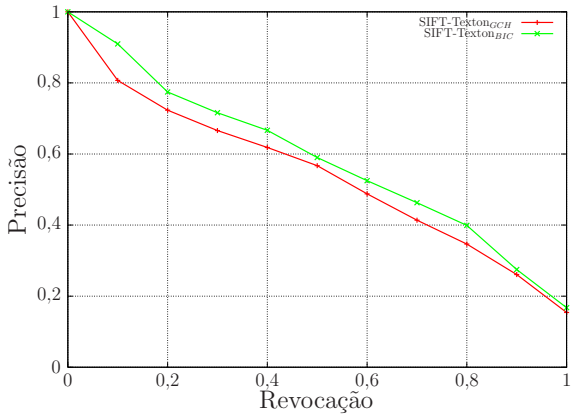


Figura A.8: Comparação entre as propostas SIFT-Texton$_{GCH}$ e SIFT-Texton$_{BIC}$.

A Figura A.9 apresenta os resultados obtidos pelo SIFT-Texton$_{BIC}$ aumentando-se o tamanho da vizinhança (16×16, 24×24 e 32×32) analisada ao redor de cada ponto característico. Observando o gráfico, pode-se notar que, independente da área analisada, os resultados se mantêm proximos.

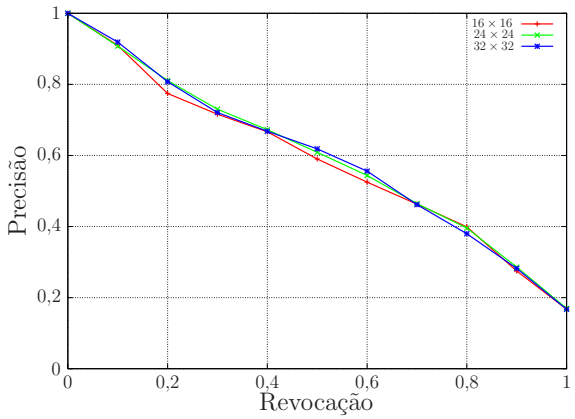


Figura A.9: Eficácia de recuperação do SIFT-Texton$_{BIC}$ variando-se a área analisada ao redor de cada ponto característico extraído.

Referências Bibliográficas

[1] F. A. Andaló. Descritores de forma baseados em *Tensor Scale*. Master's thesis, Instituto de Computação, Unicamp, Campinas, SP, March 2007.

[2] S. 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.

[3] N. Arica and F. T. Y. Vural. BAS: a perceptual shape descriptor based on the beam angle statistics. *Pattern Recognition Letters*, 24(9–10):1627–1639, June 2003.

[4] R. A. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1999.

[5] R. A. Baeza-Yates, B. Bustos, E. Chávez, N. Herrera, and G. Navarro. Clustering in metric spaces with applications to information retrieval. In W. Wu, H. Xiong, and S. Shekhar, editors, *Clustering and Information Retrieval*, volume 11 of *Network Theory and Applications*, chapter 1, pages 1–34. Kluwer Academic Publishers, Norwell, MA, USA, 2004.

[6] M. S. Bazaraa, J. J. Jarvis, and H. D. Sherali. *Linear Programming and Network Flows*. John Wiley and Sons Inc., New York, NY, USA, 1990.

[7] P. Berkhin. A survey of clustering data mining techniques. In J. Kogan, C. Nicholas, and M. Teboulle, editors, *Grouping Multidimensional Data: Recent Advances in Clustering*, volume 12, chapter 2, pages 25–71. Springer, Berlin, Heidelberg, Germany, 2006.

[8] S. Bhatia. Hierarchical clustering for image databases. In *Proceedings of the IEEE International Conference on Electro Information Technology*, pages 6–12, Lincoln, NE, USA, May 22–25 2005. IEEE Computer Society.

[9] A. Bimbo. *Visual information retrieval*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1999.

[10] M. Brown and D. G. Lowe. Unsupervised 3D object recognition and reconstruction in unordered datasets. In *Proceedings of the IEEE International Conference on 3D Digital Imaging and Modeling*, pages 56–63, Ottawa, Ontario, Canada, June 13–16 2005. IEEE Computer Society.

[11] M. Brown and D. G. Lowe. Recognising panoramas. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1218–1227, Nice, France, October 14–17 2003. IEEE Computer Society.

[12] D. Cai, X. He, Z. Li, W. Ma, and J. Wen. Hierarchical clustering of WWW image search results using visual, textual and link information. In H. Schulzrinne, N. Dimitrova, A. Sasse, S. B. Moon, and R. Lienhart, editors, *Proceedings of the ACM International Conference on Multimedia*, pages 952–959, New York, NY, USA, October 10–16 2004. ACM Press.

[13] C. Carson, M. Thomas, S. Belongie, J. M. Hellerstein, and J. Malik. Blobworld: A system for region-based image indexing and retrieval. In D. P. Huijsmans and A. W. M. Smeulders, editors, *Proceedings of the International Conference on Visual Information and Information Systems*, volume 1614 of *Lecture Notes in Computer Science*, pages 509–516, Amsterdam, The Netherlands, June 2–4 1999. Springer.

[14] M. Charikar. Similarity estimation techniques from rounding algorithms. In *Proceedings of the ACM International Symposium on Theory of Computing*, pages 380–388, Quebec, Canada, May 19–21 2002. ACM Press.

[15] E. Chávez, G. Navarro, R. A. Baeza-Yates, and J. L. Marroquín. Searching in metric spaces. *ACM Computing Surveys*, 33(3):273–321, September 2001.

[16] M. D. Cooper, J. Foote, A. Girgensohn, and L. Wilcox. Temporal event clustering for digital photo collections. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 1(3):269–288, August 2005.

[17] L. M. Cura. *Um Modelo para Recuperação por Conteúdo de Imagens de Sensoriamento Remoto*. PhD thesis, Instituto de Computação, Unicamp, Campinas, SP, December 2002.

[18] R. A. Elmasri and S. B. Navathe. *Fundamentals of Database Systems*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2005.

[19] P. Ferragina and A. Gulli. A personalized search engine based on web-snippet hierarchical clustering. In A. Ellis and T. Hagino, editors, *Proceedings of the ACM*

International Conference on World Wide Web, pages 801–810, Chiba, Japan, May 10–14 2005. ACM Press.

[20] G. W. Furnas, T. K. Landauer, L. M. Gomez, and S. T. Dumais. The vocabulary problem in human-system communication. *Communications of the ACM*, 30(11): 964–971, November 1987.

[21] V. Gaede and O. Günther. Multidimensional access methods. *ACM Computing Surveys*, 30(2):170–231, June 1998.

[22] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2001.

[23] K. Grauman and T. Darrell. Fast contour matching using approximate earth mover's distance. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, pages 220–227, Washington, DC, USA, June 27 – July 2 2004. IEEE Computer Society.

[24] K. Grauman and T. Darrell. Efficient image matching with distributions of local invariant features. In *Proceedings of the IEEE International Conference on on Computer Vision and Pattern Recognition*, pages 627–634, San Diego, CA, USA, June 20–26 2005. IEEE Computer Society.

[25] T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning: data mining, inference, and prediction*. Springer-Verlag, New York, NY, USA, 2001.

[26] K. A. Heller and Z. Ghahramani. Bayesian hierarchical clustering. In L. Raedt and S. Wrobel, editors, *Proceedings of the ACM International Conference on Machine Learning*, pages 297–304, Bonn, Germany, August 7–11 2005. ACM Press.

[27] J. Hong, H. Chen, and J. Hsiang. A digital museum of taiwanese butterflies. In *Proceedings of the ACM International Conference on Digital Libraries*, pages 260–261, San Antonio, TX, USA, June 2–7 2000. ACM Press.

[28] M. Iwayama and T. Tokunaga. Hierarchical bayesian clustering for automatic text classification. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 1322–1327, Montreal, Quebec, Canada, August 20–25 1995. Morgan Kaufmann.

[29] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: A review. *ACM Computing Surveys*, 31(3):264–323, September 1999.

[30] I. T. Jolliffe. *Principal Component Analysis*. Springer-Verlag, New York, NY, USA, 2002.

[31] C. T. Jr., A. J. M. Traina, B. Seeger, and C. Faloutsos. Slim-trees: High performance metric trees minimizing overlap between nodes. In C. Zaniolo, P. C. Lockemann, M. H. Scholl, and T. Grust, editors, *Proceeding of the International Conference on Extending Database Technology*, volume 1777 of *Lecture Notes in Computer Science*, pages 51–65, Konstanz, Germany, March 27–31 2000. Springer.

[32] C. T. Jr., A. J. M. 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, March 2002.

[33] A. Kak and C. Pavlopoulou. Content-based image retrieval from large medical databases. In *Proceedings of the IEEE International Symposium on 3D Data Processing Visualization and Transmission*, pages 138–149, Padova, Italy, June 19–21 2002. IEEE Computer Society.

[34] Y. Ke and R. Sukthankar. PCA-SIFT: A more distinctive representation for local image descriptors. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, pages 506–513, Washington, DC, USA, June 27 – July 2 2004. IEEE Computer Society.

[35] D. Kim and C. Chung. Qcluster: Relevance feedback using adaptive clustering for content-based image retrieval. In A. Y. Halevy, Z. G. Ives, and A. Doan, editors, *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 599–610, San Diego, CA, USA, June 9–12 2003. ACM Press.

[36] D. Kinoshenko, V. Mashtalir, E. Yegorova, and V. Vinarsky. Hierarchical partitions for content image retrieval from large-scale database. In P. Perner and A. Imiya, editors, *Proceedings of the International Conference on Machine Learning and Data Mining in Pattern Recognition*, volume 3587 of *Lecture Notes in Computer Science*, pages 445–455, Leipzig, Germany, July 9–11 2005. Springer.

[37] R. R. Korfhage. *Information Storage and Retrieval*. John Wiley and Sons Inc., New York, NY, USA, 1997.

[38] F. Korn, N. Sidiropoulos, C. Faloutsos, E. Siegel, and Z. Protopapas. Fast nearest neighbor search in medical image databases. In T. M. Vijayaraman, A. P. Buchmann, C. Mohan, and N. L. Sarda, editors, *Proceedings of the International Conference on Very Large Data Bases*, pages 215–226, Mumbai (Bombay), India, September 3–6 1996. Morgan Kaufmann.

[39] S. Kullback and R. A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22(1):79–86, March 1951.

[40] B. Leibe and B. Schiele. Analyzing appearance and contour based methods for object categorization. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, pages 409–415, Madison, WI, USA, June 16–22 2003. IEEE Computer Society.

[41] J. Li, J. Z. Wang, and G. Wiederhold. IRM: Integrated region matching for image retrieval. In *Proceedings of the ACM International Conference on Multimedia*, pages 147–156, Los Angeles, CA, USA, October 30 – November 3 2000. ACM Press.

[42] Y. Li and L. G. Shapiro. Consistent line clusters for building recognition in CBIR. In *Proceedings of the IEEE International Conference on Pattern Recognition*, volume 3, pages 952–956, Quebec, Canada, August 11–15 2002. IEEE Computer Society.

[43] Z. N. Li, O. R. Zaïane, and B. Yan. C-BIRD: Content-based image retrieval from digital libraries using ilumination invariance recognition kernel. Technical Report CMPT98-03, School of Computing Science, Simon Fraser University, 1998.

[44] H. Lieberman, E. Rozenweig, and P. Singh. Aria: An agent for annotating and retrieving images. *IEEE Computer*, 34(7):57–62, September 2001.

[45] T. Lindeberg. Scale-space theory: A basic tool for analysing structures at different scales. *Journal of Applied Statistics*, 21(2):224–270, April 1994.

[46] F. Long, H. J. Zhang, and D. Feng. Fundamentals of content-based image retrieval. In D. Feng, W. C. Siu, and H. J. Zhang, editors, *Multimedia Information Retrieval and Management: Technological Fundamentals and Applications*, volume 17 of *Signals and Communication Technology*, chapter 1, pages 1–26. Springer, Berlin, Heidelberg, Germany, 2003.

[47] D. G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1150–1157, Kerkyra, Corfu, Greece, September 20–25 1999. IEEE Computer Society.

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

[49] H. Lu, B. C. Ooi, and K. Tan. Efficient image retrieval by color contents. In W. Litwin and T. Risch, editors, *Proceedings of the International Conference on Applications of Databases*, volume 819 of *Lecture Notes in Computer Science*, pages 95–108, Vadstena, Sweden, June 21–23 1994. Springer.

[50] K. Mikolajczyk. *Detection of local features invariant to affine transformations*. PhD thesis, Institut National Polytechnique de Grenoble, France, July 2002.

[51] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, October 2005.

[52] W. Niblack, R. Barber, W. Equitz, M. Flickner, E. H. Glasman, D. Petkovic, P. Yanker, C. Faloutsos, and G. Taubin. The QBIC project: Querying images by content, using color, texture, and shape. In W. Niblack, editor, *Proceedings of the SPIE International Conference on Storage and Retrieval for Image and Video Databases*, volume 1908, pages 173–187, San Jose, CA, USA, January 31 – February 5 1993. SPIE.

[53] V. E. Ogle and M. Stonebraker. Chabot: Retrieval from a relational database of images. *IEEE Computer*, 28(9):40–48, September 1995.

[54] G. Pass, R. Zabih, and J. Miller. Comparing images using color coherence vectors. In *Proceedings of the ACM International Conference on Multimedia*, pages 65–73, Boston, MA, USA, November 18–22 1996. ACM Press.

[55] J. Puzicha, T. Hofmann, and J. M. Buhmann. Non-parametric similarity measures for unsupervised texture segmentation and image retrieval. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, pages 267–272, San Juan, Puerto Rico, June 17–19 1997. IEEE Computer Society.

[56] R. Ramakrishnan and J. Gehkre. *Database Management Systems*. McGraw-Hill Co., Inc.„ New York, NY, USA, 2003.

[57] Y. Rubner and C. Tomasi. Texture-based image retrieval without segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1018–1024, Kerkyra, Corfu, Greece, September 20–25 1999. IEEE Computer Society.

[58] Y. Rubner, C. Tomasi, and L. J. Guibas. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121, November 2000.

[59] N. Sahoo, J. Callan, R. Krishnan, G. T. Duncan, and R. Padman. Incremental hierarchical clustering of text documents. In P. S. Yu, V. J. Tsotras, E. A. Fox, and B. Liu, editors, *Proceedings of the ACM International Conference on Information and Knowledge Management*, pages 357–366, Arlington, Virginia, USA, November 6–11 2006. ACM Press.

[60] J. A. Sánchez, C. A. Flores, and J. L. Schnase. Mutant: Agents as guides for multiple taxonomies in the floristic digital library. In *Proceedings of the ACM International Conference on Digital Libraries*, pages 244–245, Berkeley, CA, USA, August 11–14 1999. ACM Press.

[61] J. Seo and B. Shneiderman. Interactive exploration of multidimensional microarray data: Scatterplot ordering, gene ontology browser, and profile search. Technical Report HCIL-2003-25, CS-TR-4486, UMIACS-TR-2003-55, ISR-TR-2005-68, Department of Computing Science, University of Maryland, 2003.

[62] M. Shyu, S. Chen, M. Chen, and C. Zhang. A unified framework for image database clustering and content-based retrieval. In S. Chen and M. Shyu, editors, *Proceedings of the ACM International Workshop on Multimedia Databases*, pages 19–27, Washington, DC, USA, November 13 2004. ACM Press.

[63] 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, December 2000.

[64] T. R. Smith. A digital library for geographically referenced material. *IEEE Computer*, 29(5):54–60, May 1996.

[65] R. O. Stehling. *Recuperação por Conteúdo em Grandes Coleções de Imagens Heterogêneas*. PhD thesis, Instituto de Computação, Unicamp, Campinas, SP, October 2002.

[66] R. O. Stehling, M. A. Nascimento, and A. X. Falcão. An adaptive and efficient clustering-based approach for content-based image retrieval in image databases. In *Proceedings of the IEEE International Database Engineering and Applications Symposium*, pages 356–365, Grenoble, France, July 16–18 2001. IEEE Computer Society.

[67] R. 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 ACM International Conference on Information and Knowledge Management*, pages 102–109, McLean, VA, USA, November 4–9 2002. ACM Press.

[68] R. O. Stehling, M. A. Nascimento, and A. X. Falcão. Techniques for color-based image retrieval. In C. Djeraba, editor, *Multimedia Mining - A Highway to Intelligent Multimedia Document*, volume 22 of *Multimedia Systems and Applications*, chapter 4, pages 61–80. Kluwer Academic Publishers, Norwell, MA, USA, 2002.

[69] R. 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, September 2003.

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

[71] C. Thies, A. Malik, D. Keysers, M. Kohnen, B. Fischer, and T. M. Lehmann. Hierarchical feature clustering for content-based retrieval in medical image databases. In M. Sonka and J. M. Fitzpatrick, editors, *Proceedings of the SPIE International Conference on Medical Imaging 2003: Image Processing*, volume 5032, pages 598–608, San Jose, CA, USA, May 15 2003. SPIE.

[72] R. S. Torres and A. X. Falcão. Content-based image retrieval: Theory and applications. *Revista de Informática Teórica e Aplicada*, 13(2):161–185, Special Issues 2006.

[73] R. S. Torres, E. M. Picado, A. X. Falcão, and L. F. Costa. Effective image retrieval by shape saliences. In *Proceedings of the IEEE Brazilian Symposium on Computer Graphics and Image Processing*, pages 167–174, Sao Carlos, Brazil, October 12–15 2003. IEEE Computer Society.

[74] R. S. Torres, A. X. Falcão, B. Zhang, W. Fan, E. A. Fox, M. A. Gonçalves, and P. Calado. A new framework to combine descriptors for content-based image retrieval. In *Proceedings of the ACM International Conference on Information and Knowledge Management*, pages 335–336, Bremen, Germany, October 31 – November 5 2005. ACM Press.

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

[76] R. C. Veltkamp and M. Tanase. Content-based image retrieval systems: A survey. Technical Report UU-CS-2000-34, Department of Computing Science, Utrecht University, 2000.

[77] Y. Zhao, G. Karypis, and U. M. Fayyad. Hierarchical clustering algorithms for document datasets. *Data Mining and Knowledge Discovery*, 10(2):141–168, March 2005.

[78] B. Zhu, M. Ramsey, and H. Chen. Creating a large-scale content-based airphoto image digital library. *IEEE Transactions on Image Processing*, 9(1):163–167, January 2000.