Recuperação de Imagens com Realimentação de Relevância Baseada em Programação Genética

Este exemplar corresponde à redação final da Dissertação devidamente corrigida e defendida por Cristiano Dalmaschio Ferreira e aprovada pela Banca Examinadora.

Campinas, 31 de julho de 2007.

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

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

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA DO IMECC DA UNICAMP

Bibliotecária: Maria Júlia Milani Rodrigues – CRB8a / 2116

Ferreira, Cristiano Dalmaschio

F413r Recuperação de imagens com realimentação de relevância baseada em programação genética / Cristiano Dalmaschio Ferreira -- Campinas, [S.P. :s.n.], 2007.

Orientador : Ricardo da Silva Torres

Dissertação (mestrado) - Universidade Estadual de Campinas, Instituto de Computação.

1. Programação genética (Computação). 2. Processamento de imagens. 3. Sistemas de recuperação de informação. I. Torres, Ricardo da Silva. II. Universidade Estadual de Campinas. Instituto de Computação. III. Título.

Título em inglês: Image retrieval with relevance feedback based on genetic programing.

Palavras-chave em inglês (Keywords): 1. Genetic programming. 2. Image processing. 3. Information storage and retrieval systems.

Área de concentração: Banco de dados

Titulação: Mestre em Ciência da Computação

Banca examinadora: Prof. Dr. Ricardo da Silva Torres (IC-UNICAMP)
Prof. Dr. Altigran Soares da Silva (DCC-UFAM)
Prof. Dr. Marcos André Gonçalves (DCC-UFMG)
Prof. Dr. Neucimar Jerônimo Leite (IC-UNICAMP)

Data da defesa: 31-07-2007

Programa de Pós-Graduação: Mestrado em Ciência da Computação

TERMO DE APROVAÇÃO

Dissertação Defendida e Aprovada em 31 de julho de 2007, pela Banca examinadora composta pelos Professores Doutores:

Prof. Dr. Altigran Soares da Silva
Universidade Federal do Amazonas

Prof. Dr. Marcos André Gonçalves
Universidade Federal de Minas Gerais .

Prof. Dr. Neucimar Jerônimo Leite
IC - UNICAMP.

Prof. Dr. Ricardo da Silva Torres
IC - UNICAMP.

Instituto de Computação
Universidade Estadual de Campinas

Recuperação de Imagens com Realimentação de Relevância Baseada em Programação Genética

Cristiano Dalmaschio Ferreira[1]

Agosto de 2007

Banca Examinadora:

- Prof. Dr. Ricardo da Silva Torres
  Instituto de Computação – Unicamp (Orientador)
- Prof. Dr. Altigran Soares da Silva
  Departamento de Ciência da Computação – UFAM
- Prof. Dr. Marcos André Gonçalves
  Departamento de Ciência da Computação – UFMG
- Prof. Dr. Neucimar Jerônimo Leite
  Instituto de Computação – Unicamp
- Prof. Dr. Cid Carvalho de Souza (Suplente)
  Instituto de Computação – Unicamp

[1]Suporte financeiro de: CNPq, 10/2005 à 02/2006; Fapesp, 03/2006 à 07/2007; Microsoft Escience; Tablet PC Technology Higher Education projects; CAPES; e FAEPEX.

Resumo

A técnica de *realimentação de relevância* tem sido utilizada com o intuito de incorporar a subjetividade da percepção visual de usuários à recuperação de imagens por conteúdo. Basicamente, o processo de realimentação de relevância consiste na: (i) exibição de um pequeno conjunto de imagens; (ii) rotulação dessas images pelo usuário, indicando quais são relevantes ou não; (iii) e finalmente, aprendizado das preferências do usuário a partir das imagens rotuladas e seleção de um novo conjunto de imagens para exibição. O processo se repete até que o usuário esteja satisfeito.

Esta dissertação apresenta dois arcabouços para recuperação de imagens por conteúdo com realimentação de relevância. Esses arcabouços utilizam programação genética para assimilar a percepção visual do usuário por meio de uma combinação de descritores. A utilização de programação genética é motivada pela sua capacidade exploratória do espaço de busca uma vez que esse espaço se adequa ao objetivo principal dos arcabouços propostos: encontrar, dentre todas as possíveis funções de combinação de descritores, aquela que melhor representa as características visuais que um usuário deseja ressaltar na realização de uma consulta.

Os arcabouços desenvolvidos foram validados por meio de uma série de experimentos, envolvendo três diferentes bases de imagens e descritores de cor, forma e textura para a caracterização do conteúdo dessas imagens. Os arcabouços propostos foram comparados com três outros métodos de recuperação de imagens por conteúdo com realimentação de relevância, considerando-se a eficiência e a efetividade no processo de recuperação. Os resultados experimentais mostraram a superioridade dos arcabouços propostos.

As contribuições dessa dissertação são: (i) estudo sobre diferentes técnicas de realimentação de relevância; (ii) proposta de dois arcabouços para recuperação de imagens por conteúdo com realimentação de relevância baseado em programação genética; (iii) implementação dos métodos propostos, validando-os por meio de uma série de experimentos e comparações com outros métodos.

Abstract

Relevance Feedback has been used to incorporate the subjectivity of user visual perception in content-based image retrieval tasks. The relevance feedback process consists in the following steps: (i) showing a small set of images; (ii) indication of relevant or irrelevant images by the user; (iii) and finally, learning the user needs from her feedback, and selecting a new set of images to be showed. This procedure is repeated until the user is satisfied.

This dissertation presents two content-based image retrieval frameworks with relevance feedback. These frameworks employ Genetic Programming to discover a combination of descriptors that characterize the user perception of image similarity. The use of genetic programming is motivated by its capability of exploring the search space, which deals with the major goal of the proposed frameworks: find, among all combination functions of descriptors, the one that best represents the user needs.

Several experiments were conducted to validate the proposed frameworks. These experiments employed three different images databases and color, shape and texture descriptors to represent the content of database images. The proposed frameworks were compared with three other content-based image retrieval methods regarding their efficiency and effectiveness in the retrieval process. Experiment results demonstrate the superiority of the proposed methods.

The contributions of this work are: (i) study of different relevance feedback techniques; (ii) proposal of two content-based image retrieval frameworks with relevance feedback, based on genetic programming; (ii) implementation of the proposed methods and their validation with several experiments, and comparison with other methods.

Agradecimentos

Agradeço...

Aos meus pais, Ferreira e Beth, e ao meu irmão Fausto, sempre presentes na minha vida, por toda confiança e apoio incondicionais, muito importantes para alcançar meus objetivos.

Ao meu orientador, Prof. Ricardo, pelos ensinamentos, paciência, compreensão e motivação, essenciais para o desenvolvimento deste trabalho e para meu crescimento profissional.

À minha namorada Jaudete, pela compreensão e paciência, sempre companheira nas dificuldades e desafios presentes no decorrer deste período de minha vida.

À Prof.^{a} Cláudia, pelos exemplos e ensinamentos dados, por me acolher em seu grupo de pesquisa.

Aos professores da UFV, por embasar meu caminho até este ponto, em especial ao Prof. Marcus, por despertar meu interesse por pesquisa.

Aos companheiros do LIS, pelos auxílios prestados e por tornar mais agradável o dia-a-dia de trabalho.

Ao meu tio João e à minha madrinha Ocilene (*Nina*), pela constante preocupação e apoio.

Este trabalho foi parcialmente financiado pela FAPESP (process no. 05/58228-0), CNPq, *Microsoft Escience project*, *Tablet PC Technology Higher Education projects*, CAPES e FAEPEX.

Sumário

Lista de Tabelas

Lista de Figuras

Capítulo 1

Introdução

Atualmente, um grande conjunto de imagens digitais vem sendo gerado, manipulado e armazenado em bancos de imagens. Esses acervos são utilizados em várias aplicações, tais como sensoriamento remoto, medicina e bibliotecas digitais [12, 35, 40]. Dado o tamanho desses bancos, prover meios de recuperar imagens de tais acervos de forma eficiente e eficaz é essencial.

Os primeiros sistemas de recuperação se baseavam na anotação manual de imagens. Nesses sistemas, o processo de recuperação consistia em comparar os termos de uma consulta textual, definida por um usuário, com as anotações associadas às imagens e, a partir dessa comparação, retornar um conjunto de imagens. Essa abordagem apresenta dois problemas principais: o primeiro diz respeito ao grande esforço despendido na anotação manual das imagens; o segundo, às possíveis inconsistências derivadas da subjetividade das anotações, pois pessoas diferentes poderiam associar anotações distintas a uma mesma imagem.

Sistemas de *Recuperação de Imagens por Conteúdo* (*Content-based Image Retrieval - CBIR*) [27, 44, 47] visam contornar esses problemas. Nesses sistemas, a anotação manual não é necessária. O processo de busca consiste basicamente em, dado um padrão de consulta (por exemplo uma imagem), calcular a sua similaridade em relação às imagens armazenadas na base, e exibir as mais similares.

A tarefa de comparar duas imagens é realizada por meio de descritores [2, 52, 53, 60]. Um descritor pode ser caracterizado por: (i) um *algoritmo de extração de características*, baseado em técnicas de processamento de imagens, que codifica as propriedades da imagem em um *vetor de características*; e (ii) uma *medida de similaridade* (função de distância) que computa a similaridade entre duas imagens como uma função de distância entre seus vetores de características correspondentes. Os vetores de características sintetizam determinadas propriedades da imagem, como cor, textura e forma, por meio da análise de seu *conteúdo*, ou seja, seus *pixels*. Esses vetores representam pontos no *espaço de*

características.

Diferentes descritores codificam propriedades distintas de uma imagem. Porém, na maioria das vezes, deseja-se recuperar uma imagem em função de múltiplas propriedades. Assim, descritores são combinados com o intuito de suprir tais necessidades. Grande parte dos métodos que realizam essa combinação são baseados na atribuição de pesos pré-determinados aos descritores [56]. Esses pesos definem a relevância de cada descritor na composição.

Porém, em geral, nesse tipo de abordagem não há nenhuma interação humano - computador. Esse fato dificulta a aquisição de imagens relevantes quando se consideram vários usuários. Isso acontece porque diferentes usuários têm percepções visuais distintas de um mesmo objeto. Além disso, há uma lacuna semântica (*semantic gap*) entre as propriedades visuais de alto nível, percebidas pelo usuário, e a descrição de baixo nível utilizada para a representação das imagens. Nesse cenário, verifica-se a necessidade de aprimorar os sistemas de recuperação de imagens para que se adaptem a diferentes usuários. Para suprir essas necessidades pode-se utilizar uma técnica de recuperação de imagem interativa chamada *realimentação de relevância* ou *retroalimentação de relevância (relevance feedback)* [48, 61]. Nessa técnica, o usuário interage com o computador e assim refina o resultado de uma determinada consulta de acordo com sua necessidade.

Realimentação de relevância é uma técnica inicialmente utilizada na recuperação de informações por texto [32, 58], mas que atualmente também é alvo de pesquisa na área de recuperação de imagem por conteúdo [9,21,37,41,54]. O processo de recuperação por meio dessa técnica consiste basicamente na: (i) realização de uma consulta em que é retornado um pequeno número de imagens; (ii) indicação das imagens relevantes e/ou irrelevantes pelo usuário; (iii) aprendizado, considerando a indicação do usuário, e seleção de novas imagens para exibição. Esse processo é repetido até que um resultado satisfatório seja obtido.

Vários métodos de recuperação de imagens com realimentação de relevância baseiam o aprendizado das necessidades de um usuário na atribuição de pesos para os descritores empregados [21, 46, 48]. Porém, essa estratégia permite apenas uma combinação linear dos valores de similaridade. É possível que uma combinação mais complexa desses valores seja necessária para expressar a percepção visual dos usuários.

Outros métodos baseiam-se fortemente na utilização dos vetores de características das imagens ou fixam uma métrica de distância específica, sem utilizar as funções de distância definidas para os descritores [34, 46, 54].

A eficácia de um descritor, no entanto, não depende apenas do poder de discriminação do vetor de característica extraído, mas também da função de distância definida. A Figura 1.1 ilustra o impacto da utilização de uma função de distância inapropriada para dois descritores. Essa figura apresenta curvas *precisão vs. revocação* calculadas na base

MPEG7 parte B [39] para quatro diferentes configurações de descritores. A curva *precisão vs. revocação* exibe a relação entre a razão do número de imagens relevantes pelo número de imagens retornadas (*precisão*) e a porcentagem de imagens relevantes encontradas (*revocação*), em um conjunto de imagens ordenado pela similaridade em relação à imagem de consulta (para mais detalhes, veja Capítulo 4). É desejável que uma curva apresente um valor de *precisão* igual a 1 (valor máximo) para todos os valores de *revocação*.

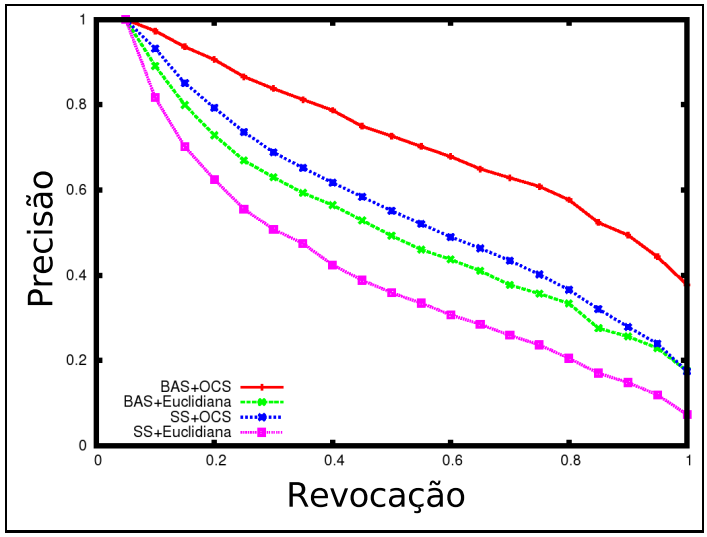


Figura 1.1: Curvas *precisão vs. revocação* na base MPEG7 parte B para os descritores BAS e SS usando duas funções de similaridade diferentes.

Em duas das curvas apresentadas na Figura 1.1, o algoritmo de extração definido pelo descritor de forma *Beam Angle Statistics (BAS)* [3] foi utilizado para extrair os vetores de características das imagens da base. Para as outras duas, foi utilizado o algoritmo de extração do descritor de forma *Saliências de Segmento (SS)* [16]. Para cada configuração baseada no BAS, duas funções de similaridade distintas fora utilizadas. A função de similaridade *OCS* [57], definida para o descritor, foi empregada em uma delas (curva BAS+OCS) e a distância Euclidiana foi utilizada na outra (BAS+Euclidiana). Similarmente, em uma das configurações baseadas no SS, foi utilizada a função de similaridade definida para o descritor, *OCS* (SS+OCS) e, novamente, a distância Euclidiana foi utilizada na outra (SS+Euclidiana). Como pode ser visto na Figura 1.1, as configurações que utilizaram as funções de similaridades definidas para os descritores apresentaram os melhores resultados.

O objetivo desta dissertação é desenvolver dois novos arcabouços para recuperação de imagens por conteúdo com realimentação de relevância. Os métodos propostos adotam a

técnica de Programação Genética para aprender as preferências do usuário na realização de uma consulta. Programação Genética (PG) [38] é uma técnica de Aprendizado de Máquinas com aplicações em várias áreas, como mineração de dados, recuperação de informação, e regressão [6, 25, 59], dentre outras. Essa técnica se baseia na Teoria da Evolução para realizar uma exploração direcionada no espaço de busca, a fim de otimizar a solução de um problema.

O objetivo dos arcabouços propostos é encontrar uma função que combina os valores de similaridades definidos por diferentes descritores. A cada iteração, o algoritmo de aprendizado baseado em PG encontra a função de combinação que melhor representa a percepção visual do usuário. Dessa forma, os valores de similaridades definidos pelos descritores são preservados e utilizados para ordenar as imagens da base. Além disso, os métodos propostos adotam uma estratégia baseada no uso de várias imagens no padrão de consulta com intuito de melhorar a exploração do espaço de características.

A utilização de programação genética é motivada pela sua capacidade exploratória do espaço de busca que se adequa ao objetivo principal do método proposto: encontrar, dentre todas as possíveis funções de combinação, aquela que melhor representa a percepção visual subjetiva de cada usuário. Além disso, a utilização dessa técnica tem obtido sucesso em outra aplicações, como recuperação de informação [19, 25] e também tem alcançado bons resultados na recuperação de imagens por conteúdo [14, 15].

Oa arcabouçoa desenvolvidos foram validados por meio de uma série de experimentos. Nesses experimentos foram empregadas três diferentes bases de imagens e descritores de cor, forma e textura foram utilizados no processo de descrição das imagens. Além disso, os arcabouços propostos foram comparados com três outros métodos de recuperação de imagens por conteúdo com realimentação de relevância [46, 48, 54], considerando-se a eficiência e a efetividade no processo de recuperação.

As principais contribuições desta dissertação são:

- Estudo sobre diferentes técnicas de realimentação de relevância (Capítulo 2);
- Proposta de dois arcabouços para recuperação de imagens por conteúdo com realimentação de relevância baseado em programação genética (Capítulo 3);
- Implementação dos métodos propostos, validando-os através de uma série de experimentos e comparações com outros métodos (Capítulos 4 e 5).

A dissertação está organizada da seguinte forma: o Capítulo 2 apresenta os conceitos básicos e trabalhos relacionados. O Capítulo 3 apresenta os arcabouços propostos. Os experimentos realizados e os resultados obtidos são apresentados nos Capítulos 4 e 5, respectivamente. Finalmente, o Capítulo 6 apresenta as conclusões e possíveis trabalhos futuros motivados pela pesquisa descrita nesta dissertação.

Capítulo 2

Conceitos básicos

Este capítulo apresenta os conceitos básicos utilizados no decorrer desta dissertação. A Seção 2.1 apresenta o modelo de recuperação de imagens por conteúdo adotado neste trabalho. A Seção 2.2 discute métodos de realimentação de relevância aplicados à recuperação de imagens. A Seção 2.3 apresenta conceitos básicos sobre programação genética.

2.1 Recuperação de Imagens por Conteúdo

Um sistema de recuperação de imagens por conteúdo (*Content-based Image Retrieval System – CBIRS*) é centrado na noção de similaridade de imagens — dado um banco com um grande número de imagens, o usuário deseja recuperar as imagens mais similares a um padrão de consulta (normalmente uma imagem definida como exemplo). O processo de recuperação é baseado na extração de características visuais (cor, textura e forma) e na comparação das imagens por meio de descritores [1, 27, 28, 43, 44].

A Seção 2.1.1 apresenta a arquitetura típica de um sistema de recuperação de imagens por conteúdo. A formalização do modelo de recuperação de imagens por conteúdo utilizado neste trabalho será apresentada na Seção 2.1.2.

2.1.1 Arquitetura Típica de um CBIRS

A Figura 2.1 mostra a arquitetura típica de um sistema de recuperação de imagens por conteúdo [17]. Essa arquitetura possui duas funcionalidades principais: a inserção de dados e o processamento de consultas.

O subsistema de inserção de dados, representados por módulos e setas tracejados, é responsável por extrair os vetores de características das imagens e armazená-los na base de imagens. Geralmente, esse processo é realizado uma única vez para cada imagem e para

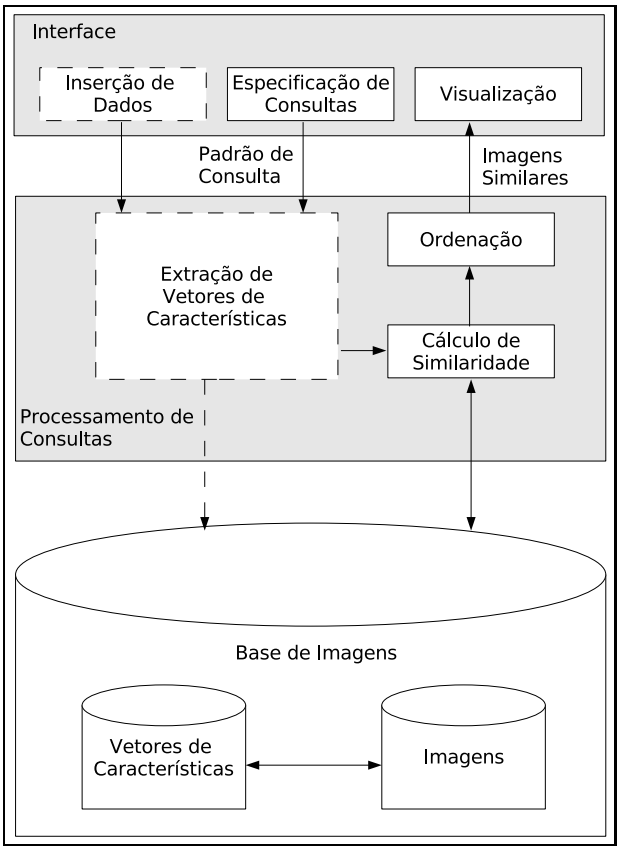


Figura 2.1: Arquitetura típica de um sistema de recuperação de imagens por conteúdo [17].

cada descritor, sendo utilizado de maneira *offline*. Os vetores armazenados são usados posteriormente no processamento de consultas.

O processamento de consultas é organizado da seguinte forma: a interface permite ao usuário especificar uma consulta por meio de um padrão de consulta (por exemplo, uma imagem – *query by visual example* [33]) e visualizar as imagens recuperadas. O módulo de processamento de consultas extrai o vetor de características do padrão de consulta e aplica uma métrica de distância, como a distância Euclidiana, para avaliar a similaridade entre a imagem de consulta e as imagens da base. Em seguida, esse módulo ordena as imagens da base de acordo com a similaridade e retorna as mais similares para o módulo de interface. Esse processo pode ser otimizado pela utilização de estruturas de indexação, como a *M-Tree* [8] e a *Slim-Tree* [55].

2.1.2 Formalização

A seguir é apresentada a formalização do modelo de recuperação de imagens por conteúdo [14, 15] adotado nesta dissertação.

Definição 1 *Uma* **imagem** $\hat{I}$ *é definida como um par* $(D_I, \vec{I})$*, onde:*

- D_I *é um conjunto finito de* pixels *(pontos em* $\mathbb{Z}^2$*, tal que,* $D_I \subset \mathbb{Z}^2$*), e*
- $\vec{I} : D_I \rightarrow \mathsf{D}'$ *é uma função que atribui a cada* pixel p *em* D_I *um vetor* $\vec{I}(p)$ *de valores em algum espaço arbitrário* D' *(por exemplo,* $\mathsf{D}' = \mathbb{R}^3$ *quando uma cor é atribuída a um* pixel *no sistema RGB).*

Definição 2 *Um* ***descritor simples*** *(ou simplesmente,* ***descritor****)* D *é definido como um par* (ϵ_D, δ_D)*, onde:*

- $\epsilon_D : \hat{I} \rightarrow \mathbb{R}^n$ *é uma função que extrai um* vetor de características $\vec{v}_{\hat{I}}$ *de uma* imagem $\hat{I}$*.*
- $\delta_D : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$ *é uma* função de similaridade *(por exemplo, baseada em uma medida de distância) que computa a similaridade entre duas imagens como o inverso da distância entre seus* vetores de características *correspondentes.*

Definição 3 *Um* ***vetor de características*** $\vec{v}_{\hat{I}}$ *de uma imagem* $\hat{I}$ *é definido como um ponto no espaço* $\mathbb{R}^n$*:* $\vec{v}_{\hat{I}} = (v_1, v_2, ..., v_n)$*, onde* n *é a dimensão do vetor.*

Exemplos de vetores de características possíveis são um histograma de cor [52], uma curva fractal multi-escala [13] e os de coeficientes de Fourier [45]. Basicamente, esses descritores codificam propriedades das imagens como cor, forma e textura. Note que diferentes tipos de vetores de características podem necessitar de funções de similaridade distintas. Cada dimensão do vetor de característica é chamada *bin*.

A Figura 2.2 ilustra o uso de um descritor simples D para computar a similaridade entre duas imagens $\hat{I}_A$ e $\hat{I}_B$. Inicialmente, o algoritmo de extração ϵ_D é usado para computar os vetores de características $\vec{v}_{\hat{I}_A}$ e $\vec{v}_{\hat{I}_B}$ associados às imagens. Em seguida, a função de similaridade δ_D é utilizada para calcular o valor da similaridade d entre as imagens.

Definição 4 *Um* ***descritor composto*** $\hat{D}$ *é definido como um par* $(\mathcal{D}, \delta_{\mathcal{D}})$ *(veja Figura 2.3), onde:*

- $\mathcal{D} = \{D_1, D_2, \ldots, D_k\}$ *é um conjunto de* k *descritores simples pré-definidos.*
- $\delta_{\mathcal{D}}$ *é um função de similaridade que combina os valores de similaridade obtidos de cada descritor* $D_i \in \mathcal{D}$*,* $i = 1, 2, \ldots, k$*.*

A Figura 2.3 mostra a utilização de um descritor composto $\hat{D}$ para computar a similaridade entre duas imagens $\hat{I}_A$ e $\hat{I}_B$. Inicialmente, os descritores simples $D_1, D_2, \ldots, D_k$ são utilizados para calcular as similaridades $d_1, d_2, \ldots, d_k$ entre as imagens. Em seguida, essas similaridades são combinadas pela função $\delta_{\mathcal{D}}$, determinando a similaridade final d entre as imagens $\hat{I}_A$ e $\hat{I}_B$.

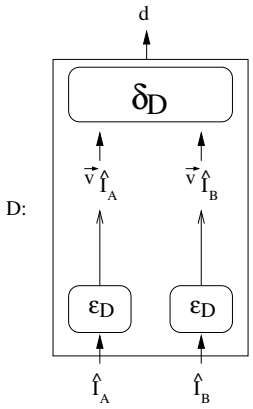


Figura 2.2: O uso de um descritor simples D para computar a similaridade entre duas imagens.

2.2 Realimentação de Relevância

A técnica de *realimentação de relevância* [10, 22, 29, 34, 41, 48, 54, 61] tem sido utilizada com bastante êxito na interação humano-computador. Esse mecanismo tem por objetivo possibilitar que o usuário expresse a sua necessidade na especificação de uma consulta, sem recorrer a ajustes de propriedades de baixo nível utilizadas na representação de imagens. Para isso, o usuário apenas precisa indicar as imagens relevantes e, em certos casos, também as irrelevantes dentre um conjunto retornado pelo sistema. A cada iteração, o algoritmo de realimentação de relevância "aprende" quais propriedades visuais melhor definem as imagens relevantes a partir das informações fornecidas pelo usuário, ou seja, das imagens por ele indicadas. E assim, após um determinado número de iterações, o sistema retorna as imagens mais similares à imagem de consulta.

Realimentação de relevância endereça duas questões referentes ao processo de recuperação de imagens por conteúdo. A primeira delas reside na lacuna semântica (*semantic gap*) entre as propriedades visuais de alto nível, através das quais o usuário tem a percepção da informação visual, e a descrição de baixo nível utilizada para a representação das imagens. A outra diz respeito ao caráter subjetivo da percepção da imagem pelo usuário. Diferentes pessoas, ou a mesma em diferentes circunstâncias, podem ter percepções visuais distintas de uma mesma imagem. Com realimentação de relevância essas duas questões são contornadas de forma transparente para o usuário.

As próximas seções caracterizam a abordagem de realimentação de relevância para a recuperação de imagens por conteúdo. A Seção 2.2.1 apresenta a arquitetura típica de um sistema de recuperação de imagens com realimentação de relevância. Na Seção 2.2.2 são

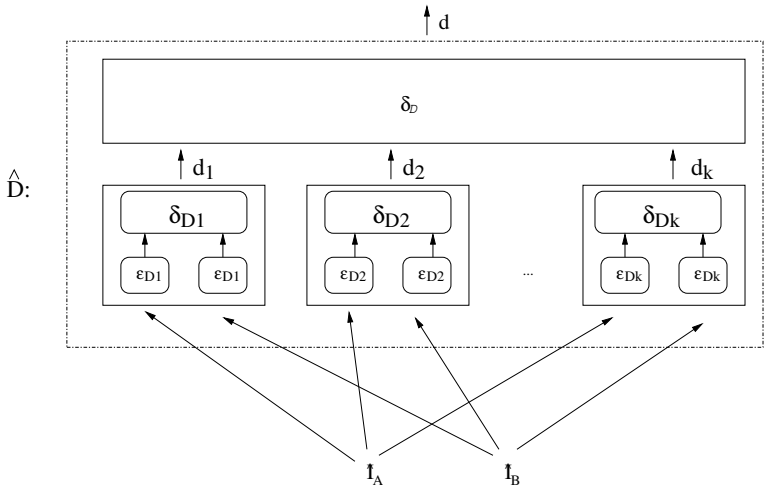


Figura 2.3: Uso de um descritor composto $\hat{D}$ para computar a similaridade entre duas imagens.

descritas características da técnica de realimentação de relevância aplicada na recuperação de imagens por conteúdo. As Seções 2.2.3, 2.2.4 e 2.2.5, abordam algumas questões relativas à modelagem de mecanismos de realimentação de relevância, aos critérios de seleção de imagens e às técnicas de aprendizado usualmente utilizados, respectivamente. Por fim, a Seção 2.2.6 apresenta uma comparação entre os trabalhos apresentados.

2.2.1 Arquitetura Típica de um CBIRS com Realimentação de Relevância

A Figura 2.4 apresenta uma extensão à arquitetura apresentada na Seção 2.1 (Figura 2.1) incorporando realimentação de relevância a um sistema de recuperação de imagens por conteúdo. Essa extensão é composta por dois módulo principais: a *interface* e o *módulo de processamento de consultas*.

Assim como na arquitetura anteriormente vista, a interface deve possibilitar a especificação do padrão de consulta e a visualização dos resultados. Porém, uma nova funcionalidade foi adicionada: permitir a indicação das imagens relevantes e irrelevantes.

Um novo módulo foi acrescido ao processamento de consultas: o módulo de realimentação de relevância. Esse novo módulo é responsável por receber as imagens indicadas pelo usuário como relevantes e irrelevantes, reformular a consulta com base nessa informação e selecionar as imagens que serão retornadas. Essas tarefas são divididas por dois outros sub-módulos: o sub-módulo de *Aprendizado* e o sub-módulo de *Seleção*, como

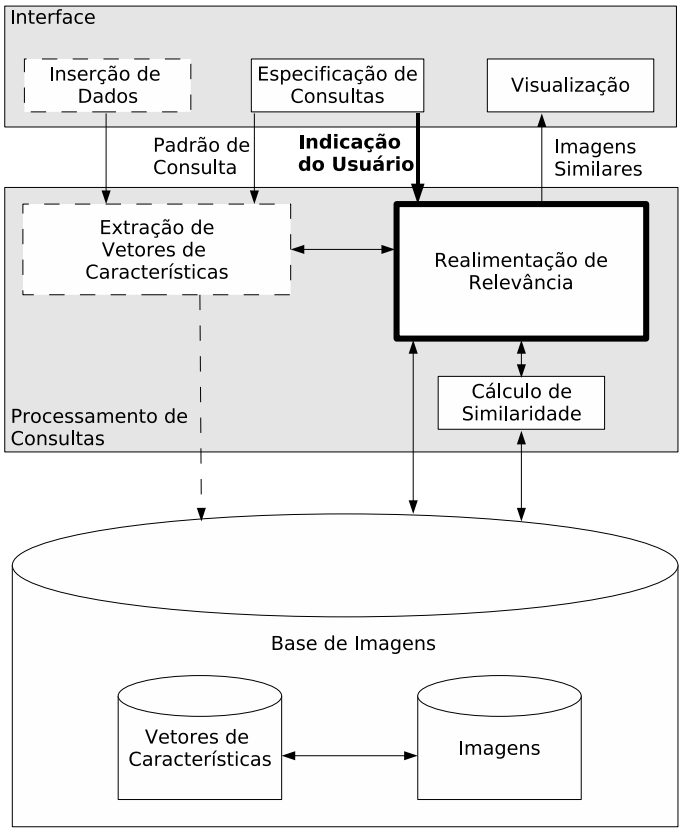


Figura 2.4: Arquitetura de um sistema de recuperação de imagens por conteúdo com realimentação de relevância.

mostrado na Figura 2.5.

O sub-módulo de *Aprendizado* é responsável por assimilar a percepção visual do usuário, por meio das imagens indicadas como relevantes e irrelevantes. Exemplos de técnicas usualmente utilizadas para aprendizado são apresentadas na Seção 2.2.5. O sub-módulo de *Seleção* determina quais imagens serão exibidas para o usuário em cada iteração. A Seção 2.2.4 discute critérios de seleção das imagens.

2.2.2 Características Essenciais

Recuperação de imagens com realimentação de relevância possui certas características essenciais que precisam ser observadas [61]: o tamanho restrito do conjunto de treinamento, a assimetria do conjunto de treinamento e a necessidade de um baixo tempo de execução de uma consulta.

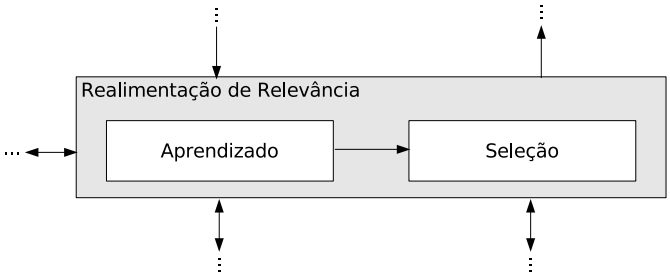


Figura 2.5: Módulo de realimentação de relevância.

Para tornar o processo de recuperação viável, é necessário que o número de imagens retornadas para o usuário a cada iteração seja pequeno. Isso porque um número muito grande poderia dificultar a interação do usuário com o sistema. Além do conjunto de imagens retornadas ser pequeno, o número de imagens marcadas pelo usuário tende a ser ainda menor. Dessa forma, o algoritmo de aprendizado terá um pequeno número de imagens para treinamento.

Uma outra característica diz respeito à assimetria do conjunto de treinamento. O conjunto de treinamento nem sempre reflete a distribuição de imagens no banco. Para uma determinada consulta, o número de imagens irrelevantes, dentre todas da base, tende a ser extraordinariamente maior do que o número de relevantes. Porém, essa distribuição normalmente não se reflete no grupo de imagens marcadas pelo usuário, ou seja, o pequeno número de imagens marcadas como irrelevantes não é representativo quando se considera a distribuição real na base.

Uma terceira característica se refere ao tempo de execução. A execução do algoritmo deve ser rápida para viabilizar a interação humano-computador em tempo real. Também com esse intuito, um resultado satisfatório deve ser atingido após um número pequeno de iterações.

2.2.3 Modelagem

Certas características devem ser consideradas quando da definição de um mecanismo de realimentação de relevância: (a) o tipo de consulta adotado; (b) a informação provida pelo usuário; e (c) como essa informação é utilizada no processo de aprendizado.

Grande parte dos sistemas assumem que o objetivo de uma consulta é obter um conjunto de itens relevantes (*category search*). Essa relevância pode ser definida como propriedades compartilhadas por um determinado grupo de imagens, o que também caracteriza uma classe de itens similares. Seguindo essa linha, para uma consulta, pode ser necessário

apenas separar as imagens relevantes das demais, o que caracteriza um problema de classificação (*classification problem*) [29, 34, 54]. Por outro lado, o objetivo de uma consulta pode ser encarado como um problema de ordenação (*ranking problem*) [41, 48]. Nesse caso, o sistema deve retornar para o usuário as imagens ordenadas em ordem decrescente em relação ao seu grau de relevância. Outros trabalhos assumem que o resultado de uma consulta é uma única imagem (*target search*) [22], ao invés de um conjunto de itens similares. Com isso, imagens marcadas como relevantes não são necessariamente imagens que satisfazem a necessidade do usuário, e sim, imagens "próximas" àquela procurada.

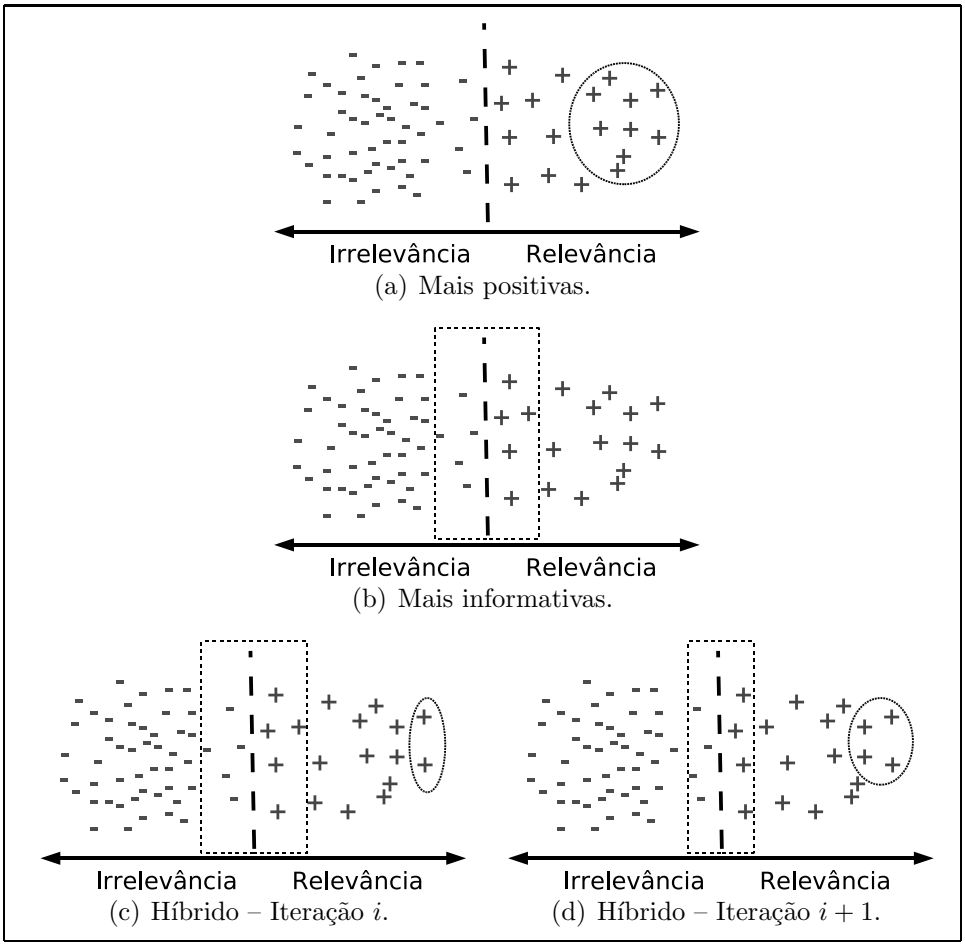
Uma outra característica se refere à informação provida pelo usuário. Alguns trabalhos permitem a indicação apenas das imagens relevantes [29]. Outros, consideram além dos exemplos positivos, os negativos [34, 41, 54]. Com isso, é possível extrair mais informação a respeito das características visuais importantes para a consulta. Há ainda algoritmos que definem graus de relevância ou irrelevância para as imagens marcadas [48].

Uma terceira característica diz respeito à forma como o mecanismo de realimentação de relevância irá lidar com a informação provida pelo usuário. Em geral, duas abordagens são utilizadas. O aprendizado *short-term* considera apenas a informação *intra-consulta*, ou seja, aquela obtida na sessão de consulta corrente, para assimilar a percepção visual do usuário. Já o aprendizado *long-term* utiliza não apenas a informação intra-consulta, mas também a informação acumulada durante todas as consultas já realizadas, o que caracteriza a informação *inter-consulta*.

2.2.4 Seleção das Imagens

Outro ponto importante diz respeito a quais imagens devem ser retornadas para o usuário a cada iteração com o sistema. Duas estratégias são comumente utilizadas como critério de seleção: as *imagens mais positivas (MP)* [41,48] e as *imagens mais informativas (MI)* [54]. Há ainda uma terceira possibilidade que é a combinação desses dois critérios.

Na primeira, o algoritmo seleciona as imagens mais positivas (MP), de acordo com as informações obtidas até a presente iteração. Nesse tipo de seleção, é possível observar o estado corrente de conhecimento do sistema a cada iteração. Além disso, essa estratégia tem a vantagem de mostrar resultados parciais mais satisfatórios logo nas primeiras iterações. Com isso, o usuário pode terminar o processo de consulta a qualquer momento, pois os melhores resultados sempre são os exibidos. Essa abordagem tem a desvantagem de, em geral, necessitar de um número maior de iterações para alcançar uma identificação completa dos itens semelhantes à imagem de consulta. A Figura 2.6(a) ilustra esse critério de seleção. Nessa figura, as imagens definidas como relevantes pelo algoritmo de aprendizado são representadas pelo símbolo + e as irrelevantes por −. Quanto mais à direita, mais positivas as imagens. A elipse pontilhada indica as imagens selecionadas.



(a) Mais positivas.
(b) Mais informativas.
(c) Híbrido – Iteração i.
(d) Híbrido – Iteração $i + 1$.

Figura 2.6: Critério de seleção de imagens.

Na segunda abordagem, as imagens mais informativas (MI) são retornadas, ou seja, as imagens a partir das quais o algoritmo de aprendizado pode extrair maiores informações a cada iteração. As imagens selecionadas devem potencializar a diferenciação entre a classe das imagens relevantes e a classe das irrelevantes, o que maximiza a transferência de informação do usuário para o sistema. Normalmente, as imagens mais informativas são as mais ambíguas, ou seja, aquelas que compartilham características comuns tanto com as imagens relevantes quanto com as irrelevantes, e por conseqüência, são as mais difíceis de serem classificadas. Por possibilitar a extração de mais informações, se comparada com MP, essa abordagem tende a definir melhor a classe das imagens relevantes após um número menor de iterações. A Figura 2.6(b) ilustra o critério de seleção MI. Nessa figura, a linha tracejada determina a fronteira entre as imagens relevantes e irrelevantes. As imagens contidas no retângulo tracejado são as mais próximas dessa fronteira, sendo

consideradas as mais informativas, e assim, as selecionadas para exibição.

Com o intuito de combinar as vantagens oferecidas por MP e MI, é possível ainda sugerir uma estratégia *híbrida*. Nessa estratégia, considerando n imagens retornadas por iteração, k dessas imagens são selecionadas segundo MP e as $n - k$ imagens restantes são selecionadas de acordo com MI. A cada nova iteração, o valor de k aumenta. Com isso, um número maior de imagens ambíguas são retornadas nas primeiras rodadas, o que contribui para uma identificação mais rápida da classe das imagens relevantes. A medida que o processo avança, o número de imagens mais positivas retornadas aumenta. Em [58] essa estratégia foi utilizada na recuperação de textos e apresentou um desempenho superior aos das técnicas MP e MI. Um ponto importante a ser observado nessa abordagem é a escolha dos parâmetros que determinam a variação de k no decorrer do processo. As Figuras 2.6(c) e 2.6(d) ilustram a seleção de imagens utilizando a abordagem híbrida. O retângulo tracejado indica a seleção das imagens mais informativas e a elipse pontilhada mostra as imagens mais positivas selecionadas. A Figura 2.6(c) exibe as imagens selecionadas na iteração i. Na próxima iteração, $i + 1$, o número de imagens informativas selecionadas diminui, e o número de positivas aumenta, como mostrado na Figura 2.6(d).

2.2.5 Aprendizado

O algoritmo de aprendizado é um ponto crucial para a definição de um mecanismo de realimentação de relevância. Em alguns trabalhos, o aprendizado consiste em estimar o vetor de característica que melhor representa o padrão de consulta. Essa estratégia é denominada *Movimento do Ponto de Consulta (MPC)* [37, 41]. Em outros, atribuem-se pesos para cada posição do vetor de características e para cada descritor utilizado [18, 46, 48]. Assim, o aprendizado consiste em estimar esses pesos, de forma a melhor representar a percepção visual do usuário. Outros métodos utilizam uma técnica de aprendizado chamada *Máquinas de Vetores de Suporte (Support Vector Machines - SVM)* para encontrar hiperplanos de separação que identificam as imagens relevantes e as irrelevantes [29,34,54]. Há ainda métodos que utilizam um arcabouço probabilístico para definir a probabilidade de cada imagem da base ser relevante [10, 22]. Alguns métodos utilizam uma combinação entre essas estratégias para aprender a percepção visual do usuário.

A seguir, são apresentados métodos que utilizam as abordagens de aprendizado citadas. A seção 2.2.5.1 descreve trabalhos que baseiam o aprendizado na estimativa de pesos. A seção 2.2.5.2 aborda o aprendizado baseado na estimativa do vetor de características que melhor reflete a necessidade do usuário. A seção 2.2.5.4 apresenta trabalhos baseados na separação entre imagens relevantes e irrelevantes por meio de um hiperplano. Finalmente, a seção 2.2.5.3 ilustra trabalhos baseados em aprendizado probabilístico.

2.2.5.1 Atribuição de Pesos

Em vários métodos, o aprendizado baseia-se na estimativa de pesos associados a descritores e aos *bins* do vetor de características. Basicamente, esses trabalhos buscam ressaltar de forma automática as características visuais almejadas pelo usuário na consulta. A Figura 2.7 ilustra a utilização desses pesos em conformidade com modelo apresentado na Seção 2.1.

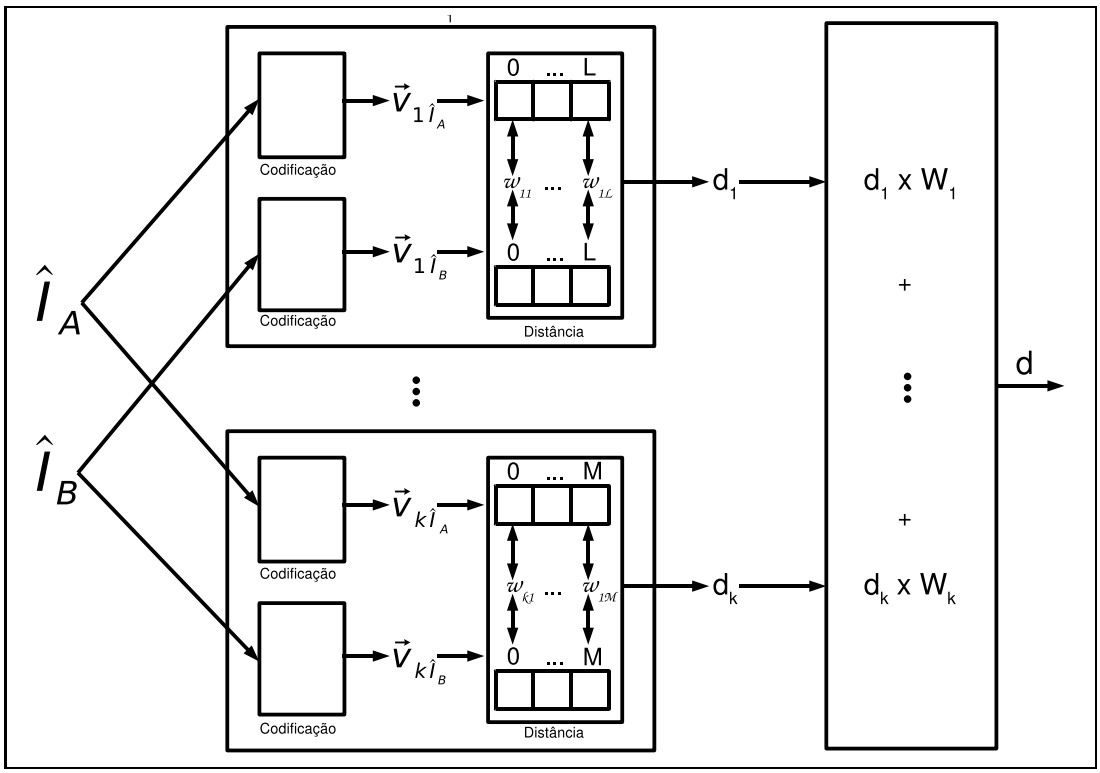


Figura 2.7: Modelo para atribuição de pesos.

Essa figura mostra que os pesos w_{ij} associados aos *bins* dos vetores de características são utilizados no cálculo da distância entre duas imagens e os pesos W_i são utilizados na combinação das distâncias definidas por cada descritor.

Um exemplo de função de distância com pesos é a *Weighted Euclidean Distance* [46], definida como

$$d_i = \sum_{j=0}^{L} w_{ij} \times (\vec{v}_{i\hat{I}_A}[j] - \vec{v}_{i\hat{I}_B}[j])^2, \tag{2.1}$$

onde $\vec{v}_{i\hat{I}_A}$ e $\vec{v}_{i\hat{I}_B}$ são vetores de características, de tamanho L, das imagens $\hat{I}_A$ e $\hat{I}_B$ codificados pelo descritor D_i, e $w_{i1}, w_{i2}, \ldots, w_{iL}$ são pesos associados aos *bins* desses vetores.

Em geral, os pesos atribuídos aos descritores são utilizados para combiná-los linearmente, e assim, definir a distância final entre duas imagens segundo esses descritores, como mostrado na Figura 2.7. Essa combinação se dá pela seguinte equação

$$d = \sum_{i=0}^{k} W_i \times d_i, \tag{2.2}$$

onde $d_1, d_2, \ldots, d_k$ são as distâncias definidas pelos descritores $D_1, D_2, \ldots, D_k$, respectivamente, e $W_1, W_2, \ldots, W_k$ são pesos associados a cada descritor. Essa atribuição de pesos promove a discriminação entre os espaços de características, conferindo maior destaque aquele que mais se aproxima da percepção de similaridade do usuário.

Um dos primeiros trabalhos que utiliza realimentação de relevância para recuperação de imagens por conteúdo foi proposto em [48]. Nesse trabalho, atribuem-se pesos a cada descritor (*interweight*), e também a cada *bin* do vetor de características (*intraweight*). Cabe ao usuário marcar cada uma das imagens retornadas em cada iteração como *muito relevante, relevante, sem opinião, irrelevante* e *muito irrelevante*. Feito isso, o algoritmo de realimentação de relevância heuristicamente atualiza os pesos.

Para recalcular os pesos *interweights*, cada descritor D é utilizado em uma consulta distinta. Para cada imagem I retornada, se I foi marcada pelo usuário como (muito) relevante, o peso de D é incrementado. Por outro lado, se I foi marcada como (muito) irrelevante, o peso de D é decrementado.

A atualização dos pesos *intraweights* funciona da seguinte forma: para cada descritor utilizado, organizam-se os vetores de características das imagens marcadas como (muito) relevantes em uma matriz. Dessa forma, cada *bin* j encontra-se representado em uma coluna c_j dessa matriz. Assim, o peso de um *bin* j é atualizado com o inverso do desvio-padrão dos elementos da coluna c_j.

Li et al. [41] também utilizam o inverso do desvio-padrão para atualizar os pesos *intraweight*. Já o peso W_i de cada descritor i é atualizado pela equação

$$W_i = 1 - \frac{\bar{d}_i - d_{min}}{d_{max} - d_{min}}, \tag{2.3}$$

onde, $\bar{d}_i$, d_{max} e d_{min} são a média, a maior e a menor das distâncias entre o padrão de consulta e as imagens marcadas como relevantes, segundo o descritor i, respectivamente.

Em [46], a atualização de pesos é novamente empregada para prover aprendizado. Porém, esse trabalho utiliza um arcabouço de otimização para definir os pesos. Esse arcabouço é baseado na minimização da distância Euclidiana generalizada (*Generalized*

Euclidean Distance) entre a imagem de consulta e as imagens definidas pelo usuário como relevantes a cada iteração.

Em [21], a estratégia *Query-Sample Scaling (QSS)* é apresentada. Esse método também se baseia na atribuição de pesos para prover aprendizado. Porém, apenas os pesos atribuídos às posições dos vetores de características são utilizados. Esses pesos são estimados por meio da maximização da correlação cruzada normalizada entre o padrão de consulta e as imagens relevantes.

Silva et al. [18] propõem um método de recuperação de imagens baseado em *Similaridade Local de Padrões (Local Similarity Patterns - LSP)*. O modelo LSP utiliza um processo de descrição local, particionando uniformemente cada imagem em regiões. Dessa forma, ao invés de se extrair os vetores de características considerando a imagem inteira, há um vetor de características associado a cada região, para cada descritor empregado. As similaridades individuais entre as regiões de duas imagens definem seu grau de similaridade. Nesse método, atribuem-se pesos para cada descritor utilizado em cada região e um peso para cada região da imagem. Esses pesos são estimados por meio de uma técnica baseada em algoritmos genéticos.

Cura [20] propõe um modelo para recuperação por conteúdo de imagens de sensoriamento remoto com realimentação de relevância. Esse modelo é baseado em consulta por padrões, que por sua vez, são compostos por regiões de uma ou várias imagens. Admite-se ainda a utilização de várias formas de representação (vetores de características definidos por descritores distintos) para cada padrão. Nesse modelo, atribuem-se pesos para cada representação de cada padrão. A cada iteração, os novos pesos são calculados a partir da correção dos pesos anteriores, considerando todo o conhecimento acumulado em iterações prévias.

2.2.5.2 Movimento do Ponto de Consulta

A estratégia de Movimento do Ponto de Consulta [37, 41, 46] se divide em dois tipos: o *Movimento do Ponto de Consulta Único (MPCU)* [41, 46] e o *Movimento dos Múltiplos Pontos de Consulta (MMPC)* [37].

O *Movimento do Ponto de Consulta Único* [41,46] representa o padrão de consulta por apenas um único ponto. Essa estratégia tem por objetivo mover o ponto que caracteriza o padrão de consulta no espaço de características (*feature space*). Esse movimento busca aproximar esse padrão dos pontos que representam as imagens relevantes. Com isso, pretende-se estimar o vetor de características que melhor caracteriza as propriedades visuais das imagens almejadas pelo usuário. A Figura 2.8 ilustra o MPCU. Nessa figura, busca-se aproximar o padrão de consulta Q das imagens positivas, representadas na figura por $+$.

A outra abordagem para MPC, o *Movimento dos Múltiplos Pontos de Consulta* [37],

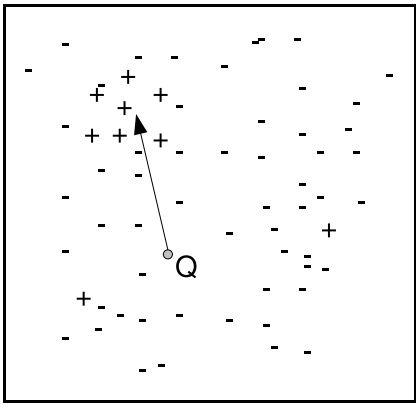


Figura 2.8: Movimento do Ponto de Consulta Único.

considera que imagens relevantes podem estar em pontos dispersos do espaço de características adotado. Assim, utiliza múltiplos pontos para a caracterização do padrão de consulta. A Figura 2.9 mostra o movimento dos múltiplos pontos de consulta Q_1, Q_2, Q_3 e Q_4 em direção aos agrupamentos de imagens relevantes.

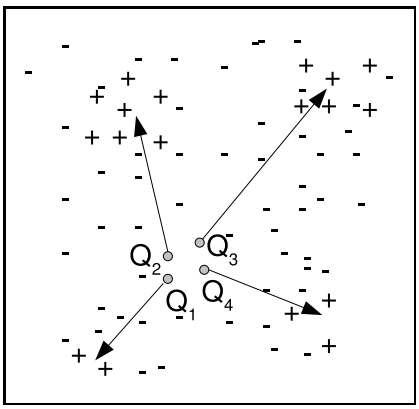


Figura 2.9: Movimento de Múltiplos Pontos de Consulta.

O mecanismo de realimentação de relevância proposto em [41] utiliza a abordagem MPCU. Nesse método, o vetor de características do padrão de consulta é atualizado a partir da diferença ponderada entre o vetor médio definido pelas imagens relevantes e o vetor médio definido pelas imagens irrelevantes. Em [46], MPCU também é utilizada, mas o padrão de consulta é representado pelo centróide definido pelas imagens relevantes apenas.

Kim et al. [37] propõem um mecanismo de realimentação de relevância baseado em múltiplos pontos de consulta. Nessa abordagem, os centróides dos agrupamentos definidos pelas imagens indicadas como relevantes representam o padrão de consulta. Em cada iteração, um classificador Bayesiano é aplicado às imagens recém-anotadas como relevantes. Assim, essas imagens ou são classificadas em relação aos agrupamentos existentes, ou dão origem a um novo. Em seguida, é aplicado um algoritmo para reduzir o número de agrupamentos. Por fim, uma função de distância de agregação é empregada para calcular a distância entre cada imagem da base e o padrão de consulta formado por múltiplos pontos, e retornar as mais relevantes.

Além da atribuição de pesos, como mencionado na seção 2.2.5.1, o modelo proposto por Cura [20] utiliza múltiplos pontos de consulta, representados pelos padrões anteriormente descritos, para caracterizar a percepção visual do usuário. Em cada iteração, as regiões indicadas como relevantes são incorporadas na definição de cada padrão. O objetivo desse mecanismo é permitir o refinamento da consulta inicial, possibilitando que a necessidade do usuário seja melhor caracterizada a cada iteração.

2.2.5.3 Aprendizado Probabilístico

O aprendizado probabilístico aplicado em realimentação de relevância é definido pela aplicação da *regra de Bayes* [10, 22]. Por meio de um *framework* Bayesiano, obtém-se a probabilidade de cada uma das imagens do banco ser a imagem almejada. Basicamente, um *framework* de classificação Bayesiano trata as imagens positivas e negativas como duas classes, Ω^+ e Ω^- respectivamente. Então, duas probabilidades são definidas *a priori* ($p(\Omega^+)$ e $p(\Omega^-)$) para as duas classes e, a partir da estratégia estatística escolhida, as funções de densidade de probabilidade são obtidas para essas duas classes. Depois, em cada iteração, a probabilidade de cada imagem do banco é calculada por meio da regra Bayesiana. Em seguida, as imagens são ordenadas em ordem decrescente em relação às suas probabilidades. Com isso, a distribuição de probabilidades é utilizada para exibir as imagens mais positivas para o usuário. A cada iteração, essa distribuição é recalculada com base na indicação do usuário do conjunto de resultados relevantes. É importante observar que diferentes estratégias estatísticas originam diferentes mecanismos de realimentação de relevância.

O sistema *PicHunter* [10] utiliza um *framework Bayesiano* para aprendizado e supõe a busca por uma única imagem. Esse mecanismo busca predizer quais imagens são relevantes utilizando-se das imagens que o usuário indica como mais próximas às suas necessidades. A regra de Bayes é utilizada para predizer qual imagem o usuário deseja obter na consulta. A cada iteração, as imagens mais positivas são selecionadas para a visualização.

Em [22], uma outra abordagem para realimentação de relevância utilizando inferência Bayesiana é proposta, a *rich get richer (RGR)*. Esse método considera a consistência entre

as indicações sucessivas do usuário para prover o aprendizado. Assim, se a indicação do usuário for consistente com a indicação da iteração anterior, as imagens marcadas como relevantes à imagem objetivo serão atualizadas com uma maior probabilidade.

2.2.5.4 Máquinas de Vetores de Suporte

A técnica de Máquinas de Vetores de Suporte [29, 49] tem sido bastante utilizada em mecanismos de realimentação de relevância. SVMs utilizam hiperplanos para classificar as imagens como relevantes ou irrelevantes. Porém, como os vetores de características das imagens podem não ser linearmente separáveis segundo essa classificação, é necessário mapear suas representações para um espaço de características de maior dimensão, *higher-dimensional feature space (HDFS)*, usando uma transformação não linear associada a um *kernel*. Esse mapeamento é realizado por uma função *kernel* de produto interno $K(\vec{v}_1, \vec{v}_2)$. Dois exemplos de *kernels* são o *kernel polinomial* $K(\vec{v}_1, \vec{v}_2) = (\vec{v}_1 \cdot \vec{v}_2 + 1)^p$ que induz fronteiras polinomiais de grau p no espaço original, e o *kernel função de base radial* (ou *Gaussiano*) $K(\vec{v}_1, \vec{v}_2) = (e^{-\gamma||\vec{v}_1 - \vec{v}_2||^2})$, com um valor fixo para o parâmetro de escala γ (o inverso da variância).

O processo de aprendizado consiste em encontrar o hiperplano de partição que melhor separa as imagens marcadas como relevantes das irrelevantes no HDFS. Assim, é possível realizar implicitamente uma classificação linear das imagens relevantes e não-relevantes nesse espaço. O hiperplano obtido durante o aprendizado é utilizado para ordenar as imagens da base. Essa ordenação é realizada em função das distâncias entre vetores de características das imagens e o hiperplano de partição no HDFS. As imagens consideradas mais relevantes são aquelas mais distantes do hiperplano, pelo lado positivo. Em geral, os métodos de realimentação de relevância baseados em SVM utilizam apenas os vetores de características das imagens nos processos de aprendizado e classificação, não envolvendo as funções de similaridade definidas pelos descritores empregados no processo de recuperação.

A Figura 2.10 mostra o resultado da aplicação de SVM para a separação de duas classes, indicadas por + e −. Nesse exemplo, foi utilizado um *kernel* polinomial de grau três. O hiperplano encontrado separa as duas classes.

No trabalho proposto em [34], SVM é utilizado para atualizar automaticamente os pesos para as imagens relevantes. A função *kernel* polinomial com $p = 1$ é utilizada para o aprendizado. *Feedbacks* positivos e negativos são utilizados para obter a informação a respeito da classe desejada pelo usuário. Com os experimentos realizados em [34] foi possível concluir que é necessário um certo número de exemplos positivos e negativos para proporcionar um bom aprendizado com uma SVM. Esse número foi escolhido heuristicamente como 4 tanto para exemplos positivos quanto para os negativos.

Em [54] é proposto um algoritmo com *support vector machine active learning* para separar as imagens relevantes das demais. A função *kernel* de base radial é utilizada para

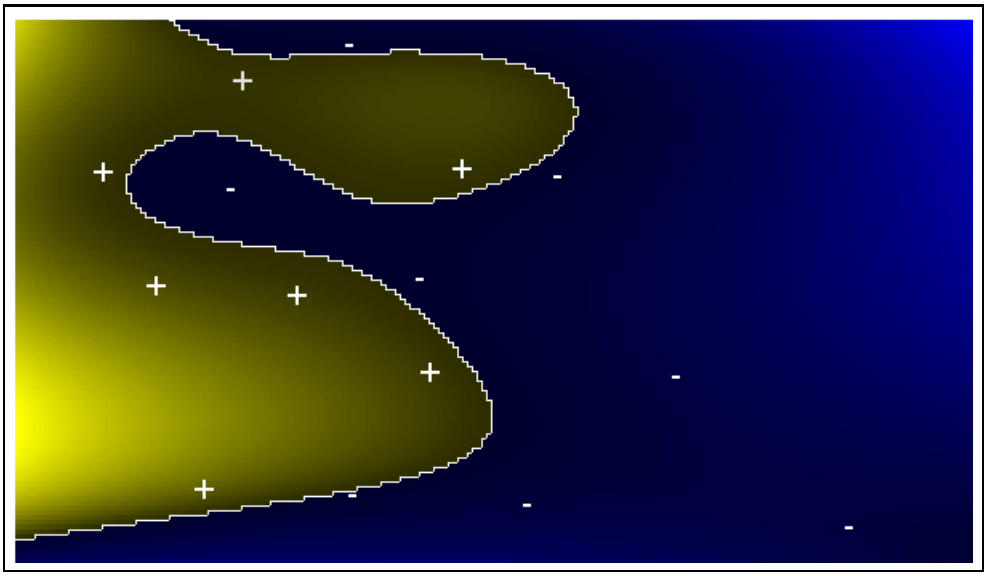


Figura 2.10: Separação de classes promovida por SVM.

o cálculo do produto interno em um espaço de maior dimensão. Na primeira iteração, as imagens exibidas para o usuário são escolhidas aleatoriamente. A partir da segunda, as imagens mais informativas são retornadas para o usuário. Essas imagens são obtidas através da distância de seus vetores de características em relação ao hiperplano de separação. As mais próximas desse hiperplano são consideradas as mais ambíguas, e portanto, as mais informativas. Ao final do processo, as imagens relevantes mais positivas são exibidas para o usuário.

Em [29], Gondra el al. propõem um método que utiliza as informações intra e inter-consulta para o aprendizado. A cada iteração, as imagens marcadas como relevantes são utilizadas para definir um classificador *One-Class SVM*. Esse classificador determina uma hiper-esfera e assim, define o conjunto das imagens relevantes. Além disso, utiliza um classificador *fuzzy* para definir o grau de pertinência da imagem de consulta em cada uma das hiper-esferas encontradas em consultas anteriores. Assim, as imagens são exibidas para o usuário a partir do conhecimento intra e inter-consulta combinados. *Reinforcement Learning* é utilizado para determinar a proporção entre a informação intra-consulta e inter-consulta a ser utilizada.

2.2.6 Comparação entre Métodos

A Tabela 2.1 resume as principais características dos trabalhos mencionados. As características exibidas são: a abordagem utilizada para o aprendizado; o critério de seleção;

o tipo de consulta adotado; o tipo de informação fornecida pelo usuário; e a forma como o mecanismo de realimentação de relevância irá lidar com a informação provida pelo usuário.

Tabela 2.1: Resumo das características dos trabalhos correlatos em realimentação de relevância.

Trabalhos	**Aprendizado**	**Seleção**	**Consulta**	***Indicação***	**Informação**
Rui et al. [48]	Pesos	MP	Ordenação	Positivo/Negativo com grau de relevância	Intra-consulta
Rui et al [46]	Pesos/MPCU	MP	Ordenação	Positivo com grau de relevância	Intra-consulta
Doulamis et al. [21]	Pesos	MP	Ordenação	Positivo	Intra-consulta
Silva et al. [18]	Pesos	MP	Ordenação	Positivo	Intra-consulta
Li et al. [41]	Pesos/MPCU	MP	Ordenação	Positivo/Negativo	Intra-consulta
Kim et al. [37]	MMPC	MP	Ordenação	Positivo	Intra-consulta
Cura [20]	Pesos/MMPC	MP	Ordenação	Positivo	Intra-consulta
Cox et al. [10]	Bayesiano	MP	*Target Search*	Positivo/Negativo	Intra-consulta
Duan et al. [22]	Bayesiano	MP	*Target Search*	Positivo	Intra-consulta
Hong et al. [34]	SVM	MP	Classificação	Positivo/Negativo	Intra-consulta
Tong et al. [54]	SVM	MI	Classificação	Positivo/Negativo	Intra-consulta
Gondra et al. [29]	SVM	MP	Classificação	Positivo	Inter-consulta

A Tabela 2.2 resume as características dos experimentos realizados nos trabalhos citados. As características presentes na tabela são: número de imagens da base e como foram obtidas; as propriedades visuais (*features*) utilizadas; o número de imagens retornadas por iteração; o número máximo de iterações realizadas; o tipo de usuário; e se o tempo de execução foi avaliado. Em algumas posições dessa tabela aparece N/D. Essa marcação significa *não definido*. Por exemplo, em [22] não fica claro quais propriedades foram codificadas.

Na Tabela 2.2 pode ser observado que vários trabalhos utilizam um subconjunto da base Corel. Em geral, é utilizado apenas um subconjunto devido a problemas de pré-classificação dessa base do ponto de vista de recuperação por conteúdo, como reportado em [42]. Essa pré-classificação tem forte cunho semântico. Por exemplo, há uma classe com imagens da *Austrália*, que vão desde prédios até animais e paisagens, ou seja, com características visuais bem distintas. Por isso, em geral é selecionado um subconjunto dessa base, quando se deseja avaliar mecanismos de recuperação de imagens por conteúdo.

Um outro fato que merece ser comentado é a utilização de usuários reais nos experimentos realizados em [10], ao contrário dos demais trabalhos que utilizaram usuários simulados por computador. Os experimentos foram realizados com seis usuários. Cada usuário repetiu o processo de recuperação 15 vezes, considerando 15 imagens-alvo distintas. Todos os usuários concluíram com sucesso a recuperação, ou seja, encontraram a

Tabela 2.2: Resumo das características dos experimentos realizados nos trabalhos apresentados.

Trabalhos	Base	Propriedades	Imagens retornadas	Num. iterações	Usuário	Tempo
Rui et al. [48]	286 MESL[1]/ 70000 Corel	Cor, forma e textura	10-1100	3	Simulado	Não
Rui et al. [46]	17000 Corel	Cor, forma e textura	N/D	3	Simulado	Não
Doulamis et al. [21]	15000	Cor e textura	50	5/10	Simulado	Não
Silva et al. [18]	60/167 Vistex 208 Brodatz 1000 Corel 12750	Cor, textura e forma	N/D	N/D	Simulado	Sim
Li et al. [41]	10000 Corel	Cor e textura	50	10	N/D	Não
Kim et al. [37]	30000 Corel	Cor e textura	100	5	Simulado	Sim
Cura [20]	150 LANDSAT	Cor e textura	6	N/D	N/D	Não
Cox et al. [10]	4522 Corel	Cor	9	35.1	Real	Não
Duan et al. [22]	800/1000/ 1500 Corel	N/D	25/100	30	Simulado	Não
Hong et al. [34]	17000 Corel	Cor e textura	20	N/D	N/D	Não
Tong et al. [54]	602/1277/ 1920 Corel	Cor e textura	20	5	Simulado	Sim
Gondra et al. [29]	20000 Letter	N/D	20	N/D	N/D	Não

[1]Museum Educational Site Licensing Project.

imagem buscada. Para isso, foram necessárias, na média, 35.1 iterações.

Nos experimentos com usuários simulados, é necessária que a base de imagens utilizada seja dividida em classes de itens similares. Dessa forma, o comportamento do usuário pode ser simulado por meio da marcação das imagens pertencentes à mesma classe da imagem de consulta como relevantes e das demais como irrelevantes.

2.3 Programação Genética

Programação genética (PG) [38] é uma técnica da Inteligência Artificial que visa a resolução de problemas baseada nos princípios biológicos de herança genética e evolução. Em PG, os indivíduos representam programas que através de recombinações e perturbações sucessivas são refinados, originando melhores soluções para um dado problema. Os melhores indivíduos tendem a se recombinar entre si. A qualidade de um indivíduo é mensurada quantitativamente por uma função de adequação (*fitness*). Dessa forma, programação genética pode ser caracterizada como uma busca no espaço de todos os possíveis programas. Essa busca é orientada para encontrar o indivíduo (programa) que melhor resolve um determinado problema. O algoritmo básico de evolução utilizado por programação genética é apresentado no Algoritmo 1.

No início do processo de evolução, uma população inicial é criada (linha 1). A seguir,

Algoritmo 1 Algoritmo básico de evolução para programação genética.

```
1 Gere a população inicial de indivíduos
2 para N gerações faça
3     Calcule a adequação de cada indivíduo
4     Selecione os indivíduos para operações genéticas
5     Aplique reprodução
6     Aplique recombinação
7     Aplique mutação
8 fim para
9 retorne os melhores indivíduos
```

sucessivos passos se repetem para evoluir esses indivíduos: o cálculo da *adequação* de cada indivíduo (linha 3); a *seleção* dos indivíduos (linha 4), baseada em sua adequação, para a geração de uma nova população, através da aplicação de *operadores genéticos* (linha 5–7).

A seguir, cada etapa do processo de evolução será apresentada em maiores detalhes: a Seção 2.3.1 descreve a representação dos indivíduos; a Seção 2.3.2 apresenta diferentes métodos para a geração da população inicial de indivíduos; a Seção 2.3.3 descreve, em linha gerais, o processo de cálculo da adequação de um indivíduo; a Seção 2.3.4 caracteriza diferentes métodos para a seleção de indivíduos; e a Seção 2.3.5 apresenta os principais operadores genéticos utilizados por programação genética.

2.3.1 Indivíduos

Geralmente, um indivíduo em programação genética é representado por uma árvore. Essa árvore é formada por dois tipos de nodos: os *terminais* (folhas) e as *funções* (nodos internos).

Os terminais podem ser entradas de um programa, constantes, ou funções sem argumentos [5].

As funções podem ser qualquer tipo de função definida em um programa. Elas recebem um determinado número de entradas e produzem uma saída. Exemplos de funções são operadores aritméticos ou lógicos, estruturas de controle, estruturas de repetição, subrotinas de programas, dentre outras.

A Figura 2.11 apresenta um exemplo de indivíduo. Os terminais desse indivíduo são $\{x, y, z\}$, e seu conjunto de funções é $\{+, *, /, \sqrt{}\}$. Esse indivíduo representa a função

$$f(x, y, z) = \sqrt{z} + \frac{x * y}{x}.$$

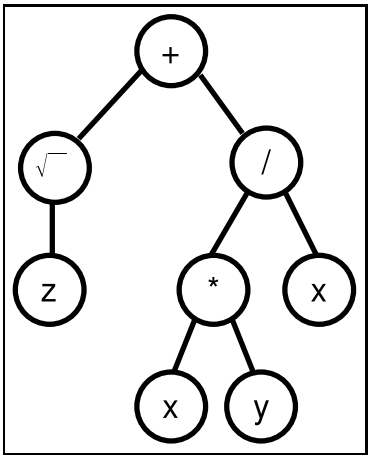


Figura 2.11: Exemplo de indivíduo.

2.3.2 População Inicial

No início da execução de um processo evolutivo, é necessário gerar uma população inicial de indivíduos. Dois diferentes métodos de inicialização são normalmente utilizados, o *crescimento (grow)* e o *cheio (full)* [38]. Para ambos os métodos, é comum definir uma altura máxima para as árvores geradas [5].

O método *crescimento* gera cada indivíduo da população escolhendo aleatoriamente como raiz um terminal ou uma função. Esse processo se repete recursivamente para todos ramos da raiz, se existirem. O crescimento continua enquanto uma função for escolhida como novo nodo da árvore, ou até que a altura máxima seja alcançada. Nesse método, as árvores (indivíduos) geradas podem não ser balanceadas e não ter sua altura ou número de nodos máximos.

O método *cheio* escolhe apenas funções até que a altura máxima da árvore seja alcançada. Quando isso ocorre, os terminais são selecionados. Esse método dá origem apenas a árvores balanceadas e com a altura máxima definida.

É importante observar que todo o processo evolutivo se realizará a partir dessa população inicial. Por isso, é desejável que essa população seja diversificada [5]. Uma população uniforme é mais propensa a ter problemas com ótimos locais. Para aumentar a diversidade na população inicial, utiliza-se uma forma híbrida de inicialização, chamada *half-and-half*, unindo os dois métodos mencionados: 50% dos indivíduos são gerados pelo método *crescimento* e os outros 50% são gerados pelo método *cheio*.

Além disso, é possível utilizar uma faixa de alturas máximas para as árvores da população inicial, ao invés de um único valor. Por exemplo, se for utilizada uma faixa 3–6, 25% do indivíduos terão altura máxima 3, outros 25% terão altura máxima 4, e assim

sucessivamente.

2.3.3 Cálculo da Adequação

A adequação de um indivíduo é uma medida quantitativa que determina a eficácia desse indivíduo na resolução de um determinado problema. Essa medida possibilita ao processo evolutivo escolher quando um indivíduo será excluído da população, ou quando a partir dele se originarão novos indivíduos.

A adequação de um indivíduos é geralmente calculada a partir de um *conjunto de treinamento*. O conjunto de treinamento é um conjunto de entradas de um problema para as quais as saídas são previamente conhecidas.

O cálculo da adequação é bem específico, uma vez que a qualidade de uma solução é inerente ao problema abordado. Assim, para exemplificar o cálculo da adequação, será considerado um problema de regressão, sendo o conjunto de treinamento a ser utilizado mostrado na Tabela 2.3.

Tabela 2.3: Conjunto de treinamento.

Entradas		**Saída**
0	0	2
10	3	4,6
5	9	4,8

O conjunto de funções utilizado é $\{+, -, *, /\}$ e os terminais são $1, x, y$. Será calculada a adequação dos indivíduos ind_1 e ind_2, apresentados na Figura 2.12. Esses indivíduos representam as funções

$$f_{ind_1}(x, y) = \frac{x + y}{1} + 2$$

$$f_{ind_2}(x, y) = \frac{x + y}{4} + 2$$

A adequação de cada indivíduo será calculada pela função

$$f_A = \frac{1}{s + 1}, \tag{2.4}$$

onde s é a soma do módulo das diferenças entre o valor calculado pelo indivíduo e o valor esperado, dado pelo conjunto de treinamento.

A Tabela 2.4 mostra o desempenho dos indivíduos ind_1 e ind_2 no conjunto de treinamento. Nessa tabela, as colunas x e y representam as entradas do conjunto de treinamento; f_{ind_1} e f_{ind_2} mostram as saídas calculadas pelos indivíduos ind_1 e ind_2, respectivamente; e

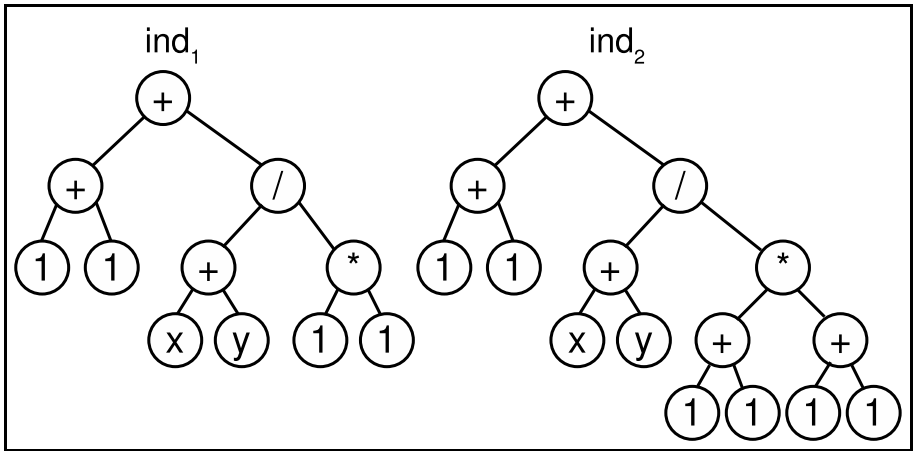


Figura 2.12: Indivíduos para cálculo da adequação.

é a saída esperada para cada entrada; $|e-f_{ind_1}|$ e $|e-f_{ind_2}|$ denotam o módulo da diferença entre cada saída esperada e a calculada pelos indivíduos ind_1 e ind_2, respectivamente; e s é a soma das diferenças $|e - f_{ind_1}|$ e $|e - f_{ind_2}|$.

Tabela 2.4: Desempenho dos indivíduos ind_1 e ind_2 no conjunto de treinamento.

x	**y**	f_{ind_1}	f_{ind_2}	e	$\|e - f_{ind_1}\|$	$\|e - f_{ind_2}\|$
0	0	2	2	2	0	0
10	3	15	5,25	4,6	10,4	0,65
5	9	16	5,50	4,8	11,2	0,7
s					21,6	1,35

Assim, os valores de adequação dos indivíduos ind_1 e ind_2 são aproximadamente $\frac{1}{21,6+1} = 0,044$ e $\frac{1}{1,35+1} = 0,43$, respectivamente. A partir desses valores pode-se concluir que o indivíduo ind_2 é mais apto que o indivíduo ind_1, ou seja, o indivíduo ind_2 é mais eficaz para resolver o problema proposto.

2.3.4 Seleção de Indivíduos

Os métodos de seleção são responsáveis por definir quais indivíduos sofrerão as operações genéticas, ou seja, aqueles que serão a base para a próxima geração, e aqueles que serão excluídos da população. Os métodos de seleção mais comuns [4] são o proporcional à adequação (roleta), o baseado em torneio e o baseado em ordenação (*rank-based selection*).

O método da roleta atribui a cada indivíduo uma probabilidade de ser selecionado. Essa probabilidade é dada pela fórmula

$$p_i = \frac{f_{A_i}}{\sum_j f_{A_j}}, \tag{2.5}$$

onde p_i é a probabilidade atribuída ao indivíduo i, f_{A_i} é a adequação do indivíduo i e $\sum_j f_{A_j}$ é a soma dos valores de adequação de todos os indivíduos da população.

A seleção baseada em torneio escolhe aleatoriamente um determinado número de indivíduos da população e seleciona, dentre os indivíduos escolhidos, aquele com maior adequação.

A seleção baseada em ordenação ordena os indivíduos da população pelos seus valores de adequação. Baseado na sua posição nessa ordenacão, cada indivíduo recebe uma probabilidade de ser selecionado.

2.3.5 Operadores Genéticos

Os operadores genéticos são responsáveis por introduzir variabilidade na população, possibilitando a geração de indivíduos (soluções) melhores durante o processo de evolução. Os operadores genéticos mais comuns são a *reprodução (reproduction)*, a *recombinação (crossover)* e a *mutação (mutation)*.

O operador de reprodução simplesmente copia um indivíduo selecionado e insere a cópia na população.

O operador de recombinação permuta sub-árvores de dois indivíduos escolhidos originando dois novos. O processo de recombinação é exemplificado na Figura 2.13. Nessa figura, as sub-árvores em destaque foram escolhidas aleatoriamente de cada pai. Essas sub-árvores são trocadas, dando origem a dois novos indivíduos.

Em geral, o processo de evolução sugere a recombinação de sub-árvores entre os melhores indivíduos para originar descendentes ainda melhores.

O operador de mutação insere uma perturbação nos indivíduos da população. Dado um indivíduo, o operador de mutação escolhe aleatoriamente uma sub-árvore desse indivíduo e a troca por uma nova sub-árvore gerada de forma aleatória. A Figura 2.14 ilustra o processo de mutação.

A Figura 2.14 (a) mostra o indivíduo que sofrerá mutação, destacando a sub-árvore que será substituída. A Figura 2.14 (b) mostra a sub-árvore gerada aleatoriamente e a Figura 2.14 (c) exibe o indivíduo resultante da operação de mutação.

A cada operador genético é associada uma taxa que, em geral, assume dois significados. Ela pode indicar a proporção da população da nova geração que será criada a partir da

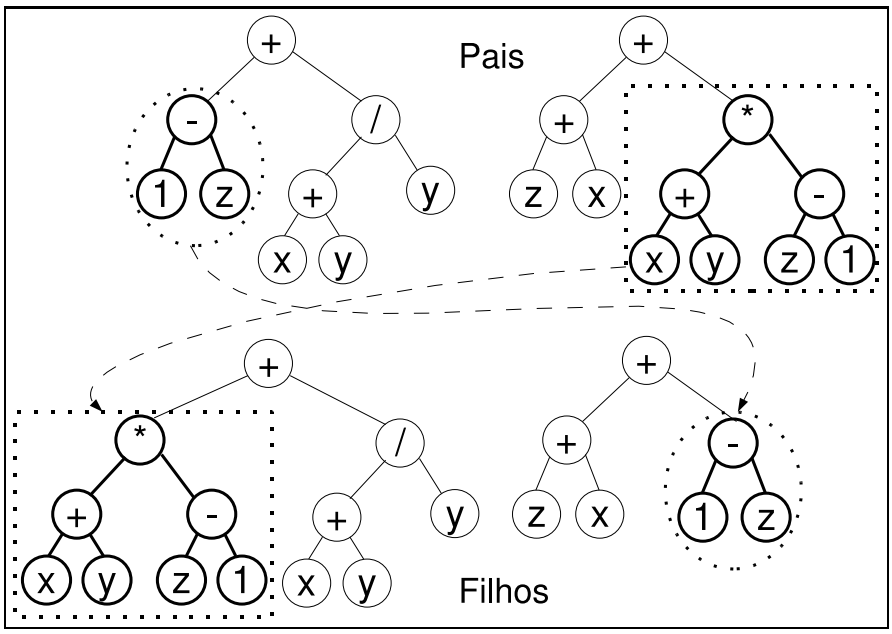


Figura 2.13: Exemplo de recombinação.

aplicação do operador, ou ainda definir a probabilidade de se utilizar o operador para gerar cada um dos novos indivíduos.

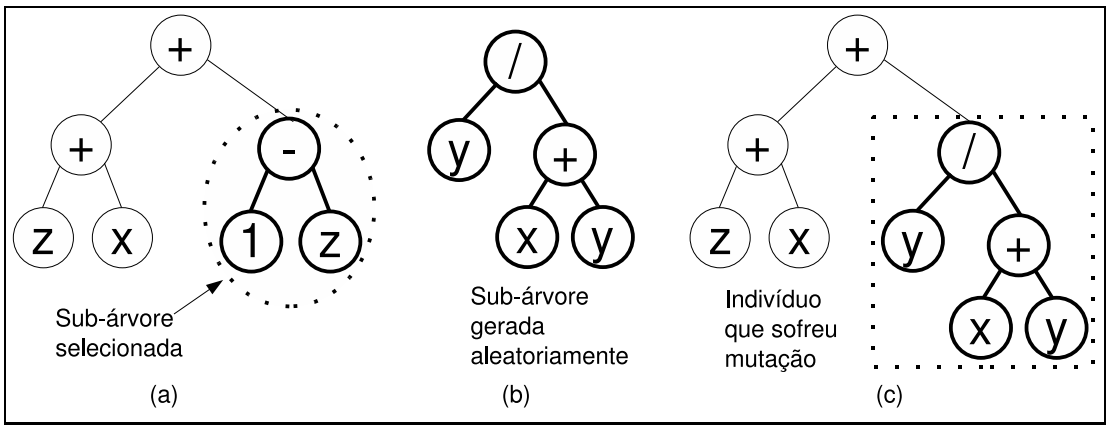


Figura 2.14: Exemplo de mutação.

Capítulo 3

Realimentação de Relevância Baseada em Programação Genética

Este capítulo propõe dois arcabouços para recuperação de imagens por conteúdo com realimentação de relevância baseada em programação genética. O primeiro, PG^{+}, utiliza apenas a informação provida pelas imagens indicadas pelo usuário como positivas no processo de aprendizado. O segundo, $PG^{\pm}$, considera também as imagens negativas para assimilar a necessidade do usuário.

A seção 3.1 apresenta o arcabouço PG^{+}. A seção 3.2 discute $PG^{\pm}$, ressaltando as modificações necessárias no arcabouço PG^{+} para incorporar as imagens irrelevantes ao processo de aprendizado e de seleção de imagens.

3.1 Arcabouço PG^{+}

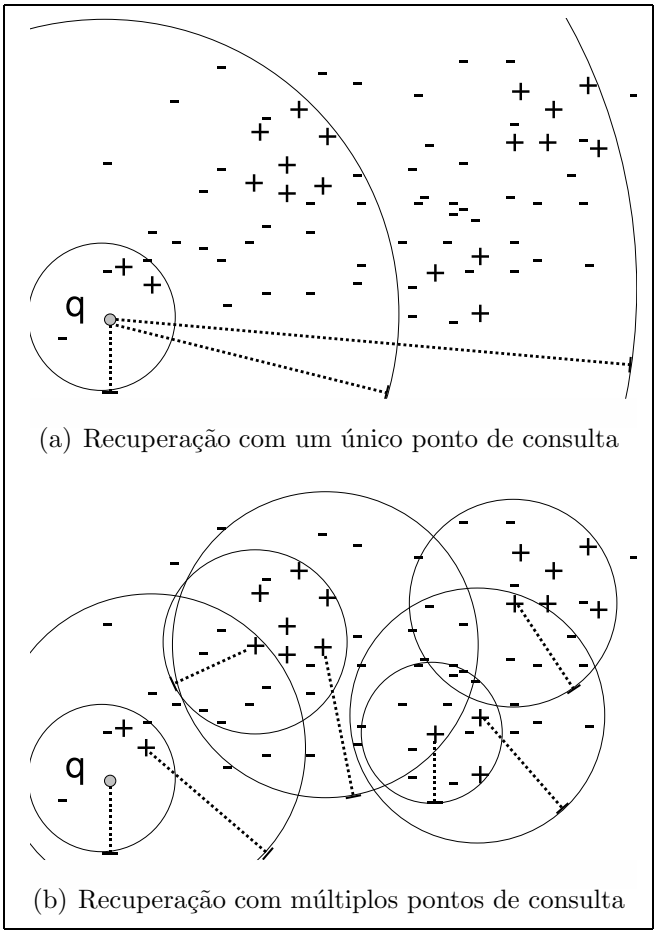
Seja $\hat{D} = (\mathcal{D}, \delta_{\mathcal{D}})$ (veja Seção 2.1) um descritor composto utilizado para ordenar as imagens da base, representadas por $DB = \{db_1, db_2, \ldots, db_N\}$. O principal objetivo do algoritmo de aprendizado proposto é encontrar um conjunto $\{\delta_{\mathcal{D}}\}$ de funções de combinação que melhor traduzam a percepção visual do usuário. Dessa forma, almeja-se redefinir as relações de similaridade entre as imagens, utilizando-se as funções de combinação encontradas, para que essas relações melhor se adeqüem à necessidade do usuário.

O conjunto de K descritores simples que compõem $\hat{D}$ é representado por $\mathcal{D} = \{D_1, D_2, \ldots, D_K\}$. A similaridade entre duas imagens I_j e I_k, calculada por D_i, é representada por $d_{iI_jI_k}$. Todas as similaridades são normalizadas no intervalo $[0, 1]$. A normalização Gaussiana [48] é utilizada para normalizar esses valores.

Seja L o número de imagens exibidas para o usuário em cada iteração. Seja o padrão de consulta $Q = \{q_1, q_2, \ldots, q_M\}$, onde M é o número de elementos em Q, formado pela imagem de consulta q_i e todas as imagens rotuladas como relevante durante uma sessão

de consulta.

Observe que o padrão de consulta utilizado em PG^+ é composto não apenas pela imagem de consulta definida inicialmente pelo usuário, mas também por todas as imagens marcadas como relevante pelo usuário em todas as iterações. Dessa forma, optou-se por uma abordagem baseada em múltiplos pontos de consulta para o processo de recuperação. Essa abordagem é mais robusta às imperfeições na definição do espaço de características, em relação à percepção do usuário. Por exemplo, considere a Figura 3.1.



(a) Recuperação com um único ponto de consulta

(b) Recuperação com múltiplos pontos de consulta

Figura 3.1: Distribuição de imagens no espaço de características.

A Figura 3.1(a) mostra as imagens distribuídas em um espaço de características definido por um determinado descritor. Como pode ser observado, esse descritor não reflete a noção de similaridade do usuário considerado nesse exemplo, pois as imagens relevantes, representadas pelo símbolo +, estão dispersas em diferentes agrupamentos nesse espaço de características. Além disso, a imagem de consulta q está distante desses agrupamen-

tos. Dessa forma, encontrar as imagens relevantes a partir de q é uma tarefa difícil. A Figura 3.1(b) apresenta a mesma distribuição de imagens. Porém, essa figura considera a utilização de múltiplos pontos de consulta, conforme a abordagem adotada no método proposto. Como pode ser observado, utilizando-se múltiplos pontos no padrão de consulta, as distâncias mínimas para as imagens relevantes são menores a cada iteração, facilitando o processo de recuperação.

Dessa forma, acredita-se que a utilização de múltiplos pontos de consulta em conjunto com a redefinição da relação de similaridade proporcionada pelo aprendizado baseado em PG, possibilita uma melhor caracterização da percepção visual do usuário.

O Algoritmo 2 apresenta uma visão geral do processo de recuperação proposto. Nesse algoritmo, as interações do usuário são indicadas em itálico. No começo do processo de recuperação, o usuário define a imagem de consulta q_1 (linha 1). A partir dessa imagem, um conjunto inicial de imagens é selecionado para ser exibido para o usuário (linha 2). Com isso, o processo de realimentação de relevância se inicia, com a indicação das imagens relevantes pelo usuário, considerando as imagens do conjunto inicial retornado. Em cada iteração, as seguintes etapas são realizadas: indicação das imagens relevantes pelo usuário (linha 4); atualização do padrão de consulta (linha 5); aprendizado da percepção visual do usuário utilizando programação genética (linha 6); ordenação das imagens da base (linha 7); e exibição das imagens mais similares (linha 8).

Algoritmo 2 O processo de realimentação de relevância baseado em programação genética.

1 *Definição da imagem de consulta* q_1
2 Exiba o conjunto inicial de imagens
3 **enquanto** o usuário não estiver satisfeito **faça**
4 *Indicação do usuário*
5 Atualize o padrão de consulta Q
6 Utilize PG para encontrar os melhores indivíduos (funções de combinação de similaridades) – veja o Algoritmo 1 na Seção 2.3
7 Ordene as imagens da base
8 Exiba as L imagens mais similares
9 **fim enquanto**

As seções seguintes apresentam em detalhes a estratégia de seleção do conjunto inicial de imagens (Seção 3.1.1), a utilização de PG para encontrar as funções de combinação de similaridade (Seção 3.1.2) e o algoritmo para ordenar as imagens da base (Seção 3.1.3).

3.1.1 Seleção do Conjunto Inicial de Imagens

O conjunto inicial de imagens exibidas para o usuário é definido pela ordenação das imagens da base db_i em relação a sua similaridade em relação à imagem de consulta q_1. Esse processo é realizado em duas etapas. Inicialmente, cada descridor simples $D_j \in \mathcal{D}$ é utilizado para calcular a similaridade $d_{jq_1db_i}$. Em seguida, os valores calculados são combinados pela sua média aritmética, ou seja

$$\delta_{MEDIA}(q_1, db_i) = \frac{\sum_{j=1}^{K} d_{jq_1db_i}}{K} \tag{3.1}$$

Essa combinação utiliza todos os descritores empregados no processo de recuperação e atribui o mesmo grau de importância para cada um. Essa forma de combinação permite a definição do conjunto inicial de imagens retornadas para o usurário sem a necessidade de conhecimento prévio sobre o desempenho dos descritores.

A partir da similaridade calculada por δ_{MEDIA} em relação à imagem de consulta, as imagens da base são ordenadas e as L primeiras são exibidas . Assim, o usuário identifica dentre essas o conjunto $R = \{R_1, R_2, \ldots, R_P\}$ com as P imagens relevantes. O próximo passo é inserir todas as imagens $\{R_i | R_i \notin Q\}$ no padrão de consulta Q.

3.1.2 Busca pelas Melhores Combinações de Similaridades – O Aprendizado PG^+

Como mencionado anteriormente, o objetivo do mecanismo de aprendizado consiste na busca de funções de combinação de similaridades que melhor codificam as necessidades do usuário. O arcabouço proposto utiliza PG para encontrar essas combinações. Conforme mostrado na Seção 2.3, PG requer a definição de vários componentes. Os únicos entre esses componentes que necessitam de uma definição específica para o processo de aprendizado proposto são a representação e o cálculo da adequação dos indivíduos. Os demais componentes utilizados são definidos na Seção 2.3.

A Seção 3.1.2.1 discute a representação do indivíduo adotada e a Seção 3.1.2.2 apresenta o processo de cálculo da adequação de um indivíduo.

3.1.2.1 Representação do Indivíduo

No método proposto, cada indivíduo representa uma função $\delta_{\mathcal{D}}$, ou seja, uma função de combinação de similaridades. Para isso, é utilizada uma representação codificada em uma estrutura de árvore, como proposto em [14, 15].

O conjunto de funções (nodos internos) desse indivíduo é composto por operadores aritméticos. O conjunto de terminais (folhas) é formado por valores de similaridade $d_{iI_jI_k}$. A Figura 3.2 mostra um exemplo de indivíduo segundo essa representação. Essa figura considera a utilização de três descritores distintos e o conjunto de operadores $\{+, /, \log, \sqrt{}\}$ como nodos internos. O indivíduo mostrado representa a função $f(d_{1I_jI_k}, d_{2I_jI_k}, d_{3I_jI_k}) = (d_{1I_jI_k} + d_{3I_jI_k}) * \sqrt{\frac{d_{2I_jI_k}}{d_{1I_jI_k}}}$.

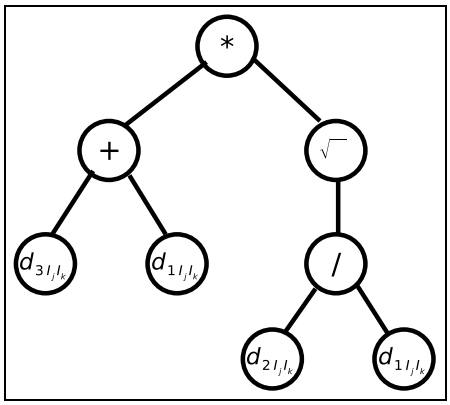


Figura 3.2: Exemplo de indivíduo.

Essa representação do indivíduo se adequa perfeitamente aos propósitos do método proposto, pois é uma forma de codificação direta para uma função de combinação de similaridades $\delta_{\mathcal{D}}$.

3.1.2.2 Cálculo da Adequação de Indivíduos

O objetivo do processo de cálculo da adequação proposto é atribuir os maiores valores de adequação para os indivíduos que melhor codificam as preferências do usuário. A abordagem proposta baseia esse cálculo na ordenação das imagens da base definida por cada indivíduo da PG. Os indivíduos que inserem as imagens marcadas como relevantes nas primeiras posições da ordenação recebem um alto valor de adequação. O processo de cálculo da adequação proposto é baseado nesse critério objetivo.

Definição do conjunto de treinamento: A adequação de um indivíduo é calculada a partir de seu desempenho em um conjunto de treinamento. No método proposto, o conjunto de treinamento é definido como a seguir.

Definição 5 *O **conjunto de treinamento** é definido como um par $\mathcal{T}^+ = (T, r^+)$ onde:*

- *o conjunto das **imagens de treinamento** $T = \{t_1, t_2, \ldots, t_{N_T}\}$ é um conjunto composto por N_T imagens distintas.*

- $r^+ : T \rightarrow \{0, 1\}$ *é uma função que indica a rotulação do usuário em relação a cada imagem* $t \in T$.

A função $r^+(t_i)$, onde $t_i \in T$, é definida como

$$r^+(t_i) = \begin{cases} 1, & \text{se } t_i \text{ é relevante.} \\ 0, & \text{caso contrário.} \end{cases} \tag{3.2}$$

É importante observar que o número de imagens do conjunto de treinamento N_T precisa ser pequeno ($N_T << N$) para permitir o cálculo eficiente da adequação de cada indivíduo. Como mencionado na Seção 2.2.2, o tempo de execução de cada iteração deve ser pequeno, para viabilizar a interação do usuário com o sistema em tempo real. Por outro lado, o conjunto de imagens de treinamento T deve conter um número de imagens suficiente para representar as necessidades do usuário e as imagens da base.

Na abordagem proposta, T é composto pelas M imagens marcadas como relevantes, pelas últimas $L - M$ imagens exibidas para o usuário e por outras $N_T - L$ escolhidas aleatoriamente da base. Essa formação não admite imagens repetidas. Além disso, caso $M \geq L$, então T é formado por L imagens escolhidas aleatoriamente dentre as relevantes e pelas $N_T - L$ escolhidas aleatoriamente da base.

Em relação ao tamanho do conjunto de treinamento N_T, os experimentos mostram que valores entre 0.5% e 5% do tamanho da base são adequados para N_T. Observe que a composição do conjunto de treinamento especificada representa tanto a necessidade do usuário, pela escolha das imagens relevantes e das últimas exibidas, quanto a base, pela escolha aleatória de algumas de suas imagens. A adição de imagens não rotuladas da base insere ruídos no conjunto de treinamento, aumentando o poder de generalização dos indivíduos obtidos durante o processo de aprendizado.

Cálculo da adequação: A adequação de um indivíduo δ_i é calculada com base na similaridade entre as imagens do padrão de consulta e todas as imagens do conjunto de treinamento. Observe que essa similaridade define a ordenação das imagens de treinamento. O processo de cálculo da adequação de um indivíduo é realizada em três fases. Na primeira, são definidas M listas ordenadas, cada uma considerando a similaridade entre todas as imagens do conjunto de treinamento e cada imagem do padrão de consulta, segundo δ_i. Na segunda fase, essas ordenações são avaliadas. Finalmente, na última fase, o valor final da adequação do indivíduo δ_i é calculado. A Figura 3.3 ilustra esse processo aplicado a um indivíduo δ_i.

Fase 1: Como pode ser visto na Figura 3.3, na primeira fase, para cada imagem do padrão de consulta q_j, as imagens de treinamento $t_k \in T$ são ordenadas em relação a suas similaridades em relação q_j calculadas a partir do indivíduo (função de combinação) $\delta_i(q_j, t_k)$. As L primeiras imagens definem uma lista ordenada $rk_{j\delta_i}$. Dessa forma, M listas são obtidas, cada uma relacionada a uma imagem do padrão de consulta q_j.

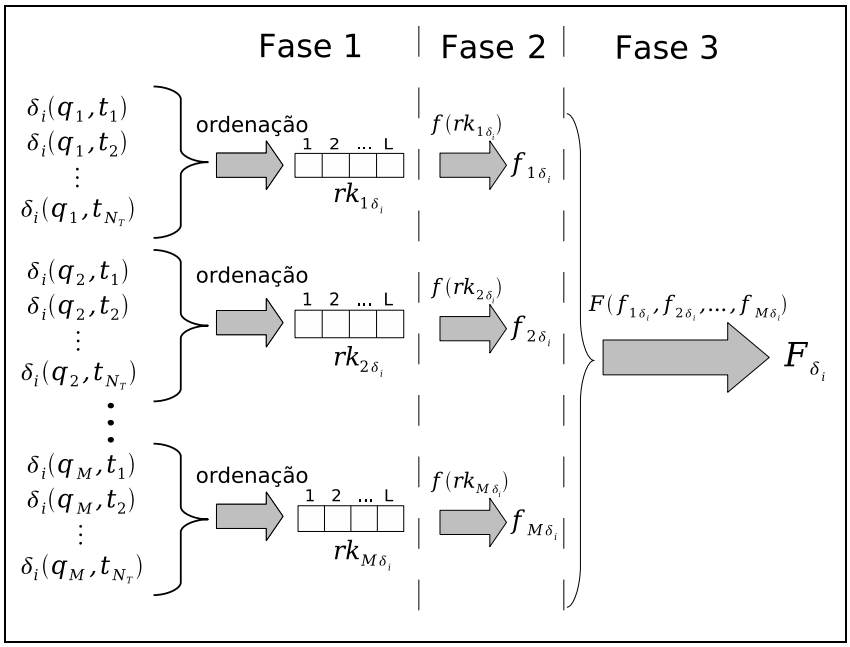


Figura 3.3: Cálculo da adequação de um indivíduo δ_i.

Fase 2: Uma vez que as listas ordenadas são obtidas, a segunda fase se inicia. O objetivo dessa fase é avaliar cada lista $rk_{j\delta_i}$ gerada na Fase 1. Essa avaliação consiste em atribuir valores altos para as listas em que as imagens relevantes presentes no conjunto de treinamento estão posicionadas nas primeiras posições. Essa avaliação é realizada por uma função de avaliação $f(rk_{j\delta_i})$.

Na abordagem proposta, a função $f(rk_{j\delta_i})$ segue os princípios da *teoria de utilidade (utility theory)* [26]. Segundo essa teoria, há uma *função de utilidade (utility function)* que atribui um valor a um item, de acordo com a preferência do usuário. Em geral, assume-se que a utilidade de um item é dada em função de sua posição em uma lista ordenada [24]. Ela diminui a medida que os itens se posicionam mais próximos do final da lista. Formalmente, dados dois itens It_i e It_{i+1}, onde i é uma posição em uma lista ordenada, a seguinte condição deve ser satisfeita por uma função de utilidade $U(x)$: $U(It_i) > U(It_{i+1})$. No contexto desta dissertação, cada item é uma imagem.

A função $f(rk_{j\delta_i})$ tem a forma

$$f(rk_{j\delta_i}) = k \times \sum_{l=1}^{L} r(rk_{j\delta_i}[l]) \times g(l) \tag{3.3}$$

onde k é uma constante, $rk_{j\delta_i}[l]$ é a l-ésima imagem na lista ordenada $rk_{j\delta_i}$, $r(\cdot)$ é uma função que indica a rotulação do usuário em relação a cada imagem do conjunto de treinamento e $g(\cdot)$ é uma função decrescente.

Um exemplo de função de avaliação $f(rk_{j\delta_i})$ é

$$f(rk_{j\delta_i}) = \sum_{l=1}^{L} r(rk_{j\delta_i}[l]) \times k_1 \times \log_{10}(1000/l) \text{ [24]} \tag{3.4}$$

onde k_1 é uma constante definida experimentalmente em [24] como 2 e $r(rk_{j\delta_i}) = 1$, se $rk_{j\delta_i}[l] \in R$ ou $r(rk_{j\delta_i}) = 0$, caso contrário.

Dessa forma, a aplicação da função $f(rk_{j\delta_i})$ para cada lista ordenada $rk_{j\delta_i}$ define M valores $f_{1\delta_i}, f_{2\delta_i}, \ldots, f_{M\delta_i}$.

Fase 3: Na terceira fase, o valor final da adequação F_{δ_i} do indivíduo δ_i é calculada como a média aritmética dos valores $f_{j\delta_i}$, ou seja

$$F(f_{1\delta_i}, f_{2\delta_i}, \ldots, f_{M\delta_i}) = \frac{\sum_{j=1}^{M} f_{j\delta_i}}{M}. \tag{3.5}$$

O Algoritmo 3 sumariza o processo de cálculo da adequação dos indivíduos. As linhas 3–8 se referem à primeira fase do processo, a definição das listas ordenadas. A Fase 2, o processo de avaliação das listas, corresponde à linha 9. Por fim, o valor de adequação final é calculado na linha 11 (Fase 3).

Algoritmo 3 O processo de cálculo da adequação de um indivíduo.

```
1  Entrada: indivíduo δ_i, padrão de consulta Q, conjunto de treinamento T.
2  Saída: o valor de adequação do indivíduo δ_i.
3  para todos q_j ∈ Q faça
4      para todos t_k ∈ T faça
5          rk_{jδ_i}[k].chave ← δ_i(q_j, t_k)
6          rk_{jδ_i}[k].elemento ← t_k
7      fim para
8      Ordene rk_{jδ_i}
9      f_{jδ_i} ← f(rk_{jδ_i})
10 fim para
11 F_{δ_i} ← (Σ_{j=1}^{M} f_{jδ_i}) / M
12 retorne F_{δ_i}
```

Um importante aspecto do processo apresentado é a incorporação de todo conhecimento acumulado durante as iterações ao processo de aprendizado, pela utilização de todas as imagens relevantes conhecidas, tanto como imagens do conjunto de treinamento quanto como imagens de consulta no cálculo da adequação. Essa solução endereça um

problema inerente ao uso de realimentação de relevância em sistemas de recuperação de imagens por conteúdo: o tamanho pequeno do conjunto de treinamento provido para o processo de aprendizado [61].

3.1.3 Ordenação das Imagens da Base

Uma vez calculada a adequação dos indivíduos, é possível definir aquele que apresentou a melhor adequação e que será utilizado para ordenar as imagens da base. No entanto, é possível que mais de um indivíduo apresente um alto valor de adequação. De fato, se o tamanho M do padrão de consulta é pequeno, há uma grande probabilidade que vários indivíduos apresentem uma boa adequação. Dessa forma, é necessário definir um critério para selecionar quais serão utilizados para ordenar a base.

No método proposto, não apenas um indivíduo é utilizado para ordenar as imagens da base, mas todos aqueles que apresentam um alto valor de adequação. Para realizar essa tarefa, a estratégia para ordenação proposta combina as ordenações da base segundo cada indivíduo considerado "bom". Essa combinação é realizada por meio de um esquema de *votação*. A seleção de indivíduos para votação é discutida na Seção 3.1.3.1 e o esquema de votação utilizado é apresentado na Seção 3.1.3.2;

3.1.3.1 Seleção de Indivíduos

O primeiro passo é definir quais indivíduos serão selecionados para ordenar a base. Seja δ_{melhor} o melhor indivíduo obtido do processo de aprendizado baseado em PG (veja Seção 3.1) na iteração corrente. O conjunto S de indivíduos selecionados para participar é definido como

$$S = \{\delta_i | \frac{F_{\delta_i}}{F_{\delta_{melhor}}} \geq \alpha\} \tag{3.6}$$

onde $\alpha \in [0, 1]$ é uma constante. Essa constante é chamada *limitante para votação*.

3.1.3.2 Votação

Cada um dos indivíduos selecionados para a votação atribui votos para L (número de imagens exibidas) imagens. O Algoritmo 4 apresenta o processo de votação dos indivíduos selecionados.

Inicialmente, as imagens da base são ordenadas utilizando-se os indivíduos selecionados δ_i, de acordo com a similaridade $Sim^+_{\delta_i}(Q, db_j)$, entre cada imagem da base db_j e o padrão de consulta Q (linhas 3–8). Dado um conjunto de imagens $IMG = \{img_1, img_2, \ldots, img_n\}$ e uma imagem I, seja $max_{\delta_i}(IMG, I)$ a maior similaridade entre I e as imagens do conjunto IMG, conforme definido na Equação 3.7.

Algoritmo 4 O processo de votação utilizado para ordenar as imagens da base.

1 **Entrada**: Conjunto S dos indivíduos selecionados, base de imagens DB,padrão de consulta Q, número de imagens exibidas L.
2 **Saída**: Lista ordenada com as imagens para serem exibidas.
3 **para todos** $\delta_i \in S$ **faça**
4 **para todos** $db_j \in DB$ **faça**
5 $rk_i[j].chave \leftarrow Sim_{\delta_i}(Q, db_j)$
6 $rk_i[j].elemento \leftarrow db_j$
7 **fim para**
8 Ordene $rk_i[j]$
9 **for** $j \leftarrow 1$ to L **faça**
10 $votos[rk_i[j].elemento] \leftarrow votos[rk_i[j].elemento] + 1/j$
11 **fim para**
12 **fim para**
13 Ordene DB em relação a *votos*
14 **retorne** as L imagens mais votadas

$$max_{\delta_i}(IMG, I) = \{\delta_i(img_k, I) \mid \delta_i(img_k, I) > \delta_i(img_l, I) \\ \forall\, img_k, img_l \in IMG \wedge k \neq l\} \tag{3.7}$$

A função de similaridade $Sim^{+}_{\delta_i}(Q, db_j)$ é definida como o maior valor entre as similaridades entre db_j e cada imagem do padrão de consulta, ou seja, $max(\delta_i(Q, db_j))$, conforme descrito na Equação 3.8. Dessa forma, cada indivíduo selecionado δ_i define uma lista ordenada com as imagens da base.

$$Sim^{+}_{\delta_i}(Q, db_j) = max(\delta_i(Q, db_j)) \tag{3.8}$$

Em seguida, cada imagem nas L primeiras posições em cada lista recebe um voto inversamente proporcional a sua posição (linhas 9–11). Por exemplo, a primeira imagem recebe um voto igual a 1; a segunda recebe 1/2; a terceira, 1/3 e assim por diante. Finalmente, as L imagens mais votadas são selecionadas para serem exibidas para o usuário.

O esquema de votação proposto para ordenação das imagens da base é exemplificado na Figura 3.4. Nesse exemplo, há dez imagens na base ($DB = \{img_1, img_2, img_3, img_4, img_5, img_6, img_7, img_8, img_9, img_{10}\}$), cinco indivíduos selecionados para votação ($S = \{\delta_1, \delta_2, \delta_3, \delta_4, \delta_5\}$) e três imagens são exibidas para o usuário em cada iteração ($L = 3$). Na ordenação da base definida pelo indivíduo δ_1, as imagens img_1, img_2 e img_3 aparecem, nessa ordem, nas três primeiras posições. Assim, a imagem img_1 recebe um voto igual a 1, a imagem img_2, por sua vez, recebe 1/2, enquanto 1/3 é adicionado a soma de votos

da imagem img_3. Esse processo se repete para cada ordenação definida pelos outros indivíduos. A tabela à esquerda dessa figura mostra o total de votos que cada imagem candidata recebeu. Finalmente, a tabela à direita apresenta as imagens da base ordenadas em relação às suas somas totais de votos. As primeiras imagens, img_1, img_3 e img_5, são exibidas para o usuário.

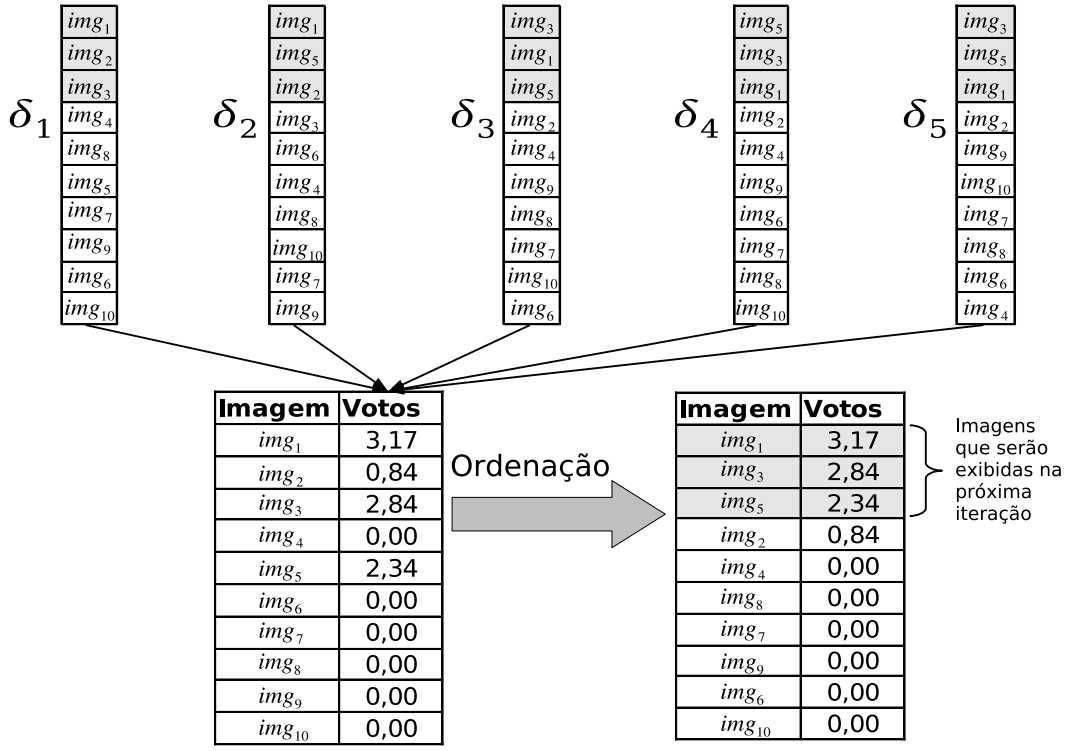


Imagem	Votos
img_1	3,17
img_2	0,84
img_3	2,84
img_4	0,00
img_5	2,34
img_6	0,00
img_7	0,00
img_8	0,00
img_9	0,00
img_{10}	0,00

Imagem	Votos
img_1	3,17
img_3	2,84
img_5	2,34
img_2	0,84
img_4	0,00
img_8	0,00
img_7	0,00
img_9	0,00
img_6	0,00
img_{10}	0,00

Figura 3.4: Exemplo de aplicação do esquema de votação.

Observe que todas as imagens do padrão de consulta são utilizadas para ordenar as imagens da base. Dessa forma, a abordagem proposta considera não apenas as imagens similares à imagem de consulta como boas candidatas para serem relevantes para o usuário, mas também aquelas similares a qualquer imagem do padrão de consulta, ou seja, aquelas similares a qualquer imagem que já tenha sido rotulada como relevante pelo usuário. Assim, é considerada a possibilidade de que as imagens relevantes estejam agrupadas em torno de vários pontos distintos no espaço de características. A Figura 3.5 ilustra essa situação.

No exemplo da Figura 3.5, as imagens relevantes estão agrupadas em quatro pontos distintos do espaço de características. O padrão de consulta é formado por quatro imagens, q_1, q_2, q_3 e q_4. Os círculos pontilhados delimitam regiões próximas de cada imagem do

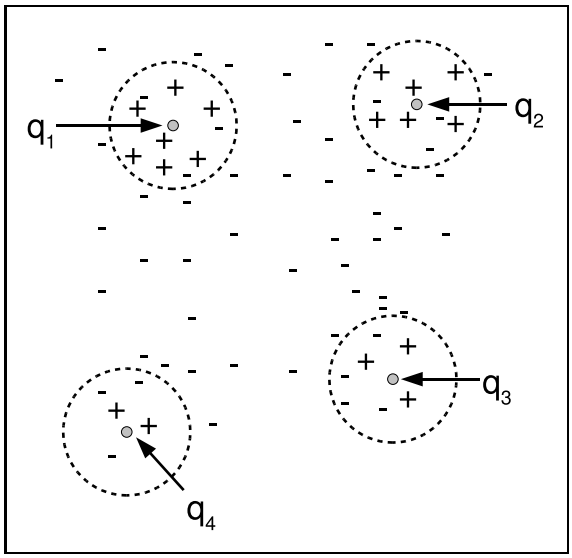


Figura 3.5: Distribuição de imagens no espaço de características.

padrão de consulta.

Outros métodos podem ser utilizados para combinar as ordenações definidas por cada indivíduo selecionado, como os apresentados em [20, 31].

3.2 Arcabouço $PG^{\pm}$

O arcabouço PG^{+} apresentado na Seção 3.1 utiliza apenas a informação provida pelas imagem marcadas como relevantes. Porém, em um processo de recuperação com realimentação de relevância, além de imagens relevantes, os usuários podem também indicar aquelas que são irrelevantes no conjunto exibido. Utilizar a informação sobre imagens irrelevantes pode trazer ganhos para o processo de recuperação. Com esse intuito, é proposto o arcabouço $PG^{\pm}$, uma extensão do arcabouço PG^{+} visando incorporar as imagens irrelevantes ao processo de aprendizado das preferências do usuário e ordenação da base.

Para realizar essa tarefa, alguns componentes de PG^{+} foram alterados. Esses componentes são: a definição do conjunto de treinamento (Seção 3.2.1) e o modo como os indivíduos selecionados para votação são utilizados para ordenar as imagens da base (Seção 3.2.2).

3.2.1 Redefinição do Conjunto de Treinamento

O novo conjunto de treinamento contém além de imagens relevantes e não-rotuladas, as imagens marcadas como irrelevantes. Seja IRR o conjunto das imagens marcadas pelo

usuário como irrelevantes ao longo de todas as iterações. Relembre que M é o tamanho do padrão de consulta Q e L é o número de imagens exibidas por iteração. O novo conjunto de treinamento é definido a seguir.

Definição 6 *O **conjunto de treinamento** é definido como um par $\mathcal{T}^{\pm} = (T, r^{\pm})$ onde:*

- *o conjunto das* imagens de treinamento $T = \{t_1, t_2, \ldots, t_{N_T}\}$ *é um conjunto composto por N_T imagens distintas.*
- $r^{\pm} : T \rightarrow \{-1, 0, 1\}$ *é uma função que indica a rotulação do usuário em relação a cada imagem $t \in T$.*

A função $r^{\pm}(t_i)$, onde $t_i \in T$, é definida como

$$r^{\pm}(t_i) = \begin{cases} 1, & \text{se } t_i \text{ é relevante.} \\ -1, & \text{se } t_i \text{ é irrelevante.} \\ 0, & \text{caso contrário.} \end{cases} \tag{3.9}$$

O conjunto de treinamento utilizado em $PG^{\pm}$ é composto por:

- Todas as imagens do padrão de consulta Q;
- $L - M$ imagens não rotuladas, escolhidas aleatoriamente da base;
- $N_T - L$ imagens irrelevantes, escolhidas aleatoriamente do conjunto IRR.

Novamente, essa formação não admite imagens repetidas. Além disso, caso $M \geq L$, então as imagens não rotuladas não são inseridas no conjunto de treinamento.

O processo de cálculo da adequação do indivíduo é idêntico ao exibido na Seção 3.1.2.2. Porém agora, os indivíduos que recebem os valores de adequação mais altos são aqueles que ao ordenar as imagens de treinamento, inserem as imagens relevantes nas primeiras posições e as irrelevantes nas últimas.

3.2.2 Ordenação das Imagens da Base

O processo de ordenação da base é semelhante ao apresentado na Seção 3.1.3. A distinção reside no modo como os indivíduos selecionados para votação são utilizados para ordenar as imagens da base. Mais especificamente, a diferença está na definição da função $Sim^{+}_{\delta_i}(Q, db_j)$ (Equação 3.8).

A nova versão dessa função, $Sim^{\pm}_{\delta_i}(Q, IRR, db_j)$, considera não apenas a similaridade das imagens da base em relação ao padrão de consulta mas também em relação às imagens rotuladas como irrelevantes. Essa função é definida como

$$Sim^{\pm}_{\delta_i}(Q, IRR, db_j) = \frac{max_{\delta_i}(Q, db_j)}{max_{\delta_i}(IRR, db_j)} \tag{3.10}$$

onde $max_{\delta_i}(Q, db_j)$ e $max_{\delta_i}(IRR, db_j)$ são os maiores valores de similaridade segundo δ_i entre db_j e Q e entre db_j e IRR, respectivamente (veja Equação 3.7).

Observe que na função $Sim^{\pm}_{\delta_i}(Q, IRR, db_j)$, a similaridade da imagem db_j com uma imagem irrelevante é considerada de uma forma "negativa" no cálculo de similaridade, ou seja, quanto maior a similaridade de db_j com uma imagem irrelevante, menor será o valor $Sim^{\pm}_{\delta_i}(Q, IRR, db_j)$. Com isso, essa função privilegia as imagens que são similares a uma relevante e não similares a uma irrelevante. A Figura 3.6 ilustra o uso da função $Sim^{\pm}$.

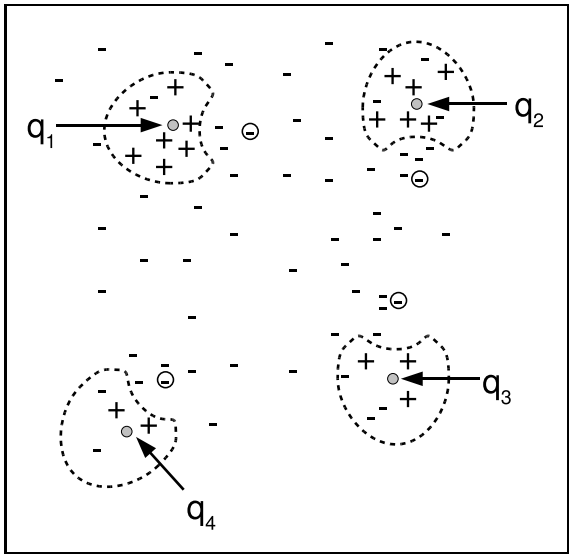


Figura 3.6: Distribuição de imagens no espaço de características.

No exemplo da Figura 3.6, os círculos com sinal de subtração indicam as imagem rotuladas pelo usuário como irrelevantes. O padrão de consulta é formado por quatro imagens: q_1, q_2, q_3 e q_4. As regiões pontilhadas delimitam regiões próximas de cada imagem do padrão de consulta. Observe que nessas regiões há uma distorção próxima as imagens irrelevantes.

Capítulo 4

Experimentos

Este capítulo apresenta a caracterização dos experimentos realizados para validar os métodos propostos. Os descritores utilizados são apresentados na Seção 4.1. A Seção 4.2 descreve as técnicas de realimentação de relevância que servirão de base para comparação com os arcabouços propostos. As bases de imagens empregadas são mostradas na Seção 4.3. A Seção 4.4 apresenta as medidas de avaliação de resultados utilizadas.

4.1 Descritores de Imagens

Os arcabouços propostos possuem grau de generalidade suficiente para não possuir restrições a respeito dos descritores que podem ser utilizados. Descritores de cor, textura e forma são os mais comuns, e são empregados nos experimentos.

Os descritores de cor utilizados são apresentados na Seção 4.1.1. A Seção 4.1.2 descreve os descritores de textura. A Seção 4.1.3 apresenta os descritores de forma utilizados.

4.1.1 Descritores Cor

Foram utilizados três descritores de cor, o *Histograma de Cor*, os *Momentos de Cor* e o *Border/Interior pixel Classification (BIC)*, descritos a seguir:

- **Histograma de Cor:** foi extraído o histograma [52] no espaço de cor *HSV*, quantizado em 16, 4, 4, para tonalidade *(Hue)*, saturação *(Saturation)* e valor *(Value)*, respectivamente, considerando-se todos os *pixels* das imagens. Esse processo de extração originou um vetor de características com 256 dimensões. A distância $L1$ foi utilizada para cálculo da similaridade.

- **Momentos de Cor:** o descritor Momentos de Cor caracteriza a distribuição das cores de uma imagem como os três primeiros momentos de cada canal de cor. Esses

momentos são a média, o desvio-padrão e a terceira raiz da obliqüidade (*skewness*). Novamente, foi utilizado o espaço de cor *HSV*. O processo de extração resultou em um vetor de características com 9 dimensões. A função de similaridade utilizada é definida como uma variação da distância $L1$, chamada d_{mon}. Essa variação atribui pesos pré-definidos para cada *bin* do vetor de características [51].

- **Border/Interior pixel Classification (BIC):** o descritor BIC [50] classifica cada *pixel* de uma imagem como interior ou borda, em um espaço de cor *RGB* quantizado em 4x4x4. Após a classificação, calcula dois histogramas, um considerando apenas os *pixels* de borda e outro considerando apenas os *pixels* interiores. Um *pixel* é classificado com interior se possuir a mesma cor dos *pixels* em sua vizinhança 4 (em cima, em baixo, à esquerda e à direita). Caso contrário, é classificado como borda. Com isso, uma imagem é descrita por dois histogramas, com 64 *bins* cada. Esses dois histogramas definem um vetor de características com 128 dimensões. A função de distância *dLog* [50] é utilizada para comparar dois vetores de características definidos pelo *BIC* em uma escala logarítmica.

4.1.2 Descritores de Textura

Para caracterizar a informação de textura das imagens foram utilizados dois descritores de textura, os *Filtros de Gabor* e o *Spline*, ambos baseados em transformadas *Wavelet*, e são descritos a seguir:

- **Filtros de Gabor:** o descritor Filtros de Gabor utiliza a média e o desvio-padrão da distribuição da energia de uma imagem para representá-la. Essa representação considera 4 escalas e 4 diferentes rotações, resultando em um vetor de características com 32 dimensões. O cálculo da similaridade é realizado pela distância Euclidiana.

- **Spline:** esse descritor utiliza uma transformação *Wavelet Spline* estruturada em árvore para decompor três sub-bandas de uma imagem. Foi empregada uma decomposição em três níveis, resultando em um vetor de características com 26 dimensões. Novamente, a distância Euclidiana foi utilizada para calcular a distância entre dois vetores de características.

4.1.3 Descritores de Forma

Foram utilizados cinco descritores de forma, os *Momentos Invariantes*, o *Descritor de Fourier*, o *Dimensão Fractal Multiescala do Contorno*, o *Beam Angle Statistics* e o *Saliências de Segmento*, descritos a seguir:

- **Momentos Invariantes:** com o descritor Momentos Invariantes, cada imagem é representada por um vetor de características com 14 dimensões, incluindo dois conjuntos de momentos invariantes normalizados [30], um considerando a borda do objeto de interesse na imagem e outro a sua silhueta sólida. A distância Euclidiana foi utilizada como medida de similaridade.

- **Descritor de Fourier:** o descritor de Fourier do contorno representa uma imagem como os 126 coeficientes mais significativos obtidos pela aplicação da transformada de Fourier no contorno do objeto de interesse na imagem, utilizando o método descrito em [30, 45]. Novamente, a distância Euclidiana foi utilizada para cálculo da similaridade.

- **Dimensão Fractal Multiescala do Contorno (MS Fractal):** foi utilizado o método descrito em [13] para extrair os valores fractais multiescalas do contorno, com polinômio fractal multiescala de grau 25, gerando um vetor de características com 25 dimensões. Novamente, a distância Euclidiana foi utilizada para medir a similaridade entre duas representações da dimensão fractal multiescala.

- **Beam Angle Statistics (BAS):** o descritor BAS [3] é baseado nos *beams* originados a partir de um *pixel* de contorno. Um *beam* é definido como um conjunto de linhas conectando determinado *pixel* de referência com os demais *pixels* ao longo do contorno. O ângulo entre um par de linhas é calculado em cada *pixel* do contorno. Com isso, o vetor de características é definido pelas estatísticas de primeira, segunda e terceira ordem de todos os ângulos, em um conjunto de sistemas de vizinhanças. A similaridade é medida pelo algoritmo de correspondência ótima de subseqüencias (OCS) [57].

- **Saliências de Segmento (SS):** o algoritmo de extração do descritor SS [16] divide o contorno de um objeto de interesse em um número pré-definido de segmentos de mesmo tamanho, e calcula os valores de saliência desses segmentos. Esses valores são calculados como a diferença entre as áreas de influência externas e internas de cada segmento. Esses valores compõem o vetor de características definido pelo descritor SS. Na implementação utilizada, o contorno foi dividido em 30 segmentos, resultando em um vetor de características com 30 dimensões. O algoritmo OCS [57] foi utilizado para cálculo da similaridade.

4.2 Técnicas de Realimentação de Relevância Utilizadas

Os arcabouços propostos foram comparados com os métodos WD_{heu} [48], WD_{opt} [46] e SVM_{active} [54]. Como mencionado no Capítulo 2, os dois primeiros métodos [46, 48] são baseados na atribuição de pesos. O último [54] utiliza SVM para classificar as imagens em relevantes e irrelevantes. A escolha desses métodos foi motivada por sua utilização como base de comparação em outros trabalhos [9, 21, 41, 46]. É importante observar que não foram utilizadas as implementações dos autores dos métodos. A seguir, esses métodos são descritos em mais detalhes.

4.2.1 Método baseado em pesos: WD_{heu}

A abordagem WD_{heu} [48] define os pesos associados aos descritores e aos *bins* dos vetores de características. O processo de aprendizado consiste em estimar esses pesos heuristicamente. Em cada iteração, o usuário rotula as imagens como *muito relevantes*, *relevantes*, *sem-opinião*, *irrelevantes* e *muito irrelevantes*.

Para o cálculo dos pesos dos descritores, seja L o número de imagens exibidas para o usuário em cada iteração. O peso de cada descritor D_i é atualizado de acordo com o número de imagens relevantes e irrelevantes que D_i é capaz de retornar nas primeiras L posições, quando utilizado sozinho. Dessa forma, quanto maior o número de imagens relevantes retornadas pelo descritor, mais alto será seu peso. Para cada uma das L primeiras imagens I, o peso do descritor aumenta em 3 se I for muito relevante e em 1 se for relevante. Caso contrário, esse peso decrementa em 3 se I for muito irrelevante e por 1 se I for irrelevante. Esses valores de incremento e decremento foram escolhidos experimentalmente [48].

Os pesos associados aos *bins* dos vetores de características são definidos como o inverso do desvio padrão do valor de cada *bin* dos vetores das imagens marcadas como muito relevantes e relevantes. O método WD_{heu} não determina a utilização de uma função de similaridade específica, mas requer que essa função considere os pesos associados a cada *bin* no cálculo da similaridade.

4.2.2 Método baseado em pesos: WD_{opt}

O método WD_{opt} [46] também é baseado na atribuição e estimativa de pesos associados aos descritores e aos *bins* dos vetores de características. Porém, esse método adota um arcabouço de otimização para estimar esses pesos. Esse arcabouço encontra os pesos que minimizam a distância Euclidiana generalizada [46] entre o padrão de consulta e todas as

imagens indicadas como relevantes pelo usuário. A aplicação desse arcabouço resulta em um vetor de pesos, cada um correspondente a um descritor, e uma matriz de pesos que é utilizada no cálculo da distância Euclidiana generalizada entre duas imagens. Essa matriz é calculada a partir da matriz de covariância dos vetores de características das imagens rotuladas como relevantes.

Além da atribuição de pesos, essa abordagem também é baseada no Movimento do Ponto de Consulta Único. O vetor de características do padrão de consulta é definido como o centróide dos pontos que representam as imagens marcadas como relevantes.

4.2.3 Método baseado em Máquinas de Vetores de Suporte: SVM_{active}

A abordagem SVM_{active} [54] utiliza SVM para identificar o conjunto de imagens relevantes existentes na base e retorná-lo ao usuário. Esse método se baseia em aprendizado ativo (*active learning*) para encontrar o hiperplano que separa as imagens relevantes das irrelevantes.

O aprendizado ativo consiste em selecionar para treinamento as imagens mais informativas, ou seja, as mais ambíguas, sendo caracterizadas por aquelas mais difíceis de serem classificadas. O método SVM_{active} considera mais ambíguas as imagens que estão mais próximas do hiperplano de partição. Dessa forma, a cada iteração, essas imagens são retornadas para o usuário rotulá-las, possibilitando que sejam utilizadas no treinamento para se encontrar o próximo hiperplano de partição.

Na última iteração, as imagens mais positivas são retornadas para o usuário. Essas imagens são definidas como as mais distantes no lado positivo do hiperplano de partição.

A biblioteca *libsvm* [7] foi utilizada na implementação desse método. Foi utilizado o *kernel* função de base radial com parâmetros sendo estimados por uma técnica baseada em validação cruzada provida na biblioteca *libsvm*.

4.3 Bases de Imagens

Três bases de imagens foram utilizadas nos experimentos, um conjunto de formas de peixes, a base *MPEG-7 Part B* e um subconjunto da *Corel GALLERY Magic - Stock Photo Library 2*, descritas a seguir.

4.3.1 Base de Formas de Peixes

A base de formas de peixes, que será denominada PEIXES, é composta por 11000 imagens contendo contornos de peixes. Essas imagens foram obtidas a partir de um conjunto origi-

nal com 1100 imagens, disponível em `www.ee.surrey.ac.uk/Research/VSSP/imagedb/demo.html`. Em cada imagem, foram aplicadas operações de rotação e escala, originando 9 outras. Cada imagem do conjunto original juntamente com as nove originadas a partir dela definem uma classe. Dessa forma, a base final contém 11000 imagens, distribuídas em 1100 classes. Exemplos de imagens dessa base são mostrados na Figura 4.1(a).

Nessa base, serão utilizados os descritores Momentos Invariantes, Descritor de Fourier e MS Fractal.

Essa base tem uma característica que lhe é peculiar em relação as demais utilizadas: o tamanho pequeno de cada classe (10) perante o número total de imagens da base (11000). Para cada imagem de consulta, há apenas 9 outras relevantes, para um universo de 10990 irrelevantes, o que representa uma razão de $\frac{9}{10990} = 0,000818926$. Os experimentos com essa base podem avaliar o desempenho dos arcabouços propostos quando o número de imagens relevantes para uma dada consulta é muito pequeno.

4.3.2 Base MPEG-7

A base *MPEG-7 Part B*, que será chamada MPEG7, é a parte principal do *Core Experiment CE-Shape-1* [39]. Possui um total de 1400 imagens de formas variadas, sendo divididas em 70 classes, cada uma contendo 20 imagens. Exemplos de imagens dessa base são apresentados na Figura 4.1(b).

Assim, além dos descritores empregados na base PEIXES, os descritores BAS e SS serão utilizados na base MPEG7. Observe que a função de similaridade definida para o BAS e o SS é a OCS, como apresentado na Seção 4.1.3.

Conforme mencionado na Seção 4.2, a abordagem WD_{heu} necessita utilizar uma função de similaridade que admita a atribuição de pesos para os *bins* dos vetores de características. Porém, não existe uma versão do algoritmo OCS compatível com essa restrição. Dessa forma, a distância Euclidiana com pesos (*Weighted Euclidean Distance*) será empregada nos experimentos com WD_{heu} como função de similaridade para os descritores BAS e SS. O método WD_{opt} também não utilizará o algoritmo OCS para esses descritores, pois determina a utilização da distância Euclidiana generalizada. E, como a abordagem SVM_{active} não faz uso de funções de similaridade, os experimentos nessa base possibilitarão verificar o impacto de não se utilizar as funções de similaridades definidas para os descritores.

4.3.3 Base Corel

A outra base utilizada é um subconjunto da coleção heterogênea de 20000 imagens coloridas que fazem parte da *Corel GALLERY Magic - Stock Photo Library 2*. Esse subconjunto

será referenciado por COREL. O subconjunto COREL é composto por 3906 imagens, divididas em 85 classes. Essas classes possuem tamanhos distintos, variando de 7 até 98. Exemplos de imagens desse subconjunto são mostrados na Figura 4.1(c).

Os três descritores de cor e os dois descritores de textura descritos na Seção 4.1 foram utilizados nessa base.

Como mostrado na Tabela 2.2, vários trabalhos utilizam um subconjunto da base COREL na avaliação dos métodos. Essa base possui uma coleção diversificada de imagens, que se aproxima de um ambiente real em que recuperação de imagens por conteúdo é empregada.

4.4 Medidas de Avaliação

Serão utilizados dois tipos de curva para avaliação de desempenho nos experimentos, *precisão vs. revocação* e *imagens relevantes retornadas vs. iterações*. Além disso, será empregado o *teste de Wilcoxon pareado* para avaliar a significância estatística [11] dos resultados.

A curva *precisão vs. revocação* é um critério de avaliação de desempenho comumente utilizado em Recuperação de Informação e que tem sido empregado para avaliar sistemas de recuperação de imagens por conteúdo.

A *precisão* pode ser definida como o número de imagens relevantes retornadas $R(q)$ em relação ao número total de imagens retornadas $N(q)$ para uma dada consulta q, ou seja

$$Pr(q) = \frac{R(q)}{N(q)} \tag{4.1}$$

A *revocação* é o número de imagens relevantes retornadas $R(q)$ em relação ao número total de imagens relevantes na base para uma dada consulta q, ou seja

$$Re(q) = \frac{R(q)}{M(q)} \tag{4.2}$$

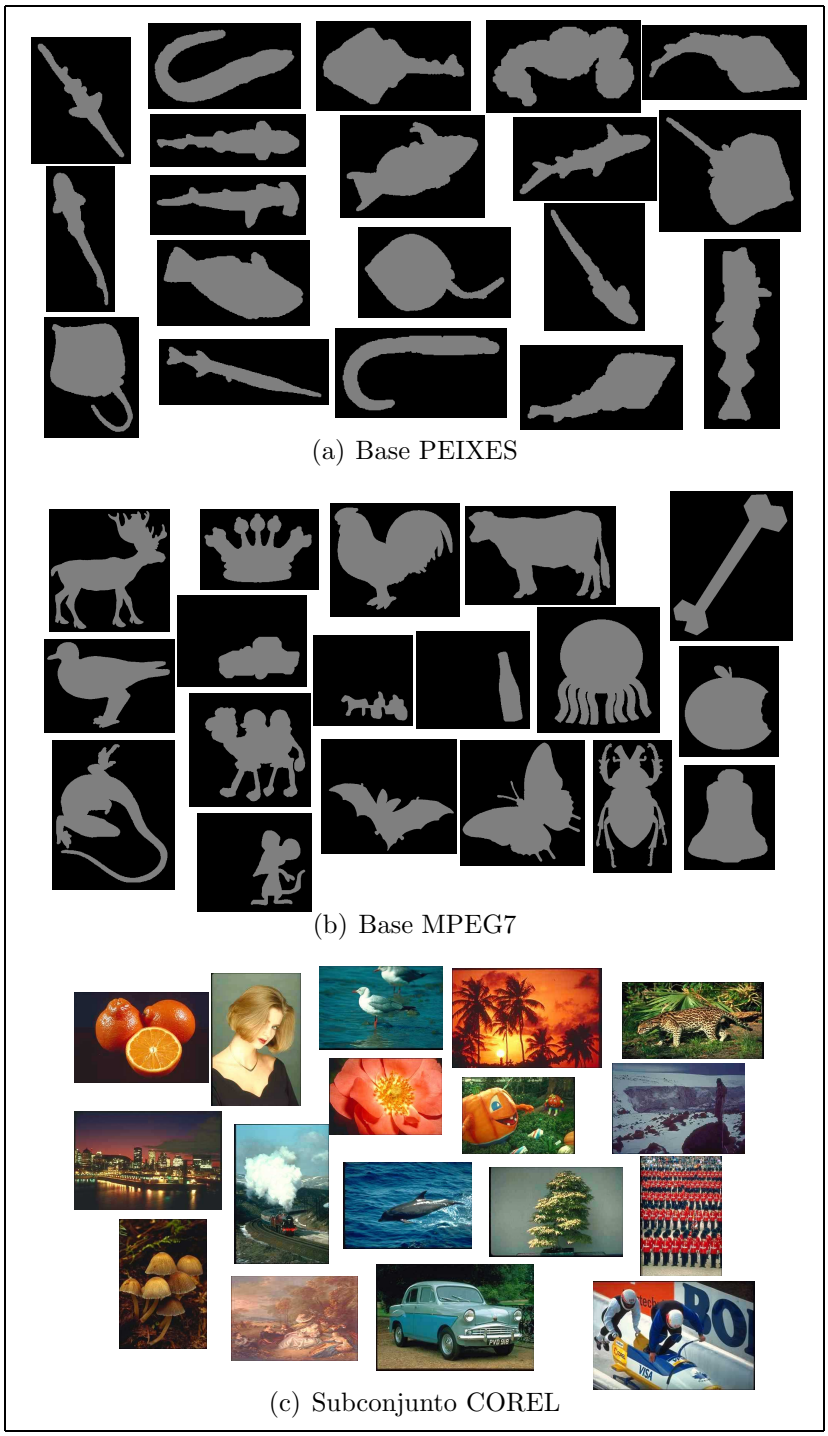
É desejável que tanto a *precisão* quanto a *revocação* sejam altos. Porém, essas duas grandezas são conflitantes. Para aumentar o número de imagens retornadas (e a *revocação*) geralmente é necessário aumentar o número de imagens retornadas e, conseqüentemente, a *precisão* diminui. Dessa forma, a curva *precisão vs. revocação* é geralmente utilizada para mostrar a relação entre essas duas medidas e assim caracterizar o desempenho de um método para recuperação de imagens por conteúdo.

O outro critério de avaliação utilizado é a curva *imagens relevantes retornadas vs. iterações*. Essa curva mostra a percentagem de imagens relevantes exibidas para o usuário

até uma determinada iteração do processo de realimentação de relevância. Essa curva permite avaliar como o número de imagens relevantes retornadas aumenta ao longo das iterações.

Ambas as curvas serão exibidas como a curva média calculada com base em cada consulta individual, ou seja, para cada imagem de consulta I utilizada, calcula-se a curva correspondente (*precisão vs. revocação* ou *imagens relevantes retornadas vs. iterações*) sobre o resultado obtido na recuperação com I. Dessa forma, haverá uma curva para cada imagem de consulta. A média entre essas curvas define a curva que será mostrada na avaliação comparativa das técnicas de realimentação de relevância implementadas. A curva *precisão vs. revocação* será calculada em função dos resultados obtidos na última iteração do processo de realimentação de relevância.

Como pode ser visto, a efetividade dos resultados obtidos será apresentada como médias calculadas a partir dos resultados para cada consulta realizada. Quando essas médias forem utilizadas para comparação entre métodos, surge a necessidade de verificar se as diferenças entre os resultados são significativas do ponto de vista estatístico. Dessa forma, será empregado o teste de *Wilcoxon* [11] pareado. Esse teste foi escolhido porque se adequa às característica dos dados envolvidos nos experimentos: as variáveis não possuem uma distribuição normal, pelo *teste Shapiro-Wilk* [11]; as comparações serão realizadas aos pares; e, como os métodos são comparados a partir do mesmo conjunto de imagens, o teste deve ser pareado [11], ou seja, há pareamento entre as variáveis aleatórias envolvidas no teste. O teste de *Wilcoxon* assim como o teste de normalidade *teste Shapiro-Wilk* foram realizados utilizando-se a ferramenta *R* (`http://www.r-project.org/`).



(a) Base PEIXES

(b) Base MPEG7

(c) Subconjunto COREL

Figura 4.1: Bases de imagens utilizadas nos experimentos.

Capítulo 5

Resultados Experimentais e Análise

Dois conjuntos de experimentos foram realizados. O primeiro, apresentado na Seção 5.1, tem por objetivo determinar os parâmetros do processo evolutivo utilizados nos arcabouços propostos. O segundo conjunto, descrito na Seção 5.2, compara PG^{+} e $PG^{\pm}$ com os métodos mostrados na Seção 4.2, em relação à eficácia e eficiência na recuperação das imagens das bases PEIXES, MPEG7 e COREL.

Nos experimentos, o comportamento do usuário é simulado por computador. Nessa simulação, todas as imagens pertencentes à mesma classe da imagem de consulta são consideradas relevantes. Foram realizadas 10 iterações em cada consulta. Em cada iteração, 40 imagens foram exibidas para o usuário. O primeiro conjunto de imagens retornado para o usuário é baseado na similaridade média calculada por cada descridor empregado, entre a imagem de consulta e as imagens da base, conforme descrito na Seção 3.1.1.

5.1 Implementação dos Arcabouços PG^{+} e $PG^{\pm}$

Os arcabouços propostos foram implementados na linguagem *C*, utilizando-se uma biblioteca para modelagem e resolução de problemas com programação genética, chamada *lilgp* [62].

Para a implementação dos arcabouços propostos é necessário atentar para uma questão inerente à utilização de programação genética: o ajuste dos parâmetros empregados no processo evolutivo. Como apresentado na Seção 2.3, vários parâmetros devem ser definidos, como o método de seleção dos indivíduos, a forma de inicialização da população, altura máxima das árvores que representam os indivíduos, dentre outros. Em geral, esses parâmetros são determinados empiricamente, por meio de experimentos. Porém, muitos desses parâmetros apresentam um forte relacionamento entre si [23], impossibilitando o ajuste de cada um individualmente e testar todas as combinações possíveis é inviável. Além disso, os arcabouços propostos possuem um requisito adicional que é o baixo tempo

de execução.

Dessa forma, esses parâmetros foram determinados por meio de uma série de experimentos em que se procurou agrupar os parâmetros que possuíam maior dependência entre si. Mais especificamente, os parâmetros a serem estimados foram divididos em cinco grupos:

- **Grupo 1:** conjunto de funções do indivíduo;
- **Grupo 2:** tamanho da população, número de gerações e altura da árvore;
- **Grupo 3:** taxas associadas aos operadores genéticos;
- **Grupo 4:** tamanho do conjunto de treinamento;
- **Grupo 5:** limitante para votação.

Primeiro, foi definido um conjunto inicial de valores para todos os parâmetros por meio de experimentos preliminares.Em seguida, foram determinados os valores que cada um dos parâmetros poderia assumir. Feito isso, para cada grupo, todas as combinações entre esses valores dos parâmetros do grupo foram testadas com os demais parâmetros fixados. Assim, o conjunto inicial de valores foi refinado, testando todas as combinações de valores entre os parâmetros de um mesmo grupo. Além dos experimentos para definir os parâmetros utilizados pelo algoritmo de aprendizado baseado em programação genética, foi realizado um outro para determinar um parâmetro introduzido pelos arcabouços propostos: o limitante para votação. Os valores iniciais atribuídos aos parâmetros são apresentados na Tabela 5.1.

Na Tabela 5.1, o parâmetro *tamanho da população* refere-se ao número de indivíduos da população. O *número de gerações* determina o número máximo de gerações utilizado. Caso algum indivíduo alcance o valor de adequação máximo, o processo de evolução se encerra. O parâmetro *população inicial* diz respeito ao método de inicialização da população inicial utilizado. A *profundidade inicial* determina a profundidade máxima das árvores que representam os indivíduos da população inicial. Já a *altura máxima* define a altura máxima que uma árvore poderá alcançar durante o processo de evolução. O *método de seleção* refere-se ao método de seleção utilizado. As *taxas de cruzamento* e *mutação* determinam a probabilidade de cada um desses operadores serem utilizados para originar um novo indivíduo. O *tamanho do conjunto de treinamento* diz respeito ao número de imagens utilizadas no treinamento do algoritmo de aprendizado. Foram utilizados dois valores distintos, um para a PEIXES e outro para a base COREL. O *limitante de votação* define o limitante para seleção dos indivíduos que serão utilizados para ordenar a base. E, finalmente, o *conjunto de funções* refere-se às funções que serão utilizadas como nodos

Tabela 5.1: Valores iniciais dos parâmetros.

Parâmetro	**Valor**
tamanho da população	30
número de gerações	10
população inicial	*half-and-half*
profundidade inicial	2 – 5
altura máxima	10
método de seleção	torneio (tamanho 2)
taxa de cruzamento	0.8
taxa de mutação	0.2
tamanho do conjunto de treinamento	55 (PEIXES) e 80(COREL)
limitante para votação	1
conjunto de funções	$+, *, /(protegida), \sqrt{}$

internos dos indivíduos. Observe que a função de divisão (/) é protegida. Nesse função, se o divisor for zero, a função retorna zero.

Os experimentos para determinação dos parâmetros foram realizados para os dois arcabouços propostos, PG^{+} e $PG^{\pm}$, ambos sobre duas bases, PEIXES e COREL. Ao fim do processo, os valores dos parâmetros que apresentaram melhor desempenho nessas bases, foram testados na base MPEG7. Dessa forma, deseja-se avaliar se os melhores parâmetros obtidos em uma base apresentam bons resultados em outras.

A seguir serão caracterizados os experimentos sobre cada um dos grupos de parâmetros definidos.

5.1.1 Grupo 1 – Conjunto de funções

O objetivo dos experimentos sobre o conjunto de funções é determinar quais funções serão utilizadas como nodos internos dos indivíduos no processo de aprendizado. O conjunto de funções utilizadas foi $\{+, *, /(protegida), \sqrt{}\}$. Esse conjunto é baseado no conjunto definido em [14,15]. Foram testadas 4 variações desse conjunto: $\{+, *\}$, $\{+, *, /(protegida)\}$, $\{+, *, \sqrt{}\}$ e o conjunto completo $\{+, *, /(protegida), \sqrt{}\}$.

Para cada uma dessas variações, 5 diferentes combinações de valores para os parâmetros *tamanho da população*, *número de gerações* e *profundidade máxima*. Essas cinco combinações foram utilizadas para evitar que o conjunto inicial de parâmetros não privilegiasse um determinado subconjunto de funções. Para os demais parâmetros, foram utilizados os valores mostrados na Tabela 5.1.

Os subconjuntos selecionados são mostrados na Tabela 5.2. Essa tabela apresenta o

subconjunto de funções utilizado em cada base e por cada arcabouço. Esses subconjuntos são utilizados nos experimentos para determinação dos parâmetros dos Grupos 2, 3, 4, e 5.

Tabela 5.2: Subconjunto de funções selecionadas.

Base/Arcabouço	PG^{+}	$PG^{\pm}$
PEIXES	$\{+, *, /(protegida), \sqrt{}\}$	$\{+, *\}$
COREL	$\{+, *, /(protegida), \sqrt{}\}$	$\{+, *, \sqrt{}\}$

5.1.2 Grupo 2 – Tamanho da População, Número de Gerações e Altura Máxima

O objetivo desse experimento é definir o tamanho da população, o número de gerações e a altura máxima das árvores durante o processo de evolução. O tamanho da população e o número de gerações definem o número de indivíduos que serão avaliados no processo de busca e a altura máxima das árvores determina o número máximo de indivíduos possíveis, para um dado conjunto de terminais e de funções. Assim, acredita-se que exista uma relação entre esses parâmetros porque, provavelmente, quanto maior o número de indivíduos possíveis, mais indivíduos precisam ser avaliados.

A Tabela 5.3 mostra os valores utilizados para cada parâmetro neste experimento. Foram testadas todas as combinações entre esses valores.

Tabela 5.3: Valores testados para os parâmetros do Grupo 2.

Parâmetro	**Valores**
tamanho da população	$30, 60, 90$
número de gerações	$10, 15, 20$
altura máxima	$5, 15, 25$

É importante observar que esses parâmetros têm um alto impacto no tempo de execução do algoritmo de aprendizado. A Tabela 5.4 ilustra esse fato. Essa tabela apresenta o tempo médio gasto em cada iteração do arcabouço PG^{+} na base COREL. A primeira linha apresenta o tempo para a combinação com os menores valores testados para cada parâmetro. A segunda linha mostra os tempos quando os segundos maiores valores foram testados. Por fim, a última linha mostra o tempo quando os valores mais altos são utilizados. Como

pode ser visto, o tempo por iteração da combinação $(30, 10, 5)$ é aproximadamente 5 vezes mais rápido que $(60, 15, 15)$ e 15 vezes mais rápido que $(90, 20, 25)$.

Tabela 5.4: Tempo médio de cada iteração para diferentes combinações de parâmetros do Grupo 2 na base COREL com PG^+.

(tamanho da população, número de gerações, profundidade máxima)	**Tempo (segundos)**
$(30, 10, 5)$	0.57
$(60, 15, 15)$	2.97
$(90, 20, 25)$	8.62

Devido a questão do tempo de execução, os valores dos parâmetros *tamanho da população*, *número de gerações* e a *altura máxima* foram limitados a 90, 20, 25, respectivamente. Os valores selecionados para os parâmetros do Grupo 2 são apresentados na Tabela 5.5. Nessa tabela, os valores dos parâmetros são indicados por triplas em que o primeiro valor refere-se ao *tamanho da população*, o segundo, ao *número de gerações* e o terceiro define a *altura máxima*. Esses valores são utilizados nos experimentos dos Grupos 3, 4 e 5.

Tabela 5.5: Valores selecionados para os parâmetros do Grupo 2.

Base/Arcabouço	PG^+	$PG^{\pm}$
PEIXES	$(60, 10, 15)$	$(30, 10, 5)$
COREL	$(60, 20, 5)$	$(60, 10, 15)$

5.1.3 Grupo 3 – Taxas Associadas aos Operadores Genéticos

O objetivo deste experimentos é determinar as taxas associadas aos operadores genéticos de *reprodução*, *recombinação*, *mutação* que serão referenciados por t_{rp}, t_{rc} e t_{mt}, respectivamente. Essas taxas definem a probabilidade de cada operador genético ser empregado para originar um novo indivíduo.

A Tabela 5.6 mostra todas as combinações testadas para os parâmetros do Grupo 3. Observe que nas combinações da primeira coluna, a taxa associada ao operador de reprodução é zero. Nos experimentos com essas combinações, o operador de reprodução não foi utilizado.

Tabela 5.6: Combinações testadas para os parâmetros do Grupo 3.

(t_{rp}, t_{rc}, t_{mt})		
$(0, 0.5, 0.5)$	$(0.05, 0.5, 0.45)$	$(0.1, 0.4, 0.5)$
$(0, 0.6, 0.4)$	$(0.05, 0.6, 0.35)$	$(0.1, 0.5, 0.4)$
$(0, 0.7, 0.3)$	$(0.05, 0.7, 0.25)$	$(0.1, 0.6, 0.3)$
$(0, 0.8, 0.2)$	$(0.05, 0.8, 0.15)$	$(0.1, 0.7, 0.2)$
$(0, 0.9, 0.1)$	$(0.05, 0.9, 0.05)$	$(0.1, 0.8, 0.1)$

Neste experimento, os valores para os parâmetros do Grupo 3 que apresentaram os melhores resultados e foram escolhidos são apresentados na Tabela 5.7. O primeiro, o segundo e o terceiro valor de cada tripla referem-se a t_{rp}, t_{rc} e t_{mt}, respectivamente. Os valores escolhidos serão utilizados nos experimentos dos Grupos 4 e 5.

Observe que em nenhuma das combinações selecionadas o operador de reprodução foi utilizado. Por meio de novos experimentos, percebeu-se que quando o operador de reprodução era utilizado, a diversidade da população caía muito rápido, ou seja, as árvores que representavam os indivíduos ficavam muito parecidas, ou até mesmo iguais. Com isso, o algoritmo de aprendizado perde mais rápido sua capacidade de exploração do espaço de busca.

Tabela 5.7: Valores selecionados para os parâmetros do Grupo 3.

Base/Arcabouço	PG^{+}	$PG^{\pm}$
PEIXES	$(0, 0.8, 0.2)$	$(0, 0.8, 0.2)$
COREL	$(0, 0.8, 0.2)$	$(0, 0.8, 0.2)$

5.1.4 Grupo 4 – Tamanho do Conjunto de Treinamento

Este experimento tem como objetivo determinar o tamanho do conjunto de treinamento a ser utilizado no processo de aprendizado. Foram testados os seguintes valores: 55, 80, 105 e 130.

O tamanho do conjunto de treinamento é outro parâmetro que influencia o tempo de execução do algoritmo de aprendizado, como pode ser visto na Tabela 5.8. Essa tabela apresenta o tempo médio de cada iteração para os tamanhos do conjunto de treinamento avaliados neste experimento, utilizando-se o arcabouço PG^{+} na base COREL. Como pode

ser visto nessa tabela, o tempo de cada iteração é diretamente proporcional ao tamanho do conjunto de treinamento.

Tabela 5.8: Tempo médio de cada iteração para diferentes tamanhos do conjunto de treinamento na base COREL com PG^+.

tamanho do conjunto de treinamento	**Tempo (segundos)**
55	1.23
80	1.79
105	2.36
130	2.96

Os valores para tamanho do conjunto de treinamento que apresentaram melhores resultados nos experimentos e foram selecionados são exibidos na Tabela 5.9. Esses valores foram utilizados nos experimentos com o Grupo 5.

Tabela 5.9: Valores selecionados para o tamanho do conjunto de treinamento.

Base/Arcabouço	PG^+	$PG^{\pm}$
PEIXES	55	130
COREL	80	80

5.1.5 Grupo 5 - Limitante para Votação

O objetivo deste experimento é definir o valor do limitante para votação a ser utilizado. Esse parâmetro foi definido como α na Equação 3.6 e determina a razão mínima entre o valor da adequação de um determinado indivíduo δ_i e a adequação do melhor indivíduo, para que δ_i seja selecionado para ordenar as imagens da base. Os valores testados foram 0.99, 0.993, 0.996, 0.999 e 1. Por questões de tempo de execução, o número de indivíduos votantes máximo foi fixado como 4.

A Tabela 5.10 apresenta os valores para limitante de votação selecionados.

5.1.6 Valores escolhidos para os parâmetros

Realizados todos os experimentos, os valores dos parâmetros escolhidos para o arcabouço PG^+ são apresentados na Tabela 5.11 e para o arcabouço $PG^{\pm}$ são mostrados na Tabela 5.12.

Tabela 5.10: Valores selecionados para limitante de votação.

Base/Arcabouço	PG^{+}	$PG^{\pm}$
PEIXES	1	0.999
COREL	0.999	1

Tabela 5.11: Valores dos parâmetros escolhidos para o arcabouço PG^{+}.

Parâmetro	**Base PEIXES**	**Base COREL**
tamanho da população	60	60
número de gerações	10	20
população inicial	*half-and-half*	*half-and-half*
profundidade inicial	2 – 5	2 – 5
altura máxima máxima	15	5
método de seleção	torneio (tamanho 2)	torneio (tamanho 2)
taxa de cruzamento	0.8	0.8
taxa de mutação	0.2	0.2
tamanho do conjunto de treinamento	55	80
limitante para votação	1	0.999
conjunto de funções	$+, *, /(protegida), \sqrt{}$	$+, *, /(protegida), \sqrt{}$

5.2 Comparação entre Métodos

Para a obtenção dos resultados apresentados nesta seção, foram realizados experimentos nas três bases descritas na Seção 4.3. Na base PEIXES, foi escolhida aleatoriamente 1 imagem de consulta em cada classe, totalizando, 1100 imagens de consulta. Foram definidas 140 imagens de consultas para a base MPEG7, sendo escolhidas aleatoriamente 2 imagens por classe. Na base COREL também foram definidas 2 imagens por classe, sendo escolhidas no total 170 imagens de consulta. Foi utilizada uma máquina com processador Pentium 4 3.0 GHz com 2 GB de memória RAM.

Os gráficos com as curvas *precisão vs. revocação* apresentam, além das curvas associadas a cada método, uma outra referente à recuperação de imagens sem realimentação de relevância, denominada *SRR*. Para o cálculo dessa curva, são simuladas 10 iterações sem a utilização de nenhum mecanismo de aprendizado, apenas retornando novas imagens a cada iteração considerando a ordenação inicial da base obtida pelo processo apresentado na Seção 3.1.1. Dessa forma será possível observar o ganho referente à utilização de mecanismos de realimentação de relevância em recuperação de imagens por conteúdo. Esse

Tabela 5.12: Valores dos parâmetros escolhidos para o arcabouço $PG^{\pm}$.

Parâmetro	**Base PEIXES**	**Base COREL**
tamanho da população	30	60
número de gerações	10	10
população inicial	*half-and-half*	*half-and-half*
profundidade inicial	2 – 5	2 – 5
altura máxima máxima	5	15
método de seleção	torneio (tamanho 2)	torneio (tamanho 2)
taxa de cruzamento	0.8	0.8
taxa de mutação	0.2	0.2
tamanho do conjunto de treinamento	130	80
limitante para votação	0.999	1
conjunto de funções	$+, *, \sqrt{}$	$+, *$

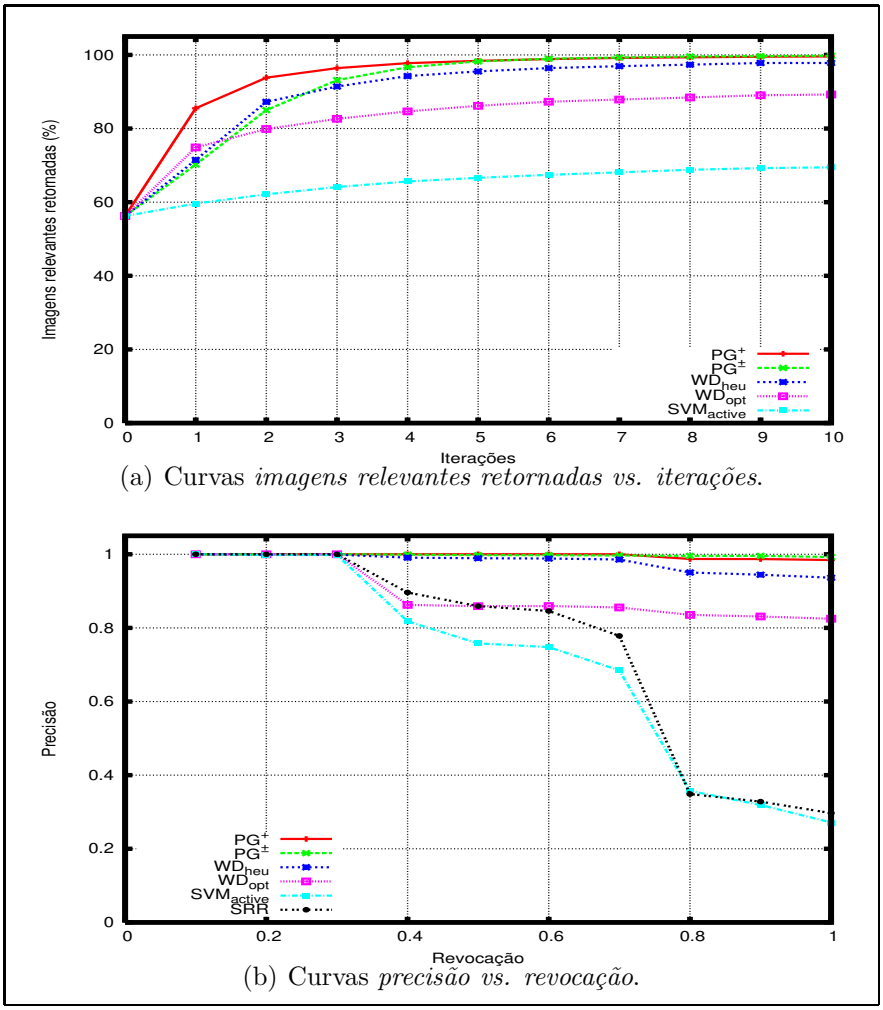
ganho também é ilustrado nos gráficos com as curvas *imagens retornadas vs.iterações*, em que cada ponto com coordenada zero no eixo das iterações indica a porcentagem de imagens relevantes retornadas no conjunto inicial.

Nas tabelas com os resultados dos testes de significância estatística, cada valor apresentado indica a probabilidade de que a diferença entre duas médias não seja significativa. Em geral, admite-se que as médias são diferentes se essa probabilidade for inferior a 0.05 ou 0.01 [11]. Nessas tabelas, o símbolo + é adicionado ao final dos valores de probabilidade que indicam diferenças significativas entre as médias testadas.

5.2.1 Resultados na Base PEIXES

A Figura 5.1 apresenta as curvas *imagens relevantes retornadas vs. iterações* (Figura 5.1(a)) e *precisão vs. revocação* (Figura 5.1(b)) para a comparação entre os métodos na base PEIXES.

Pela análise da Figura 5.1(a) pode ser observado que o arcabouço PG^{+} apresenta um desempenho superior aos demais métodos desde a primeira iteração na base PEIXES. Por outro lado, $PG^{\pm}$ encontra menos imagens relevantes do que WD_{opt} na primeira iteração, e menos que WD_{heu} até a segunda. Porém, a partir da terceira iteração, $PG^{\pm}$ passa a retornar mais imagens relevantes do que os métodos utilizados como base de comparação. É importante observar que, a partir da oitava iteração, a porcentagem de imagens relevantes que os arcabouços propostos encontram é próxima a 100%, ou seja, a maioria das imagens relevantes são retornadas, em todas as consultas executadas. Um outro aspecto que deve ser ressaltado é o desempenho ruim da técnica SVM_{active}. Apesar dessa técnica



(a) Curvas *imagens relevantes retornadas vs. iterações.*

(b) Curvas *precisão vs. revocação.*

Figura 5.1: Curvas de comparação entre os métodos na base PEIXES.

apresentar bons resultados nas outras bases, na base PEIXES seu desempenho foi inferior aos demais métodos. Acredita-se que esse fato se deve ao pequeno número de imagens relevantes existentes para cada consulta realizada nessa base, conforme mencionado na Seção 4.3.1. Esse número de imagens relevantes não foi suficiente para possibilitar o aprendizado de SVM_{active}.

A Tabela 5.13 mostra o resultado da aplicação do teste de *Wilcoxon* nos dados mostrados na Figura 5.1(a), comparando as curvas referentes aos arcabouços propostos, com às relativas aos outros métodos. Como pode ser observado nessa tabela, as diferenças entre as curvas são significativas para todos os pontos.

As curvas *precisão vs. revocação* exibidas na Figura 5.1(b) mostram que tanto PG^+

Tabela 5.13: Resultados do teste de *Wilcoxon* na base PEIXES sobre os dados apresentados na Figura 5.1(a).

	PG^{+}			$PG^{\pm}$		
Iteração	WD_{heu}	WD_{opt}	SVM_{active}	WD_{heu}	WD_{opt}	SVM_{active}
1	$10^{-16}+$	$10^{-16}+$	$10^{-16}+$	0.01440+	$10^{-8}+$	$10^{-16}+$
2	$10^{-16}+$	$10^{-16}+$	$10^{-16}+$	0.00015+	$10^{-7}+$	$10^{-16}+$
3	$10^{-16}+$	$10^{-16}+$	$10^{-16}+$	0.00087+	$10^{-16}+$	$10^{-16}+$
4	$10^{-13}+$	$10^{-16}+$	$10^{-16}+$	$10^{-7}+$	$10^{-16}+$	$10^{-16}+$
5	$10^{-11}+$	$10^{-16}+$	$10^{-16}+$	$10^{-12}+$	$10^{-16}+$	$10^{-16}+$
6	$10^{-10}+$	$10^{-16}+$	$10^{-16}+$	$10^{-12}+$	$10^{-16}+$	$10^{-16}+$
7	$10^{-10}+$	$10^{-16}+$	$10^{-16}+$	$10^{-12}+$	$10^{-16}+$	$10^{-16}+$
8	$10^{-8}+$	$10^{-16}+$	$10^{-16}+$	$10^{-12}+$	$10^{-16}+$	$10^{-16}+$
9	$10^{-8}+$	$10^{-16}+$	$10^{-16}+$	$10^{-11}+$	$10^{-16}+$	$10^{-16}+$
10	$10^{-9}+$	$10^{-16}+$	$10^{-16}+$	$10^{-12}+$	$10^{-16}+$	$10^{-16}+$

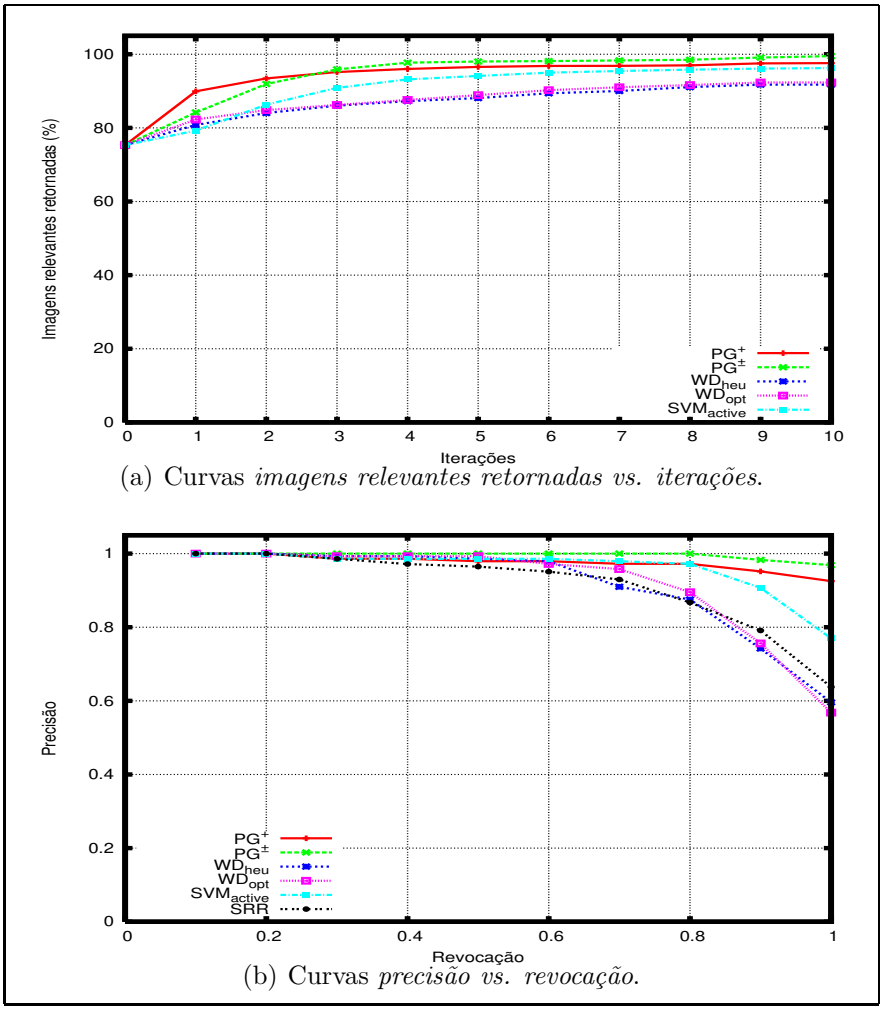
quanto $PG^{\pm}$ apresentam precisão próxima a 1 na base PEIXES para todos os pontos. O método WD_{heu} apresenta um resultado próximo aos métodos propostos até o valor de *revocação* igual a 0.5. A partir desse ponto, os valores de *precisão* de PG^{+} e $PG^{\pm}$ são superiores. A Tabela 5.14 reforça essa constatação. Essa tabela mostra o resultado da aplicação do teste de *Wilcoxon* nos dados mostrados na Figura 5.1(b). Observe que a partir do valor de *revocação* 0.5 as diferenças entre os valores de *precisão* dos arcabouços propostos o os valores dos outros métodos são significativas.

Tabela 5.14: Resultados do teste de *Wilcoxon* na base PEIXES sobre os dados apresentados na Figura 5.1(b).

	PG^{+}			$PG^{\pm}$		
Revocação	WD_{heu}	WD_{opt}	SVM_{active}	WD_{heu}	WD_{opt}	SVM_{active}
0.1	1	1	1	1	1	1
0.2	1	1	0.3711	1	1	0.3711
0.3	1	1	0.3711	1	1	0.3711
0.4	0.05922	$10^{-16}+$	$10^{-16}+$	0.05593	$10^{-16}+$	$10^{-16}+$
0.5	0.02526+	$10^{-16}+$	$10^{-16}+$	0.02382+	$10^{-16}+$	$10^{-16}+$
0.6	0.01662+	$10^{-16}+$	$10^{-16}+$	0.01579+	$10^{-16}+$	$10^{-16}+$
0.7	0.00482+	$10^{-16}+$	$10^{-16}+$	0.006931+	$10^{-16}+$	$10^{-16}+$
0.8	$10^{-6}+$	$10^{-16}+$	$10^{-16}+$	$10^{-9}+$	$10^{-16}+$	$10^{-16}+$
0.9	$10^{-7}+$	$10^{-16}+$	$10^{-16}+$	$10^{-11}+$	$10^{-16}+$	$10^{-16}+$
1.0	$10^{-8}+$	$10^{-16}+$	$10^{-16}+$	$10^{-12}+$	$10^{-16}+$	$10^{-16}+$

5.2.2 Resultados na Base MPEG7

A Figura 5.2 mostra as curvas *imagens relevantes retornadas vs. iterações* (Figura 5.2(a)) e *precisão vs. revocação* (Figura 5.2(b)) na base MPEG7.



(a) Curvas *imagens relevantes retornadas vs. iterações.*

(b) Curvas *precisão vs. revocação.*

Figura 5.2: Curvas de comparação entre os métodos na base MPEG7.

Como apresentado na Figura 5.2(a), a porcentagem de imagens relevantes encontradas pelos arcabouços PG^{+} e $PG^{\pm}$ é superior a dos demais métodos desde a primeira iteração, na base MPEG7. Nessa base, o método SVM_{active} apresenta um resultado próximo a PG^{+} a partir da sexta iteração. Observe que, a partir da nona iteração, novamente $PG^{\pm}$ encontra a maioria das imagens relevantes para todas as consultas realizadas. A Tabela 5.15 mostra o resultado da aplicação do teste de *Wilcoxon* sobre os dados apresentados na Figura 5.2(a). Como pode ser observado nessa tabela, as diferenças entre

as curvas *imagens relevantes retornadas vs. iterações* na base MPEG7 são significativas para todos os pontos.

Tabela 5.15: Resultados do teste de *Wilcoxon* na base MPEG7 sobre os dados apresentados na Figura 5.2(a).

	PG^{+}			$PG^{\pm}$		
Iteração	WD_{heu}	WD_{opt}	SVM_{active}	WD_{heu}	WD_{opt}	SVM_{active}
1	$10^{-12}+$	$10^{-12}+$	$10^{-13}+$	$0.0027+$	$0.0413+$	$10^{-6}+$
2	$10^{-12}+$	$10^{-11}+$	$10^{-10}+$	$10^{-10}+$	$10^{-10}+$	$10^{-8}+$
3	$10^{-11}+$	$10^{-10}+$	$10^{-7}+$	$10^{-11}+$	$10^{-12}+$	$10^{-8}+$
4	$10^{-11}+$	$10^{-9}+$	$10^{-6}+$	$10^{-11}+$	$10^{-11}+$	$10^{-8}+$
5	$10^{-9}+$	$10^{-8}+$	$0.0001+$	$10^{-11}+$	$10^{-10}+$	$10^{-7}+$
6	$10^{-8}+$	$10^{-7}+$	$0.0005+$	$10^{-10}+$	$10^{-9}+$	$10^{-6}+$
7	$10^{-8}+$	$10^{-7}+$	$0.0025+$	$10^{-9}+$	$10^{-9}+$	$10^{-6}+$
8	$10^{-7}+$	$10^{-7}+$	$0.0098+$	$10^{-9}+$	$10^{-8}+$	$10^{-6}+$
9	$10^{-7}+$	$10^{-7}+$	$0.0021+$	$10^{-9}+$	$10^{-9}+$	$10^{-6}+$
10	$10^{-7}+$	$10^{-7}+$	$0.0015+$	$10^{-10}+$	$10^{-11}+$	$10^{-5}+$

As curvas *precisão vs. revocação* mostram que na última iteração dos experimentos executados na base MEPG7, todos os métodos apresentam resultados muito próximos até o valor de revocação 0.7. A partir desse ponto, os arcabouços propostos e o método SVM_{active} apresentam valores de precisão superiores aos demais, sendo que, a partir do valor de revocação 0.9, PG^{+} e $PG^{\pm}$ passam a superar SVM_{active}. O teste de *Wilcoxon* apresentado na Tabela 5.16 confirma essa informação. Como pode ser observado, as diferença entre os valores de precisão dos arcabouços propostos e se o dos métodos WD_{heu} e WD_{opt} se tornam significativas a partir do valor de revocação 0.7 e 0.8, respectivamente. Em relação a SVM_{active}, essa diferença passa a ser significativa a partir do valor de revocação 0.9.

5.2.3 Resultados na Base COREL

A Figura 5.3 apresenta as curvas *imagens relevantes retornadas vs. iterações* (Figura 5.3(a)) e *precisão vs. revocação* (Figura 5.3(b)) para a comparação entre os métodos na base COREL.

A Figura 5.3(a) mostra que o arcabouço PG^{+} apresenta um desempenho superior aos demais métodos nas duas primeiras iterações. A partir da quarta iteração, SVM_{active} passa a exibir mais imagens relevantes que $PG+$. Já o arcabouço $PG^{\pm}$ apresenta um resultado semelhante a WD_{opt} e SVM_{active} na primeira iteração. A partir da segunda, $PG^{\pm}$ e SVM_{active} superam WD_{opt}. Observe que $PG^{\pm}$ e SVM_{active} apresentam desempenhos semelhantes, sendo que, na terceira iteração, SVM_{active} se sobressai e, a partir da quinta

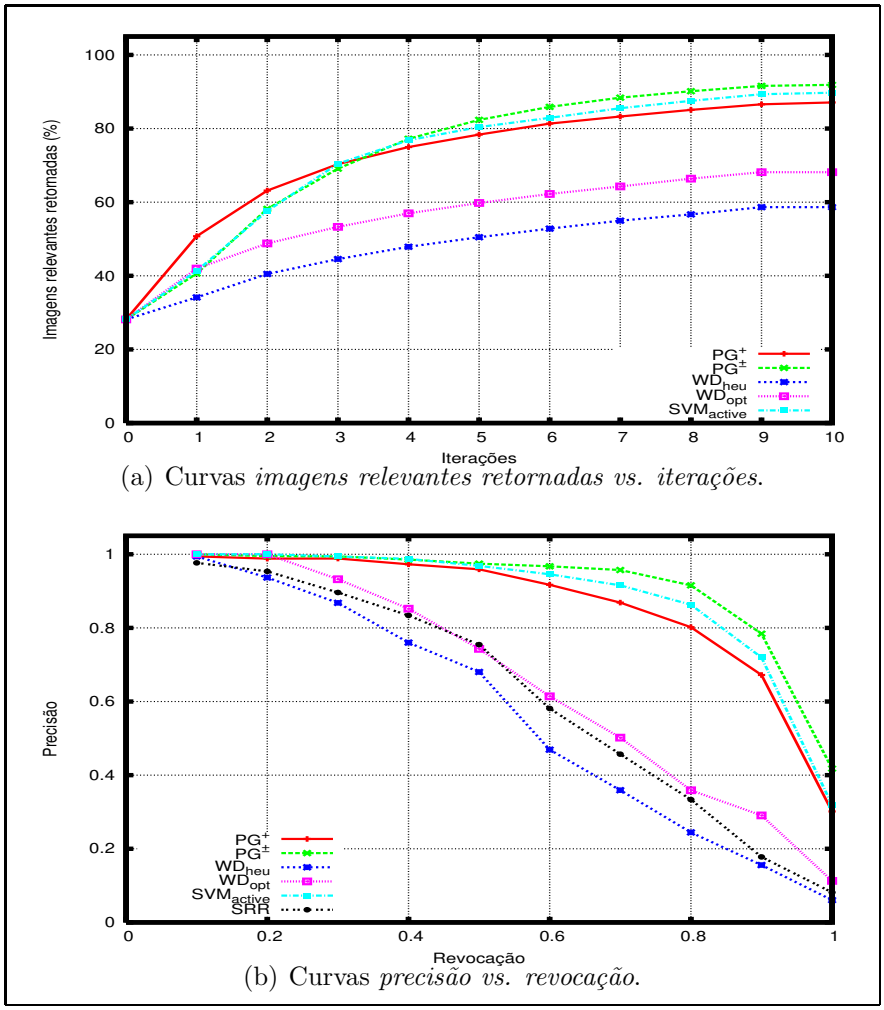
Tabela 5.16: Resultados do teste de *Wilcoxon* na base MPEG7 sobre os dados apresentados na Figura 5.2(b).

	PG^{+}			$PG^{\pm}$		
Revocação	WD_{heu}	WD_{opt}	SVM_{active}	WD_{heu}	WD_{opt}	SVM_{active}
0.1	1	1	1	1	1	1
0.2	1	1	1	1	1	1
0.3	1	1	0.3711	1	1	0.3711
0.4	1	1	0.3711	1	1	0.3711
0.5	0.7874	0.8551	1	0.3711	1	0.3711
0.6	0.5294	0.2041	1	0.1814	0.0975	0.3711
0.7	0.01347+	0.1846	0.7874	0.0016+	0.0350+	0.1814
0.8	0.00158+	0.0089+	1	0.0002+	0.0007+	0.1003
0.9	10^{-7}+	10^{-5}+	0.0049+	10^{-7}+	10^{-7}+	0.00161+
1.0	10^{-11}+	10^{-11}+	10^{-6}+	10^{-11}+	10^{-12}+	10^{-7}+

iteração, $PG^{\pm}$ é ligeiramente superior. A Tabela 5.17 exibe o resultado da aplicação do teste de *Wilcoxon* para as curvas *imagens relevantes retornadas vs. iterações* apresentadas na Figura 5.3(a). Com essa tabela pode ser observado que as curvas associadas aos arcabouços propostos possuem diferenças significativas em relação às curvas referentes aos métodos WD_{opt} e WD_{heu} para todos os pontos. Porém, em relação ao método SVM_{active}, essas diferenças são bem menos significativas. De fato, $PG^{\pm}$ somente apresenta um desempenho superior significativo, a partir da nona iteração. Ao contrário do que é mostrado na Figura 5.3(a), apenas a partir da sétima iteração a superioridade do método SVM_{active} em relação a PG^{+} é significativa.

Tabela 5.17: Resultados do teste de *Wilcoxon* na base COREL sobre os dados apresentados na Figura 5.3(a).

	PG^{+}			$PG^{\pm}$		
Iteração	WD_{heu}	WD_{opt}	SVM_{active}	WD_{heu}	WD_{opt}	SVM_{active}
1	10^{-16}+	10^{-16}+	10^{-11}+	10^{-10}+	0.0412+	0.6283
2	10^{-16}+	10^{-16}+	10^{-5}+	10^{-16}+	10^{-9}+	0.9065
3	10^{-16}+	10^{-16}+	0.8494	10^{-16}+	10^{-16}+	0.0298+
4	10^{-16}+	10^{-16}+	0.0728	10^{-16}+	10^{-16}+	0.2731
5	10^{-16}+	10^{-16}+	0.0551	10^{-16}+	10^{-16}+	0.9984
6	10^{-16}+	10^{-16}+	0.1420	10^{-16}+	10^{-16}+	0.1777
7	10^{-16}+	10^{-16}+	0.0263+	10^{-16}+	10^{-16}+	0.0874
8	10^{-16}+	10^{-16}+	0.0252+	10^{-16}+	10^{-16}+	0.0546
9	10^{-16}+	10^{-16}+	0.0200+	10^{-16}+	10^{-16}+	0.0335+
10	10^{-16}+	10^{-16}+	0.0402+	10^{-16}+	10^{-16}+	0.0348+



(a) Curvas *imagens relevantes retornadas vs. iterações.*

(b) Curvas *precisão vs. revocação.*

Figura 5.3: Curvas de comparação entre os métodos na base COREL.

As curvas *precisão vs. revocação* apresentadas na Figura 5.3(b) mostram que os arcabouços propostos, juntamente com o SVM_{active} apresentam valores de precisão superiores aos demais métodos na última iteração. Como pode ser observado, SVM_{active} é superior a PG^{+} a partir do valor de revocação 0.4. Já $PG^{\pm}$ apresenta resultados semelhantes a SVM_{active} até revocação igual a 0.5. A partir desse ponto, $PG^{\pm}$ apresenta valores de precisão superiores aos demais métodos. A Tabela 5.18 exibe o resultado da aplicação do teste de *Wilcoxon* nos dados da Figura 5.3(b) indicando se as diferenças entre as curvas associadas aos arcabouços propostos e as relativas aos demais métodos são significativas. Essa tabela mostra que essas diferenças são significativas em relação às curvas associadas a WD_{heu} e WD_{opt} a partir dos valores de revocação 0.2 e 0.3, respectivamente. No que

diz respeito ao método SVM_{active}, as diferenças são significativas a partir do valor de revocação 0.7, com exceção do ponto 0.9 em relação à PG^{+}.

Tabela 5.18: Resultados do teste de *Wilcoxon* na base COREL sobre os dados apresentados na Figura 5.3(b).

	PG^{+}			$PG^{\pm}$		
Revocação	WD_{heu}	WD_{opt}	SVM_{active}	WD_{heu}	WD_{opt}	SVM_{active}
0.1	1	0.7893	1	1	0.3711	1
0.2	0.0087+	1	0.3711	0.0032+	0.2012	1
0.3	10^{-5}+	0.0302+	0.4227	10^{-5}+	0.0064+	1
0.4	10^{-8}+	0.0002+	0.3621	10^{-8}+	0.0001+	0.8551
0.5	10^{-11}+	10^{-7}+	0.5760	10^{-11}+	10^{-8}+	0.6101
0.6	10^{-16}+	10^{-11}+	0.2196	10^{-16}+	10^{-11}+	0.2347
0.7	10^{-16}+	10^{-14}+	0.0488+	10^{-16}+	10^{-15}+	0.0121+
0.8	10^{-16}+	10^{-16}+	0.0638	10^{-16}+	10^{-16}+	0.0112+
0.9	10^{-16}+	10^{-16}+	0.6396	10^{-16}+	10^{-16}+	0.0046+
1.0	10^{-16}+	10^{-16}+	0.00723+	10^{-16}+	10^{-16}+	10^{-7}+

5.2.4 Tempos de Execução

Uma questão importante a respeito da utilização de programação genética é o tempo de execução. Em geral, em aplicações que utilizam essa técnica de aprendizado, o processo de evolução necessita de um longo tempo de execução. Isso pode ser um problema para a utilização de programação genética em mecanismos de realimentação de relevância pois requerem um baixo tempo de execução para cada iteração [61]. Para responder a essa questão, a Tabela 5.19 mostra o tempo médio gasto por iteração pelos métodos propostos e pelos demais métodos utilizados como base de comparação, para cada base utilizada. Esse tempo diz respeito ao tempo despendido no processo de aprendizado da percepção visual do usuário e seleção das imagens para serem exibidas. É importante observar que nos experimentos não foram utilizadas as implementações dos autores dos métodos WD_{heu}, WD_{opt} e SVM_{active}. Por isso, é possível que existam algumas variações nos tempos em relação às implementações originais.

Como pode ser observado nessa tabela, o tempo necessário para que os métodos propostos aprendam as necessidades do usuário é aceitável, pois na média, o tempo de cada iteração é entre 0.31 e 4.23 segundos. A obtenção desses baixos tempos de execução foi devida aos parâmetros de programação genética utilizados no processo de busca. Conforme visto na Seção 5.1, alguns parâmetros como tamanho da população, número máximo de gerações e altura da árvore podem influenciar consideravelmente o tempo de execução do algoritmo de aprendizado. Por meio de uma série de experimentos foi observado que

Tabela 5.19: Tempo médio gasto em cada iteração (em segundos).

Base	PG^{+}	$PG^{\pm}$	WD_{heu}	WD_{opt}	SVM_{active}
PEIXES	0.31	3.90	0.09	0.81	13.27
MPEG7	0.93	2.47	0.02	0.41	5.90
COREL	2.19	4.23	0.07	4.50	15.89

os processos de aprendizado propostos foram capazes de assimilar a percepção visual do usuário após poucas gerações, com uma população pequena e utilizando-se árvores baixas.

5.2.5 Análise

Conforme mostrado na Tabela 5.19, o método WD_{heu} apresenta os melhores tempos em todas as bases. Entretanto, comparando esse método com os arcabouços propostos em termos de eficácia, pode ser observado na Figura 5.1(a) que apenas na base PEIXES, nas duas primeiras iterações, WD_{heu} possui melhor desempenho que $PG^{\pm}$ comprovado estatisticamente pelos testes de significância (Tabela 5.13). Em todas as outras situações, os arcabouços propostos apresentam desempenho superior.

O tempo gasto por iteração pelo método WD_{opt} na base MPEG7 é inferior aos dos arcabouços propostos, e na base COREL é superior. Na base PEIXES, seu tempo fica entre o de PG^{+} e o de $PG^{\pm}$. Entretanto, conforme mostram as Figuras 5.1(a) e 5.3(a), apenas na primeira iteração nas bases PEIXES e COREL, WD_{opt} apresenta um desempenho melhor que $PG^{\pm}$. Em todos os demais casos, $PG\pm$ foi superior em relação a eficácia na recuperação. E quando comparado ao arcabouço PG^{+}, o desempenho do método WD_{opt} foi inferior em todas as situações.

Em termos de eficácia, os resultados dos experimentos realizados mostram que apenas o método SVM_{active} apresenta desempenho semelhante aos métodos propostos. Na base MPEG7, esse método mostra resultados próximos a PG^{+} e $PG^{\pm}$, sendo que nas curvas *precisão vs. revocação* (Figura 5.3(b)), somente a partir do valor de revocação 0.9, SVM_{active} é superado por esses arcabouços. Na base COREL, chega a apresentar desempenho estatísticamente melhor à PG^{+} a partir da sétima iteração. Porém, na base PEIXES, os arcabouços propostos são superiores. Além disso, como mostrado na Tabela 5.19, os tempos gastos por iteração pelo método SVM_{active} são consideravelmente superiores para todas as bases quando comparado aos arcabouços propostos.

Os experimentos realizados permitem também comparar os arcabouços propostos entre si. Em relação às diferentes bases utilizadas, o tempo de execução do método PG^{+} varia de acordo com o número de imagens relevantes encontradas por iteração. Esse fato se deve ao processo de cálculo da adequação do indivíduo proposto na Seção 3.1.2.2. Nesse processo,

cada indivíduo é utilizado para ordenar o conjunto de treinamento *para cada imagem no padrão de consulta*, ou seja, cada imagem rotulada como relevante. Dessa forma, quanto mais imagens relevantes, mais ordenações devem ser realizadas, aumentando o tempo gasto no aprendizado. Essa constatação é refletida nos resultados da Tabela 5.19. A base PEIXES possui apenas 10 imagens relevantes para cada consulta. A base MPEG7 possui um número maior, 20 imagens. E na base COREL, esse número varia, mas há, na média, 45 imagens relevantes para cada consulta. Como pode ser observado na Tabela 5.19, PG^{+} é mais rápido na base PEIXES e mais lento na base COREL. O tamanho da base também pode influenciar o tempo de execução desse método. Porém, nas bases utilizadas esse parâmetro não foi muito significante.

O tempo de execução do arcabouço $PG^{\pm}$ é determinado tanto pelo número de imagens relevantes quanto pelo tamanho da base. Segundo esse arcabouço, após o processo de aprendizado as imagens da base são ordenadas. Nessa ordenação, é necessário calcular a similaridade entre cada imagem da base e cada imagem indicada como relevante e cada imagem indicada como irrelevante. No entanto, o número de imagens irrelevantes tende a ser alto. Com isso, para cada imagem da base, muitos cálculos de similaridade são realizados, e quanto maior o tamanho da base, mais tempo será gasto nesse processo de ordenação. Esse fato é ilustrado pelos resultados apresentados na Tabela 5.19. O método $PG^{\pm}$ apresenta um tempo de execução mais baixo na base MPEG7 do que na base PEIXES pois a última possui um número maior de imagens. Porém, o tamanho da base PEIXES não tem maior peso no tempo de execução de $PG^{\pm}$ do que o número de imagens relevantes por consulta na base COREL. Dessa forma, o *overhead* extra do processo de ordenação da base faz com que, em geral, $PG^{\pm}$ seja mais lento que PG^{+}.

Considerando os resultados obtidos, ambos os arcabouços propostos são de fato adequados para recuperação de imagens por conteúdo. Ao serem confrontados com outros métodos de mesmo propósito, citados freqüentemente na literatura, PG^{+} e $PG^{\pm}$ apresentam um desempenho superior. Quando comparados a métodos mais rápidos, como WD_{heu} e WD_{opt}, sua eficácia na recuperação das imagens é superior para todas as bases utilizadas. E quando comparados a métodos com grau de eficácia próximo, como SVM_{active}, seu tempo gasto por iteração é inferior.

Capítulo 6

Conclusões e Trabalhos Futuros

6.1 Conclusões

Este trabalho apresentou dois arcabouços para recuperação de imagens por conteúdo com realimentação de relevância. Esses métodos utilizam programação genética para aprender as necessidades do usuário em uma consulta. Essas necessidades são assimiladas em uma função de combinação. Essa função permite combinações complexas entre valores de similaridade definidos por todos os descritores empregados no processo de recuperação. Dessa forma, o objetivo do processo de busca com programação genética é, em cada iteração, encontrar essas funções, a partir das indicações do usuário.

Com o intuito de obter uma melhor exploração do espaço de características, os métodos propostos utilizam uma abordagem baseada em múltiplos pontos para representação de padrão de consulta. Esses métodos também prevêem a possibilidade de obtenção de mais de uma função de combinação que caracterize a percepção visual do usuário. Por isso, propõem um esquema de votação que possibilita a ordenação das imagens da base considerando mais de uma função de combinação.

Foram realizados experimentos utilizando-se três diferentes bases de imagens e descritores para caracterização das propriedades de cor, textura e forma das imagens. Nos experimentos realizados, os métodos propostos foram comparados com outros três amplamente conhecidos [46, 48, 54]. Os experimentos avaliaram os métodos propostos em relação à eficácia na recuperação de imagens e o tempo gasto em cada iteração do processo de realimentação de relevância. Além disso, foram realizados testes de significância estatística na comparação entre os métodos. Considerando os resultados obtidos, ambos os arcabouços propostos são de fato adequados para recuperação de imagens por conteúdo. Quando comparados a outros métodos de mesmo propósito, citados freqüentemente na literatura, PG^{+} e $PG^{\pm}$ apresentam um desempenho superior.

As principais contribuições são:

- Estudo sobre diferentes técnicas de realimentação de relevância;
- Proposta de dois arcabouços para recuperação de imagens por conteúdo com realimentação de relevância baseado em programação genética;
- Implementação dos métodos propostos, validando-os através de uma série de experimentos e comparações com outros métodos.

6.2 Extensões

Esta seção discute possíveis extensões ao trabalho descrito nesta dissertação, que se dividem em dois conjuntos: aprimoramentos dos arcabouços, apresentados na Seção 6.2.1 e aplicações em outros domínios, mostradas na Seção 6.2.2

6.2.1 Aprimoramentos dos Arcabouços de Realimentação de Relevância

Alguns aprimoramentos possíveis aos métodos são:

- **Técnicas de agrupamento.** Uma possível extensão seria incorporar um método de agrupamento aos mecanismos de aprendizado propostos. Esse método poderia ser aplicado às imagens do padrão de consulta. Com isso, poderiam-se eliminar redundâncias no padrão de consulta – imagens muito próximas entre si – e assim, melhorar o tempo de execução do método proposto.
- ***Cache* para indivíduos**. O cálculo da adequação de um indivíduo é um processo que demanda muito processamento. Uma outra forma de tornar os métodos propostos mais rápidos seria desenvolver um mecanismo de *cache* para os indivíduos. Assim, o sistema poderia evitar o recálculo da adequação de um mesmo indivíduo caso aparecesse repetido na população.
- **Estruturas de indexação**. Em sistemas de recuperação de imagens por conteúdo, estruturas de indexação [8, 55] podem ser utilizadas para melhorar o tempo de execução de consultas por similaridade. Essas estruturas se baseiam nas distâncias entre as imagens para indexá-las no espaço de características. Porém os método propostos alteram a relação de distância entre as imagens, uma possível extensão para o trabalho seria associar os métodos propostos com estruturas de indexação.
- **Discriminação entre imagens de consulta**. Uma outra extensão é relacionada à representação do indivíduo adotada. O conjunto de terminais poderia ser estendido

para comportar as distâncias entre uma imagem da base e todas as imagens do padrão de consulta, segundo todos os descritores utilizados. Essa representação possibilitaria não apenas uma discriminação entre os espaços de características mas também, entre as imagens do padrão de consulta. Esta estratégia parece ser mais robusta à presença de *outliers*, ou seja, imagens relevantes que estão distantes de outras relevantes, mas próximas a imagens irrelevantes. A Figura 6.1 apresenta um exemplo de indivíduo com a nova extensão. Observe que o conjunto de terminais desse indivíduo é composto pelas distâncias entre a imagem I e cada uma das imagens Q_1, Q_2 e Q_3 do padrão de consulta, segundo os descritores D_1 e D_2.

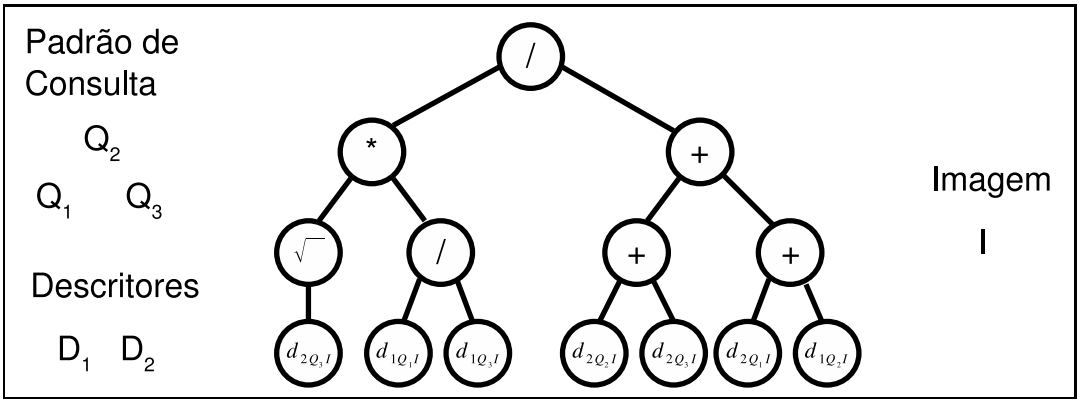


Figura 6.1: Proposta para indivíduo.

6.2.2 Outras Aplicações

Outras aplicações sugeridas para os arcabouços propostos são:

- **Uso em aplicações *Web***. Uma outra questão que mereceria ser estudada é a viabilidade de utilização dos métodos propostos na *Web*. Recuperação de imagem por conteúdo na *Web* é um tópico de pesquisa recente [36], vista a grande quantidade de recursos de *hardware* necessária para esse tipo de aplicação. Por isso, pesquisar o impacto de todo o *overhead* despendido no processo de aprendizado dos métodos propostos seria interessante.

- **Recuperação de outros objetos digitais**. Seria também interessante investigar o desempenho dos arcabouços propostos na recuperação de outros tipos de objetos digitais, por exemplo, documentos textuais e músicas. Os métodos propostos não interferem no processo de descrição dos objetos digitais, por exemplo, fixando

uma função de similaridade para ser utilizada. Essa flexibilidade permite uma fácil acomodação dos métodos para a recuperação de outros tipos de objetos digitais.

- **Integração entre recuperação por conteúdo e textual**. Uma outra variante para a recuperação de imagens é baseada em anotações textuais. Nesse tipo de abordagem, palavras-chaves são manualmente associadas às imagens da base. As consultas são realizadas de forma textual, por meio da comparação dos termos da consulta com as palavras-chaves atribuídas às imagens. Essa vertente possui dois problemas principais: o alto tempo necessário para a anotação manual das imagens e as possíveis inconsistências nas anotações. Porém, essa abordagem introduz a informação semântica ao processo de recuperação de imagens, sendo assim, complementar à recuperação de imagens por conteúdo. Dessa forma, uma outra extensão ao trabalho seria novamente alterar a representação do indivíduo. O conjunto de terminais poderia ser estendido para comportar além de distâncias entre imagens, alguma métrica de distância entre palavras-chaves. Isso possibilitaria combinar consulta por conteúdo com consultas textuais, incorporando semântica ao processo de recuperação de imagens.

Apêndice A

Comparação entre Métodos Exibindo 20 Imagens por Iteração

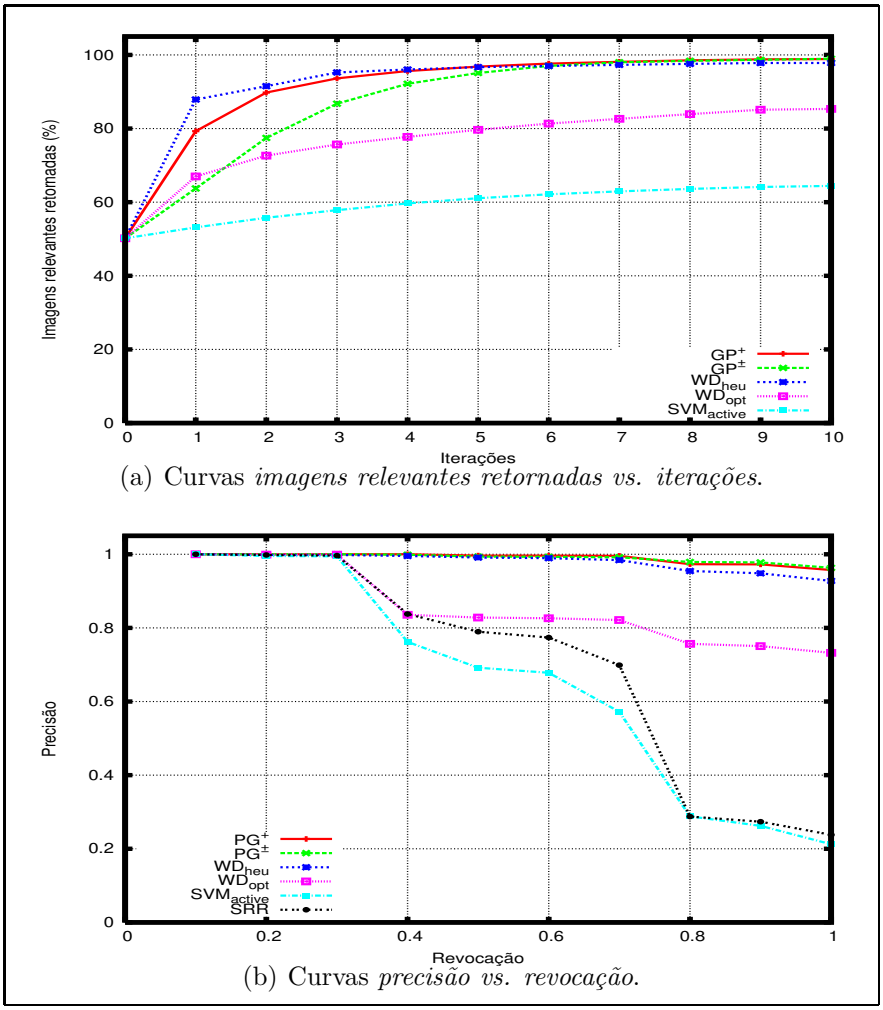
Este apêndice apresenta uma comparação entre métodos similar à mostrada na Seção 5.2, mas agora considerando a exibição de 20 imagens por iteração ao invés de 40.

A.1 Resultados na Base PEIXES

A Figura A.1 apresenta as curvas *imagens relevantes retornadas vs. iterações* (Figura A.1(a)) e *precisão vs. revocação* (Figura A.1(b)) para a comparação entre os métodos na base PEIXES.

A Figura A.1(a) mostra que nas primeiras iterações, o método WD_{heu} apresenta um desempenho superior aos arcabouços propostos. Analisando o resultado do teste de significância sobre os dados dessa figura, apresentado na Tabela A.1, pode ser observado que essa superioridade ocorre apenas nas três primeiras iterações, em relação à PG^{+} e até a quinta iteração, em comparação à $PG^{\pm}$. A partir da sétima iteração, PG^{+} e $PG^{\pm}$ apresentam melhor desempenho que WD_{heu}. Em relação aos demais métodos, PG^{+} é superior a ambos desde a primeira iteração e $PG^{\pm}$ é inferior a WD_{opt} apenas na primeira iteração.

As curvas *precisão vs. revocação* relativas à última iteração do processo de recuperação são exibidas na Figura A.1(b). Essas curvas mostram que tanto PG^{+} quanto $PG^{\pm}$ apresentam um desempenho superior aos demais métodos na última iteração na base PEIXES para todos os pontos. O método WD_{heu} apresenta um resultado próximo à PG^{+} até o valor de *revocação* igual a 0.4 e próximo a $PG^{\pm}$ até o valor de *revocação* igual a 0.6. A partir desses pontos, os valores de *precisão* de PG^{+} e $PG^{\pm}$ são superiores. A Tabela A.2 reforça essa constatação. Essa tabela mostra o resultado da aplicação do teste de *Wilcoxon* nos dados mostrados na Figura A.1(b). Observe que a partir do valor de *revocação*



(a) Curvas *imagens relevantes retornadas vs. iterações.*

(b) Curvas *precisão vs. revocação.*

Figura A.1: Curvas de comparação entre os métodos na base PEIXES.

0.4 para PG^{+} e 0.7 para $PG^{\pm}$ as diferenças entre os valores de *precisão* dos arcabouços propostos e os valores dos outros métodos são significativas.

Em relação aos resultados apresentados na Seção 5.2.1, a redução do número de imagens exibidas por iteração não causou um grande impacto no desempenho dos métodos na base PEIXES. A maior distinção entre os resultados se deu em relação ao método WD_{heu}, que retornou um percentual de imagens relevantes maior que PG^{+} até a iteração 4 e que $PG^{\pm}$ até a iteração 6.

Tabela A.1: Resultados do teste de *Wilcoxon* na base PEIXES sobre os dados apresentados na Figura A.1(a).

	PG^{+}			$PG^{\pm}$		
Iteração	WD_{heu}	WD_{opt}	SVM_{active}	WD_{heu}	WD_{opt}	SVM_{active}
1	$10^{-16}+$	$10^{-16}+$	$10^{-16}+$	$10^{-16}+$	$10^{-6}+$	$10^{-16}+$
2	$0.0004+$	$10^{-16}+$	$10^{-16}+$	$10^{-16}+$	$10^{-6}+$	$10^{-16}+$
3	$10^{-5}+$	$10^{-16}+$	$10^{-16}+$	$10^{-16}+$	$10^{-16}+$	$10^{-16}+$
4	0.1116	$10^{-16}+$	$10^{-16}+$	$10^{-14}+$	$10^{-16}+$	$10^{-16}+$
5	0.9116	$10^{-16}+$	$10^{-16}+$	$0.0001+$	$10^{-16}+$	$10^{-16}+$
6	0.0912	$10^{-16}+$	$10^{-16}+$	0.9094	$10^{-16}+$	$10^{-16}+$
7	$0.0147+$	$10^{-16}+$	$10^{-16}+$	$0.0216+$	$10^{-16}+$	$10^{-16}+$
8	$0.0015+$	$10^{-16}+$	$10^{-16}+$	$0.0004+$	$10^{-16}+$	$10^{-16}+$
9	$0.0007+$	$10^{-16}+$	$10^{-16}+$	$0.0001+$	$10^{-16}+$	$10^{-16}+$
10	$0.0002+$	$10^{-16}+$	$10^{-16}+$	$10^{-5}+$	$10^{-16}+$	$10^{-16}+$

A.1.1 Resultados na Base MPEG7

A Figura A.2 mostra as curvas *imagens relevantes retornadas vs. iterações* (Figura A.2(a)) e *precisão vs. revocação* (Figura A.2(b)) na base MPEG7.

Pela análise da Figura A.2(a) pode ser observado que os arcabouços retornam mais imagens relevantes que os demais métodos em todas as iterações nos experimentos com a base MPEG7. A Tabela A.3 reforça esse fato. Essa tabela apresenta o resultado do teste de *Wilcoxon* sobre os dados da Figura A.2(a). Como pode ser observado nessa tabela, a superioridade de PG^{+} e $PG^{\pm}$ mostradas pelas curvas *imagens relevantes retornadas vs. iterações* na base MPEG7 são significativas para todos os pontos.

As curvas *precisão vs. revocação* apresentadas na Figura A.2(b) mostram que na última iteração dos experimentos executados na base MEPG7, todos os métodos apresentam resultados muito próximos até o valor de revocação 0.4. Essas curvas em conjunto com o teste de *Wilcoxon* apresentado na Tabela A.4, mostram que a partir do valor de revocação 0.6, os arcabouços propostos e o método SVM_{active} apresentam valores de precisão significativamente superiores aos demais, sendo que, esse último passa a ser superado por PG^{+} e $PG^{\pm}$ a partir dos valores de revocação 0.9 e 0.8, respectivamente.

Em relação aos resultados apresentados na Seção 5.2.2, a redução do número de imagens exibidas por iteração não causou um grande impacto no desempenho dos métodos na base MPEG7. Uma distinção que pode ser feita entre os resultados é que a distância entre as curvas referentes aos arcabouços propostos e o método SVM_{active} é um pouco maior.

Tabela A.2: Resultados do teste de *Wilcoxon* na base PEIXES sobre os dados apresentados na Figura A.1(b).

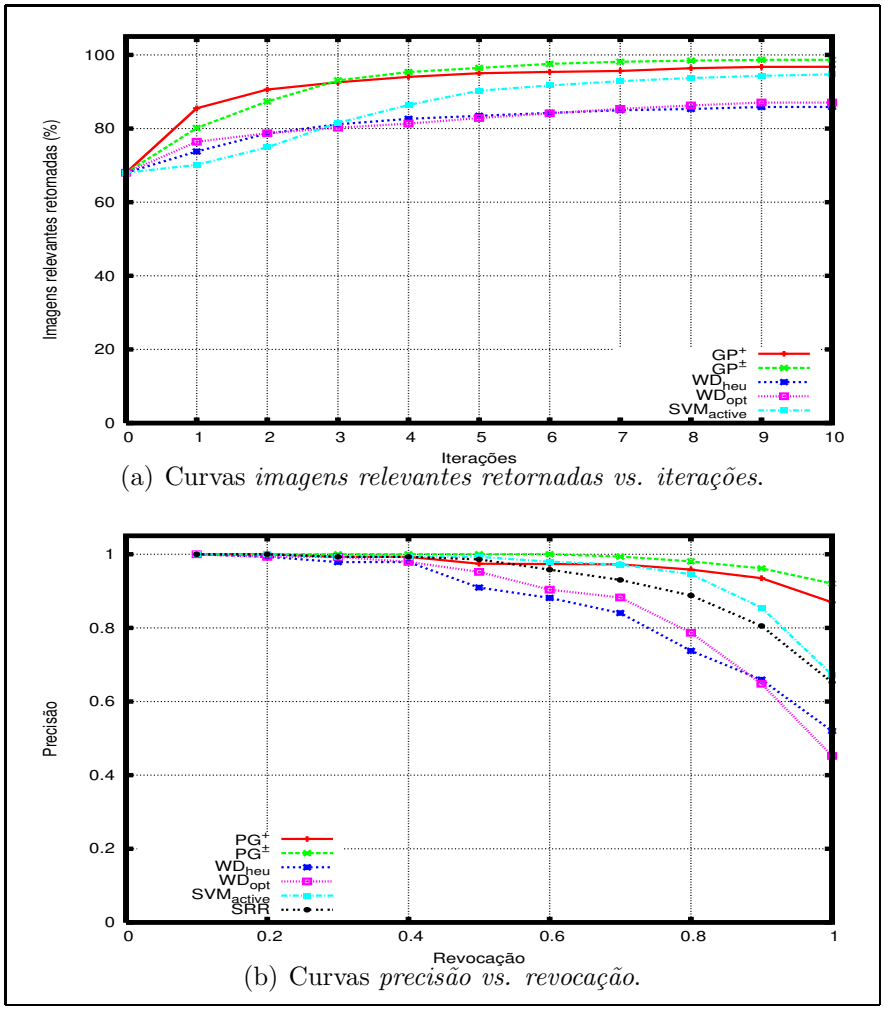
	PG^{+}			$PG^{\pm}$		
Revocação	WD_{heu}	WD_{opt}	SVM_{active}	WD_{heu}	WD_{opt}	SVM_{active}
0.1	1	1	1	1	1	1
0.2	0.3711	1	0.0590	0.3711	1	0.0590
0.3	0.3711	1	0.0360+	0.3711	1	0.0360+
0.4	0.0590	10^{-16}+	10^{-16}+	0.1508	10^{-16}+	10^{-16}+
0.5	0.0206+	10^{-16}+	10^{-16}+	0.2184	10^{-16}+	10^{-16}+
0.6	0.0144+	10^{-16}+	10^{-16}+	0.1507	10^{-16}+	10^{-16}+
0.7	0.0010+	10^{-16}+	10^{-16}+	0.0382+	10^{-16}+	10^{-16}+
0.8	0.0297+	10^{-16}+	10^{-16}+	0.0007+	10^{-16}+	10^{-16}+
0.9	0.0129+	10^{-16}+	10^{-16}+	0.0002+	10^{-16}+	10^{-16}+
1.0	0.0013+	10^{-16}+	10^{-16}+	0.0002+	10^{-16}+	10^{-16}+

A.1.2 Resultados na Base COREL

A Figura A.3 apresenta as curvas *imagens relevantes retornadas vs. iterações* (Figura A.3(a)) e *precisão vs. revocação* (Figura A.3(b)) para a comparação entre os métodos na base COREL.

A Figura A.3(a) mostra que o arcabouço PG^{+} apresenta um desempenho superior aos demais métodos nas quatro primeiras iterações. A partir da sexta iteração, SVM_{active} passa a exibir mais imagens relevantes que PG^{+}. Já o arcabouço $PG^{\pm}$ apresenta um resultado semelhante à WD_{opt} e à SVM_{active} na primeira iteração. A partir da segunda, $PG^{\pm}$ e SVM_{active} superam WD_{opt}. Observe que $PG^{\pm}$ e SVM_{active} apresentam desempenhos semelhantes, sendo que a partir da quinta iteração $PG^{\pm}$ é ligeiramente superior. A Tabela A.5 exibe o resultado da aplicação do teste de *Wilcoxon* para as curvas *imagens relevantes retornadas vs. iterações* apresentadas na Figura A.3(a). A partir dessa tabela pode ser observado que a curva associada ao arcabouço PG^{+} possui diferenças significativas em relação às curvas referentes aos métodos WD_{heu} e WD_{opt} para todas as iterações. Apenas na primeira iteração WD_{opt} apresenta desempenho semelhante a $PG^{\pm}$. Em todas as outras, $PG^{\pm}$ retorna mais imagens relevantes que WD_{heu} e WD_{opt}. Em relação ao método SVM_{active}, PG^{+} apresenta um melhor desempenho nas quatro primeiras iterações. A partir da quinta, as diferenças entre o desempenhos desses métodos não são significativas, embora o resultado exibido na Figura A.3(a) sugira que SVM_{active} seja superior a PG^{+} a partir da sexta iteração. $PG^{\pm}$ e SVM_{active} apresentam desempenhos semelhantes em todas as iterações, sendo $PG^{\pm}$ significantemente superior na segunda, nona e na décima iteração.

As curvas *precisão vs. revocação* apresentadas na Figura A.3(b) mostram que os



(a) Curvas *imagens relevantes retornadas vs. iterações.*

(b) Curvas *precisão vs. revocação.*

Figura A.2: Curvas de comparação entre os métodos na base MPEG7.

arcabouços propostos, juntamente com o SVM_{active} apresentam valores de precisão superiores aos demais métodos na última iteração na base COREL. Como pode ser observado, SVM_{active} é superior a PG^{+} do valor de revocação 0.3, até 0.7, A partir do valor de revocação 0.8, PG^{+} passa a apresentar precisão mais alta que SVM_{active}. Já $PG^{\pm}$ apresenta resultados semelhantes a SVM_{active} até revocação igual a 0.6. A partir desse ponto, $PG^{\pm}$ apresenta valores de precisão superiores aos demais métodos. A Tabela 5.18 exibe o resultado da aplicação do teste de *Wilcoxon* nos dados da Figura 5.3(b) indicando se as diferenças entre as curvas associadas aos arcabouços propostos e as relativas aos demais métodos são significativas. Essa tabela mostra que essas diferenças são significativas em relação às curvas associadas a WD_{heu} e WD_{opt} a partir dos valores de revocação 0.1 e

Tabela A.3: Resultados do teste de *Wilcoxon* na base MPEG7 sobre os dados apresentados na Figura A.2(a).

	PG^{+}			$PG^{\pm}$		
Iteração	WD_{heu}	WD_{opt}	SVM_{active}	WD_{heu}	WD_{opt}	SVM_{active}
1	$10^{-14}+$	$10^{-10}+$	$10^{-16}+$	$10^{-07}+$	$0.0007+$	$10^{-14}+$
2	$10^{-13}+$	$10^{-12}+$	$10^{-16}+$	$10^{-09}+$	$10^{-9}+$	$10^{-14}+$
3	$10^{-12}+$	$10^{-13}+$	$10^{-16}+$	$10^{-11}+$	$10^{-13}+$	$10^{-14}+$
4	$10^{-12}+$	$10^{-13}+$	$10^{-13}+$	$10^{-12}+$	$10^{-14}+$	$10^{-13}+$
5	$10^{-11}+$	$10^{-13}+$	$10^{-8}+$	$10^{-12}+$	$10^{-13}+$	$10^{-10}+$
6	$10^{-11}+$	$10^{-12}+$	$10^{-6}+$	$10^{-12}+$	$10^{-13}+$	$10^{-10}+$
7	$10^{-11}+$	$10^{-12}+$	$10^{-5}+$	$10^{-12}+$	$10^{-12}+$	$10^{-9}+$
8	$10^{-11}+$	$10^{-11}+$	$0.0002+$	$10^{-12}+$	$10^{-13}+$	$10^{-9}+$
9	$10^{-11}+$	$10^{-11}+$	$0.0004+$	$10^{-12}+$	$10^{-13}+$	$10^{-8}+$
10	$10^{-11}+$	$10^{-11}+$	$0.0011+$	$10^{-12}+$	$10^{-13}+$	$10^{-7}+$

0.2, respectivamente. Como pode ser observado nessa tabela, a superioridade do método SVM_{active} em relação à PG^{+} somente é significante para o valor de revocação 0.6, sendo que PG^{+} apresenta precisão mais alta para revocação igual a 1. Em comparação com $PG^{\pm}$, SVM_{active} apresenta desempenho significativamente inferior a partir do valor de revocação 0.8.

A redução do número de imagens exibidas por iteração ocasionou um grande impacto no desempenho dos métodos na base COREL. Em comparação com os resultados apresentados na Seção 5.2.3, a porcentagem de imagens relevantes retornadas por iteração diminuiu em torno de 10 pontos percentuais.

A.1.3 Tempos de Execução

A Tabela A.7 mostra o tempo médio gasto por iteração pelos métodos propostos e pelos demais métodos utilizados como base de comparação, para cada base utilizada quando 20 imagens são exibidas para o usuário em cada iteração. Esse tempo diz respeito ao tempo despendido no processo de aprendizado da percepção visual do usuário e seleção das imagens para serem exibidas.

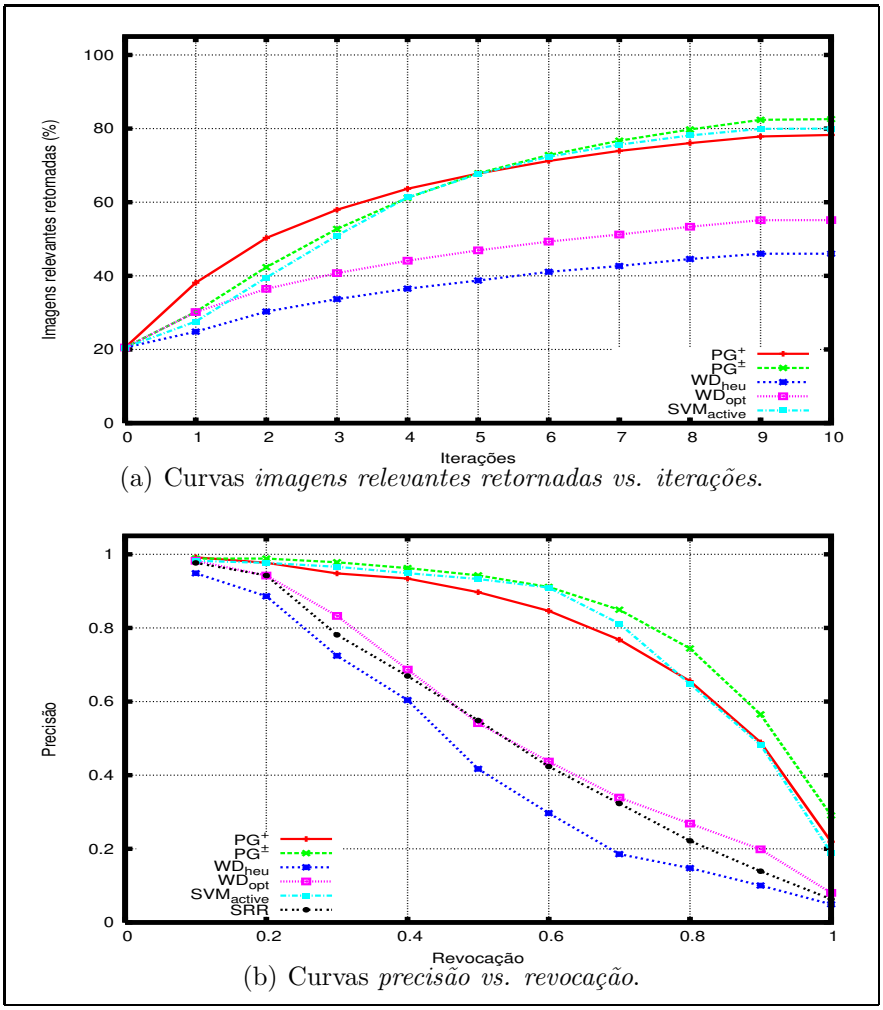
Em comparação com os tempos apresentados na Tabela 5.19, pode ser observado que PG^{+} e os métodos WD_{heu}, WD_{opt} e SVM_{active} apresentam uma pequena diminuição no tempo de cada iteração quando 20 imagens são exibidas para o usuário. Essa redução é maior para o arcabouço $PG^{\pm}$, devido, principalmente, ao menor número de imagens rotuladas como irrelevantes ao longo das iterações. Como mencionado na Seção 5.2.5, quanto maior o número de imagens irrelevantes encontradas, maior o tempo necessário para seleção das imagens a serem exibidas para o usuário. Como o número de imagens

Tabela A.4: Resultados do teste de *Wilcoxon* na base MPEG7 sobre os dados apresentados na Figura A.2(b).

	PG^{+}			$PG^{\pm}$		
Revocação	WD_{heu}	WD_{opt}	SVM_{active}	WD_{heu}	WD_{opt}	SVM_{active}
0.1	1	1	1	1	1	1
0.2	1	1	1	1	1	1
0.3	0.1814	1	1	0.1814	1	1
0.4	0.1814	0.4227	1	0.1814	0.1814	1
0.5	0.0025+	0.1029	0.5896	0.0016+	0.0224+	1
0.6	0.0003+	0.0020+	1	0.0003+	0.0010+	0.1814
0.7	10^{-5}+	0.0027+	0.7998	10^{-5}+	0.0002+	0.1056
0.8	10^{-8}+	10^{-5}+	0.5049	10^{-8}+	10^{-6}	0.0249+
0.9	10^{-9}+	10^{-9}+	0.0003+	10^{-9}+	10^{-9}+	10^{-5}+
1.0	10^{-12}+	10^{-14}+	10^{-7}+	10^{-12}+	10^{-14}+	10^{-9}+

exibidas por iteração diminuiu de 40 para 20, conseqüentemente o número de imagens rotuladas como irrelevantes é menor, possibilitando um tempo mais baixo para seleção das imagens.

No que diz respeito ao arcabouço PG^{+}, a maior diferença no tempo de cada iteração ocorreu nos experimentos com base COREL. Conforme exposto na Seção 5.2.5, esse tempo é sensível ao número de imagens relevantes encontradas. Essa imagens influenciam o tempo despendido no processo de cálculo da adequação dos indivíduos. Pela comparação entre as Figuras 5.3(a) e A.3(a) pode ser observado que a porcentagem de imagens relevantes encontradas em cada iteração por esse método é muito inferior quando são exibidas 20 imagens ao invés de 40. Dessa forma, o processo de cálculo da adequação de um indivíduo é mais rápido e o tempo médio por iteração é menor.



(a) Curvas *imagens relevantes retornadas vs. iterações.*

(b) Curvas *precisão vs. revocação.*

Figura A.3: Curvas de comparação entre os métodos na base COREL.

Tabela A.5: Resultados do teste de *Wilcoxon* na base COREL sobre os dados apresentados na Figura A.3(a).

	PG^{+}			$PG^{\pm}$		
Iteração	WD_{heu}	WD_{opt}	SVM_{active}	WD_{heu}	WD_{opt}	SVM_{active}
1	$10^{-16}+$	$10^{-16}+$	$10^{-15}+$	$10^{-8}+$	0.4407	0.0613
2	$10^{-16}+$	$10^{-16}+$	$10^{-11}+$	$10^{-16}+$	$10^{-7}+$	0.0335+
3	$10^{-16}+$	$10^{-16}+$	$10^{-7}+$	$10^{-16}+$	$10^{-16}+$	0.1751
4	$10^{-16}+$	$10^{-16}+$	0.0144+	$10^{-16}+$	$10^{-16}+$	0.8673
5	$10^{-16}+$	$10^{-16}+$	0.2038	$10^{-16}+$	$10^{-16}+$	0.8773
6	$10^{-16}+$	$10^{-16}+$	0.8032	$10^{-16}+$	$10^{-16}+$	0.9753
7	$10^{-16}+$	$10^{-16}+$	0.6997	$10^{-16}+$	$10^{-16}+$	0.5346
8	$10^{-16}+$	$10^{-16}+$	0.3087	$10^{-16}+$	$10^{-16}+$	0.2058
9	$10^{-16}+$	$10^{-16}+$	0.2608	$10^{-16}+$	$10^{-16}+$	0.0424+
10	$10^{-16}+$	$10^{-16}+$	0.2885	$10^{-16}+$	$10^{-16}+$	0.0317+

Tabela A.6: Resultados do teste de *Wilcoxon* na base COREL sobre os dados apresentados na Figura A.3(b).

	PG^{+}			$PG^{\pm}$		
Revocação	WD_{heu}	WD_{opt}	SVM_{active}	WD_{heu}	WD_{opt}	SVM_{active}
0.1	0.3613	0.7893	0.3613	0.0091+	0.3613	0.3613
0.2	0.0683+	1	0.4185	$10^{-5}+$	0.0108+	0.1003
0.3	$10^{-5}+$	0.0302+	0.3268	$10^{-9}+$	$10^{-6}+$	0.1508
0.4	$10^{-10}+$	0.0002+	0.3133	$10^{-13}+$	$10^{-10}+$	0.2553
0.5	$10^{-15}+$	$10^{-7}+$	0.0612	$10^{-16}+$	$10^{-15}+$	0.5075
0.6	$10^{-16}+$	$10^{-11}+$	0.0139+	$10^{-16}+$	$10^{-16}+$	0.6224
0.7	$10^{-16}+$	$10^{-14}+$	0.3435	$10^{-16}+$	$10^{-16}+$	0.0628
0.8	$10^{-16}+$	$10^{-16}+$	0.2299	$10^{-16}+$	$10^{-16}+$	0.0004+
0.9	$10^{-16}+$	$10^{-16}+$	0.0726	$10^{-16}+$	$10^{-16}+$	$10^{-5}+$
1.0	$10^{-16}+$	$10^{-16}+$	$10^{-7}+$	$10^{-16}+$	$10^{-16}+$	$10^{-10}+$

Tabela A.7: Tempo médio gasto em cada iteração (em segundos).

Base	PG^{+}	$PG^{\pm}$	WD_{heu}	WD_{opt}	SVM_{active}
PEIXES	0.28	2.05	0.08	0.81	12.8
MPEG7	0.84	1.78	0.02	0.40	5.34
COREL	1.75	2.92	0.07	4.43	15.04

Apêndice B

Ilustração de Consulta

Este apêndice ilustra o resultado de uma consulta na base de imagens COREL utilizando PG^{+} (Seção B.1) e $PG^{\pm}$ (Seção B.2). Nessa consulta são utilizados os parâmetros definidos na Tabela 5.11 para o método PG^{+} e para $PG^{\pm}$ os parâmetros definidos na Tabela 5.12. Também são apresentados exemplos de indivíduos utilizados para ordenar a base em cada iteração.

B.1 Consulta com PG^{+}

A Figura B.1 mostra o resultado de uma consulta utilizando o método PG^{+}. Observe que apenas uma imagem relevante foi retornada no conjunto inicial, a própria imagem de consulta. Ao final do processo de recuperação, na décima iteração, 36 entre as 40 imagens exibidas foram relevantes.

A Figura B.2 apresenta alguns dos indivíduos utilizados para ordenar as imagens da base em cada iteração para a consulta ilustrada na Figura B.1. Além dos indivíduos, é exibido também o número de novas imagens relevantes encontradas em cada iteração. Em algumas iterações, mais de um indivíduo foram utilizados para ordenar as imagens da base, como nas iterações $1, 2$ e 4.

A Tabela B.1 apresenta a freqüência de ocorrência de cada terminal nos indivíduos da Figura B.2, em cada iteração, em valores percentuais. A coluna *Total* exibe essa frequência considerando todos os indivíduos obtidos ao longo das iterações. Observe que os terminais definidos pelos descritores BIC e Histograma de Cor apresentam um número de ocorrências mais alto. Esses descritores obtiveram o melhor desempenho na classe das imagens da Figura B.1, como mostrado na Figura B.3. Essa figura apresenta as curvas *precisão vs. revocação* referentes aos descritores de cor e textura utilizados nos experimentos, para as imagens da classe exibidas na Figura B.1.

Tabela B.1: Freqüência de ocorrência de cada terminal nos indivíduos da Figura B.2 em valores percentuais.

Terminais \ Iterações	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**Total**
F0	16,67	21,21	16,67	0,00	15,38	0,00	2,38	0,00	8,33	25,00	10,24
F1	12,50	15,15	33,33	31,25	46,15	33,33	50,00	50,00	31,25	50,00	33,17
F2	29,17	33,33	33,33	56,25	30,77	33,33	26,19	25,00	41,67	12,50	33,66
F3	16,67	12,12	0,00	0,00	0,00	0,00	2,38	0,00	14,58	0,00	7,80
F4	25,00	18,18	16,67	12,50	7,69	33,33	19,05	25,00	4,17	12,50	15,12

B.2 Consulta com $PG^{\pm}$

A Figura B.4 mostra o resultado de uma consulta utilizando o arcabouço $PG^{\pm}$, considerando a mesma imagem de consulta utilizada na Figura B.1. Assim como no exemplo anterior, apenas a imagem de consulta foi retornada no conjunto inicial de imagens. Na última iteração, todas as 40 imagens retornadas são relevantes.

A Figura B.5 apresenta os indíviduos utilizados para selecionar as imagens para serem exibidadas para o usuário em cada iteração no processo de recuperação ilustrado na Figura B.4. Essa figura destaca também o número de imagens relevantes encontradas em cada iteração.

A Tabela B.2 apresenta a freqüência de ocorrência de cada terminal nos indivíduos da Figura B.5, em cada iteração, em valores percentuais. A coluna *Total* exibe essa frequência considerando todos os indivíduos obtidos ao longo das iterações. Novamente, os descritores que apresentaram maior incidência sobre os terminais dos indivíduos foram o BIC e o Histograma de Cor.

Tabela B.2: Freqüência de ocorrência de cada terminal nos indivíduos da Figura B.5 em valores percentuais.

Terminais \ Iterações	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**Total**
F0	0,00	20,00	33,33	0,00	15,38	9,68	16,67	7,69	0,00	17,65	11,76
F1	0,00	53,33	50,00	40,00	53,85	41,94	33,33	61,54	50,00	35,29	45,10
F2	100,00	6,67	16,67	40,00	30,77	16,13	33,33	30,77	35,71	29,41	23,53
F3	0,00	13,33	0,00	20,00	0,00	8,06	0,00	0,00	0,00	11,76	6,54
F4	0,00	6,67	0,00	0,00	0,00	24,19	16,67	0,00	14,29	5,88	13,07

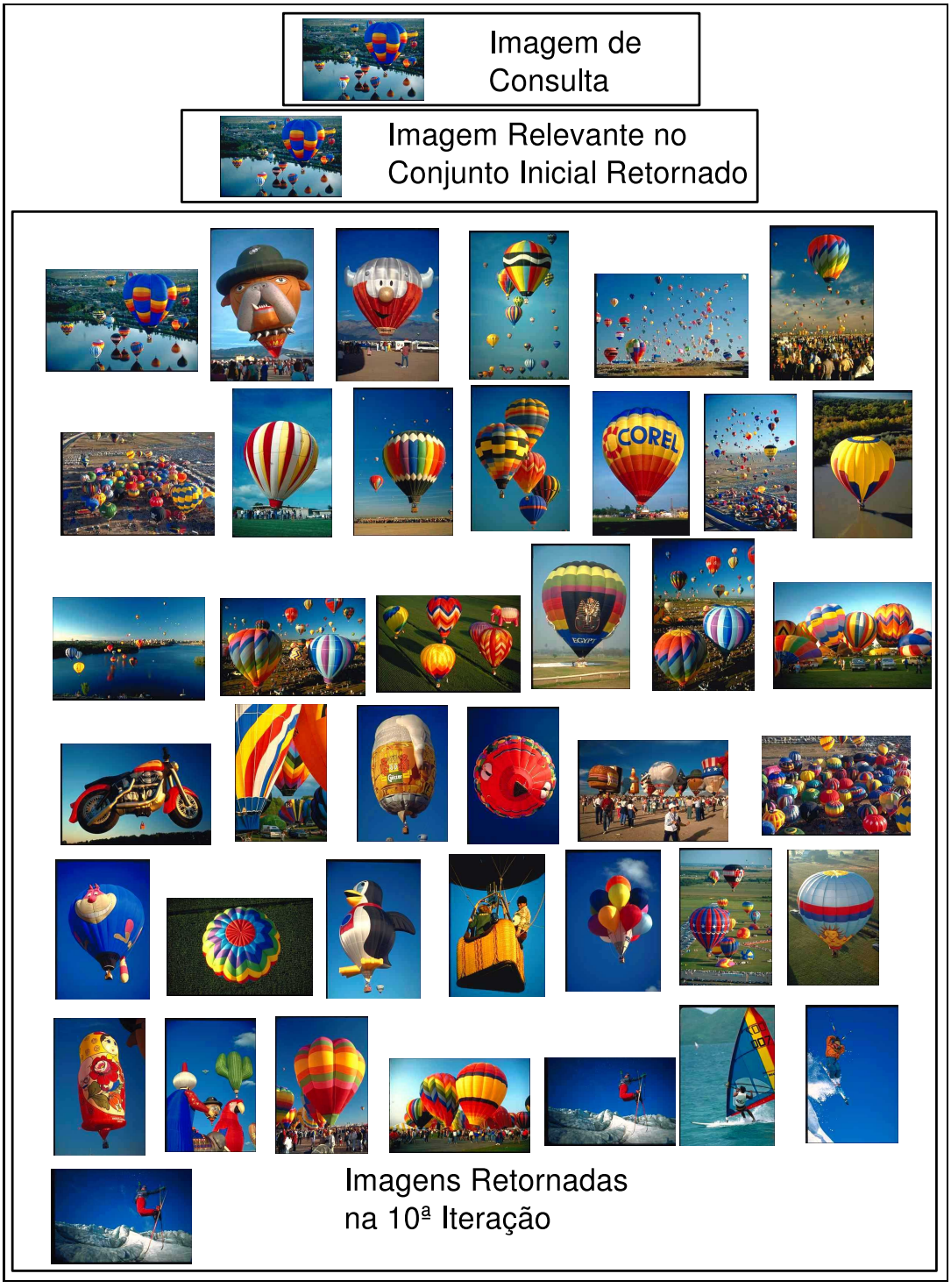


Figura B.1: Resultado de uma consulta.

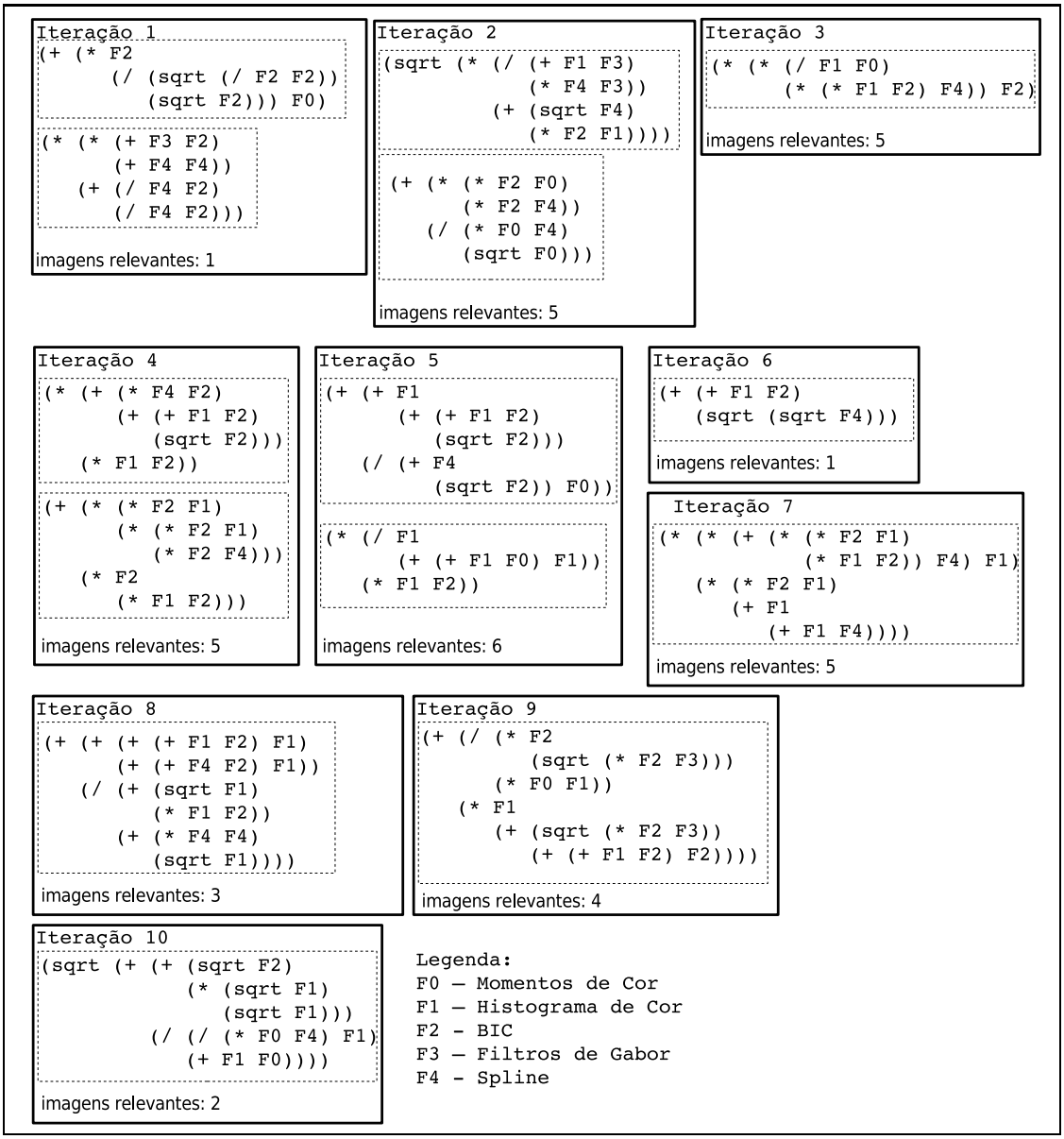


Figura B.2: Indivíduos gerados na consultada da Figura B.1.

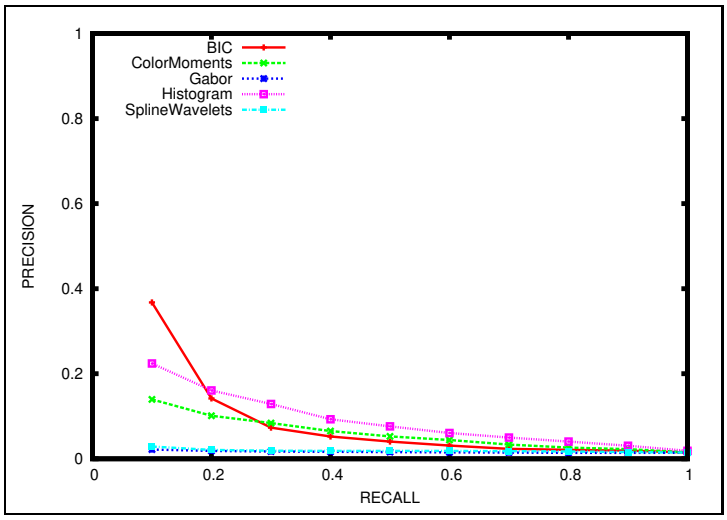


Figura B.3: Curvas *precisão vs. revocação* Figura B.1.

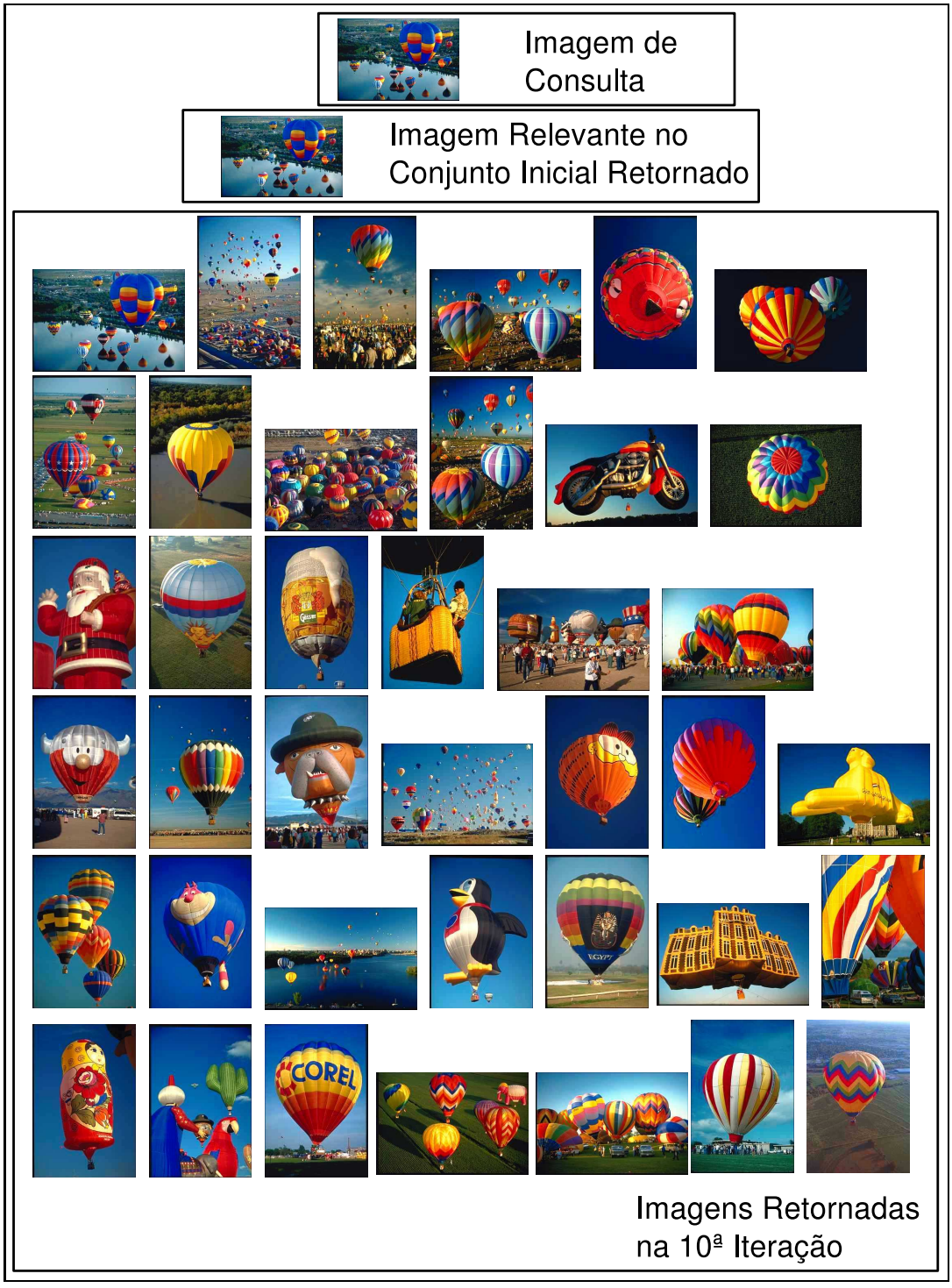


Figura B.4: Resultado de uma consulta.

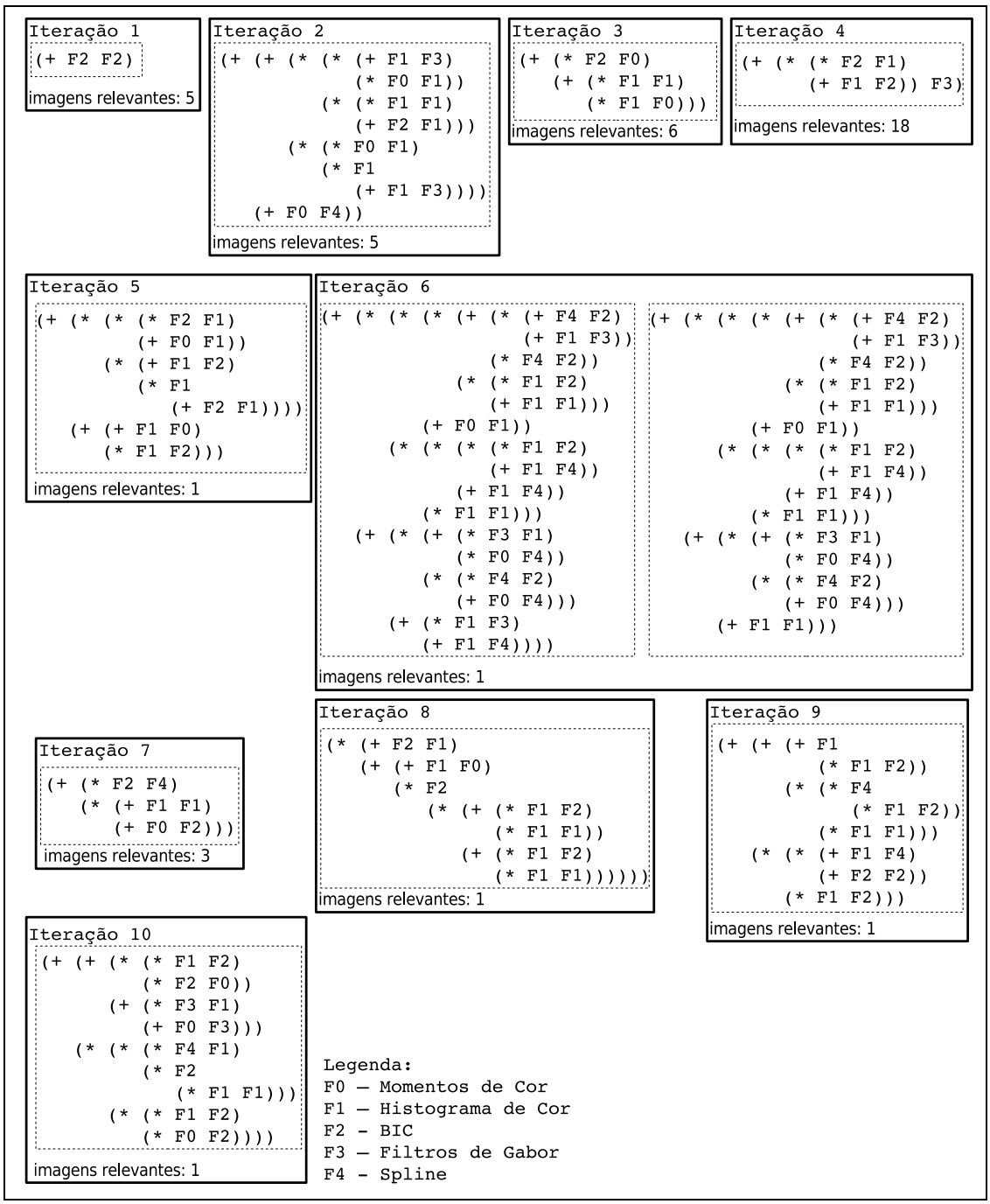


Figura B.5: Indivíduos gerados na consultada da Figura B.1.

Referências Bibliográficas

[1] P. Alshuth, T. Hermes, J. Kreyb, and M. Roper. *Image and Mult-media Search*, volume 8, chapter Intelligent Retrieval for Images and Videos. A. W. M. Smeulders and R. Jain, World Scientific, 1997.

[2] F. A. Andaló. Descritores de forma baseados em *Tensor Scale*. Master's thesis, Universidade Estadual de Campinas, Março 2007.

[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] T. Bäck, D. B. Fogel, and Z. Michalewicz. *Evolutionary Computation 1 Basics Algorithms and Operators*. Institute of Physics Publishing, 2002.

[5] W. Banzhaf, P. Nordin, R. Keller, and F. Francone. *Genetic Programming - An Introduction*. Morgan Kaufmann Publishers, Inc, San Francisco, CA, 1998.

[6] B. Bhanu and Y. Lin. Object Detection in Multi-Modal Images Using Genetic Programming. *Applied Soft Computing*, 4(2):175–201, May 2004.

[7] Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm` (acessado em julho/2007).

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

[9] M. Cord, P. H. Gosselin, and S. Philipp-Foliguet. Stochastic exploration and active learning for image retrieval. *Image and Vision Computing*, 25(1):14–23, 2007.

[10] I. J. Cox, M. L. Miller, T. P. Minka, T. V. Papathomas, and P. N. Yianilos. The Bayesian Image Retrieval System, PicHunter: Theory, Implementation, and Psychophysical Experiments. *IEEE Transactions on Image Processing*, 9(1):20–37, January 2000.

[11] D. da S. Andrade. Testes de significância estatísticos e avaliação de um modelo de recuperação de imagens por conteúdo. Master's thesis, Universidade Estadual de Campinas, Julho 2004.

[12] R. da S. Torres. *Integrating Image and Spatial Data for Biodiversity Information Management*. PhD thesis, Institute of Computing, University of Campinas, 2004.

[13] R. da S. Torres, A. X. Falcão, and L. da F. Costa. A Graph-based Approach for Multiscale Shape Analysis. *Pattern Recognition*, 37(6):1163–1174, June 2004.

[14] R. da S. Torres, A. X. Falcão, M. A. Gonçalves, B. Zhang, W. Fan, and E. A. Fox. A New Framework to Combine Descriptors for Content-based Image Retrieval. Technical Report IC-05-21, Institute of Computing, State University of Campinas, Campinas, Brazil, 2005.

[15] R. da 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 14th ACM International Conference on Information and Knowledge Management*, pages 335–336. ACM Press, 2005.

[16] R. da S. Torres and Alexandre X. Falcão. Contour Salience Descriptors for Effective Image Retrieval and Analysis. *Image and Vision Computing*, 25(1):3–13, January 2007.

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

[18] S. F. da Silva, C. A. Z. Barcelos, and M. A. Batista. An image retrieval system adaptable to user's interests by the use of relevance feedback via genetic algorithm. In *Proceedings of the 12th Brazilian Symposium on Multimedia and the Web*, pages 45–52, 2006.

[19] H. M. de Almeida. Uma abordagem de componentes combinados para a geração de funções de ordenação usando programação genética. Master's thesis, Universidade Federal de Minas Gerais, 2007.

[20] L. M. del Val Cura. *Um modelo para recuperação por conteúdo de Imagens de Sensoriamento Remoto*. PhD thesis, Universidade Estadual de Campinas, 2002.

[21] N. Doulamis and A. Doulamis. Evaluation of relevance feedback schemes in content-based in retrieval systems. *Signal Processing: Image Communication*, 21(4):334–357, April 2006.

[22] L. Duan, W. Gao, W. Zeng, and D. Zhao. Adaptive relevance feedback based on Bayesian inference for image retrieval. *Signal Processing*, 85(2):395–399, February 2005.

[23] A. E. Eiben, R. Hinterding, and Z. Michalewicz. Parameter control in evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 3(2):124–141, 1999.

[24] W. Fan, E. A. Fox, P. Pathak, and H. Wu. The Effects of Fitness Functions on Genetic Programming-Based Ranking Discovery for Web Search. *Journal of the American Society for Information Science and Technology*, 55(7):628–636, 2004.

[25] W. Fan, M. D. Gordon, and P. Pathak. A generic ranking function discovery framework by genetic programming for information retrieval. *Information Processing & Management*, 40(4):587–602, July 2004.

[26] P. C. Fishburn. *Non-Linear Preference and Utility Theory*. Johns Hopkins University Press, Baltimore, 1988.

[27] M. Flickner, H. Sawhney, W. Niblack, Q. Huang J. Ashley, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker. Query by Image and Video Content: the QBIC System. *IEEE Computer*, 28(9):23–32, Sep 1995.

[28] I. Gagliardi G. Ciocca and R. Schettini. Quicklook^2: An Integrated Multimedia System. *Journal of Visual Languages & Computing*, 12(1):81–103, February 2001.

[29] I. Gondra and D. Heisterkamp. Adaptive and Efficient Image Retrieval with One-Class Support Vector Machines for Inter-Query Learning. *WSEAS Transactions on Circuits and Systems*, 3(2):324–329, April 2004.

[30] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Addison-Wesley, Reading, MA, USA, 1992.

[31] S. Guha, N. Koudas, A. Marathe, and D. Srivastava. Merging the results of approximate match operations. In *Proceedings of the 30th International Conference on Very Large Data Bases*, pages 636–647, 2004.

[32] D. Harman. Relevance feedback revisited. In *Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1–10, Copenhagen, Denmark, 1992. ACM Press.

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

[34] P. Hong, Q. Tian, and T. S. Huang. Incorporate support vector machines to content-based image retrieval with relevant feedback. In *Proceedings of the 7th IEEE International Conference on Image Processing*, pages 750–753, 2000.

[35] A. Kak and C. Pavlopoulou. Content-based image retrieval from large medical databases. *3D Data Processing Visualization and Transmission*, pages 138–147, June 2002.

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

[37] D.-H. Kim, C.-W. Chung, and K. Barnard. Relevance feedback using adaptive clustering for image similarity retrieval. *Journal of Systems and Software*, 78(1):9–23, October 2005.

[38] J. R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, Cambridge, MA, USA, 1992.

[39] L. J. Latecki and R. Lakamper. Shape Similarity Measure Based on Correspondence of Visual Parts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10):1185–1190, 2000.

[40] M. S. Lew, editor. *Principles of Visual Information Retrieval – Advances in Pattern Recognition*. Springer-Verlag, London Berlin Heidelberg, 2001.

[41] B. Li and S. Yuan. A novel relevance feedback method in content-based image retrieval. In *Proceedings of International Conference on Information Technology: Coding an Computing*, pages 120–123, 2004.

[42] Y. Liu, D. Zhang, G. Lu, and W-Y Ma. A survey of content-based image retrieval with high-level semantics. *Pattern Recognition*, 40(1):262–282, January 2007.

[43] W. Y. Ma and B. S. Manjunath. Netra: A Toolbox for Navigating Large Image Databases. In *IEEE International Conference on Image Processing*, pages 256–268, 1997.

[44] V. E. Ogle and M. Stonebraker. Chabot: Retrieval from Relational Database of Images. IEEE *Computer*, 28(9):40–48, Sep 1995.

[45] E. Persoon and K. Fu. Shape Discrimination Using Fourier Descriptors. *IEEE Transanctions on Systems, Man, and Cybernetics*, 7(3):170– 178, 1977.

[46] Y. Rui and T. Huang. Optimizing learning in image retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 236–245, 2000.

[47] Y. Rui, T. S. Huang, and S. F. Chang. Image Retrieval: Current Techniques, Promising Directions, and Open Issues. *Journal of Communications and Image Representation*, 10(1):39–62, March 1999.

[48] Y. Rui, T. S. Huang, M. Ortega, and S. Mehrotra. Relevance Feedback: A Power Tool for Interactive Content-Based Image Retrieval. *IEEE Transactions on Circuits and Systems for Video Technology*, 8(5):644–655, 1998.

[49] B. Schoelkopf and A. J. Smola. *Learning with Kernels*. The MIT Press, Cambridge, MA, 2002.

[50] 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 11th ACM International Conference on Information and Knowledge Management*, pages 102–109, McLean, Virginia, USA, November 2002. ACM Press.

[51] M. A. Stricker and M. Orengo. Similarity of Color Images. In *Storage and Retrieval for Image and Video Databases (SPIE)*, pages 381–392, 1995.

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

[53] H. Tamura, S. Mori, and T. Yamawaki. Texture features corresponding to visual perception. *IEEE Transactions on Systems, Man and Cybernetics*, 8(6):460–473, 1978.

[54] S. Tong and E. Y. Chang. Support vector machine active learning for image retrieval. In *Proceedings of 9th ACM international conference on Multimedia*, pages 107–118, New York, NY, USA, 2001. ACM Press.

[55] C. Traina, B. Seeger, C. Faloutsos, and A. Traina. Fast Indexing and Visualization of Metric Datasets Using Slim-Trees. *IEEE Transactions on Knowledge and Data Engineering*, 14(2):244–60, March/April 2002.

[56] A. Vadivel, A.K. Majumdar, and S. Sural. Characteristics of weighted feature vector in content-based image retrieval applications. *Intelligent Sensing and Information Processing*, 1(18):127–132, 2004.

[57] Y. P. Wang and T. Pavlids. Optimal Correspondence of String Subsequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(11):1080–1087, 1990.

[58] Z. Xu, X. Xu, K. Yu, and V. Tresp. A Hybrid Relevance-Feedback Approach to Text Retrieval. *Proceedings of the 25th European Conference on Information Retrieval Research, Lecture Notes in Computer Science*, 2633:81–293, April 2003.

[59] B. Zhang, M. A. Gonçalves, W. Fan, Y. Chen, E. A. Fox, P. Calado, and M. Cristo. Combining structural and citation-based evidence for text classification. In *Proceedings of the 13th ACM Conference on Information and Knowledge Management*, pages 162–163, 2004.

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

[61] X. S. Zhou and T. S. Huang. Relevance feedback in image retrieval: A comprehensive review. *Multimedia System*, 8(6):536–544, 2003.

[62] D. Zonker and B. Punch. lil-gp 1.01 User's Manual. Technical report, College of Engineering, Michigan State University, Michigan, USA, Marxch 1996.