

UNIVERSIDADE FEDERAL DE SÃO CARLOS

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**USO DE MÚLTIPLOS DESCRITORES COM
CONDIÇÕES DE CONTORNO E
VISUALIZAÇÃO HIERÁRQUICA EM CBIR**

LUIZ GUSTAVO DOS SANTOS REAL

ORIENTADOR: PROFA. DRA. MARCELA XAVIER RIBEIRO

São Carlos – SP

Março/2017

UNIVERSIDADE FEDERAL DE SÃO CARLOS

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**USO DE MÚLTIPLOS DESCRITORES COM
CONDIÇÕES DE CONTORNO E
VISUALIZAÇÃO HIERÁRQUICA EM CBIR**

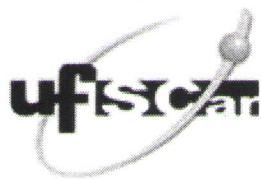
LUIZ GUSTAVO DOS SANTOS REAL

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de São Carlos, como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação, área de concentração: Banco de Dados.

Orientador: Profa. Dra. Marcela Xavier Ribeiro

São Carlos – SP

Março/2017

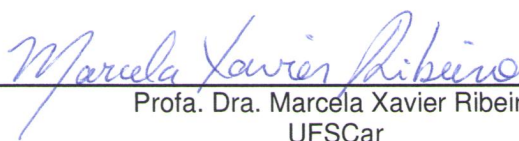


UNIVERSIDADE FEDERAL DE SÃO CARLOS


Centro de Ciências Exatas e de Tecnologia
Programa de Pós-Graduação em Ciência da Computação

Folha de Aprovação

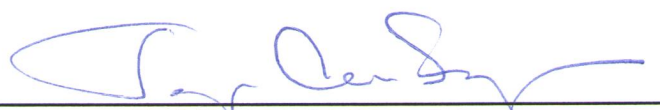
Assinaturas dos membros da comissão examinadora que avaliou e aprovou a Defesa de Dissertação de Mestrado do candidato Luiz Gustavo dos Santos Real, realizada em 23/03/2017:



Profa. Dra. Marcela Xavier Ribeiro
UFSCar



Prof. Dr. Renato Bueno
UFSCar



Prof. Dr. Joaquim Cezar Felipe
USP

A Deus, de quem provém todo conhecimento.

AGRADECIMENTOS

Aos colegas de trabalho da Diretoria Técnica de Informática da Unesp - FOA (Faculdade de Odontologia de Araçatuba) e aos diretores da faculdade, pelo apoio, paciência e compreensão.

Aos amigos que ajudaram durante todo o processo de busca pelo título de mestre. Em especial ao Flávio por todas as maratonas de programação, ao Cauê pelas traduções e revisões e ao Nelson por todo o suporte.

Aos meus avós, pelo apoio que me deram. Eu não teria conseguido sem eles. Aos meus pais, tios e irmãs, pelo incentivo e apoio.

Aos colegas da UFSCar, principalmente à minha compreensiva orientadora Profa. Dra. Marcela Xavier Ribeiro e aos docentes, discentes e demais pesquisadores ligados ao Laboratório de Banco de Dados e Engenharia de Software (LABDES) e ao Grupo de Banco de Dados (GBD).

Aos meus professores da graduação, em especial ao meu orientador Prof. Me. Anderson Pazin e ao coordenador Prof. Me. Eduardo Bergamo, que também fizeram mestrado na UFSCar, e ao meu amigo Prof. Me. João Artur Izzo, pelo apoio, incentivo e exemplo.

A todos que de alguma forma contribuíram para a realização deste projeto.

Tudo me é permitido, mas nem tudo me convém.

São Paulo

RESUMO

Múltiplos descritores têm sido testados e utilizados em sistemas de recuperação de imagens por conteúdo (do inglês, Content-Based Image Retrieval - CBIR). Cada descritor é composto de um extrator de característica associado a uma função de distância. Um extrator é em geral adequado para representar um subconjunto específico de imagens em uma base. As condições de contorno são informações utilizadas para detectar esse subconjunto. O uso de visualização em CBIR possibilita representar de uma forma pictórica a relação de similaridade existente entre as imagens presentes na base, melhorando o entendimento do usuário sobre o funcionamento do sistema CBIR, podendo este modificar parâmetros para se obter melhores resultados. É comprovado que o uso de múltiplos descritores com condições de contorno tende a melhorar a precisão das consultas em CBIR, porém não há dados sobre o impacto que o mesmo gera na visualização. Este trabalho utiliza múltiplos descritores com condições de contorno para gerar uma visualização baseada em árvore de similaridade Neighbor Joining. Os experimentos demonstraram que a qualidade da visualização pode estar relacionada com a qualidade do resultado da consulta. Quanto mais precisos os resultados de uma consulta, melhor é a organização dos elementos na visualização. No contexto de árvore de similaridade, foi comprovado que a qualidade da árvore gerada acompanha o nível de precisão medido na consulta.

Palavras-chave: Content-Based Image Retrieval, CBIR, Visualização Hierárquica, Árvore de Similaridade, Neighbor Joining, Múltiplos Descritores, Condições de Contorno.

ABSTRACT

Multiple descriptors have been tested and used in Content-Based Image Retrieval (CBIR) systems. Each descriptor consists of a feature extractor associated with a distance function. An extractor is generally suitable for representing a specific subset of images on a base. The boundary conditions are information used to detect this subset. The use of visualization in CBIR enables to represent pictorially the similarity relationship between the images present in the base, improving the user's understanding of the CBIR system, which can modify parameters to obtain better results. It is proven that the use of multiple descriptors with boundary conditions tends to improve the accuracy of CBIR queries, but there is no data on the impact that the technique generates on visualization. This work uses multiple descriptors with boundary conditions to generate a Neighbor Joining similarity tree-based view. Tests have shown that the quality of the visualization may be related to the quality of the query result. The more accurate the results of a query, the better the organization of the elements in the visualization. In the context of similarity tree, it was verified that the quality of the generated tree follow the level of precision measured in the query.

Keywords: Content-Based Image Retrieval, CBIR, Hierarchical Visualization, Similarity Tree, Neighbor Joining, Multiple Descriptors, Boundary Conditions

LISTA DE FIGURAS

2.1	Visão geral de um sistema CBIR (RIBEIRO, 2008).	20
2.2	Exemplo de histograma de cor de uma imagem com 256 níveis de cinza (BARI- ONI, 2006).	21
2.3	Exemplo de imagens (a, b, c, d) com o mesmo histograma (e) (RIBEIRO, 2008).	22
2.4	Exemplos de texturas.	22
2.5	Gráfico de precisão e revocação (BAEZA-YATES; RIBEIRO-NETO, 1999).	28
2.6	Exemplo de visualização em Grid (CáCERES, 2010)	29
2.7	Exemplo de visualização em Anéis Concêntricos (CáCERES, 2010)	30
2.8	Exemplo de visualização utilizando Transformação Geométrica (DIAS, 2013) .	31
2.9	Exemplo de <i>hyperbolic-tree</i> (LAMPING; RAO; PIROLI, 1995).	32
2.10	Exemplo de <i>cone-tree</i> (CRUZ, 2012)	32
2.11	Exemplo de visualização em Árvore (DIAS, 2013)	33
2.12	Árvore de similaridade do conjunto de 700 imagens COIL usando o algoritmo NJ. Cada ramo apresenta um agrupamento distinto, com as imagens do mesmo grupo igual ao espaço original (imagens recuperadas perto de cada ramo) (CRUZ, 2012).	34
2.13	Comparação entre uma projeção LSP e uma árvore NJ, para uma coleção de 675 documentos textuais (PAIVA, 2012).	35
2.14	Exemplo de uma árvore NJ para a coleção de imagens COREL (PAIVA, 2012). .	36
2.15	Operação de promoção de nós. Círculos preenchidos representam nós da coleção, e triângulos representam subárvores (PAIVA et al., 2011).	38

2.16	Comparação entre árvores NJ e PNJ para a coleção COREL, destacando os nós virtuais e arestas (PAIVA et al., 2011).	38
3.1	Visão geral do método proposto.	42
3.2	Exemplo implementado do método desenvolvido.	43
3.3	Exemplo de parte de uma matriz de distâncias.	44
3.4	Exemplo de grid contendo o resultado de uma consulta por similaridade.	44
3.5	Exemplo de árvore de similaridade construída pelo algoritmo NJ a partir de uma matriz de distâncias.	45
3.6	Exemplo de imagens da Base Balan.	45
3.7	Imagem 0037 da Base Balan, pertencente à classe 5.	46
3.8	Resultado das consultas da imagem 0037 da classe 5 utilizando cada descritor individualmente.	47
3.9	Gráfico de precisão e revocação medidas da consulta da imagem 0037 da classe 5 utilizando cada descritor individualmente.	47
3.10	Visualização gerada nas consulta da imagem 0037 da classe 5 utilizando cada descritor individualmente.	48
3.11	Gráfico de <i>neighborhood hit</i> da consulta da imagem 0037 da classe 5 utilizando cada descritor individualmente.	48
3.12	Média de precisão medida para consultas de imagens da Classe 2.	50
3.13	Média de precisão medida para consultas de imagens da Classe 3.	50
3.14	Média de precisão medida para consultas de imagens da Classe 5.	51
3.15	Média de precisão medida para consultas de imagens da Classe 7.	51
3.16	Árvores geradas para consulta de imagens da classe 2.	52
3.17	Neighborhood Hit para uma consulta de 5 vizinhos da classe 2.	53
3.18	Gráfico de precisão e revocação para quatro testes realizados com uma imagem da classe 2.	54
3.19	Árvores geradas para consulta de imagens da classe 5.	55
3.20	Neighborhood Hit para uma consulta de 5 vizinhos da classe 5.	56

3.21 Gráfico de precisão e revocação para quatro testes realizados com uma imagem da classe 5.	56
--	----

LISTA DE TABELAS

2.1	Comparação entre tempos de Geração do Layout (segundos), considerando as abordagens de geração rápida de árvores NJ e a técnica de projeção LSP (PAIVA et al., 2011).	40
3.1	Melhor combinação de pesos de descritores obtida para cada classe testada. . .	49
A.1	Lista de opções de linha de comando para extração de características com jFeatureLib	67

SUMÁRIO

CAPÍTULO 1 – INTRODUÇÃO	15
1.1 Contexto	15
1.2 Motivação	16
1.3 Objetivos	17
1.4 Organização do Trabalho	17
CAPÍTULO 2 – REFERENCIAL TEÓRICO	19
2.1 Considerações Iniciais	19
2.2 Recuperação por Conteúdo	19
2.3 CBIR	20
2.3.1 Extração de Características	20
2.3.1.1 Cor	21
2.3.1.2 Textura	22
2.3.1.3 Forma	23
2.3.2 Similaridade	23
2.3.3 Múltiplos Descritores e Condições de Contorno	24
2.3.4 Avaliação dos resultados de uma consulta em CBIR	26
2.4 Visualização de Dados e Imagens	28
2.4.1 Considerações Iniciais	28
2.4.2 Técnicas de Visualização de Imagens	29

2.4.3	Árvore de Similaridade Neighbor Joining	32
2.4.3.1	Melhorias na construção de árvores de similaridade	37
2.5	Considerações Finais	40
CAPÍTULO 3 – TRABALHO DESENVOLVIDO		41
3.1	Considerações Iniciais	41
3.2	Visão Geral do Método Desenvolvido	41
3.3	Experimentos e Resultados	44
3.3.1	Experimento 1	46
3.3.1.1	Testes	46
3.3.2	Experimento 2A	48
3.3.3	Experimento 2B - Neighborhood Hit	53
3.3.3.1	Teste com imagens da classe 2	53
3.3.3.2	Teste com imagens da classe 5	53
3.4	Considerações Finais	54
CAPÍTULO 4 – CONCLUSÕES E TRABALHOS FUTUROS		57
4.1	Considerações Iniciais	57
4.2	Principais Contribuições	58
4.3	Trabalhos Futuros	59
4.4	Considerações Finais	59
REFERÊNCIAS		60
GLOSSÁRIO		64
APÊNDICE A – UTILIZANDO A BIBLIOTECA JFEATURELIB PARA EXTRAIR VETORES DE CARACTERÍSTICAS DE IMAGENS EM JAVA		65
A.1	Extratores de características implementados	65

A.2	Como utilizar	66
A.2.1	Extrair características de várias classes de imagens:	66
A.2.2	Opções	67
A.3	Código de demonstração e Wiki	67

Capítulo 1

INTRODUÇÃO

1.1 Contexto

Sistemas de Recuperação de Imagens por Conteúdo (*Content-Based Image Retrieval - CBIR*) permitem que sejam realizadas consultas em bases de imagens, retornando as imagens com maior grau de similaridade em relação à imagem de consulta (KUMAR et al., 2013). Essa similaridade entre as imagens da base e a imagem de consulta é verificada através de características como cor, forma e textura, que são extraídas e armazenadas em uma estrutura conhecida como vetor de características (BUENO et al., 2011). Depois, uma métrica estipulada (função de distância) avalia a distância entre os vetores selecionados.

A combinação de um algoritmo extrator de característica e uma função de distância é geralmente referido na literatura como descritor (TORRES et al., 2009; BARROSO et al., 2015).

Os sistemas CBIR possuem o Gap Semântico (inconsistência entre a percepção do ser humano na avaliação de similaridade entre imagens comparada com a computada por sistemas CBIR) como uma das suas principais limitações (LIU et al., 2007). Para diminuir o gap semântico em consultas por similaridade, o uso de combinações de múltiplos descritores tem sido considerado por demonstrar bons resultados (BARROSO et al., 2015), pois subconjuntos de imagens de um mesmo conjunto podem ser melhor representados por características diferentes.

Barroso et al. (2013) propõe o uso de condições de contorno para encontrar esses subconjuntos e escolher a melhor combinação de descritores para cada um deles. Condições de contorno são quaisquer informações associadas à imagem que possam ser utilizadas para delimitar subconjuntos de imagens.

A última etapa de um sistema CBIR é a visualização dos resultados de uma consulta. Cada sistema CBIR possui uma forma diferente de apresentar esses resultados (CÁCERES, 2010).

Um dos desafios das pesquisas conhecidas na área de recuperação de imagens é exibir grandes quantidades de imagens de maneira intuitiva para o usuário. Vários autores defendem que uma representação visual auxilia a análise e a identificação de padrões existentes na coleção de dados, permitindo que o usuário entenda o comportamento do espaço de características dos dados (CRUZ, 2012).

Uma possibilidade de visualização de imagens é disponibilizá-las de forma hierárquica (ELER et al., 2009). Ou seja, fazer uma projeção das imagens através da representação encontrada na fase de extração de características, levando-se em conta uma hierarquia estabelecida pela similaridade entre as mesmas. Desta forma, a construção dessas relações de similaridade pode assumir uma estrutura semelhante à de uma árvore, a qual pode ser estruturada segundo diferentes metodologias.

1.2 Motivação

É comprovado que o uso de múltiplos descritores com condições de contorno tende a melhorar a precisão das consultas em CBIR (BARROSO et al., 2015), porém não foi analisado o impacto que esse uso gera na visualização.

A qualidade da visualização pode estar relacionada com a qualidade da resposta da consulta de sistemas CBIR. Assim, a primeira hipótese deste trabalho é:

Hipótese 1: Quanto mais precisos os resultados de uma consulta, melhor pode ser a organização das imagens na visualização.

No caso da hipótese 1 ser verdadeira, definimos:

Hipótese 2: O uso de múltiplos descritores com condição de contorno pode melhorar a qualidade da visualização hierárquica de imagens em CBIR, pois já foi comprovado que seu uso melhora o resultado de consultas por similaridade de imagens.

Dentre as técnicas de visualização de imagens existentes, a visualização hierárquica foi escolhida em detrimento as outras técnicas, por exemplo, as técnicas baseadas em projeções multidimensionais, porque até o momento são poucos trabalhos na literatura que tratam da visualização hierárquica em sistemas CBIR (CRUZ, 2012; PAIVA, 2012) e não foi encontrado nenhum trabalho que trata desse assunto aliado ao uso de múltiplos descritores com condições de contorno (BARROSO et al., 2013, 2015).

Ademais, a noção de hierarquia pode indicar as imagens mais representativas de uma classe, o que pode ser interessante em CBIR.

1.3 Objetivos

Um dos objetivos do trabalho é demonstrar que a qualidade da visualização está relacionada com a qualidade da consulta (visualização de má qualidade está relacionada a um resultado da consulta de qualidade ruim; visualização de boa qualidade está relacionada a um resultado da consulta também de qualidade boa).

Também espera-se demonstrar que o uso de múltiplos descritores com condições de contorno melhora a qualidade da visualização hierárquica de imagens em CBIR. O uso de combinações de múltiplos descritores pode representar melhor as características distintas encontradas em subconjuntos de imagens de uma base de imagens. Já as condições de contorno são utilizadas para encontrar esses subconjuntos e escolher a melhor combinação de descritores para cada um deles.

A técnica de visualização hierárquica escolhida para o trabalho é a *Neighbor Joining* (SAITOU; NEI, 1987). É um dos métodos mais utilizados para a construção de uma árvore filogenética. Essa técnica, adaptada para a criação de árvores de similaridade, utiliza a ideia de encontrar pares de instâncias mais próximas em uma coleção de dados, de modo a minimizar o comprimento e o número de ramos da árvore gerada (PAIVA, 2012). Ela tem sido utilizada no contexto de classificação visual de dados (ELER et al., 2008, 2009; PAIVA et al., 2011; CRUZ, 2012; PAIVA, 2012).

Ao posicionar os elementos da consulta em ramos de uma árvore, a similaridade é organizada em níveis, representando uma abordagem natural para a interpretação de graus de similaridade.

A medida de qualidade da visualização de imagens usada no trabalho é a *neighborhood hit* (PAULOVICH et al., 2008), que mede a porcentagem de vizinhos mais próximos de uma imagem na visualização, que pertencem à mesma classe. A precisão final representa a média das precisões para cada ponto. Isso permite avaliar numericamente a separabilidade das classes pré-existentes no *layout*.

1.4 Organização do Trabalho

Primeiramente, o capítulo 2 trata de todos os conceitos relacionados ao trabalho, como CBIR, uso de múltiplos descritores com condições de contorno, visualização de dados e imagens e *Neighbor Joining*. No capítulo 3 são apresentados alguns trabalhos relacionados ao uso de múltiplos descritores e visualização hierárquica. O capítulo 4 apresenta o trabalho desenvol-

vido, inclusive os experimentos realizados e seus resultados. Por último, o capítulo 5 apresenta as conclusões e trabalhos futuros.

Capítulo 2

REFERENCIAL TEÓRICO

2.1 Considerações Iniciais

Neste capítulo são apresentados os principais conceitos relacionados ao projeto de mestrado. Primeiramente é introduzido o conceito de Recuperação por Conteúdo e aprofundado especificamente o conceito de recuperação de imagens por conteúdo, expondo todas as suas etapas, terminando com uma breve introdução de visualização de dados e imagens. Em seguida é apresentado o conceito de Visualização Hierárquica, utilizando árvore de similaridade *Neighbor Joining*, método utilizado neste trabalho de mestrado. Por fim, são apresentados os Trabalhos Correlatos, que justificam a escolha do tema deste projeto.

2.2 Recuperação por Conteúdo

Dados complexos, que são dados que não são de tipos simples (inteiro, *string*, etc.), são armazenados atualmente em bancos de dados como atributos de tipo BLOB (*Binary Large Object*). Estando em forma binária, esses dados não são interpretáveis, necessitando de outros atributos textuais e/ou numéricos para auxiliar em sua recuperação (BARIONI, 2006; RIBEIRO, 2008).

Para comparar dados multimídia, são utilizadas as características extraídas desses dados. Esse tipo de busca, através das características, é denominado Recuperação por Conteúdo (BARIONI, 2006). A recuperação de dados complexos por conteúdo é utilizada para vários tipos de dados, como imagens e áudio.

2.3 CBIR

Recuperação de imagem por conteúdo, ou CBIR, é uma técnica criada para auxiliar o usuário a encontrar imagens similares à imagem de consulta em diversos tipos de aplicações multimídia (AKGÜL et al., 2011; KUMAR et al., 2013).

Os sistemas CBIR realizam consultas em uma base de imagens com base no critério de semelhança. Esses sistemas envolvem um conjunto de métodos que processam as imagens a fim de obter a representação das mesmas em vetores de características. Esses vetores de características são utilizados no lugar da imagem para a execução de consultas (RIBEIRO, 2008).

Em geral, os sistemas CBIR envolvem o pré-processamento da imagem, a extração de características, técnicas de indexação e avaliação de similaridade entre as imagens. O resultado de uma consulta em um sistema CBIR é um conjunto de imagens ordenadas de acordo com sua similaridade à imagem de consulta (DIAS, 2013).

A visão geral com os passos executados por um sistema CBIR podem ser vistos na Figura 2.1.

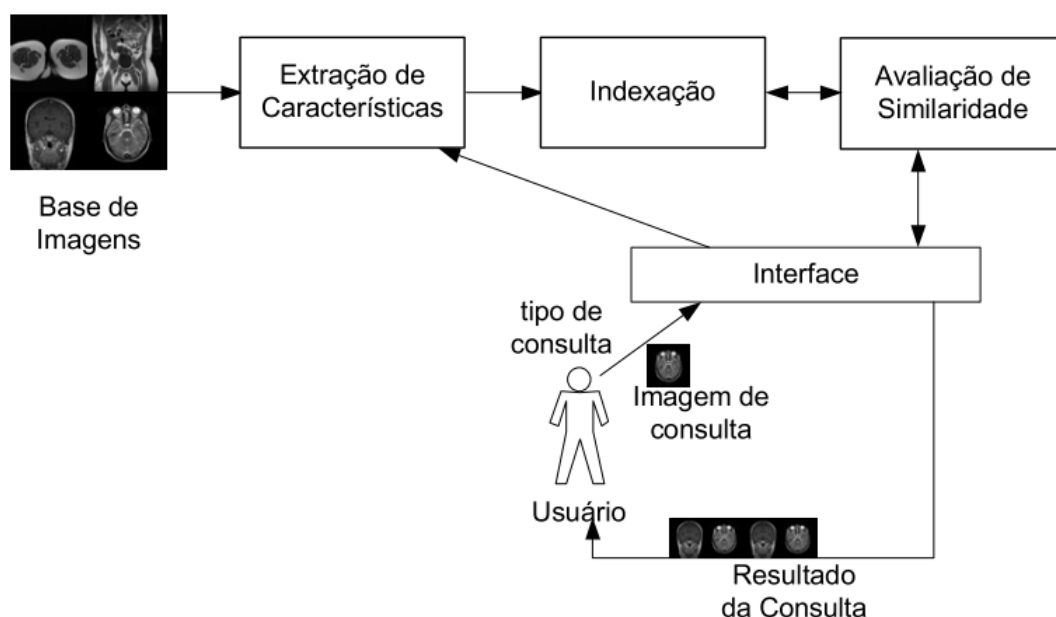


Figura 2.1: Visão geral de um sistema CBIR (RIBEIRO, 2008).

2.3.1 Extração de Características

Para que as imagens possam ser trabalhadas computacionalmente, é necessário encontrar uma representação numérica que sintetize o conteúdo das mesmas, levando-se em consideração características visuais específicas, como cor, textura e forma (RIBEIRO, 2008).

As imagens podem ser submetidas a um pré-processamento (transformada, segmentação) ou a extratores de características, que são algoritmos que extraem as características visuais da imagem. Essas características então são utilizadas para criar o vetor de características (BHATT; KANKANHALLI, 2011).

2.3.1.1 Cor

Dentre as características utilizadas na recuperação de imagens, a cor é a mais conhecida, sendo adotada desde os primeiros sistemas CBIR propostos. Uma caracterização global da imagem pode ser obtida através do agrupamento de componentes de cor de cada pixel num histograma (AKGÜL et al., 2011). Um histograma de cor possui informações do número de *pixels* para cada cor (ou nível de cinza) em uma imagem, podendo ser usado na comparação com outro histograma de cor pela soma das diferenças absolutas ou quadráticas do número de *pixels* de cada cor (RIBEIRO, 2008). Um exemplo de histograma de cor pode ser visto na Figura 2.2.

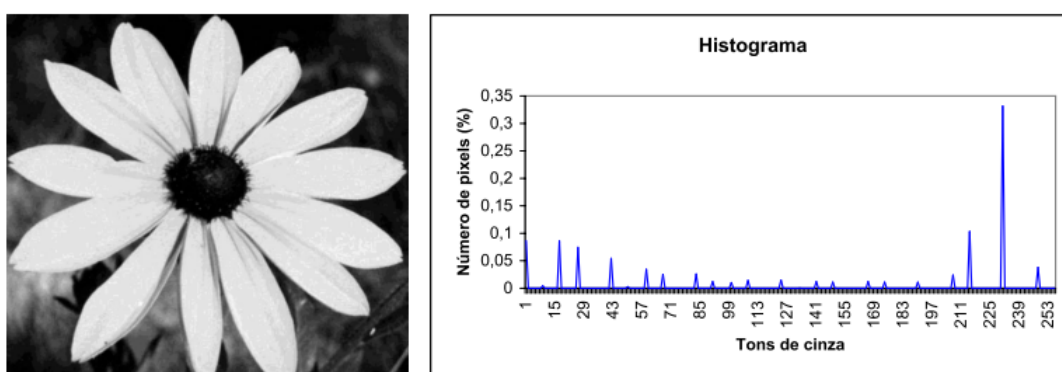


Figura 2.2: Exemplo de histograma de cor de uma imagem com 256 níveis de cinza (BARIONI, 2006).

Apesar de ser simples, o histograma de cor pode elevar o custo computacional e se tornar inadequado na indexação, caso tenha um grande número de cores. Ele também possui baixa capacidade de discriminação, ou seja, não trata as informações especiais dos *pixels*, assim, imagens diferentes podem ter distribuições de cores semelhantes (BARIONI, 2006), como pode ser observado na Figura 2.3.

Enquanto a cor é uma das características visuais mais utilizadas para a descrição de imagens, a maioria das imagens da área da saúde são em tons de cinza. Em alguns casos a cor pode ser utilizada no diagnóstico, como em oftalmologia, patologia e dermatologia. Entretanto, para a maioria das imagens dessa área, as características de cor não são úteis (AKGÜL et al., 2011).

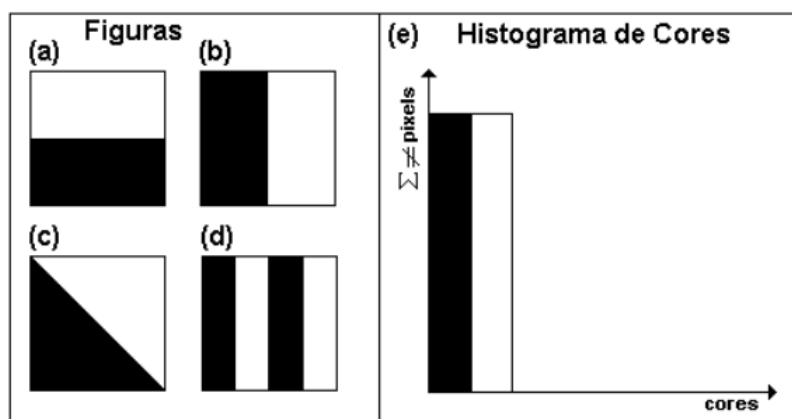


Figura 2.3: Exemplo de imagens (a, b, c, d) com o mesmo histograma (e) (RIBEIRO, 2008).

2.3.1.2 Textura

Textura é uma característica relacionada à organização espacial dos valores de pixel de uma região da imagem (AKGÜL et al., 2011). Não existe uma definição formal para textura, de forma intuitiva este extrator fornece medidas de propriedades tais como suavidade, aspereza, e regularidade (HUANG; DAI, 2003). Textura se refere à repetição de elementos básicos da imagem chamados *textels* (RIBEIRO, 2008). Alguns exemplos de texturas podem ser vistos na Figura 2.4.

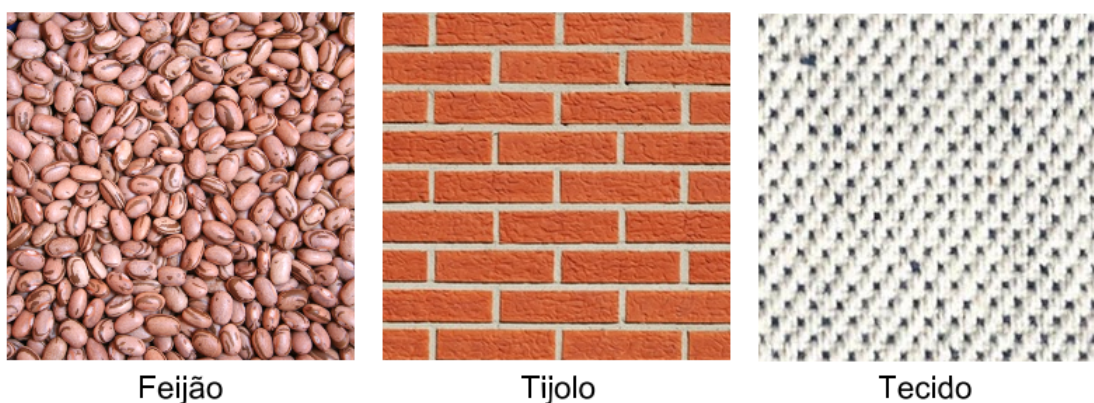


Figura 2.4: Exemplos de texturas.

Os métodos utilizados para representar a textura podem ser classificados nas seguintes categorias: estatísticos, sintáticos e híbridos (SONKA; HLAVAC; BOYLE, 1999).

Os métodos estatísticos inclinam-se a descrever texturas pequenas. Eles caracterizam a textura calculando propriedades como contraste, granularidade, periodicidade, entre outras. Matrizes de co-ocorrência, transformadas de Gabor e Wavelets são exemplos de métodos estatísticos.

Os métodos sintáticos identificam elementos e indicam a disposição espacial dos mesmos. Já nos métodos híbridos, compostos pela combinação dos dois métodos anteriores, os elemen-

tos são exatamente definidos e o relacionamento espacial entre esses elementos é baseado em probabilidades.

Os métodos sintáticos e híbridos inclinam-se a descrever texturas regulares e não são utilizados com a mesma frequência que os métodos estatísticos.

Na área da saúde, os extratores baseados em textura tornam-se particularmente importantes, já que podem refletir os detalhes contidos dentro da estrutura de imagem. Por exemplo, cada tipo de lesão pode conter um conjunto de características distinto (AKGÜL et al., 2011).

2.3.1.3 Forma

Uma das possibilidades encontradas na atividade de extração de características é a extração de formas presentes em imagens. O termo forma se refere à informação que se pode deduzir diretamente de imagens e que não é representada por cor ou textura, com bordas / contornos (AKGÜL et al., 2011).

Alguns métodos para representar forma são os métodos geométricos para detecção de borda e os métodos escalares. Os métodos geométricos detectam o comprimento de borda, curvatura e assinatura (sequência de distâncias entre pontos da borda). Os escalares detectam região, como área, excentricidade (razão entre o maior e o menor eixo, por exemplo) e retangularidade (SONKA; HLAVAC; BOYLE, 1999).

Dentre os métodos utilizados para representar imagens com base nos contornos presentes, estão os momentos de Zernike (KHOTANZAD; HONG, 1990) e o modelo de contornos ativos Snakes (KASS; WITKIN; TERZOPOULOS, 1988).

Segundo Barioni (2006), um aspecto importante na recuperação por forma é a escolha da característica a ser utilizada para delimitar as formas nas imagens. É comum que seja utilizada a cor, porém, em imagens médicas a textura é uma característica mais interessante.

A escolha de técnicas de extração de características está relacionada com o contexto em que um determinado conjunto de imagens está inserido, possibilitando encontrar uma representação numérica que atenda as características visuais específicas deste contexto (DESELAERS; KEYSERS; NEY, 2004, 2008).

2.3.2 Similaridade

As consultas realizadas por aplicações que manipulam dados complexos são, na maioria dos casos, baseadas na noção de similaridade (BARIONI, 2006). No domínio de imagens, o conceito

de similaridade é o mais adequado, pois o objetivo é encontrar imagens semelhantes e não idênticas. Na área médica, por exemplo, pode haver a necessidade de encontrar imagens com uma lesão similar à lesão presente na imagem de consulta, o que não seria possível utilizando o conceito de igualdade, pois até duas imagens de uma mesma lesão tiradas em momentos diferentes possuem características distintas.

As medidas de similaridade de imagem geralmente avaliam a distância entre vetores de características. As distâncias mais curtas correspondem a maior similaridade e distâncias maiores correspondem a maior dissimilaridade. A comparação entre os vetores é feita a partir do uso de funções de distância sobre os vetores de um par de elementos, retornando um valor numérico que corresponde a distância entre esses dois elementos (BARROSO et al., 2013).

Algumas funções de distância utilizadas são: a distância Manhattan, ou L_1 (Equação 2.1), distância Euclidiana, ou L^2 (Equação 2.2) e a distância Canberra (Equação 2.3). É calculada a distância (d) entre dois vetores $P = (p_1, p_2, \dots, p_n)$ e $Q = (q_1, q_2, \dots, q_n)$.

$$d_{(p,q)} = \|p - q\| = \sum_{i=1}^n |p_i - q_i| \quad (2.1)$$

$$d_{(p,q)} = \sqrt{\sum_{i=1}^n (p_i - q_i)^2} \quad (2.2)$$

$$d_{(p,q)} = \sum_{i=1}^n \frac{|p_i - q_i|}{|p_i| + |q_i|} \quad (2.3)$$

2.3.3 Múltiplos Descritores e Condições de Contorno

A combinação de um extrator de característica e uma função de distância é geralmente referenciada na literatura como descritor (TORRES et al., 2009; BARROSO et al., 2013, 2015).

Um problema enfrentado por sistemas CBIR é o Gap Semântico, que é a inconsistência existente na percepção do ser humano na avaliação de similaridade entre imagens pelo sistema (LIU et al., 2007).

Para se combater o gap semântico em consultas por similaridade, o uso de combinações de múltiplos descritores tem sido considerado por demonstrar bons resultados (BARROSO et al., 2015), pois subconjuntos de imagens de um mesmo conjunto podem ser melhor representados por características diferentes. Barroso et al. (2013) propõe o uso de condições de contorno para encontrar esses subconjuntos e escolher a melhor combinação de descritores para cada um

deles.

Condições de contorno em consultas por similaridade de imagens, conforme entendido por Barroso et al. (2013), é qualquer informação associada à imagem que possa ser utilizada para delimitar subconjuntos de imagens. Por exemplo, para consultas contendo imagens médicas as condições de contorno podem ser informações como: hipótese de diagnóstico, laudos radiológicos ou mesmo o tipo de exame.

Método para combinação de múltiplos descritores partindo do melhor descritor

Considerando as imagens x e y , e as métricas $\delta_1, \dots, \delta_n$ dos descritores a serem combinados, definidas sobre os domínios dos respectivos vetores de características, a composição da função de distância para combinação dos descritores δ_c é dada pela Equação (2.4):

$$\delta_c(x, y) = \sum_{i=1}^n \omega_i * \frac{\delta_i(x, y)}{dmax_i} \quad (2.4)$$

onde ω_i é o peso atribuído ao i -ésimo descritor e $dmax_i$ representa a distância máxima obtida pela comparação entre todos os elementos do conjunto utilizando este mesmo descritor, que normaliza as distâncias entre 0 e 1.

Os algoritmos de busca exaustiva para combinação linear utilizados em Barroso et al. (2013) testam todas as combinações possíveis (a partir de uma quantidade determinada de possíveis pesos) a fim de encontrar a melhor combinação linear entre os descritores. O extenso espaço de busca torna o processo bastante custoso computacionalmente, com um crescimento exponencial de acordo com o número de descritores utilizados.

Essa técnica foi então otimizada (BARROSO et al., 2015), porpondo-se a realização de combinações lineares dos descritores a partir do descritor que apresente o melhor desempenho individual, avaliado pela sua precisão média na fase de treinamento. Este descritor assume então o peso $\omega_1 = 1$.

Em uma fase que precede ao balanceamento são definidos os descritores candidatos que poderão ser combinados. Os descritores podem ser definidos por um especialista no domínio da aplicação. Os descritores candidatos são então ranqueados pela precisão média. Partindo do descritor melhor avaliado, inicia-se a combinação com o segundo descritor melhor avaliado. Considerando as imagens x e y , e as métricas dos descritores candidatos ordenados, $\delta_1, \dots, \delta_n$, definidas sobre os domínios dos respectivos vetores de características, a composição da função de distância δ_{c_1} para combinação do primeiro descritor com o segundo descritor é dada pela Equação (2.5):

$$\delta_{c_1}(x, y) = \frac{\delta_1(x, y)}{dmax_1} + \omega_2 * \frac{\delta_2(x, y)}{dmax_2} \quad (2.5)$$

onde ω_2 é o peso dado ao segundo descritor e $dmax_1$ e $dmax_2$ representam as distâncias máximas encontradas pela comparação entre todos os elementos do conjunto utilizando o respectivo descritor, utilizadas para normalizar a participação de cada descritor na combinação.

Após a definição da melhor combinação entre os dois primeiros descritores, parte-se para a combinação com o terceiro melhor descritor, utilizando a Equação (2.6):

$$\delta_{c_2}(x, y) = \delta_{c_1}(x, y) + \omega_3 * \frac{\delta_3(x, y)}{dmax_3} \quad (2.6)$$

onde ω_3 é o peso dado ao terceiro melhor descritor e $dmax_3$ representa a distância máxima encontrada pela comparação entre todos os elementos do conjunto utilizando este descritor.

Dessa forma, os descritores são combinados aos pares sequencialmente, sempre testando os pesos do próximo descritor $i + 1$ na combinação com o melhor balanceamento já definido para os i descritores anteriores, de acordo com a Equação (2.7):

$$\delta_{c_i}(x, y) = \delta_{c_{i-1}}(x, y) + \omega_{i+1} * \frac{\delta_i(x, y)}{dmax_{i+1}} \quad (2.7)$$

2.3.4 Avaliação dos resultados de uma consulta em CBIR

Para se avaliar se os elementos presentes no resultado de uma consulta em CBIR efetivamente representam o objeto de consulta, ou seja, se são relevantes, utiliza-se as medidas de revocação (*recall*) e precisão (*precision*) (BAEZA-YATES; RIBEIRO-NETO, 1999). Com essas informações, é possível construir um gráfico de precisão e revocação (P&R).

Considere uma dada consulta, onde R seja o número total de itens relevantes existentes na base, R_R o número total de itens relevantes recuperados e I_R o total de itens recuperados.

Revocação é a fração do conjunto de elementos relevantes (R) que foram recuperados na consulta.

$$Recall = \frac{R_R}{R} \quad (2.8)$$

Precisão é a fração do conjunto de elementos recuperados (I_R) que são relevantes.

$$Precision = \frac{R_R}{I_R} \quad (2.9)$$

O primeiro passo para se calcular P&R é ordenar os elementos recuperados da base de acordo com sua distância em relação ao objeto de consulta.

Como exemplo (BAEZA-YATES; RIBEIRO-NETO, 1999), considere uma determinada consulta q onde existe na base um conjunto R_q de 10 elementos relevantes, composto da seguinte forma:

$$R_q = \{e_5, e_{13}, e_{17}, e_{20}, e_{31}, e_{36}, e_{42}, e_{47}, e_{55}, e_{61}\}$$

Considere que um algoritmo de busca retornou um conjunto de elementos I_q , referentes à consulta q , cujos elementos e suas respectivas relevâncias são:

- | | |
|----------------------|------------------|
| 1. $e_{42} \bullet$ | 2. e_{54} |
| 3. $e_{17} \bullet$ | 4. e_{15} |
| 5. e_2 | 6. $e_5 \bullet$ |
| 7. e_{25} | 8. e_{27} |
| 9. $e_{13} \bullet$ | 10. e_{52} |
| 11. $e_{55} \bullet$ | 12. e_{67} |

Os elementos relevantes à consulta q estão indicados com o símbolo \bullet . Analisando o conjunto I_q dos elementos recuperados, verifica-se que o primeiro elemento (e_{42}) é um dos elementos relevantes à consulta. Neste momento, o valor de precisão é de 100% - pois todos os elementos analisados até aqui, são relevantes à consulta - e o valor de revocação é de 10% - pois até este ponto, apenas um elemento relevante foi recuperado dentre um conjunto de dez elementos.

O próximo elemento relevante da lista é o terceiro elemento (e_{17}). Para este elemento, o valor de precisão é de aproximadamente 66% (dois elementos relevantes em três verificados) e o valor de revocação é de 20% (dois entre dez elementos relevantes). A análise prossegue desta maneira até que todos os elementos relevantes sejam verificados. Os valores de precisão e revocação são descritos no gráfico de P&R, conforme visto na Figura 2.5.

Para uma avaliação efetiva dos resultados obtidos por um determinado sistema CBIR, é necessário que diversas operações de consulta sejam realizadas, construindo assim uma curva de precisão e revocação que represente a média dos desempenhos das diversas consultas realizadas. Isso geralmente é feito calculando-se valores de precisão para níveis determinados de revocação, como a cada intervalo de 10% de revocação, por exemplo.

A avaliação de desempenho é realizada observando-se as curvas obtidas. Quanto mais próxima do topo do gráfico a curva estiver, melhor será o resultado da operação de busca.

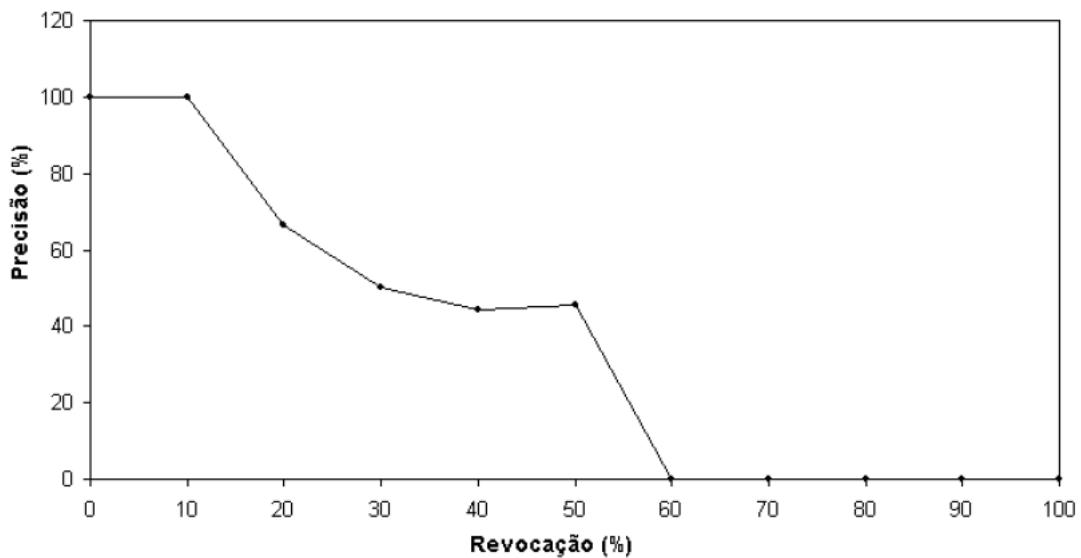


Figura 2.5: Gráfico de precisão e revocação (BAEZA-YATES; RIBEIRO-NETO, 1999).

Sendo assim, a curva ideal para uma consulta apresenta 100% de precisão para todos os valores de revocação.

2.4 Visualização de Dados e Imagens

2.4.1 Considerações Iniciais

Segundo Cruz (2012), a visualização de dados refere-se a uma representação gráfica na qual dados multidimensionais são posicionados na tela de um computador, por exemplo, a fim de refletir suas relações de similaridade. Assim, esses dados podem ser analisados e reportados atendendo a necessidades específicas e suas aplicações.

A representação visual de coleções de dados comunica claramente ao usuário o conteúdo informacional desses dados, reduzindo o trabalho cognitivo necessário para realizar diversas tarefas. Além disso, o usuário se torna um agente ativo no processo de mineração das informações, pois consegue, além de visualizar as relações entre os dados, interagir com o layout, tendo uma visão geral, ou concentrando-se em fenômenos particulares. Isso resulta em um processo exploratório mais rápido e com resultados melhores, em especial quando procedimentos automáticos falham (PAIVA, 2012).

A visualização de grandes conjuntos de dados de imagens apresenta vários desafios. Um deles é a identificação de informações ou conhecimento relevante na visualização, permitindo a discriminação dos objetos em diferentes classes ou grupos sem perder a capacidade de identificar elementos não facilmente agrupáveis. Outro desafio é encontrar representações simples que

sejam fiéis ao conjunto de dados completo e às relações entre eles, considerando a limitação da baixa dimensionalidade dos dispositivos de apresentação. Neste sentido existem diversos métodos para representar dados multidimensionais (CRUZ, 2012). No presente trabalho serão utilizadas as estruturas baseadas em árvore.

2.4.2 Técnicas de Visualização de Imagens

Neste capítulo são apresentadas algumas técnicas de visualização de imagens geralmente utilizadas em sistemas CBIR.

Grid

Esta é a técnica mais comum para a visualização de resultados de consultas em sistemas CBIR. As imagens são apresentadas de maneira matricial, levando em conta o padrão ocidental (da esquerda para a direita, de cima para baixo, como na Figura 2.6) de acordo com seu grau de relevância (CÁCERES, 2010).

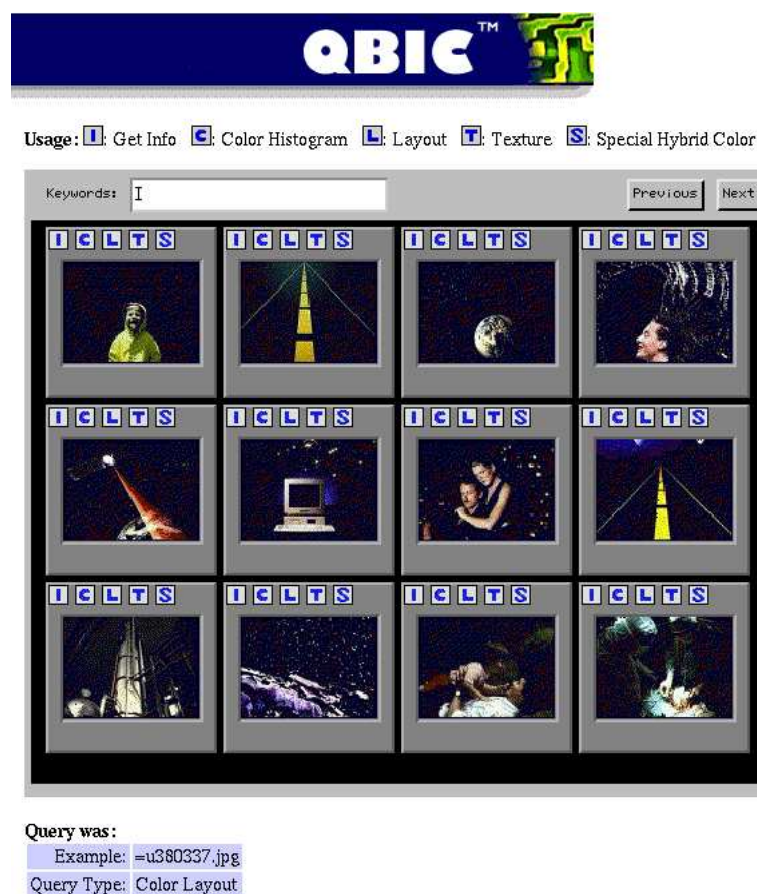


Figura 2.6: Exemplo de visualização em Grid (CÁCERES, 2010)

Anéis Concêntricos

Nessa técnica, as imagens são distribuídas sobre uma série de anéis concêntricos, como pode ser visto na Figura 2.7. Cada anel é o círculo, definido em coordenadas polares, formado pelo conjunto de pontos com raio $r = k$, onde k é uma constante. A constante k incrementa-se para cada anel. Este incremento diminui a cada anel, ou seja, anéis sucessivos estarão mais próximos conforme k é incrementado (CÁCERES, 2010).

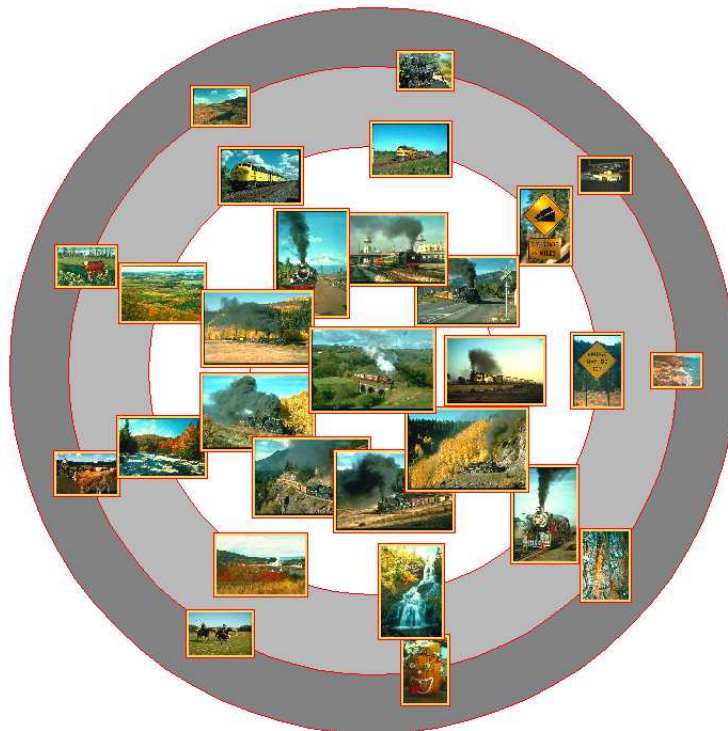


Figura 2.7: Exemplo de visualização em Anéis Concêntricos (CÁCERES, 2010)

Transformação Geométrica

A visualização de dados multidimensionais através da utilização de transformações geométricas implica em realizar um mapeamento da informação em um espaço visual de dimensionalidade inferior, mantendo, na medida do possível, as relações de distância definidas no espaço original. O subespaço de projeção é, normalmente, unidimensional, bidimensional ou tridimensional para facilitar a compreensão humana (DIAS, 2013). Um exemplo pode ser visto na Figura 2.8.

Árvore

Uma forma de mapeamento de dados é a imposição de uma hierarquia sobre algum relacionamento significativo dos dados, como o grau da similaridade calculado sobre as coordenadas da representação vetorial (CRUZ, 2012). Para se expressar dados hierárquicos multidimensionais na visualização, é comum utilizar árvores (PAIVA et al., 2011). A hierarquia apresentada em forma de árvore, levando em conta o grau de similaridade, é chamada de “árvore de similaridade”.

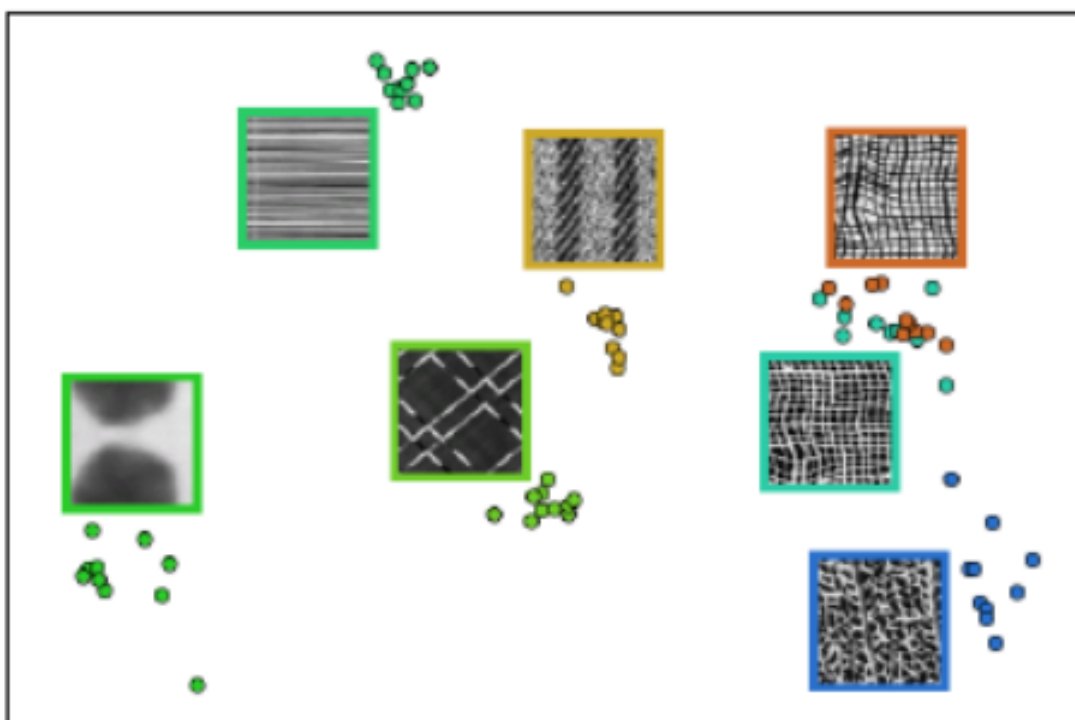


Figura 2.8: Exemplo de visualização utilizando Transformação Geométrica (DIAS, 2013)

dade”.

As árvores têm uma estrutura hierárquica que organiza as entidades de dados por meio de conexões pai-filho. Para ser uma árvore clássica, cada entidade filho deve ter apenas um pai e cada filho posiciona-se “embaixo” de um ancestral comum. No qual os nós representam objetos e as arestas representam a ligação entre eles (CRUZ, 2012). Dentre as técnicas encontradas para se representar hierarquia em árvores, destaca-se: *hyperbolic-tree* (LAMPING; RAO; PIROLI, 1995), que pode ser vista na Figura 2.9, e *cone-tree* (ROBERTSON; MACKINLAY; CARD, 1991), apresentada na Figura 2.10.

Uma outra abordagem encontrada na literatura para a construção de uma árvore de similaridade é através do conceito de árvore filogenética (DIAS, 2013), utilizando o método *Neighbor Joining* (NJ) (SAITOU; NEI, 1987), através do cálculo da matriz de distância. Um exemplo dessa técnica pode ser observado na Figura 2.11.

Poucos trabalhos na literatura tratam da visualização hierárquica em sistemas CBIR (CRUZ, 2012; PAIVA, 2012). A técnica NJ tem sido muito utilizada no contexto de classificação visual de dados (ELER et al., 2008, 2009; PAIVA et al., 2011; CRUZ, 2012; PAIVA, 2012). Por esse motivo ela foi escolhida neste trabalho.

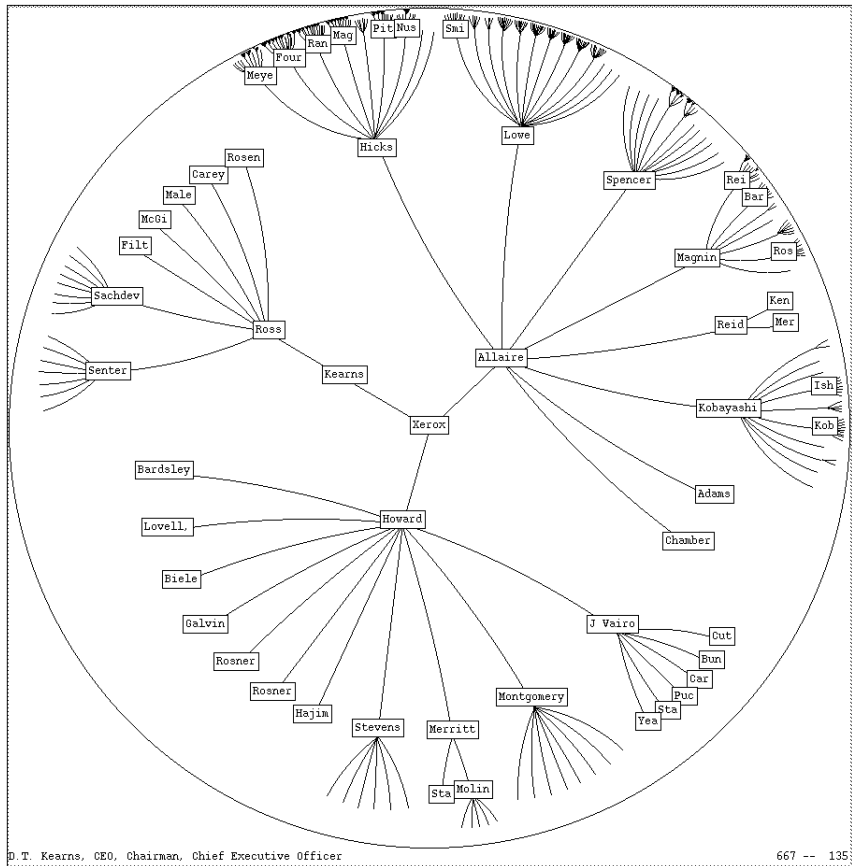


Figura 2.9: Exemplo de *hyperbolic-tree* (LAMPING; RAO; PIROLI, 1995).

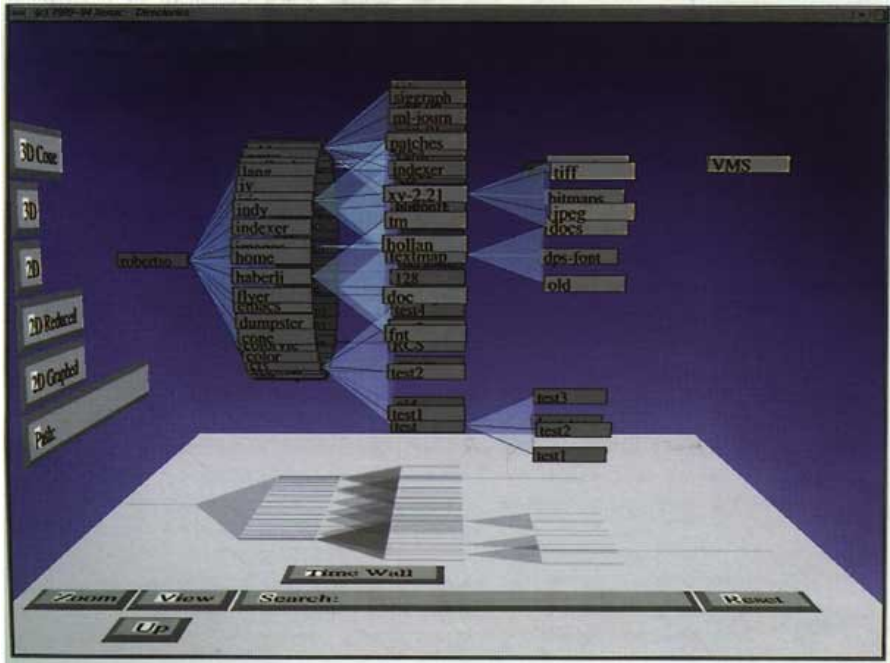


Figura 2.10: Exemplo de *cone-tree* (CRUZ, 2012)

2.4.3 Árvore de Similaridade Neighbor Joining

Árvores de similaridade podem ser construídas a partir de matrizes de distâncias, por meio de algoritmos de reconstrução de árvores filogenéticas. Essas árvores buscam refletir relações

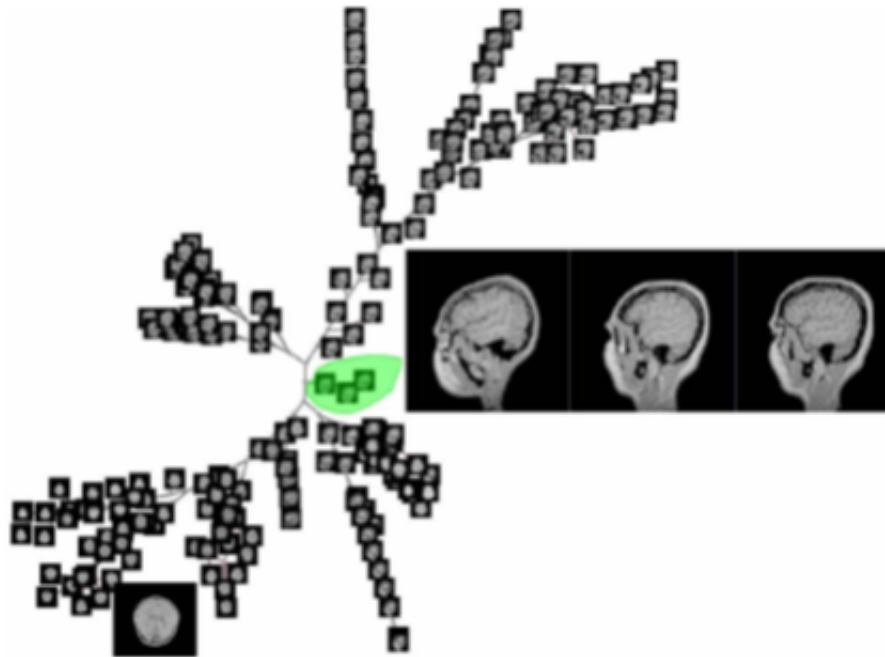


Figura 2.11: Exemplo de visualização em Árvore (DIAS, 2013)

de similaridade a partir de princípios de evolução mínima - que tenta minimizar a soma dos tamanhos de todos os nós da árvore - entre os organismos vivos, codificando relações de ancestralidade entre espécies. Assim, em uma árvore filogenética as folhas representam os objetos originais da matriz de distâncias, os nós internos são ancestrais hipotéticos e os ramos (arestas) indicam a distância evolutiva entre os objetos (CRUZ, 2012). Uma ideia dessa árvore pode ser vista na Figura 2.12.

Um dos métodos mais utilizados para a construção de uma árvore filogenética é o NJ, criado por Saitou e Nei (1987). Uma árvore filogenética tem o propósito de expressar relações de similaridade evolucionária aplicando o princípio da evolução mínima (DIAS, 2013). Essa técnica, adaptada para a criação de árvores de similaridade, utiliza a ideia de encontrar pares de instâncias mais próximas em uma coleção de dados, de modo a minimizar o comprimento e o número de ramos da árvore gerada (PAIVA, 2012).

O Algoritmo 1 apresenta as etapas do método NJ. É gerada uma árvore sem raiz, representando apenas as distâncias entre as instâncias. O algoritmo recebe como entrada uma matriz de distâncias (similaridade) $D_{n \times n}$, com as distâncias entre todos os pares de instâncias da coleção, de acordo com alguma medida de similaridade, e produz uma árvore de similaridade com n nós-folha, e $n-2$ nós virtuais com grau 3. R_i é a distância média do nó i para todos os outros nós em D , capturando a noção de mudança evolucionária. Em cada passo do algoritmo, os dois nós mais próximos em D são removidos da matriz, e substituídos pelo nó virtual x , para o qual

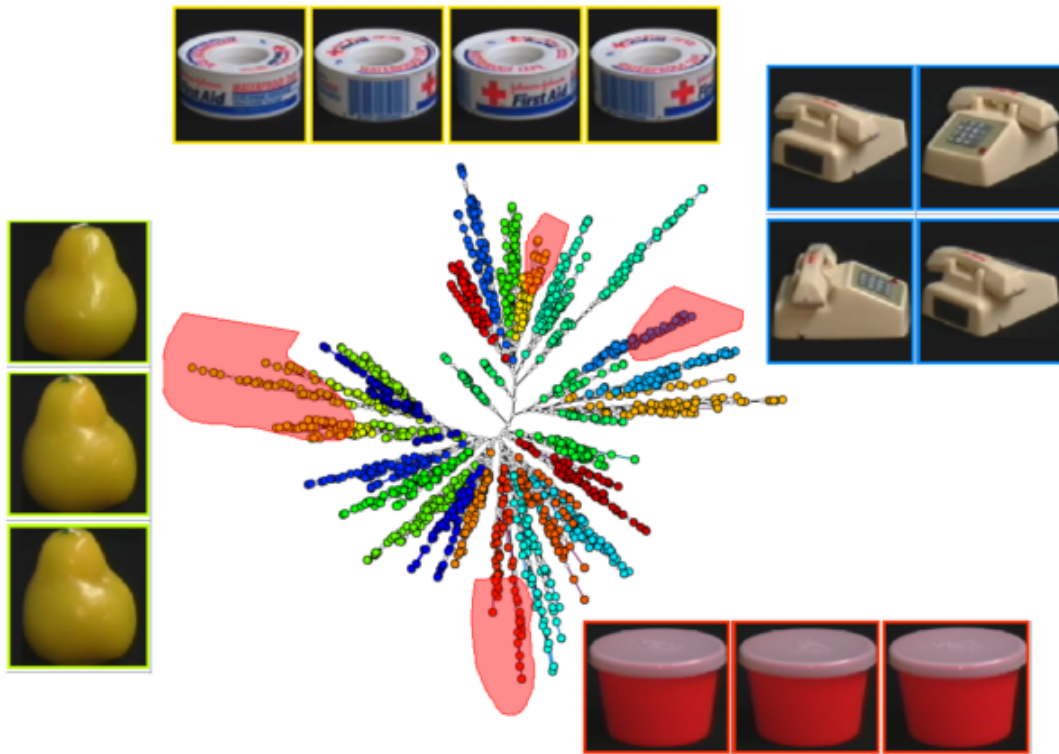


Figura 2.12: Árvore de similaridade do conjunto de 700 imagens COIL usando o algoritmo NJ. Cada ramo apresenta um agrupamento distinto, com as imagens do mesmo grupo igual ao espaço original (imagens recuperadas perto de cada ramo) (CRUZ, 2012).

uma nova linha é inserida em D . As novas distâncias de x para todos os outros nós restantes na matriz é calculada de acordo com as fórmulas apresentadas no algoritmo (PAIVA, 2012).

Algoritmo 1 Neighbor Joining

Considere $n =$ número de elementos da matriz D ;

Associe cada linha i da matriz D com um nó folha i ;

repita

 Selecione o par de nós (i, j) com o mínimo valor de S_{ij} ;

 Crie um novo nó x que conecte os nós i e j , com arestas de tamanhos L_{ix} e L_{jx} , respectivamente;

 Adicione a linha x a D , com valores D_{xk} para cada coluna $k \neq i, j$;

 Remova as linhas i e j de D ;

até $n = 3$;

Conecte os três nós restantes na árvore;

$$S_{ij} = D_{ij} - R_i - R_j; R_y = \frac{1}{n-2} \sum_k D_{yk}$$

$$L_{ix} = \frac{1}{2}(D_{ij} + R_i - R_j); L_{jx} = D_{ij} - L_{ix}; D_{xk} = \frac{1}{2}(D_{ik} + D_{jk} - D_{ij})$$

O NJ, além de construir a topologia no formato de uma árvore sem raiz, também fornece o valor do comprimento dos ramos da árvore resultante. A interpretação permite que se analisem os dados em grupos identificados pelos ramos externos da árvore. Desta forma, o algoritmo ocupa melhor o espaço em uma projeção, pois é possível expandir o desenho da árvore de forma a reduzir significativamente ou eliminar sobreposições (CRUZ, 2012).

A Figura 2.13 mostra um comparativo entre um layout produzido pela LSP (*Least-Square Projection*) (2.13a), uma técnica de projeção, e uma árvore de similaridade NJ (2.13b, 2.13c), para uma coleção de dados textuais. Esse exemplo mostra que a organização das instâncias em ramos é consistente com a organização da projeção LSP. No entanto, a árvore adicionalmente estrutura os grupos em níveis de similaridade, e reduz consideravelmente a sobreposição e confusão visual. Além disso, as vizinhanças locais são claramente visualizadas na árvore, de acordo com a matriz de similaridade.

Após a construção de uma árvore NJ, um algoritmo de desenho radial de grafos pode ser aplicado, utilizando por exemplo um procedimento simplificado de construção de layout baseado em força, como mostrado na Figura 2.13c, minimizando a sobreposição de nós e permitindo uma inspeção da organização dos ramos (PAIVA, 2012).

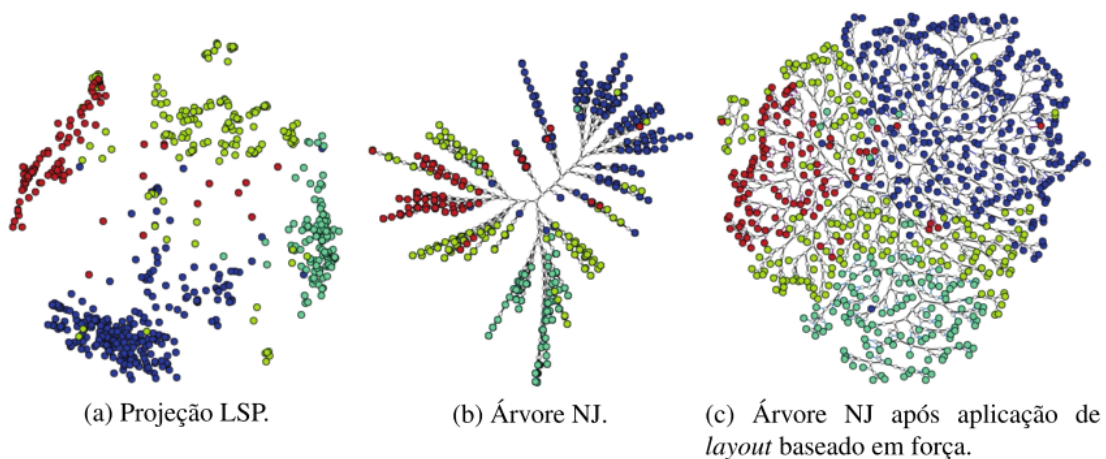


Figura 2.13: Comparação entre uma projeção LSP e uma árvore NJ, para uma coleção de 675 documentos textuais (PAIVA, 2012).

Um dos desafios das pesquisas conhecidas na área de recuperação de imagens é exibir grandes quantidades de imagens por similaridade. Vários autores defendem que uma representação visual auxilia a análise e a identificação de padrões existentes na coleção de dados, permitindo que o usuário entenda o comportamento do espaço de características dos dados (CRUZ, 2012).

Existem muitas aplicações e sistemas que sustentam a exploração de coleções, em que os usuários possuem acesso a uma interface visual na qual podem interagir e reconhecer simila-

ridade, fazer seleções, ampliar detalhes, recuperar informações, entre outras atividades. Além disso, permitem que o usuário tenha uma visão geral de toda a coleção de imagens, por meio da qual se pode criar um processo de busca para recuperar uma ou várias imagens desejadas, ou ainda detectar tendências e características particulares que seriam extremamente difíceis de serem detectadas analisando imagem a imagem. Um desses sistemas é proposto no trabalho de Rafael Dias (DIAS, 2013).

A Figura 2.14 mostra um exemplo de árvore NJ para a coleção de imagens COREL. Na Figura 2.14a, as cores indicam as classes das instâncias. Na Figura 2.14b, as instâncias são exibidas como miniaturas das imagens que representam. É possível perceber que os ramos organizam bem a maioria das classes da coleção, e que quase todas as instâncias cujas características não expressam claramente os padrões de suas classes situam-se no núcleo da árvore.

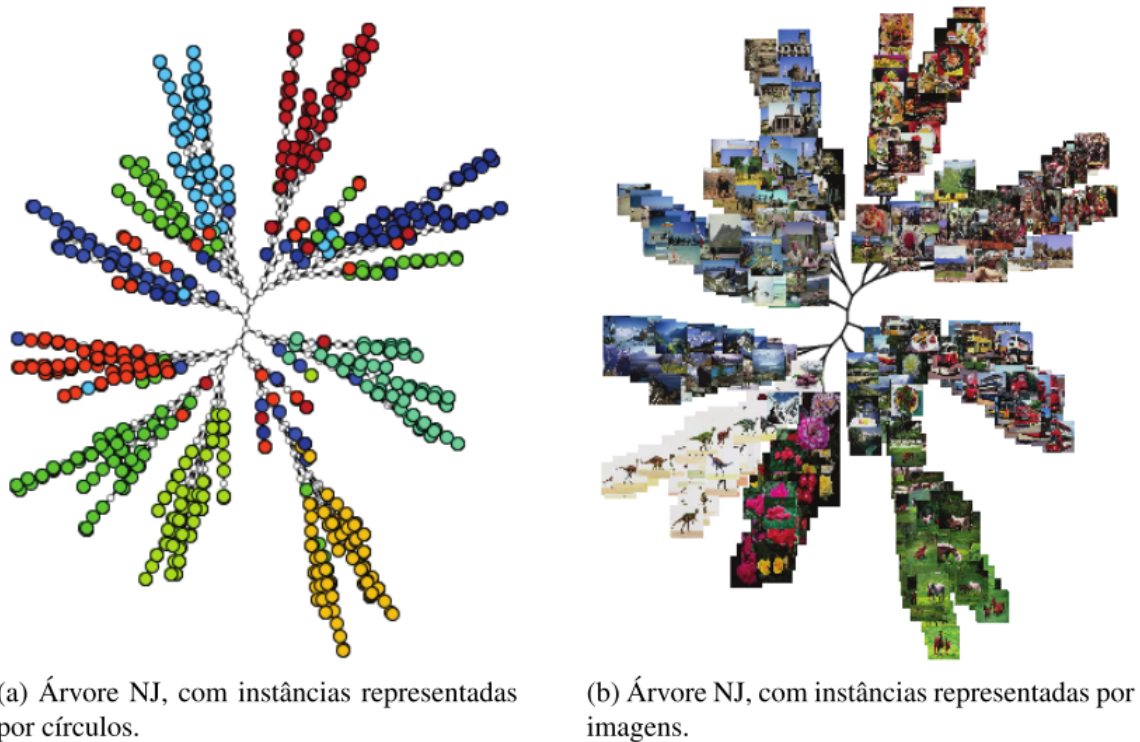


Figura 2.14: Exemplo de uma árvore NJ para a coleção de imagens COREL (PAIVA, 2012).

É importante ressaltar que a medida de distância utilizada para o cálculo da vizinhança na árvore NJ é diferente daquela utilizada para as projeções. Essas últimas organizam o layout de forma que pontos com distância Euclidiana pequena apresentam similaridade maior do que pontos com uma distância Euclidiana grande. Para as árvores NJ, esse raciocínio não é válido, pois a proximidade no layout é definida pelo algoritmo de desenho da árvore, e não possui nenhuma relação com a distância Euclidiana entre um ponto e os demais. Assim, é possível, por exemplo, que dois pontos pertencentes a ramos diferentes, e por isso com pouca similaridade

entre si, sejam posicionados próximos um do outro no layout. Dessa forma, a distância entre dois pontos a e b na árvore NJ foi definida como a soma dos pesos das arestas que formam o menor caminho conectando a a b (PAIVA, 2012).

2.4.3.1 Melhorias na construção de árvores de similaridade

No trabalho de Paiva et al. (2011), são propostas duas melhorias na construção de árvores de similaridade NJ: promoção de nós e algoritmos mais rápidos.

Promoção de nós em árvores NJ

Um problema com o layout produzido pela técnica NJ é o volume de pontos e arestas gerado no espaço de visualização. Como apresentado na seção 2.4.3, para n instâncias em uma coleção, $(n-2)$ nós virtuais são criados. Esses nós não fazem parte da coleção, mas ocupam boa parte do espaço de visualização, gerando um layout poluído.

De modo a reduzir o número de nós virtuais nas árvores NJ, foi implementada uma operação determinística de reescrita de grafos baseada na substituição de nós virtuais por nós da coleção, sempre que detectada uma configuração de nós específica. Essa operação é chamada de promoção de nós.

Seja uma árvore NJ T , na qual existe um par de nós-folha u e v , conectados a um nó virtual a , e que outro nó virtual b conecte esse nó virtual a e um outro nó-folha w , como mostrado na Figura 2.15a. A operação de promoção de nós baseia-se no fato de que nenhum outro nó é tão similar a w do que a (durante a construção de T), e por isso nenhum outro nó é mais similar de u e v do que w . Assim, a pode ser substituído por w e b pode ser removido sem perda no poder representacional. A relação entre u , v , w e a é válida apenas para a topologia criada pelo algoritmo NJ, não tendo valor para o relacionamento induzido pela matriz de similaridade, no caso de algum relacionamento existir.

A promoção de nós pode ser formalmente definida em termos da ocorrência de um padrão e uma substituição na árvore, como mostra as Figuras 2.15b e 2.15c, e consiste em substituir cada ocorrência desse padrão, em ordem decrescente da distância do nó a no padrão para o nó que reside no centro da árvore. Os pesos das arestas, durante a substituição, são chamados de ωr , e são definidos de acordo com os pesos que ocorrem no padrão, chamados de ωp , computados pelo algoritmo de construção da árvore. Chamando de $T1$, $T2$ e $T3$ os nós que conectam a subárvore ao resto da árvore, é possível obter $\omega r(w, T1) = \omega p(b, T1) + \omega p(a, b)/2 + \omega p(w, b)/2$ e $\omega r(w, Ti) = \omega p(a, Ti) + \omega p(a, b)/2 + \omega p(w, b)/4$, para $i=2, 3$. A árvore resultante é chamada de *Promoting Neighbor Joining* (PNJ).

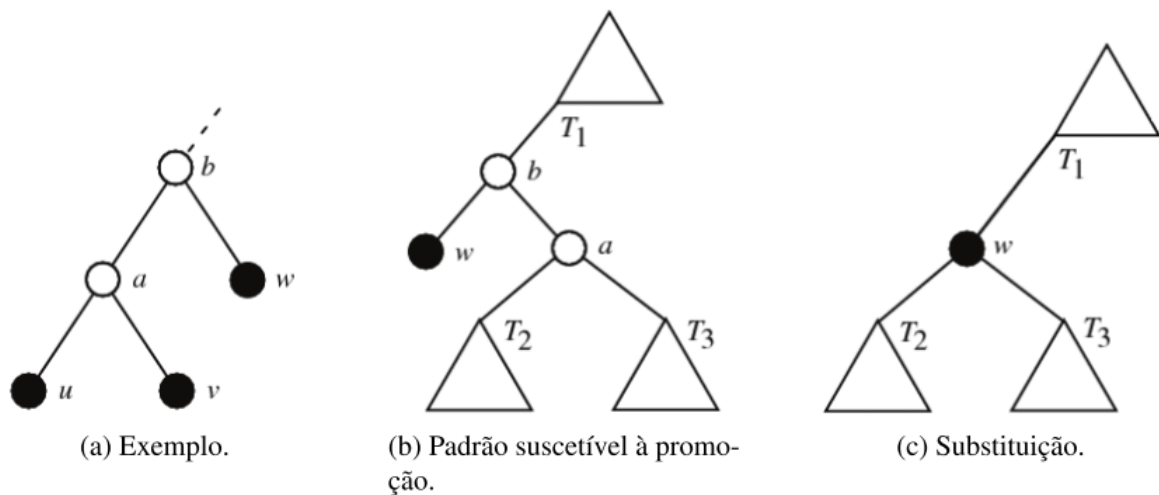


Figura 2.15: Operação de promoção de nós. Círculos preenchidos representam nós da coleção, e triângulos representam subárvores (PAIVA et al., 2011).

A promoção de nós permite que as capacidades de organização e exploração das árvores NJ sejam mantidas, mas reduzindo o número de nós virtuais e conseqüentemente de arestas desnecessárias, resultando em um layout mais limpo, que permite um acesso mais direto às instâncias da coleção através dos ramos da árvore. Nas árvores PNJ, a utilização do espaço de visualização é mais racional, e a poluição visual da construção é reduzida, como pode ser visto na Figura 2.16, que mostra uma comparação entre os *layouts* produzidos pela técnica NJ original (2.16a) e PNJ (2.16b).

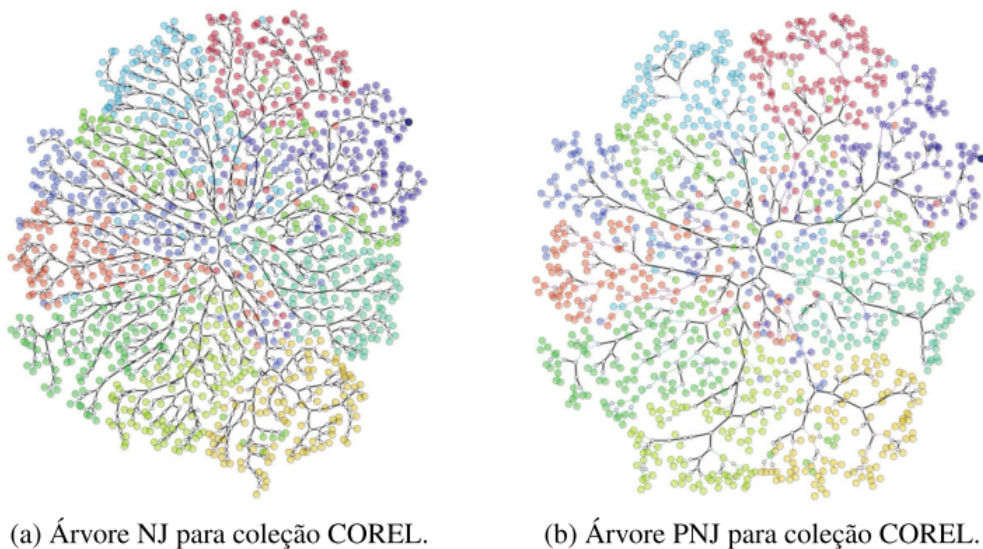


Figura 2.16: Comparação entre árvores NJ e PNJ para a coleção COREL, destacando os nós virtuais e arestas (PAIVA et al., 2011).

Algoritmos mais rápidos para NJ

A necessidade de se recalcularem parte da matriz de distâncias a cada iteração do processo de construção da árvore NJ, devido à inserção do nó virtual gerado na iteração anterior, pode resultar em um alto custo computacional na geração da árvore, mesmo para coleções pequenas. Além disso, a procura pelo valor de S_{ij} mínimo na matriz D , para determinar o próximo par de nós a serem combinados, exige uma verificação em cada posição de D , resultando em $O(n^2)$, com n igual ao número de instâncias.

Para amenizar tais problemas, alguns algoritmos que modificam o processo de montagem da árvore foram propostos. De um modo geral, esses algoritmos utilizam duas estratégias, baseadas em estruturas de dados especializadas, ou heurísticas que sacrificam a precisão, e podem ser diferenciados basicamente por algumas modificações direcionadas a situações específicas. Assim, de forma a representar essas duas estratégias, e verificar o impacto na velocidade de geração e na precisão da árvore obtida, duas técnicas foram selecionadas, e são descritas a seguir. As descrições e notações utilizadas são baseadas nos passos de construção da árvore NJ apresentados no Algoritmo 1. Adicionalmente, S_{ij} é denotado como S_{min} , para uma determinada matriz D .

O primeiro algoritmo selecionado é chamado *Rapid Neighbor Joining (Rapid NJ)*. Esse algoritmo produz árvores idênticas às produzidas pelo algoritmo original. Sua ideia baseia-se no fato de que, na procura por S_{min} em determinada linha i da matriz D , através da avaliação de $S_{ij} = D_{ij} - R_i - R_j$, o valor de R_i é fixo. Assim, o algoritmo mantém uma matriz auxiliar, na qual as linhas são ordenadas, e cujas células contêm índices para D . A procura em uma linha ordenada i dessa matriz para assim que $S_{ij} - R_i - R_{max} > S_{min}$, para o qual R_{max} é o valor máximo de R_i dentre todas as linhas da matriz ($R_j \leq R_{max}$). Um trabalho extra é inserido com a ordenação das linhas nessa matriz auxiliar, bem como a manipulação de colunas removidas e avaliação de R_{max} . O algoritmo ainda é $O(n^3)$, mas os experimentos apresentados em (PAIVA et al., 2011) mostraram que a estratégia diminuiu significativamente o número de células visitadas, na busca por S_{min} , fazendo com que ele se mostre mais eficiente do que outros algoritmos de aceleração na construção da árvore.

O segundo algoritmo selecionado é chamado *Fast Neighbor Joining (Fast NJ)*. Esse algoritmo mantém um conjunto com $O(n)$ células candidatas na matriz D . A procura por S_{min} é restrita a esse conjunto, que contém, inicialmente, os valores mínimos de S_{ij} para cada linha i . Quando as linhas i e j são conectadas a um nó virtual x e removidas de D , as células candidatas com as quais eles contribuem são removidas e uma nova célula candidata, para a nova linha x é adicionada. Essa abordagem resulta em um algoritmo que não produz a mesma árvore que o algoritmo original de construção da árvore NJ, e que no pior caso, é $O(n^2)$. Os autores da

técnica mostraram que quando a medida de similaridade é próxima de ser aditiva, as árvores produzidas são idênticas às aquelas produzidas pelo algoritmo original.

A Tabela 2.1 mostra uma comparação entre os tempos de geração dos layouts utilizando os algoritmos NJ e NJ melhorados e a técnica de projeção LSP. Técnicas de projeção rápidas, tais como a LSP, que é $O(n\sqrt{n})$, levam vantagem em relação às árvores NJ com relação ao tempo de geração do layout. A árvore *Fast NJ* é mais rápida do que a técnica de projeção LSP, sendo que o tempo usado para a projeção de nós é proporcionalmente pequeno.

Técnica	COREL	MEDICAL	OBJECTS
ÁrvoreNJ	3.0474	0.5	394.383
ÁrvorePNJ	3.064	0.506	396.507
FastNJ	0.0936	0.0462	4.976
FastPNJ	0.1104	0.052	6.1
RapidNJ	2.366	0.373	261.971
RapidPNJ	2.3828	0.3788	262.178
LSP	0.7182	0.2068	61.564

Tabela 2.1: Comparação entre tempos de Geração do Layout (segundos), considerando as abordagens de geração rápida de árvores NJ e a técnica de projeção LSP (PAIVA et al., 2011).

2.5 Considerações Finais

Neste capítulo foram apresentados os conceitos relacionados ao trabalho. No capítulo 3 é demonstrado o uso em conjunto das técnicas apresentadas a fim de testar a relação entre a qualidade da resposta da consulta em CBIR e a qualidade da visualização gerada. Também é observada a qualidade da árvore NJ gerada com o uso de múltiplos descritores com condições de contorno.

Capítulo 3

TRABALHO DESENVOLVIDO

3.1 Considerações Iniciais

Neste capítulo é apresentado o método desenvolvido neste projeto de mestrado, que propõe o uso de múltiplos descritores com condições de contorno para melhorar a visualização de árvores de similaridade NJ. Barroso et al. (2013) comprovou que o uso de condições de contorno para combinar múltiplos descritores resulta em melhor precisão em sistemas CBIR. É esperado que o uso de múltiplos descritores também contribua na construção de árvores de similaridade mais bem organizadas, melhorando a visualização e o entendimento do usuário.

Inicialmente cada passo do método proposto é explicado para depois serem apresentados os experimentos realizados para avaliar se as hipóteses são verdadeiras. Tais experimentos consistem em comparações de consultas utilizando cada descritor separadamente e todos os descritores juntos, utilizando-se o método proposto por Barroso et al. (2015) para determinar o melhor peso dos descritores. Também são analisadas as árvores geradas por essas consultas de forma subjetiva e também utilizando o método *neighborhood hit* (PAULOVICH et al., 2008), que mede a porcentagem de vizinhos mais próximos de uma imagem na visualização, que pertencem à mesma classe. A precisão final representa a média das precisões para cada ponto. Isso permite avaliar numericamente a separabilidade das classes pré-existentes no *layout*.

3.2 Visão Geral do Método Desenvolvido

Como pode ser visto na Figura 3.1, que apresenta uma visão geral do método desenvolvido, o usuário fornece uma imagem de consulta - para obter as imagens mais similares a ela - e um peso para os descritores. Para o método proposto, o peso ideal para a condição de contorno

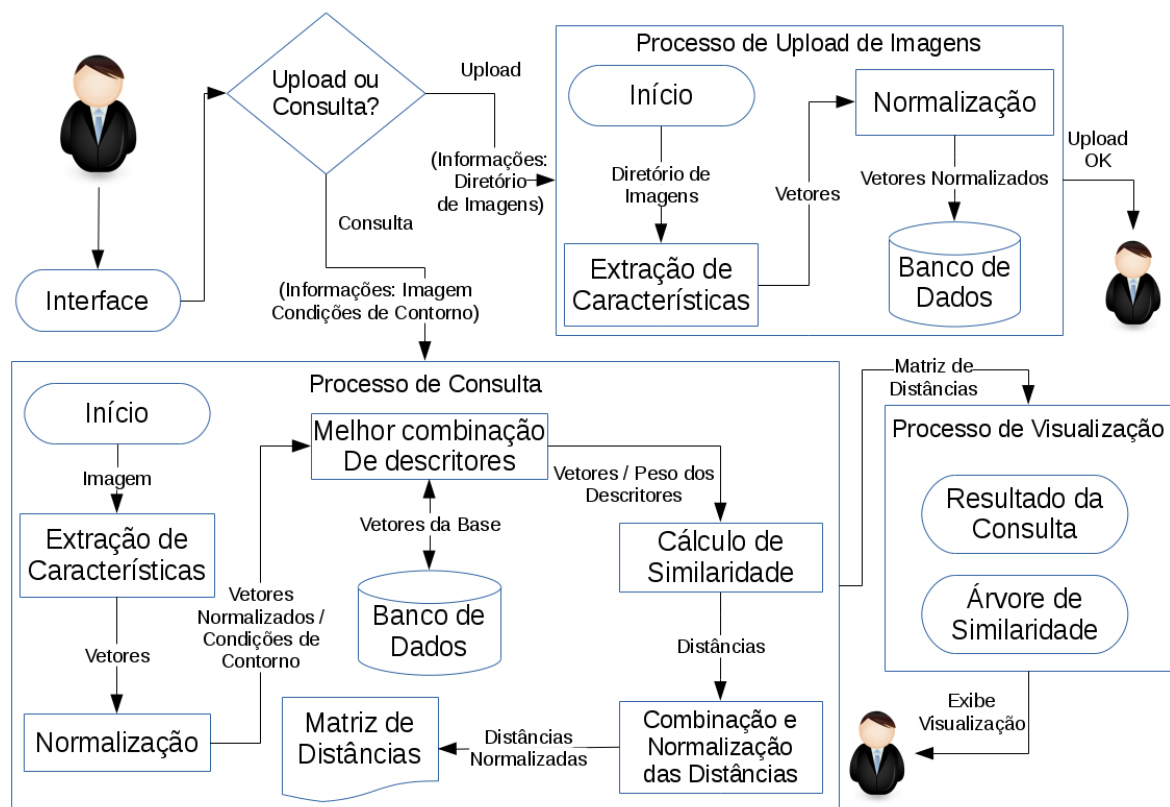


Figura 3.1: Visão geral do método proposto.

da imagem de consulta deve ser obtido previamente utilizando-se a técnica de Barroso et al. (2015). Neste método são utilizados três descritores. Cada descritor é formado por um extrator de características e uma função de distância (a similaridade entre as imagens é definida através do cálculo da distância entre os vetores de características obtidos pelo extrator). Os extratores usados são: Haralick e distância Canberra para textura; Statistical Moments e distância Manhattan (L_1) para forma; e Histograma de Cinza e distância Euclidiana (L^2) para cor. Um exemplo de consulta por similaridade de imagem utilizando o método proposto pode ser visualizado na Figura 3.2, onde o usuário também pode definir quantas imagens (k) devem ser retornadas.

A escolha pelos descritores pode ser feita pelo especialista de domínio ou aleatoriamente, apenas para fins de testes. Optou-se por utilizar, aleatoriamente, um extrator de cada tipo (textura, forma e cor) a fim de se expandir a possibilidade de melhor descrever todos os subconjuntos de imagens. Para escolher a função de distância utilizou-se como base o trabalho de Bugatti, Traina e Traina Jr. (2008).

Em seguida, os vetores de características extraídos da imagem de consulta são normalizados. Para isso, os valores do vetor (cada um representa uma determinada característica) são divididos pelo maior valor obtido de cada característica levando em conta todas as imagens utilizadas (consulta + base), dessa forma os valores são alterados para que fiquem entre 0 e 1.

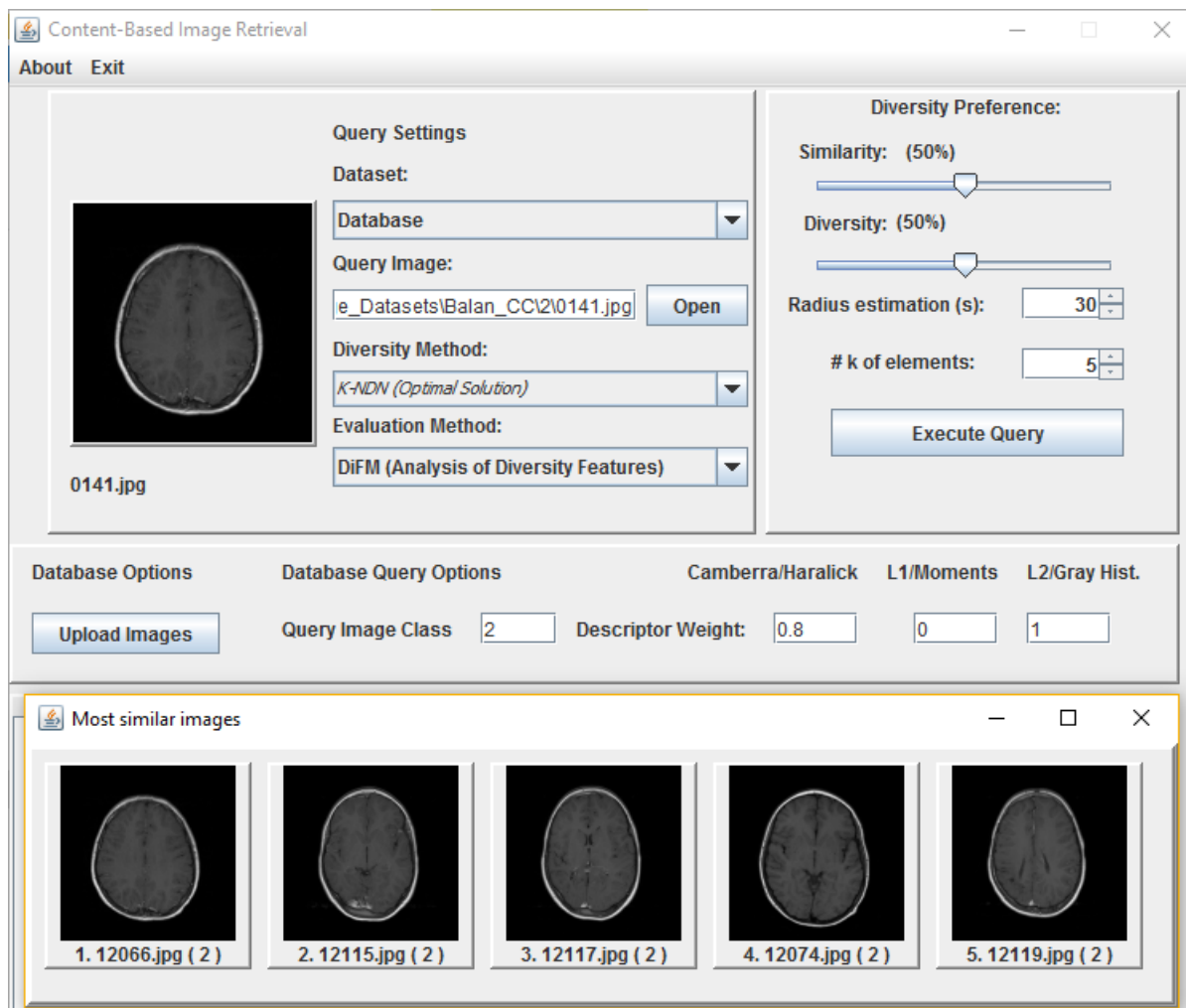


Figura 3.2: Exemplo implementado do método desenvolvido.

Os vetores de características das imagens da base já normalizados são retornados e se unem aos vetores da imagem de consulta para que seja feito o cálculo de similaridade. Uma medida de similaridade é aplicada para obter a distância entre os vetores, que depois são utilizadas para se obter um valor único que represente as três distâncias medidas (uma de cada descritor).

O cálculo de junção das distâncias pode ser visto na Equação 3.1. Para cada par de imagens (i, j) soma-se o resultado do cálculo de cada descritor (n) , de 1 a 3, que é: o peso (w_n) informado pelo usuário, multiplicado pela distância entre i e j obtida pelo descritor $(dn_{(i,j)})$, dividido pela maior distância medida pelo descritor (md_n) . A divisão pelo maior valor funciona como normalização.

$$distance_{(i,j)} = \sum_{n=1}^3 \frac{w_n * dn_{(i,j)}}{md_n} \quad (3.1)$$

As distâncias finais são utilizadas para se criar uma matriz de distâncias (exemplificada em

parte na Figura 3.3). O arquivo de matriz de distâncias (*dmat file*) contém as distâncias entre todos os pares de imagens. Essa matriz é utilizada tanto para se obter uma classificação das distâncias, determinando assim as imagens mais similares à imagem de consulta para serem exibidas ao usuário (exemplificado na Figura 3.4), quanto para se construir a árvore de similaridade NJ através do algoritmo *Fast Promoting Neighbor Joining* (FastPNJ, exemplificada na Figura 3.5).

```
4 ...
5 0.6253914023280838
6 0.5800668945934224;0.4200949963752706
7 0.6040243790354303;0.04890269705047373;0.405562746897984
8 0.6494481143389289;0.16152994481274413;0.4966129650272855;0.13449749782842277
9 0.6887878050259041;0.41526633913143773;0.4259989408681318;0.390458613534935;0.3615835164129047
10 ...
```

Figura 3.3: Exemplo de parte de uma matriz de distâncias.



Figura 3.4: Exemplo de grid contendo o resultado de uma consulta por similaridade.

3.3 Experimentos e Resultados

Este capítulo apresenta os experimentos realizados com a implementação do método proposto para avaliar se a qualidade da visualização está relacionada com a qualidade da consulta (visualização de má qualidade está relacionada a um resultado da consulta de qualidade ruim; visualização de boa qualidade está relacionada a um resultado da consulta também de qualidade boa), apresentado no capítulo 3.3.1.

Também é avaliado se o uso de múltiplos descritores com condições de contorno melhora a qualidade da visualização hierárquica de imagens em CBIR, apresentado nos capítulos 3.3.2 e 3.3.3.

O conjunto de imagens utilizado nesse experimento, chamado BalanRMI704, é composto por 704 imagens de exames de Ressonância Magnética, obtidas no Hospital das Clínicas de Ribeirão Preto da Universidade de São Paulo (USP), divididas em 8 classes, de acordo com o tipo de exame: Angiograma, Pélvis Axial, Cabeça Axial, Abdômen Axial, Abdômen Coronal,

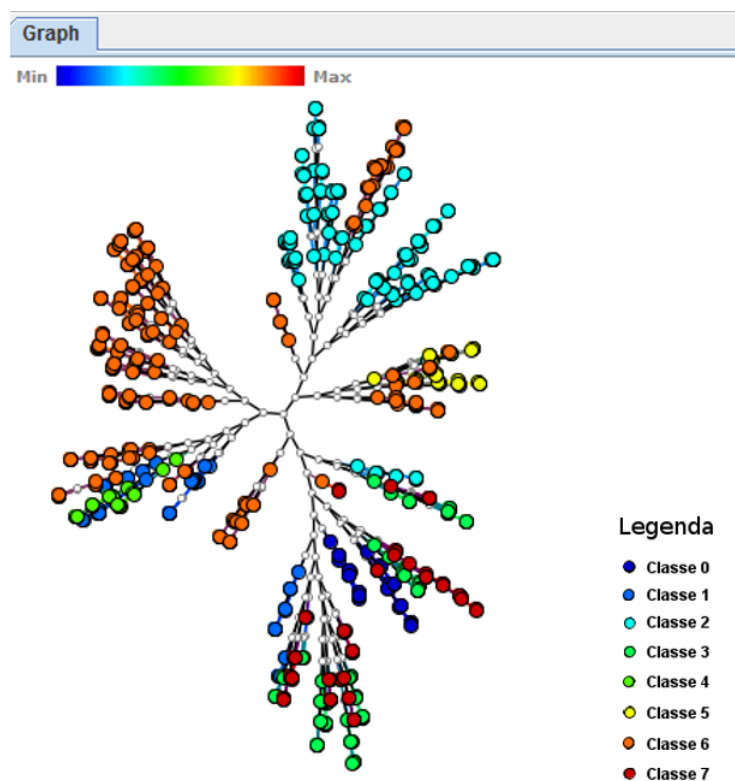


Figura 3.5: Exemplo de árvore de similaridade construída pelo algoritmo NJ a partir de uma matriz de distâncias.

Cabeça Coronal, Cabeça Sagital e Espinha Sagital. Um exemplo de algumas imagens da base pode ser visto na Figura 3.6.

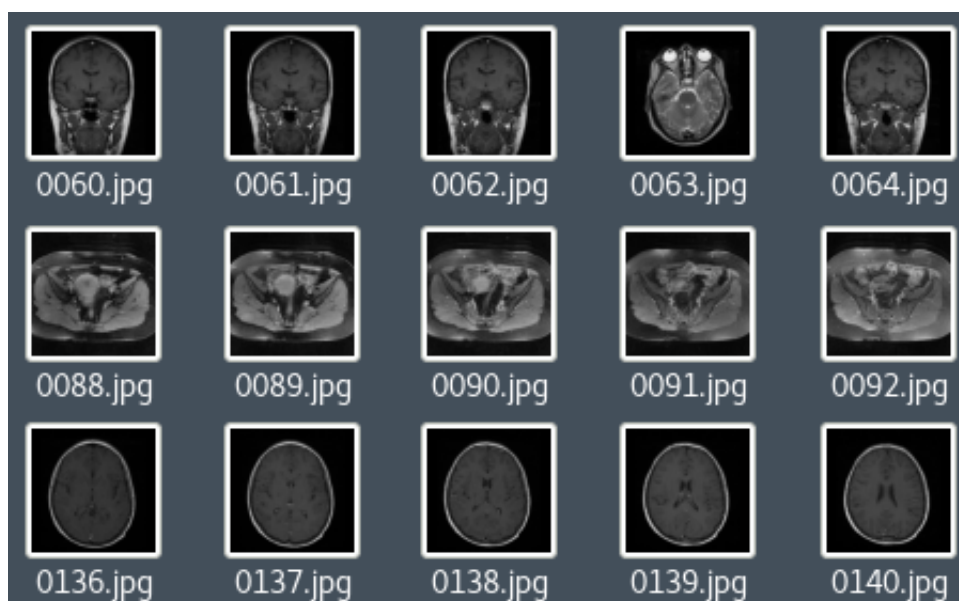


Figura 3.6: Exemplo de imagens da Base Balan.

3.3.1 Experimento 1

Este experimento está voltado a testar a primeira hipótese deste trabalho de mestrado: quanto mais precisos os resultados de uma consulta, melhor pode ser a organização das imagens na visualização. Em outras palavras, avalia se existe uma correlação entre os dois.

Para isso, uma imagem aleatória é escolhida para se fazer consultas utilizando cada descritor individualmente. A partir de cada consulta são obtidas duas informações: a precisão/revocação dos resultados desta consulta e o *neighborhood hit* da árvore NJ gerada.

É então gerado um gráfico de precisão e revocação com os valores obtidos das três consultas e um gráfico de *neighborhood hit* das três árvores NJ geradas. Dessa forma, podemos comparar a qualidade da consulta e a qualidade da visualização.

3.3.1.1 Testes

Leva-se em conta a imagem 0037 pertencente à classe 5 (Cabeça Coronal) da base, representada na Figura 3.7. São realizadas três consultas, cada uma utilizando um descritor individual, sem alterar qualquer outro parâmetro. O resultado da consulta utilizando apenas o descritor 1 (*Haralick / Canberra*) pode ser visto em (A) na Figura 3.8, o resultado relacionado ao descritor 2 (*Statistical Moments / Manhattan*) pode ser visto em (B) e o resultado do descritor 3 (*Gray Histogram / Euclidean*) está em (C).

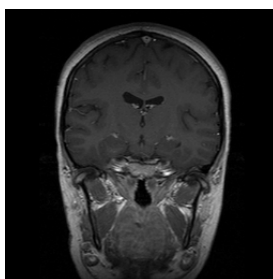


Figura 3.7: Imagem 0037 da Base Balan, pertencente à classe 5.

O gráfico de precisão e revocação gerado dessas três consultas se segue na Figura 3.9.

Pode-se relacionar as Figuras 3.8 e 3.9. A Figura 3.8 mostra que o Descritor 1 retornou 5 imagens da mesma classe que a imagem de consulta. O Descritor 2 retornou apenas uma imagem corretamente. O Descritor 3 retornou 4 das 5 imagens corretamente. Portanto, o Descritor 1 foi o mais eficiente e o Descritor 2 o menos eficiente. Esta eficiência também é vista na Figura 3.9.

A visualização gerada para cada uma das três consultas deste teste pode ser consultada na

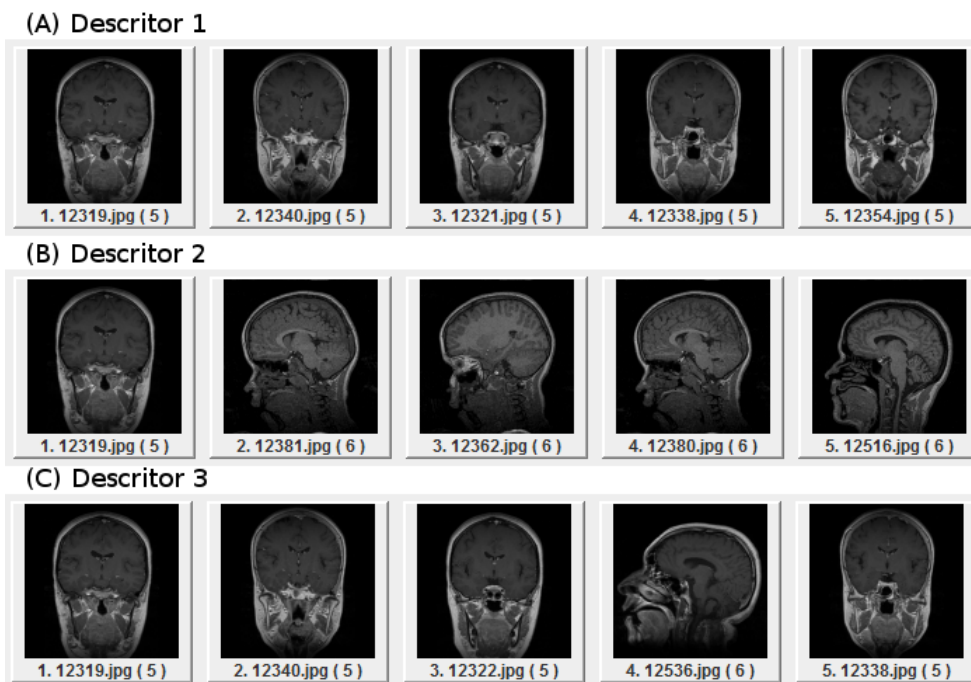


Figura 3.8: Resultado das consultas da imagem 0037 da classe 5 utilizando cada descritor individualmente.

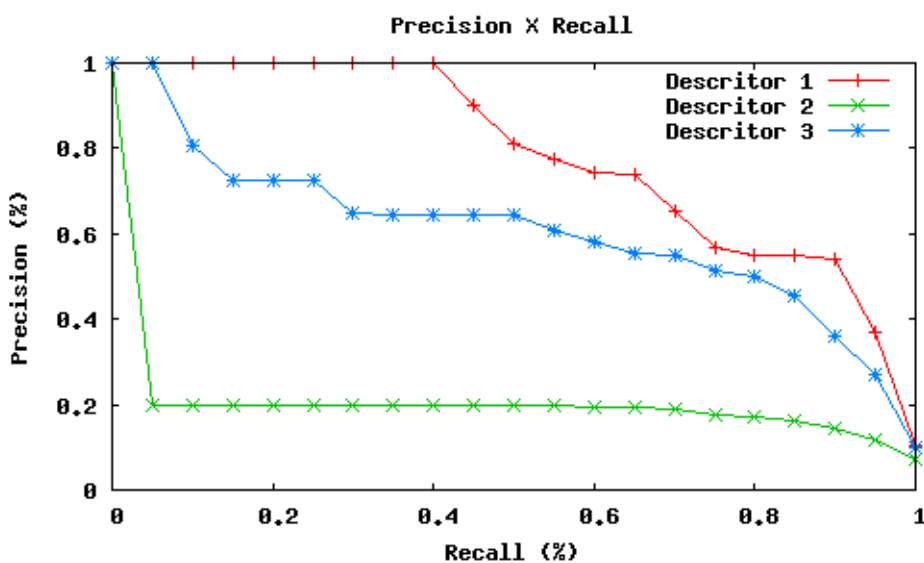


Figura 3.9: Gráfico de precisão e revocação medidas da consulta da imagem 0037 da classe 5 utilizando cada descritor individualmente.

Figura 3.10.

O gráfico de *neighborhood hit* gerado para cada visualização apresentada na Figura 3.10 encontra-se na Figura 3.11.

É possível observar uma relação entre precisão e revocação medida e a média de *neighborhood hit*, ou seja, uma relação entre a qualidade do resultado da consulta e a qualidade da

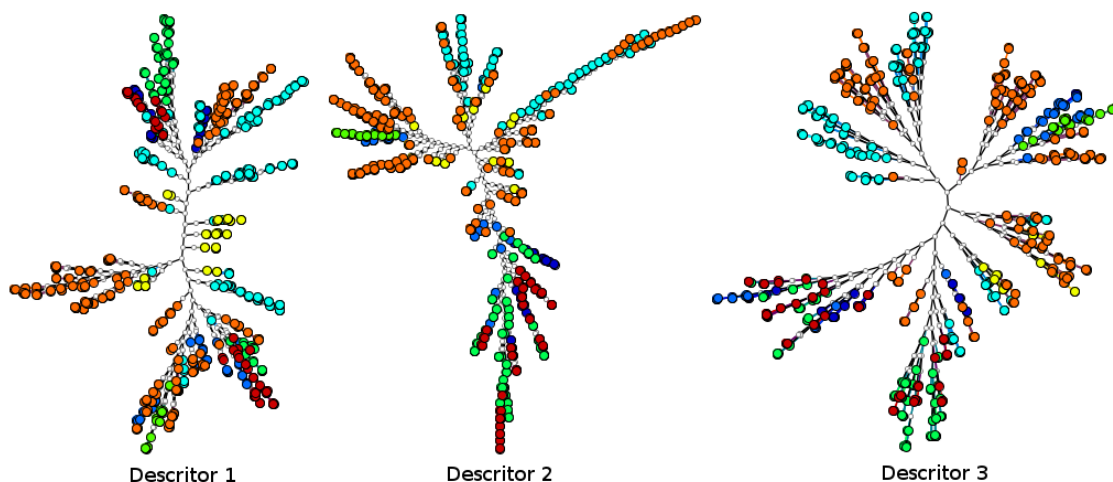


Figura 3.10: Visualização gerada nas consulta da imagem 0037 da classe 5 utilizando cada descritor individualmente.

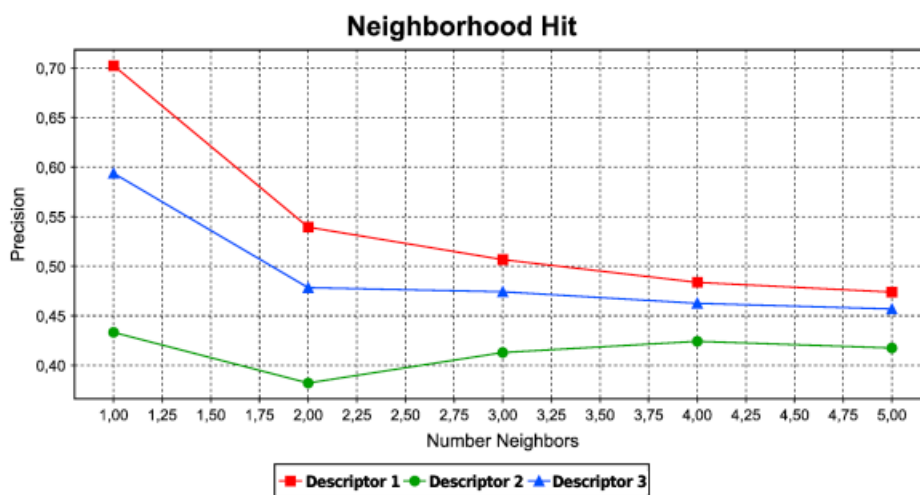


Figura 3.11: Gráfico de *neighborhood hit* da consulta da imagem 0037 da classe 5 utilizando cada descritor individualmente.

visualização gerada.

Para melhor visualizar esta relação, foram realizados outros testes com imagens aleatórias da base de dados, mostrando a correlação entre a qualidade dos resultados da consulta e a qualidade da visualização. Foi comparada a posição no *ranking* de precisão e revocação com a posição no *ranking* de *neighborhood hit*. Esse *ranking* é a ordenação dos descritores pela precisão média obtida para cada classe.

3.3.2 Experimento 2A

Este experimento avalia o impacto da utilização de múltiplos descritores com condições de contorno na qualidade da árvore de similaridade gerada.

Dez imagens aleatórias foram selecionadas de 4 das 8 classes da base de dados para se calcular a média de precisão por classe de cada um dos 3 descritores implementados. Isso é necessário para se obter a melhor combinação de descritores pelo método de Barroso et al. (2015), partindo do melhor descritor individual. As condições de contorno escolhidas para a combinação de descritores são os tipos de exame, ou seja, a própria classe das imagens.

A melhor combinação obtida para cada uma dessas 4 classes pode ser vista na Tabela 3.1. Lembrando que os descritores são Haralick e distância Canberra para textura (D1), Statistical Moments e distância Manhattan (L_1) para forma (D2) e Histograma de Cinza e distância Euclidiana (L^2) para cor (D3).

Classe	D1	D2	D3
Classe 2	0.800000	0.000000	1.000000
Classe 3	1.000000	0.400000	0.210526
Classe 5	1.000000	0.000000	0.266667
Classe 7	1.000000	3.333333	0.307692

Tabela 3.1: Melhor combinação de pesos de descritores obtida para cada classe testada.

Foi constituído o gráfico de precisão e revocação utilizando a melhor combinação gerada para cada descritor, como pode ser visto nos gráficos correspondentes à classe 2 (Figura 3.12), classe 3 (Figura 3.13), classe 5 (Figura 3.14) e classe 7 (Figura 3.15).

Foram realizadas dez consultas distintas para cada uma das 4 classes do experimento anterior, gerando suas respectivas árvores de similaridade. Dessa forma, foram geradas 4 árvores para cada consulta, uma correspondente a cada um dos três descritores individuais e também uma para a melhor combinação de descritores.

Um exemplo pode ser visto na Figura 3.16, que são as árvores geradas em uma das consultas do experimento, com uma imagem da classe 2 (Cabeça Axial), correspondentes à combinação dos descritores e aos descritores individuais.

Na Figura 3.16, as árvores geradas foram analisadas subjetivamente para se tentar identificar quais parecem ser melhor construídas. Para isso foram levadas em conta características como proximidade de nós de mesma classe (representados pela mesma cor), distância entre esses nós e também a distância entre as classes.

É preciso levar em conta a classe / condição de contorno da consulta para se analisar melhor a organização da árvore. Dessa forma podemos perceber duas ocorrências no uso de descritores individuais: nós de outras classes misturadas aos nós da classe de consulta e ramos da classe de consulta distantes entre si. Isso não ocorre no uso de múltiplos descritores. Apenas um ramo com 4 nós da classe de consulta ficou mais distante, talvez por serem imagens mais semelhantes

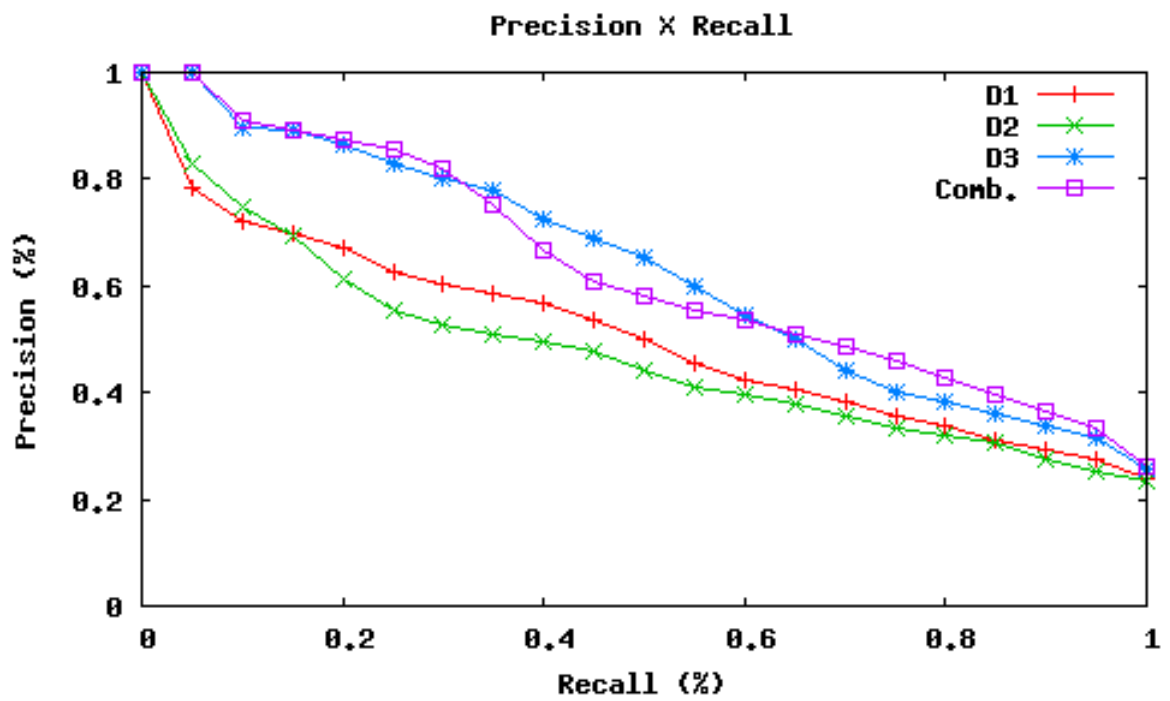


Figura 3.12: Média de precisão medida para consultas de imagens da Classe 2.

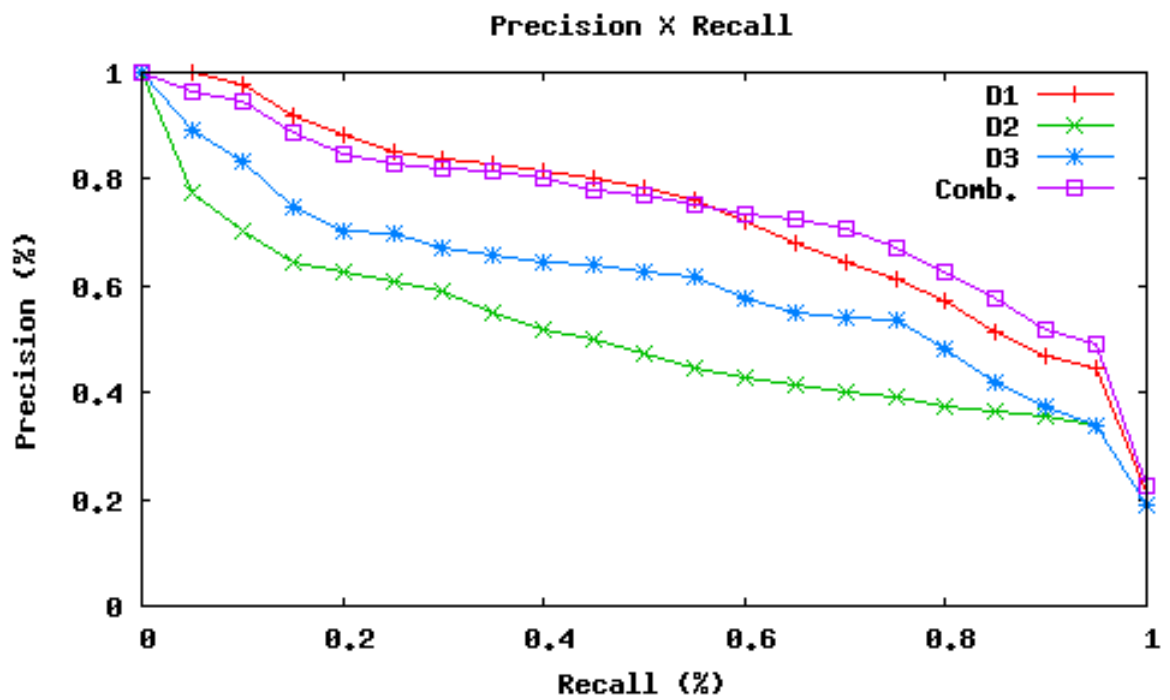


Figura 3.13: Média de precisão medida para consultas de imagens da Classe 3.

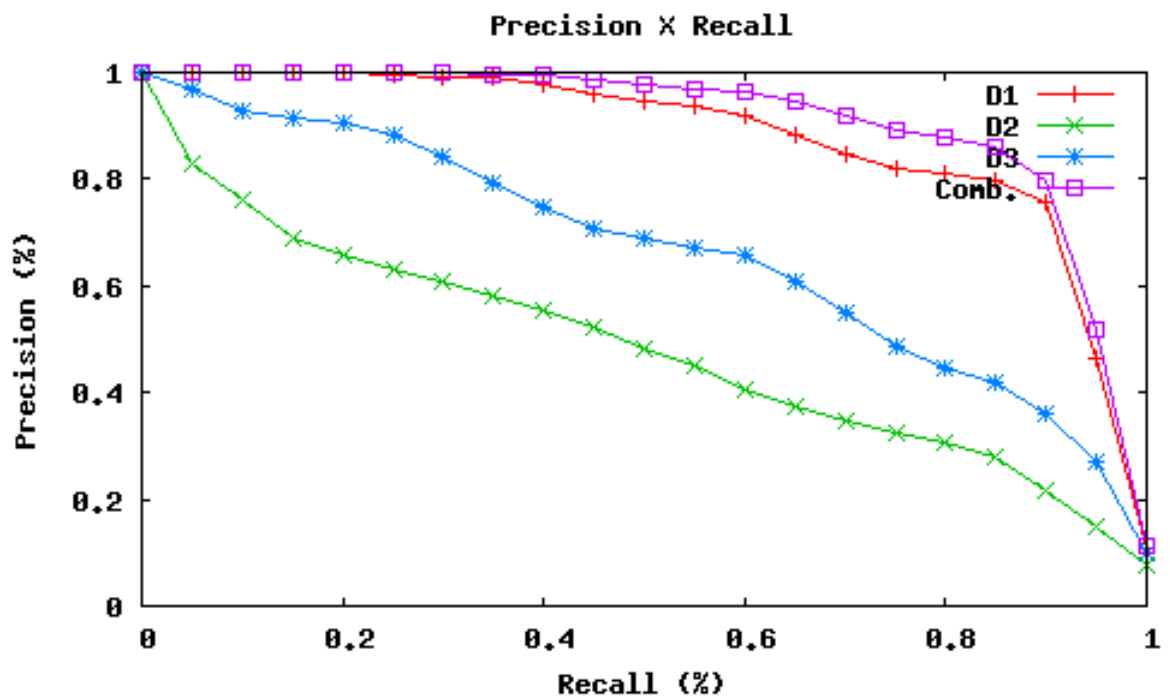


Figura 3.14: Média de precisão medida para consultas de imagens da Classe 5.

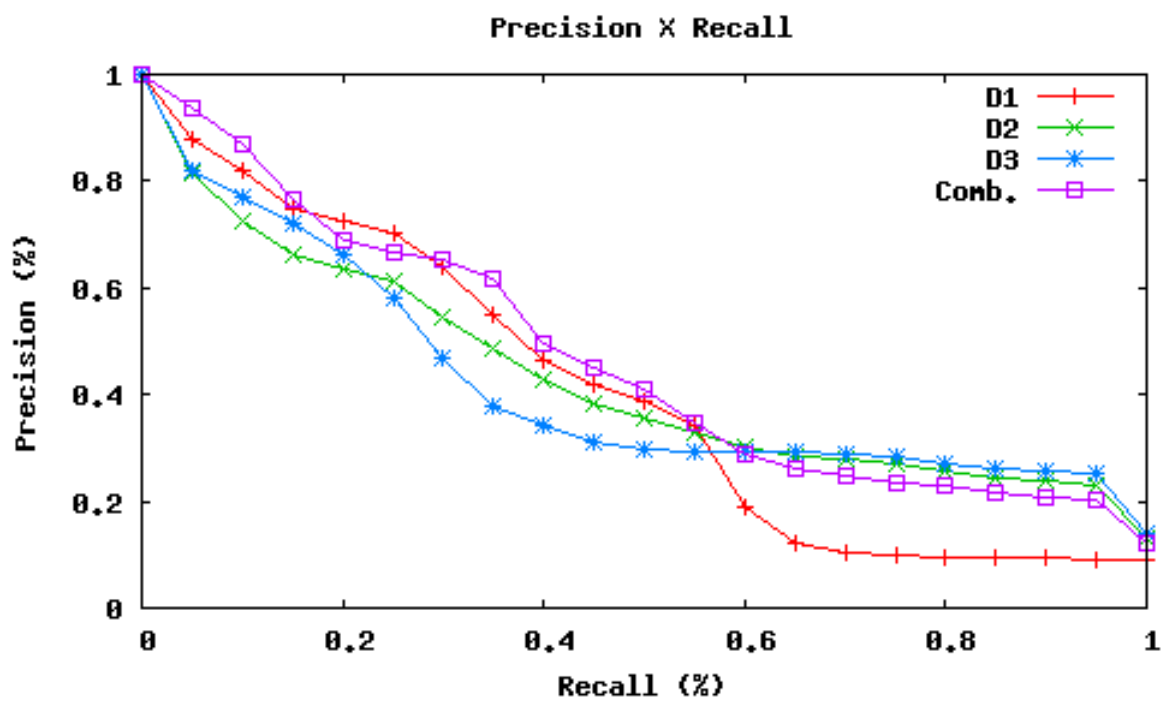


Figura 3.15: Média de precisão medida para consultas de imagens da Classe 7.

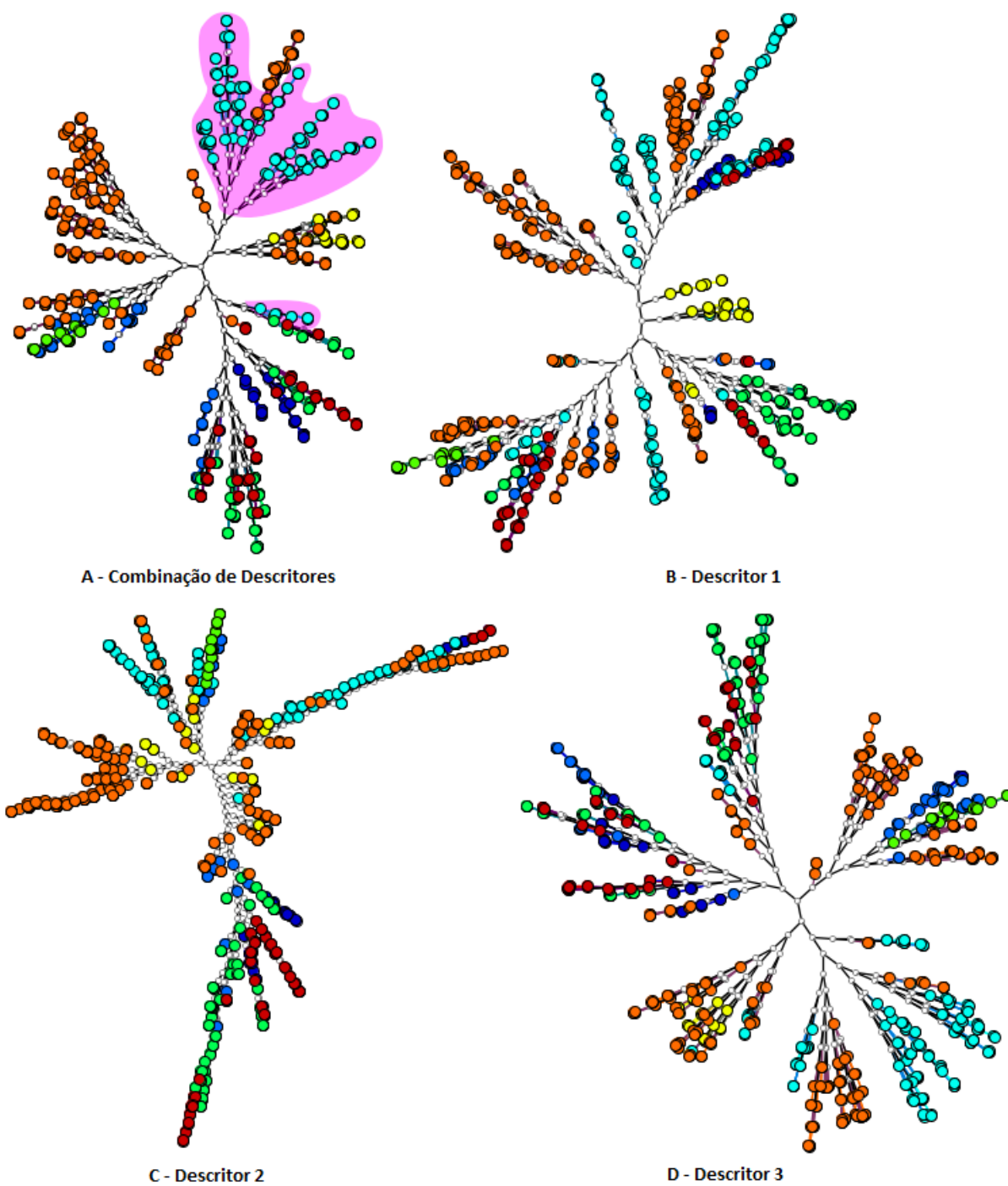


Figura 3.16: Árvores geradas para consulta de imagens da classe 2.

a imagens de outra classe. Foi destacado na imagem o conjunto de nós de classe 2 na primeira árvore.

3.3.3 Experimento 2B - Neighborhood Hit

Este experimento é uma extensão do experimento anterior, para exemplificar o resultado em classes diferentes. A fim de se avaliar cientificamente as árvores geradas, foi utilizado o método *Neighborhood Hit*, que mede a porcentagem de vizinhos mais próximos de uma instância, no espaço de visualização, que pertencem à mesma classe dessa instância.

3.3.3.1 Teste com imagens da classe 2

Neste experimento são utilizadas as quatro árvores do experimento anterior, de classe 2, presentes da Figura 3.16. A partir dessas árvores se obteve a precisão de vizinhos através do método *neighborhood hit*, que pode ser conferido na Figura 3.17. A Figura 3.18 demonstra o gráfico de precisão e revocação para as quatro consultas realizadas. Pode-se ver a correlação existente entre as três figuras.

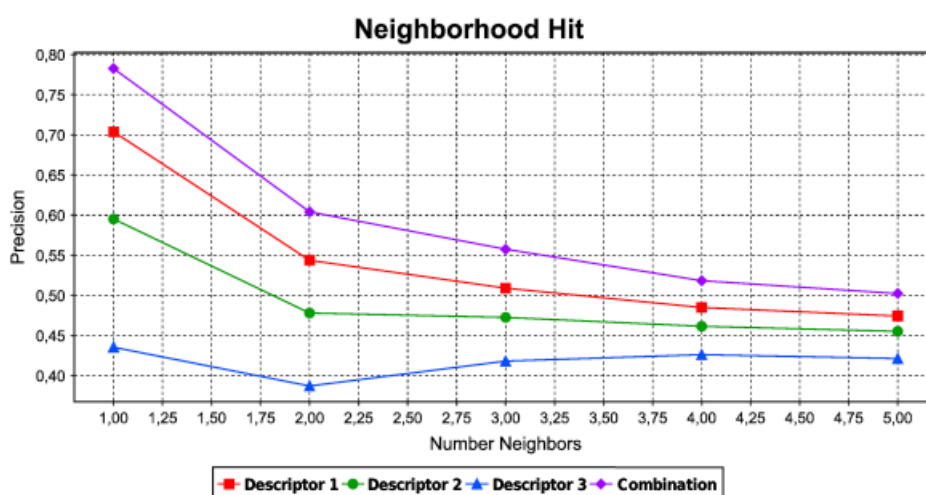


Figura 3.17: Neighborhood Hit para uma consulta de 5 vizinhos da classe 2.

3.3.3.2 Teste com imagens da classe 5

Da mesma forma que foi realizado o teste anterior, foi escolhida uma imagem aleatória da classe 5 (Cabeça Coronal) e foram geradas quatro árvores, modificando apenas o parâmetro relacionado ao descritor. A Figura 3.19 contém a árvore gerada utilizando-se múltiplos descritores (A), a árvore gerada se utilizando apenas o descritor 1 (B), descritor 2 (C) e descritor 3 (D). Novamente foi destacado na primeira árvore da figura o grupo de nós pertencentes à classe testada.

A Figura 3.20 apresenta o gráfico de *neighborhood hit* para as quatro árvores indicadas na

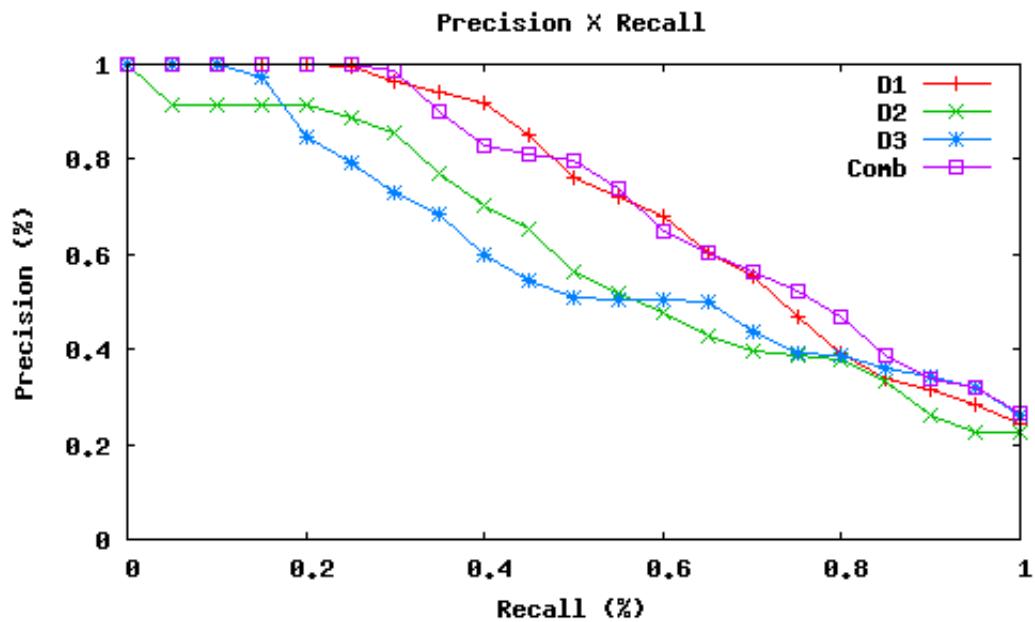


Figura 3.18: Gráfico de precisão e revocação para quatro testes realizados com uma imagem da classe 2.

Figura 3.19. Já o gráfico de precisão e revocação das quatro consultas é demonstrado na Figura 3.21.

3.4 Considerações Finais

A análise dos experimentos realizados indicam que existe uma correlação entre a qualidade do resultado da consulta por similaridade de imagens e a qualidade da visualização em árvore de similaridade NJ gerada, porém é importante testar essa correlação utilizando outros tipos de visualização.

Já os experimentos relacionados a combinação de descritores demonstram que o uso de múltiplos descritores contribui para a geração de árvores de similaridade NJ mais organizadas, comprovado por meio do método *neighborhood hit*, que mede a porcentagem de vizinhos mais próximos de uma instância, no espaço de visualização, que pertencem à mesma classe dessa instância.

Os gráficos de precisão e revocação indicam que o uso de múltiplos descritores com condições de contorno ficam próximos da precisão medida pelo melhor descritor individual por conta do tamanho do grupo de treinamento. É sugerido que se use a base toda para se obter o melhor descritor individual e daí partir para o cálculo da melhor combinação, porém isso se torna inviável em bases grandes. Apesar disso, mesmo usando um grupo de treinamento pequeno, a técnica

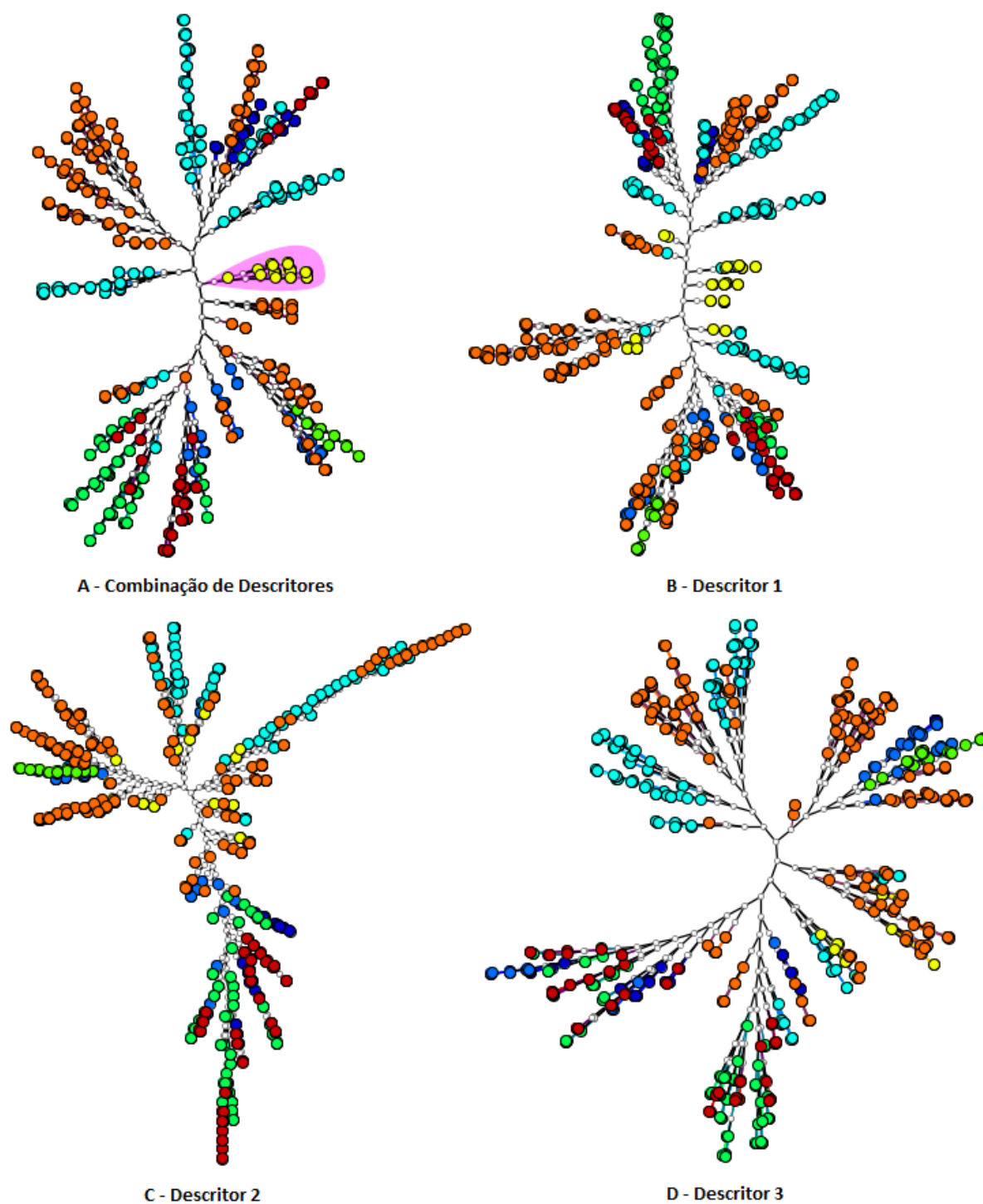


Figura 3.19: Árvores geradas para consulta de imagens da classe 5.

garantiu níveis de precisão e revocação tão bons ou melhores que o melhor descritor individual, e mesmo nos casos de aparente empate, os gráficos de *neighborhood hit* indicam alguma melhora na organização da árvore NJ gerada ao se utilizar múltiplos descritores.

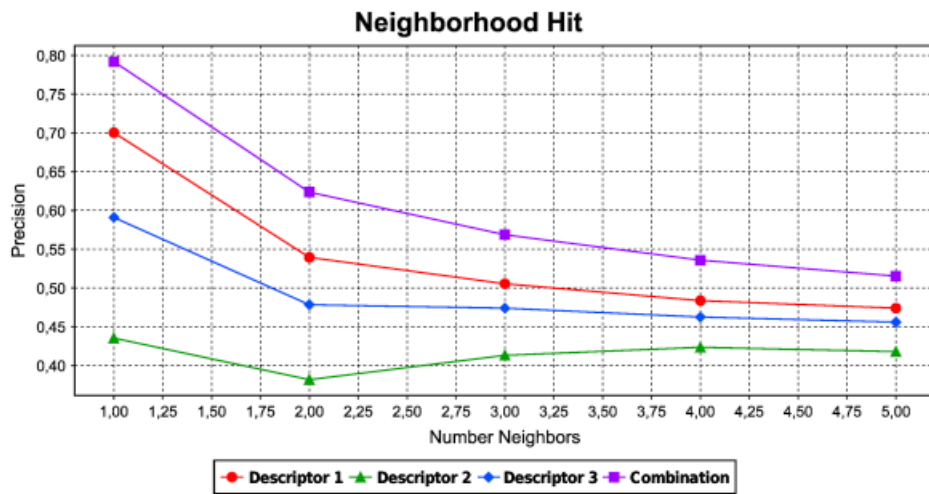


Figura 3.20: Neighborhood Hit para uma consulta de 5 vizinhos da classe 5.

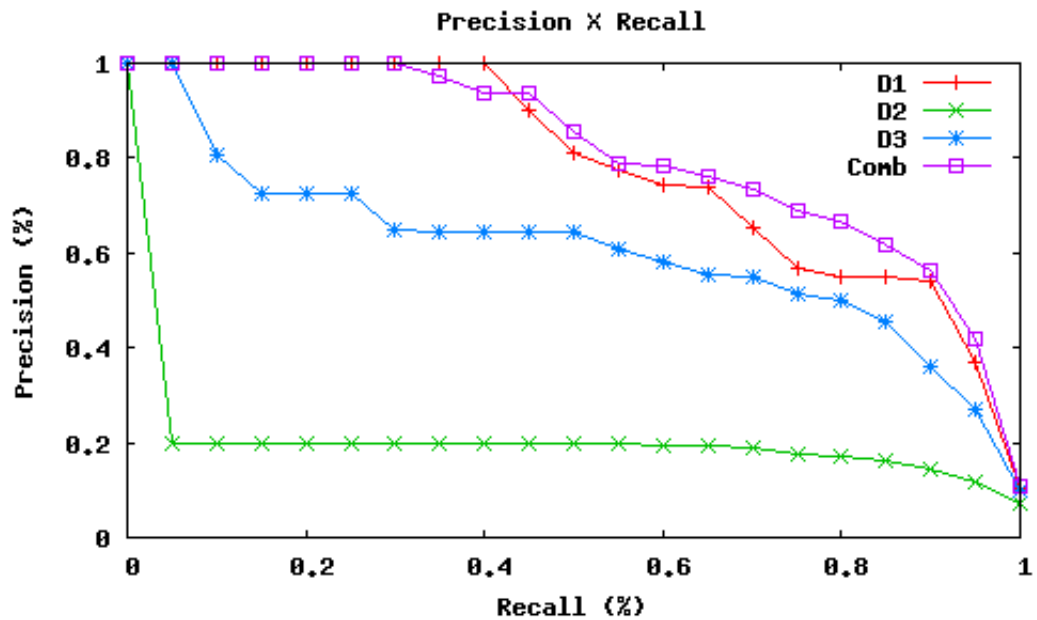


Figura 3.21: Gráfico de precisão e revocação para quatro testes realizados com uma imagem da classe 5.

Capítulo 4

CONCLUSÕES E TRABALHOS FUTUROS

4.1 Considerações Iniciais

Sistemas de Recuperação de Imagens por Conteúdo (*Content-Based Image Retrieval* - CBIR) permitem que sejam realizadas consultas em bases de imagens, retornando as imagens com maior grau de similaridade em relação à imagem de consulta (CÁCERES, 2010).

A combinação de um algoritmo extrator de característica e uma função de distância é geralmente referido na literatura como descritor (TORRES et al., 2009; BARROSO et al., 2015).

Os sistemas CBIR possuem o Gap Semântico (inconsistência entre a percepção do ser humano na avaliação de similaridade entre imagens comparada com a computada por sistemas CBIR) como uma das suas principais limitações (LIU et al., 2007). Para se combater o gap semântico em consultas por similaridade, o uso de combinações de múltiplos descritores tem sido considerado por demonstrar bons resultados (BARROSO et al., 2015), pois subconjuntos de imagens de um mesmo conjunto podem ser melhor representados por características diferentes.

Barroso et al. (2013) propõe o uso de condições de contorno para encontrar esses subconjuntos e escolher a melhor combinação de descritores para cada um deles. Condições de contorno são quaisquer informações associadas à imagem que possam ser utilizadas para delimitar subconjuntos de imagens.

A última etapa de um sistema CBIR é a visualização dos resultados de uma consulta. Cada sistema CBIR possui uma forma diferente de apresentar esses resultados (CÁCERES, 2010). A qualidade da visualização pode estar relacionada com a qualidade da resposta da consulta de sistemas CBIR.

Assim, a primeira hipótese testada neste trabalho foi a de que quanto mais precisos os

resultados de uma consulta, melhor pode ser a organização das imagens na visualização. A segunda hipótese experimentada neste trabalho foi a de que o uso de múltiplos descritores com condição de contorno pode melhorar a qualidade da visualização hierárquica de imagens em CBIR. As contribuições e ideias de trabalhos futuros relacionadas aos testes realizados com essas duas hipóteses do trabalho podem ser vistas a seguir.

4.2 Principais Contribuições

Este trabalho de mestrado demonstrou que a qualidade da visualização está relacionada com a qualidade do resultado de consulta. Ou seja, quanto mais precisos os resultados de uma consulta, melhor pode ser a organização das imagens na visualização.

Também foi explorada a hipótese de que o uso de múltiplos descritores com condição de contorno pode melhorar a qualidade da visualização hierárquica de imagens em CBIR, utilizando o algoritmo *Neighbor Joining*. Os testes demonstram que a técnica proposta por (BARROSO et al., 2015), apesar de representar um nível de precisão próximo do nível de precisão medido pelo melhor descritor individual, tende a melhorar a árvore de similaridade gerada.

A visualização hierárquica escolhida para o trabalho, a *Neighbor Joining* (SAITOU; NEI, 1987), é um dos métodos mais utilizados para a construção de uma árvore filogenética. Essa técnica, adaptada para a criação de árvores de similaridade, utiliza a ideia de encontrar pares de instâncias mais próximas em uma coleção de dados, de modo a minimizar o comprimento e o número de ramos da árvore gerada (PAIVA, 2012). Ela tem sido utilizada no contexto de classificação visual de dados (ELER et al., 2008, 2009; PAIVA et al., 2011; CRUZ, 2012; PAIVA, 2012).

Este trabalho contribui para que as técnicas de múltiplos descritores com condições de contorno e visualização baseada em árvore de similaridade NJ possam ser mais utilizadas no contexto de sistemas CBIR, pois demonstra que o uso da primeira melhora a qualidade da segunda.

A medida de qualidade da visualização de imagens *neighborhood hit* (PAULOVICH et al., 2008) indica maior porcentagem de vizinhos mais próximos de uma imagem na visualização (que pertencem à mesma classe da imagem de consulta) ao se utilizar a técnica de múltiplos descritores com condições de contorno (BARROSO et al., 2013).

4.3 Trabalhos Futuros

Os experimentos relacionados a hipótese inicial deste trabalho sobre a correlação de qualidade do resultado da consulta com a qualidade da visualização gerada testou apenas a visualização baseada em árvore de similaridade NJ. É interessante testar essa correlação utilizando outras técnicas de visualização. Outro trabalho futuro interessante é integrar a visualização NJ à consulta propriamente dita.

4.4 Considerações Finais

Neste capítulo foram apresentadas as conclusões deste trabalho de mestrado. Foram levantadas as hipóteses de correlação entre a qualidade do resultado de consulta em sistemas CBIR e a qualidade da visualização gerada nessas consultas e, também, do impacto do uso de múltiplos descritores com condições de contorno na visualização baseada em árvore de similaridade NJ. Foi relatado que experimentos realizados demonstram que a primeira hipótese pode ser verdadeira, porém, indicou-se mais testes como trabalho futuro. Por fim, conclui-se que a segunda hipótese demonstra que o uso de múltiplos descritores com condições de contorno melhora a qualidade da árvore NJ gerada na consulta por similaridade em sistemas CBIR.

REFERÊNCIAS

- AKGÜL, C. B. et al. Content-based image retrieval in radiology: Current status and future directions. *Journal of Digital Imaging*, Springer-Verlag, v. 24, n. 2, p. 208–222, 2011. ISSN 0897-1889. Disponível em: <<http://dx.doi.org/10.1007/s10278-010-9290-9>>.
- AKSOY, S.; HARALICK, R. M. Using texture in image similarity and retrieval. In: *Physical Chemistry Chemical Physics* 98(9):1185–1193. [S.l.]: World Scienti, 2000. p. 129–149.
- AKSOY, S.; HARALICK, R. M. Feature normalization and likelihood-based similarity measures for image retrieval. *Pattern Recognition Letters*, v. 22, n. 5, p. 563 – 582, 2001. ISSN 0167-8655. Image/Video Indexing and Retrieval. Disponível em: <www.sciencedirect.com/science/article/pii/S0167865500001124>.
- BAEZA-YATES, R. A.; RIBEIRO-NETO, B. *Modern Information Retrieval*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1999. ISBN 020139829X.
- BARIONI, M. C. N. *Operações de consulta por similaridade em grandes bases de dados complexos*. Tese (Doutorado) — Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2006. Disponível em: <<http://www.teses.usp.br/teses/disponiveis/55/55134/tde-28092006-151225/>>.
- BARROSO, R. F. et al. Using boundary conditions for combining multiple descriptors in similarity based queries. In: _____. *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications: 18th Iberoamerican Congress, CIARP 2013, Havana, Cuba, November 20-23, 2013, Proceedings, Part I*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. p. 375–382. ISBN 978-3-642-41822-8.
- BARROSO, R. F. et al. Speeding up the combination of multiple descriptors for different boundary conditions. In: *2015 Latin American Computing Conference (CLEI)*. [S.l.: s.n.], 2015. p. 1–11.
- BHATT, C.; KANKANHALLI, M. Multimedia data mining: state of the art and challenges. *Multimedia Tools and Applications*, Springer US, v. 51, n. 1, p. 35–76, 2011. ISSN 1380-7501. Disponível em: <<http://dx.doi.org/10.1007/s11042-010-0645-5>>.
- BUENO, R. et al. Using visual analysis to weight multiple signatures to discriminate complex data. In: *2011 15th International Conference on Information Visualisation*. [S.l.: s.n.], 2011. p. 282–287. ISSN 1550-6037.
- BUGATTI, P. H.; TRAINA, A. J. M.; TRAINA JR., C. Assessing the best integration between distance-function and image-feature to answer similarity queries. In: *Proceedings of the 2008 ACM Symposium on Applied Computing*. New York, NY, USA:

ACM, 2008. (SAC '08), p. 1225–1230. ISBN 978-1-59593-753-7. Disponível em: <<http://doi.acm.org/10.1145/1363686.1363969>>.

CRUZ, L. E. F. *Uma abordagem baseada em técnicas de visualização de informações para avaliação de características de imagens e aplicações*. Dissertação (Mestrado) — Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2012. Disponível em: <<http://www.teses.usp.br/teses/disponiveis/55/55134/tde-05122012-095421/>>.

CÁCERES, S. M. P. *Técnicas de visualização para sistemas de recuperação de imagens por conteúdo*. Dissertação (Mestrado) — Instituto de Computação, Universidade Estadual de Campinas, Campinas, 2010. Disponível em: <<http://www.bibliotecadigital.unicamp.br/document/?code=000779382>>.

DESELAERS, T.; KEYSERS, D.; NEY, H. Features for image retrieval - a quantitative comparison. In: *In DAGM 2004, Pattern Recognition, 26th DAGM Symposium*. [S.l.: s.n.], 2004. p. 228–236.

DESELAERS, T.; KEYSERS, D.; NEY, H. Features for image retrieval: An experimental comparison. *Inf. Retr.*, Kluwer Academic Publishers, Hingham, MA, USA, v. 11, n. 2, p. 77–107, abr. 2008. ISSN 1386-4564. Disponível em: <<http://dx.doi.org/10.1007/s10791-007-9039-3>>.

DIAS, R. L. *Mineração visual de imagens aliada a consultas pelos k-vizinhos diversos mais próximos: flexibilizando e maximizando o entendimento de consultas por conteúdo de imagens*. Dissertação (Mestrado) — Universidade Federal de São Carlos, São Carlos, 2013. Disponível em: <<https://repositorio.ufscar.br/handle/ufscar/557>>.

ELER, D. M. et al. Multidimensional visualization to support analysis of image collections. In: *2008 XXI Brazilian Symposium on Computer Graphics and Image Processing*. [S.l.: s.n.], 2008. p. 289–296. ISSN 1530-1834.

ELER, D. M. et al. Visual analysis of image collections. *The Visual Computer*, v. 25, n. 10, p. 923–937, 2009. ISSN 1432-2315. Disponível em: <<http://dx.doi.org/10.1007/s00371-009-0368-7>>.

ENGEL, D. et al. Structural decomposition trees. *Computer Graphics Forum*, Blackwell Publishing Ltd, v. 30, n. 3, p. 921–930, 2011. ISSN 1467-8659. Disponível em: <<http://dx.doi.org/10.1111/j.1467-8659.2011.01941.x>>.

FEKETE, J.-D. The infovis toolkit. In: *Proceedings of the IEEE Symposium on Information Visualization*. Washington, DC, USA: IEEE Computer Society, 2004. (INFOVIS '04), p. 167–174. ISBN 0-7803-8779-3. Disponível em: <<http://dx.doi.org/10.1109/INFOVIS.2004.64>>.

GRAF, F. *JFeatureLib v1.6.3*. out. 2015. Disponível em: <<http://dx.doi.org/10.5281/zenodo.31793>>.

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

- HUANG, P.; DAI, S. Image retrieval by texture similarity. *Pattern Recognition*, v. 36, n. 3, p. 665 – 679, 2003. ISSN 0031-3203. Disponível em: <www.sciencedirect.com/science/article/pii/S0031320302000833>.
- KASS, M.; WITKIN, A.; TERZOPOULOS, D. Snakes: Active contour models. *International Journal of Computer Vision*, v. 1, n. 4, p. 321–331, 1988. ISSN 1573-1405. Disponível em: <<http://dx.doi.org/10.1007/BF00133570>>.
- KHOTANZAD, A.; HONG, Y. H. Invariant image recognition by zernike moments. *IEEE Trans. Pattern Anal. Mach. Intell.*, IEEE Computer Society, Washington, DC, USA, v. 12, n. 5, p. 489–497, maio 1990. ISSN 0162-8828. Disponível em: <<http://dx.doi.org/10.1109/34.55109>>.
- KUMAR, A. et al. Content-based medical image retrieval: A survey of applications to multidimensional and multimodality data. *Journal of Digital Imaging*, Springer US, v. 26, n. 6, p. 1025–1039, 2013. ISSN 0897-1889. Disponível em: <<http://dx.doi.org/10.1007/s10278-013-9619-2>>.
- KURTZ, C. et al. A hierarchical knowledge-based approach for retrieving similar medical images described with semantic annotations. *Journal of Biomedical Informatics*, v. 49, n. 0, p. 227 – 244, 2014. ISSN 1532-0464. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1532046414000574>>.
- LAMPING, J.; RAO, R.; PIROLI, P. A focus+context technique based on hyperbolic geometry for visualizing large hierarchies. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 1995. (CHI '95), p. 401–408. ISBN 0-201-84705-1. Disponível em: <<http://dx.doi.org/10.1145/223904.223956>>.
- LIU, Y. et al. A survey of content-based image retrieval with high-level semantics. *Pattern Recognition*, v. 40, n. 1, p. 262 – 282, 2007. ISSN 0031-3203. Disponível em: <www.sciencedirect.com/science/article/pii/S0031320306002184>.
- PAIVA, J. et al. Improved similarity trees and their application to visual data classification. *Visualization and Computer Graphics, IEEE Transactions on*, v. 17, n. 12, p. 2459–2468, Dec 2011. ISSN 1077-2626.
- PAIVA, J. G. d. S. *Técnicas computacionais de apoio à classificação visual de imagens e outros dados*. Tese (Doutorado) — Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2012. Disponível em: <<http://www.teses.usp.br/teses/disponiveis/55/55134/tde-02042013-084718/>>.
- PAULOVICH, F. V. et al. Least square projection: A fast high-precision multidimensional projection technique and its application to document mapping. *IEEE Transactions on Visualization and Computer Graphics*, v. 14, n. 3, p. 564–575, May 2008. ISSN 1077-2626.
- POCO, J. et al. A framework for exploring multidimensional data with 3d projections. In: *Proceedings of the 13th Eurographics / IEEE - VGTC Conference on Visualization*. Chichester, UK: The Eurographs Association & John Wiley & Sons, Ltd., 2011. (EuroVis'11), p. 1111–1120. Disponível em: <<http://dx.doi.org/10.1111/j.1467-8659.2011.01960.x>>.

- RAHMAN, M. et al. Multimodal biomedical image retrieval using hierarchical classification and modality fusion. *International Journal of Multimedia Information Retrieval*, Springer London, v. 2, n. 3, p. 159–173, 2013. ISSN 2192-6611. Disponível em: <<http://dx.doi.org/10.1007/s13735-013-0038-4>>.
- RIBEIRO, M. X. *Suporte a sistemas de auxílio ao diagnóstico e de recuperação de imagens por conteúdo usando mineração de regras de associação*. Tese (Doutorado) — Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2008. Disponível em: <<http://www.teses.usp.br/teses/disponiveis/55/55134/tde-16022009-144432/>>.
- ROBERTSON, G. G.; MACKINLAY, J. D.; CARD, S. K. Cone trees: Animated 3d visualizations of hierarchical information. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 1991. (CHI '91), p. 189–194. ISBN 0-89791-383-3. Disponível em: <<http://doi.acm.org/10.1145/108844.108883>>.
- SAITOU, N.; NEI, M. The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Molecular Biology and Evolution*, v. 4, n. 4, p. 406, 1987. Disponível em: <[+ http://dx.doi.org/10.1093/oxfordjournals.molbev.a040454](http://dx.doi.org/10.1093/oxfordjournals.molbev.a040454)>.
- SONKA, M.; HLAVAC, V.; BOYLE, R. *Image Processing, Analysis, and Machine Vision*. PWS Pub., 1999. ISBN 9780534953935. Disponível em: <<https://books.google.com.br/books?id=TPRRAAAAMAAJ>>.
- TANBEER, S. K. et al. Efficient single-pass frequent pattern mining using a prefix-tree. *Information Sciences*, v. 179, n. 5, p. 559 – 583, 2009. ISSN 0020-0255. Special Section - Quantum Structures: Theory and Applications Workshop “Quantum Structures”. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0020025508004532>>.
- TORRES, R. da S. et al. A genetic programming framework for content-based image retrieval. *Pattern Recognition*, v. 42, n. 2, p. 283 – 292, 2009. ISSN 0031-3203. Learning Semantics from Multimedia Content. Disponível em: <www.sciencedirect.com/science/article/pii/S0031320308001623>.
- YU, Z. et al. Visual query processing for efficient image retrieval using a som-based filter-refinement scheme. *Information Sciences*, v. 203, n. 0, p. 83 – 101, 2012. ISSN 0020-0255. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S002002551200223X>>.

GLOSSÁRIO

BLOB – *Binary Large Object*

CBIR – *Content-Based Image Retrieval*

Fast NJ – *Fast Neighbor Joining*

Fast PNJ – *Fast Promoting Neighbor Joining*

LSP – *Least-Square Projection*

MAE – *Método de Acesso Espacial*

MAM – *Método de Acesso Métrico*

NJ – *Neighbor Joining*

PNJ – *Promoting Neighbor Joining*

P&R – *Precisão e Revocação*

Rapid NJ – *Rapid Neighbor Joining*

Rapid PNJ – *Rapid Promoting Neighbor Joining*

SGBD – *Sistema Gerenciador de Banco de Dados*

Apendice A

UTILIZANDO A BIBLIOTECA JFEATURELIB PARA EXTRAIR VETORES DE CARACTERÍSTICAS DE IMAGENS EM JAVA

JFeatureLib é uma biblioteca livre em Java que oferece implementações de vários extratores de características de imagens e detectores de ponto / região principalmente utilizados no área de pesquisa de Visão Computacional (GRAF, 2015).

A biblioteca está disponível através de seu projeto no GitHub, na página:
<<https://github.com/locked-fg/JFeatureLib>>. (Acesso em: 30/01/2017)

A.1 Extratores de características implementados

- AutoColorCorrelogram
- CEDD
- Color Histogram
- FCTH
- Fuzzy Histogram
- FuzzyOpponentHistogram
- Gabor

- Haralick Texture Features
- Histograms:
 - Gray-Histogram
 - RGB Histogram
 - HSB Histogram
- JCD
- LumnanceLayout
- MPEG7ColorLayout
- MPEG7EdgeHistogram
- Statistical Moments (os 4 primeiros: Mean, Standard Deviation, Skewneess e Kurtosis OpponentHistorgam)
- PHoG (Pyramid Histograms of Oriented Gradients)
- ReferenceColorSimilarity
- SURF
- SIFT
- Tamura
- Thumbnail

A.2 Como utilizar

A.2.1 Extrair características de várias classes de imagens:

Usando a linha de comando, as características podem ser extraídas pela classe *Extractor*.

Exemplo:

Neste exemplo teremos imagens de duas classes (*classA*, *classB*) e vamos extrair características de textura dessas imagens, utilizando Haralick. Os vetores de características devem ser salvos num arquivo CSV.

Passo 1: Extrair as características das imagens de *classA* e excrever em *output.csv*:

```
java -jar JFeatureLib.jar -D HARALICK -c classA -d /images/of/classA -o /write/output.csv
```

Passo 2: Extrair as características das imagens de *classB*, omitir o cabeçalho (-nh) e anexar em output.csv:

```
java -jar JFeatureLib.jar -D HARALICK -c classB -nh -d /images/of/classB --append -o /write/output.csv
```

A.2.2 Opções

As opções da classe *Extractor* podem ser exibidas simplesmente ao chamar a classe sem argumentos, ou com a opção *-help* (x.x.x é a versão da biblioteca que está sendo utilizada):

```
java -jar JFeatureLib-x.x.x-jar-with-dependencies.jar -help
```

Opção	Descrição
-append	anexa / acrescenta no arquivo de saída (padrão: falso = sobrescrever)
-help	exibe essas opções
-list-capabilities	lista os descritores registrados e a saída dos seus métodos suportados
-threads N	quantidade de <i>threads</i> (padrão: quantidade de processadores disponíveis)
-unpack-properties	extrai as propriedades padrão e de <i>logging</i> para o diretório atual
-D (-descriptor) VAL	usar este descritor de características (ex.: SIFT)
-c VAL	classe da imagem que deve ser escrita no arquivo de saída
-d (-src-dir) FILE	diretório contendo as imagens
-m (-masks-dir) FILE	diretório contendo máscaras
-nh	omitir cabeçalho
-o (-output-dir) FILE	arquivo de saída (padrão: features.csv)
-r	acessar os diretórios recursivamente (padrão: não)
-v	exibir mensagens de <i>debug</i>

Tabela A.1: Lista de opções de linha de comando para extração de características com *jFeatureLib*

A.3 Código de demonstração e Wiki

Alguns códigos de demonstração podem ser encontrados em:

<<https://github.com/locked-fg/JFeatureLib-Demo>>. Acesso em (30/01/2017)

A *wiki* do projeto com mais informações pode ser acessada em:

<<https://github.com/locked-fg/JFeatureLib/wiki>>. Acesso em (30/01/2017)