Formalização do domínio imagem para buscas por conteúdo em SGBDs relacionais

Josiel Maimoni de Figueiredo

Orientador: Prof. Dr. Caetano Traina Júnior

Tese apresentada ao ICMC-USP, como parte dos requisitos para a obtenção do título de Doutor em Ciências de Computação e Matemática Computacional.

USP - São Carlos
Maio/2005

Aluna: *Josiel Maimoni de Figueiredo*

A Comissão Julgadora:

Prof. Dr. Caetano Traina Junior

Profa. Dra. Renata Pontin de Mattos Fortes

Profa. Dra. Marina Teresa Pires Vieira

Prof. Dr. Paulo Mazzoncini de Azevedo Marques

Prof. Dr. Sergio Shiguemi Furuie

Às minhas filhas, Cecília e Raquel, que inspiram meu futuro.
À minha esposa, Marília, que cuida do meu presente.
Aos meus pais, Ângelo e Nilza, que construíram o meu passado.

Agradecimentos

Aos irmãos Daniela, Samuel, Raphael e Gabriela, e seus respectivos cônjuges Leonardo, Isabela, Lilian e Luís, que sempre estiveram prontos para ajudar nos momentos que precisei.

Aos sobrinhos Tainá, Inaê, Lia, Jaci, Cauê e Théo, que tornam a família mais alegre.

À vó Isabel, que demonstra força e alegria aos 96 anos.

Ao vô Vicente que, aos 91 anos mostra que sempre temos de acreditar no futuro.

Aos meus sogros, José Wilson e Silvia, pelo respeito e carinho que sempre demonstraram.

Aos cumpadres, Fernando e Jéssica, pelo carinho e amizade.

Ao Prof. Caetano Traina Jr, por acreditar em meu potencial e ser paciente.

À "Titia" Marina T. P. Vieira, que me ajudou a dar os primeiros passos na área acadêmica.

Ao prof Mauro Biajiz, por ser um amigo e sempre estar disponível para troca de idéias e experiências.

À profa Agma Traina por estar sempre por perto nos momentos que precisei.

Ao Prof Paulo M. A. Marques pela sua receptividade e paciência nas explicações sobre as imagens Médicas.

Aos amigos do GBDI (Grupo de Base de Dados e Imagens), que nessa fase mostraram companheirismo e amizade. Um agradecimento especial aos amigos Adriano e Marcos que mostram que a persistência é imprescindível para quem deseja o sucesso.

Às funcionárias Beth, Laura e Ana Paula, da Secretaria de Pós-Graduação do ICMC-USP, por estarem sempre dispostas a ajudar.

Ao pessoal do CHOPI (Centro Hospitalar de Pesquisas em Imagens) e do CCIM (Centro de Ciências das Imagens e Física Médica) da FMRP-USP (Faculdade de Medicina de Ribeirão Preto da Universidade de São Paulo), que mostraram muita simplicidade e amizade no curto período de convívio que tivemos.

Aos amigos do DCC-UFMT (Departamento de Ciências da Computação da Universidade Federal do Mato Grosso), espero que nossas próximas caminhadas sejam duradouras e proveitosas.

E por fim, aos Sombra e Corisco que sempre agem com alegria em suas bricadeiras, arranhadas, babadas, mordidas e bagunças.

Resumo

Sistemas de Recuperação de Imagens por Conteúdo (SiRICs) têm como objetivo realizar buscas em grandes coleções de imagens, recuperando aquelas cujos conteúdos são mais similares à uma imagem fornecida como parte do predicado de busca. A similaridade é expressada por uma função de distância (dissimilaridade) que calcula a relação entre um par de imagens para permitir que ambas sejam comparadas. Funções de distância usam características extraídas de cada imagem por um conjunto de algoritmos chamados extratores de características. Para melhorar o desempenho do processo de buscas, as características selecionadas são extraídas de cada imagem no momento em cada uma é armazenada na base de dados, criando um vetor de características para cada imagem. As operações subseqüentes da busca são realizadas usando os vetores de características no lugar das imagens. Antes de extrair as características, outros algoritmos de processamento de imagem são utilizados para pré-processar cada imagem, de forma a prepará-la para os extratores. Adicionalmente, é comum a existência de vários critérios que podem ser considerados no processo de cálculo da similaridade de duas imagens. Nos SiRICs atuais, para comparar imagens, é preciso definir (1) o critério de comparação, (2) os pré-processamentos necessários para execução dos extratores, (3) quais são os extratores utilizados, (4) quais características devem ser consideradas, (5) e qual função de distância deve ser aplicada. Todas essas definições têm de ser configuradas antes da comparação ser realizada. A complexidade desse processo levou ao desenvolvimento de SiRICs com poucas opções para configuração das operações de comparação. Além disso, não existe nenhuma representação formal do processo SiRIC como um todo. Este trabalho apresenta uma representação formal do conjunto completo de operações que compoem o processo de buscas por conteúdo em imagens, objetivando consultas por similaridade em grandes bases de dados relacionais. A aplicação desse formalismo é apresentada com os resultados experimentais gerados sobre imagens médicas de uma base de dados hospitalar.

FIGUEIREDO, J. M. de, *Formalização do domínio imagem para buscas por conteúdo em SGBDs relacionais*, São Carlos, 2005. 125 p. Tese de Doutorado - Instituto de Ciências Matemáticas e de Computação - ICMC, USP.

Abstract

Content-Based Image Retrieval Systems (CBIR) aims at searching in large collections of images, retrieving those whose contents are similar to an image given as part of the search predicate. Similarity is a relationship between a pair of images that enables their comparison, which is expressed by a distance (dissimilarity) function. Distance functions usually rely on a set of features extracted from each image by a set of image processing algorithms called feature extractors. To speed up the search process, selected features are extracted from each image when each one is stored in a image database creating a feature vector for each image. Further image searching operations are performed using the feature vectors in place of the images. Before extracting features, other image processing algorithms are usually employed to pre-process each image, preparing it for the extractors. Moreover, usually there are several criteria that can be considered when measuring how much two images are similar. In current CBIR environments, to compare images, one must define (1) the criteria, (2) the pre-processing needed before the extractors can be executed, (3) which are those extractors, (4) which features must be considered, (5) and which distance function must be used. All of these definitions must have been set before a comparison can be performed. The complexity of defining how to compare images has lead to the development of systems aiming CBIR that allow relatively few options to configure the image comparison operations. Moreover, no formal representation of the entire CBIR process exists. This work presents a formal representation of the complete set of operations composing the CBIR process, aiming to search images in large relational databases, using similarity queries. It is also reported a system developed using this formalism that enables the content-based retrieval of medical images from a hospital database, thus showing results of applying the presented formalism in a practical way.

Lista de Figuras

1.1 Arquitetura do ambiente CIRCE. 6

2.1 Fluxo de Dados pelos módulos de um SiRIC (Adaptado de [Smeulders et al., 2000]). 11

3.1 Algumas funções de distância da família L_p 35

4.1 Arquitetura Cliente/Servidor sem suporte a imagem pelo SGBD. 51
4.2 Nova Arquitetura Cliente/Servidor com suporte a imagem pelo SGBD. . . 55

5.1 Sumário das Classes de Processadores Imagem 59
5.2 Sumário dos Conceitos envolvidos em uma Expressão de Domínio 62
5.3 Exemplo de expressão de domínio. 64
5.4 Exemplo de um Ambiente Imagem I gerado pela execução de uma expressão de domínio. (a)I_0: imagem original, gerada pelo operador *Read*(). (b)I_2: imagem I_0 segmentada pelo operador *SegmentLung*(). (c)I_1: imagem I_0 segmentada pelo operador *TextureShape*() usando 5 classes. 65
5.5 Novo fluxo de dados (setas cheias) pelos módulos de um SiRIC. 69

6.1 Exemplo de declaração XML para definição de uma expressão de domínio. 73
6.2 (a)Comando SQL enviado para o SGBD. (b)Comando SQL convertido com o intuito de suportar o uso do domínio *Tomografia*. (c) Conjunto de tabelas criadas pela definição do domínio *Tomografia* e da relação *Paciente*. 77
6.3 Conjunto de tabelas criadas pela definição do domínio *Tomografia* e da relação *Paciente*. 77
6.4 Resultado fornecido pelo protótipo para uma consulta $10 - NN$, usando o critério *Cor*. 79
6.5 Expressão de domínio para processamento de CT de Pulmão. 80
6.6 Seqüência de processamento usando somente propriedades dos valores de Houusfield e configurados para execução sobre imagens de CT de pulmão. Os números representam as estapas de processamento equivalentes às linhas da Figura 6.5. 80

6.7 Expressão de domínio resultante da inclusão do operador $CountHU(valorHU \quad 1 \quad hu[1])$ na expressão de domínio $\theta_{pulmãoCT}$ (da Figura 6.5. 81
6.8 Expressão de domínio resultante da substituição do operador $SegmentHU()$ pelos operadores $Windowing()$ e $Minus()$ na expressão de domínio $\theta_{pulmãoCT}$ (da Figura 6.7). 82
6.9 Ilustração de várias seqüências de processamento no âmbito de um SiRIC. 83
6.10 Exemplo de uso do operador ConditionalCase, a expressão $\theta_{exameCT}$ usa o operador para determina qual das expressões serão chamadas no momento da execução. 84
6.11 Tempo médio de execução de 500 consultas em uma relação contendo 11.000 imagens de CT. (a) para responder consultas kNN, de acordo com os critérios Cor, $Textura$, e $TexturaObjetos$. (b)Para responder consultas Rq para um índice com imagens filtradas (com patologias) e outro índdice com todas as imagens da base. 85
6.12 Exemplo de generalização de duas expressões de domínio, com a expressão θ_A e θ_B sendo generalizadas pela expressão θ_C. 86
6.13 Ilustração representando a generalização mostrada na Figura 6.12. 87
6.14 Exemplo de especialização de um domínio em outros dois. 88
6.15 Aparência do histograma de CT de várias anatomias. (a)Exemplos de histograma completo para as anatomias indicadas. (b)Parte do histograma com projeção dos intervalos $[-1000, -800]$ e $[-200, 200]$, que são as características que influenciam na diferenciação das imagens. (c)Operadores que realizam as extração do Histograma e a projeção do intervalo equivalente aos mostrados em (b). 89

Abreviaturas

CIRCE	*Content-based Image Retrieval Core Engine.*
CT	Tomografia Computadorizada.
DICOM	*Digital Imaging and Communications in Medicine.*
GIS	*Geographical Information System.*
HIS	*Hospital Information System.*
HU	*Hounsfield Unit.*
IR	*Information Retrieval.*
kNN	*k-Nearest Neighbors Query* - Consulta por similaridade cujo tipo é aos *k*-Vizinhos Mais Próximos.
MAM	Método de Acesso Métrico.
MAE	Método de Acesso Espacial.
MAEP	Método de Acesso Espacial Pontual.
MAENP	Método de Acesso Espacial Não Pontual.
PACS	*Picture Archiving and Communication System.*
PCA	*Principal Component Analysis*
RIS	*Radiologic Information System.*
RM	Ressonância Magnética.
ROI	Região de Interesse (*Region of Interest*).
Rq	*Range Query* - Consulta por similaridade cujo tipo é por Abrangência.
SGBD	Sistema Gerenciador de Banco de Dados.
SiRIC	Sistema de Recuperação de Imagens por Conteúdo.
SQL	*Structured Query Language.*
SVD	*Singular Value Decomposition*
UDF	*User Defined Function.*
UDT	*User Defined Type.*

Sumário

Resumo i

Abstract ii

Lista de Figuras iii

Abreviaturas v

1 Introdução 1
- 1.1 Considerações Iniciais . . . 1
- 1.2 Motivação . . . 2
- 1.3 Objetivos . . . 3
- 1.4 Contexto do Trabalho . . . 5
- 1.5 Organização do Trabalho . . . 6

2 Sistemas de Recuperação de Imagens por Conteúdo 8
- 2.1 Considerações Iniciais . . . 8
- 2.2 Aspectos Gerais de um SiRIC . . . 10
  - 2.2.1 Aquisição . . . 10
  - 2.2.2 Anotação e Interação . . . 13
  - 2.2.3 Cálculo de Propriedades Locais . . . 14
  - 2.2.4 Extração de Características . . . 17
  - 2.2.5 Base de Características . . . 18
  - 2.2.6 Base de Imagens . . . 19
  - 2.2.7 Cálculo de similaridade . . . 19
  - 2.2.8 Apresentação e Resultado . . . 21
- 2.3 SiRIC e Semântica . . . 22
- 2.4 Conclusão sobre Sistemas de Recuperação de Imagens . . . 24

3 Dados Multidimensionais 26
- 3.1 Considerações Iniciais . . . 26
- 3.2 Construção . . . 27
- 3.3 Manipulação . . . 28

3.3.1 Redução de Dimensionalidade 29
3.4 Similaridade 33
3.4.1 Espaço Métrico 34
3.4.2 Consultas por Similaridade 35
3.4.3 Função de Distância 37
3.5 Armazenamento 39
3.5.1 Métodos de Acesso Convencionais 40
3.5.2 Métodos de Acesso Espaciais 41
3.5.3 Métodos de Acesso Métrico 43
3.6 Conclusões sobre Dados Multidimensionais 47

4 Domínio Imagem e SGBDs **49**
4.1 Considerações Iniciais 49
4.2 Arquitetura 50
4.3 SGBDs e Domínio de Dados 52
4.4 Consultas 52
4.5 Conclusões sobre Imagem e SGBD 55

5 Álgebra para o Domínio Imagem **57**
5.1 Considerações Iniciais 57
5.2 Operando Imagem 58
5.3 Operadores Imagem 58
5.3.1 Operadores de Manipulação 58
5.3.2 Operadores de Controle 60
5.3.3 Operadores de Persistência 61
5.4 Expressão de Domínio 62
5.5 Otimizações na execução das Expressões de Domínio 65
5.6 Conclusões sobre a Álgebra para o Domínio Imagem 68

6 Validação da Formalização do Domínio Imagem **71**
6.1 Considerações Iniciais 71
6.2 Arquitetura 71
6.2.1 Implementação de um SiRIC 72
6.2.2 Incorporação em um SGBD Relacional 74
6.2.3 Papéis dos Usuários 77
6.3 Resultados e Discussões 78
6.3.1 Estudo de caso 79
6.3.2 Operadores Imagem 81
6.3.3 Operadores de Controle 82
6.3.4 Operador de Projeção 84
6.3.5 Operadores de Persistência 84
6.3.6 Gerenciamento de Domínios 86
6.3.7 Definição e Construção de novos operadores 88
6.3.8 Construção das Expressões de Domínio 89
6.4 Conclusões sobre o uso das Expressões de Domínio 92

7 Conclusão 94
7.1 Considerações Finais 94
7.2 Principais Contribuições 96
7.2.1 Contribuições para o contexto dos SGBDs 97
7.2.2 Contribuições para o contexto dos SiRICs 97
7.2.3 Contribuições para o contexto dos PACs 98
7.3 Proposta de Trabalhos Futuros 98

Referências Bibliográficas 100

Capítulo

1

Introdução

1.1 Considerações Iniciais

Os Sistemas Gerenciadores de Banco de Dados (SGBDs) são sistemas que foram criados com o intuito de facilitar o armazenamento e a recuperação de grande quantidade de informações. No entanto, as informações a serem manipuladas por esses sistemas aumentaram demasiadamente tanto em quantidade quanto em variedade. A quantidade das informações em si não traz tantos problemas porque os sistemas atuais permitem a manipulação de informações na ordem de *Terabytes* e com respostas medidas em *milissegundos*.

O problema é que a variedade das informações exige que cada tipo seja tratado de forma específica. Por exemplo, formas diferentes de tratamento devem ser dadas para dados temporais, imagens e seqüências genômicas. Isso significa que as funcionalidades tradicionalmente implementadas nos SGBDs para garantir a eficiência na manipulação dos dados não são suficientes para atender à demanda gerada por esses novos tipos de dados.

Por outro lado, outras propriedades desses sistemas são imprescindíveis, como segurança, robustez, integridade e consistência. Essas propriedades são ortogonais ao tratamento dos dados, e implementá-las e integrá-las adequadamente é uma tarefa extremamente trabalhosa e complexa. Esses fatos aliados à onipresença dos SGBDs, faz com que, o enfoque de estender as funcionalidades de um SGBD seja a tendência mais natural para a manipulação de grande quantidade desses tipos de dados.

A necessidade de estender as funcionalidades de um SGBD, principalmente o Relacional, para inclusão de novos tipos de dados, demanda diversos esforços normal-

mente preocupados com a formalização e voltados para contextos específicos. Por exemplo, em [Gerber & Fernandes, 2004] é mostrada uma álgebra baseada na álgebra relacional para descoberta de conhecimento em base de dados, enquanto que [Eiter et al., 2000] estende a álgebra relacional para comportar valores complexos probabilísticos. Em [Chaudhuri et al., 2005] é apresentada uma integração com algoritmos da área de IR, ou seja, o tratamento de dados textuais. Para dados multimídia, em [Adali et al., 1999, Adali et al., 2000] é apresentada uma extensão da álgebra relacional para dados multimídia interativos, enquanto que [Ánafu et al., 2001] mostra uma álgebra voltada para similaridade de dados multimídia e, [Kiranyaz et al., 2003] mostra detalhes do modelo MUVIS que é um *framework* para dados multimídia.

Um das extensões mais difundidas é voltada para aplicações geográficas e usa uma implementação do padrão OpenGIS-SQL [Group, 1999]. Para essa extensão, diversas operações foram definidas com o intuito de contemplar todas as formas nas quais os dados espaciais podem ser manipulados. Isso inclui tipos de dados representando pontos, áreas, sistemas de coordenadas e diversas outras informações que são usadas não somente pelo SGBD, mas também pelos Sistemas de Informações Geográficas (GIS), como informações sobre a visualização dos dados. Para que o SGBD contemple todas essas informações, novas relações são adicionadas ao dicionário de dados, bem como é necessário a utilização de métodos de acesso específicos (como os discutido na Seção 3.5.2) objetivando melhorar o desempenho da execução das consultas. Por fim, a linguagem de acesso também é estendida com a inclusão de novos comandos para permitir ao usuário utilizar toda essa gama de novos recursos. O conjunto de operações comuns são definidas pelo padrão, contudo novas operações podem ser definidas.

Pelo exemplo do OpenGIS pode-se concluir que normalmente a inclusão de um novo tipo de dado no SGBD ocorre após a definição de um padrão para representação e manipulação desse novo tipo de dado e, depois esse padrão é incorporado ao SGBD e conseqüentemente adaptado ao padrão SQL. Este trabalho segue esse mesmo enfoque, porém voltado para o tipo imagem. Assim, uma formalização para manipulação do tipo imagem, com suporte para buscas por conteúdo, é criada e sua incorporação em um SGBD é implementada.

1.2 Motivação

Para a área médica o uso das imagens sempre foi uma necessidade [Ledley, 1987] e diversas especialidades as utilizam como principal (muitas vezes única) ferramenta de diagnóstico. O ambiente hospitalar contempla todas as complexidades envolvidas no tratamento de imagens, pois diversos tipos de dispositivos de aquisição geram grande quantidade de imagens de diversos tipos. A complexidade desse ambiente ultrapassa as questões de

variedade e de quantidade das imagens, e chega na questão do uso variado das imagens, pois uma mesma imagem é utilizada diferentemente por especialidades médicas distintas.

Em termos evolutivos, a existência de diversos dispositivos de aquisição e a necessidade de compartilhamento das imagens, culminou com a definição do padrão *Digital Imaging and Communications in Medicine* (DICOM) [Bidgood et al., 1998, Mildenberger et al., 2002, Hludov & Meinel, 1999], que é um formato de arquivo que possibilita a inclusão de informações médicas no cabeçalho do arquivo binário da imagem.

Uma imagem DICOM é gerada pelo dispositivo de aquisição e enviada para outro local em que ocorre seu armazenamento e distribuição, esse conjunto de sistemas é denominado *Picture Archiving and Communication Systems* (PACS) [Bick & Lenzen, 1999, Cabrera, 2002]. Sistemas PACS são tipicamente fechados, dedicados à armazenagem e distribuição de imagens em formatos proprietários, com funcionalidade rígida.

O problema desse conjunto de sistemas é que a capacidade de recuperação é limitada e, a recuperação de informações de uma imagem demanda obter não somente a imagem em si, mas informações que estejam relacionadas ao conteúdo da mesma. Um exemplo, é a identificação de patologias em uma imagem de Raio-X. Outra operação importante é a comparação entre imagens com intuito de identificar as semelhanças.

Esse tipo de tratamento diferenciado para as imagens é fornecido pelos Sistemas de Recuperação de Imagens por Conteúdo (SiRIC). SiRICs são sistemas que têm por objetivo realizar consultas em conjuntos de imagens usando como parâmetros informações presentes na própria imagem.

Esforços no sentido de integrar os PACS e os SiRICs têm sido realizados, como em [Bueno et al., 2002], que inclui funcionalidades de consultas por conteúdo em um ambiente PACS. No entanto, essa integração normalmente abrange somente aspectos específicos de todo o processo. O problema é que apesar de existir diversos esforços no sentido de melhorar a forma de manipulação e tratamento das imagens, nenhum trabalho conseguiu organizar o tratamento das imagens com um modelo uniforme que abrangesse todas as manipulações que as imagem sofrem nesses sistemas.

1.3 Objetivos

Este trabalho mostra o desenvolvimento de um ambiente para armazenagem e recuperação de imagens seguindo a conceituação de um sub-sistema em arquitetura aberta, como uma extensão ao Modelo Relacional, suportando o tipo imagem como mais um tipo de dado. Para isso, considera-se que as imagens são um tipo de dado, que apesar de terem um conjunto próprio de operadores, devem ter o mesmo suporte dado aos demais tipos de dados tradicionalmente suportados. Isso significa, principalmente, a possibilidade de realizarem-se consultas utilizando imagens como atributo de comparação, por exemplo,

permitindo-se que as condições de seleção sejam usadas como expressões de busca por similaridade. Este trabalho formaliza a manipulação das informações no contexto de um SiRIC, visando uniformizar todos os aspectos de processamento envolvidos com as imagens. Essa formalização possibilita que o processamento das imagens inclua condições semânticas, armazenamento e consultas por similaridade em bases de dados.

A maneira escolhida para inclusão das condições semânticas tem como um dos objetivos influenciar na velocidade de processamento dos dados. O funcionamento pretendido é análogo ao existente no núcleo de um SGBD, no qual os métodos de acesso implementados são utilizados pelos atributos de tipos de dados numéricos ou textuais, que podem ser indexados através de estruturas de dados próprias. Assim, quando uma consulta é recebida pelo interpretador de consultas, ela é analisada e planos alternativos de execução são gerados. Idealmente, o melhor plano dentre os vários gerados é escolhido. Em geral, a existência de uma estrutura de índices sobre atributos envolvidos na consulta leva a um desempenho melhor da estratégia que a utiliza, embora seu uso não seja obrigatório. Essa liberdade de escolha, entre várias alternativas, decorre do fato da linguagem de consulta (SQL) ser declarativa, ou seja, o interpretador de consultas é quem escolhe como executar a busca.

Este projeto segue essa mesma técnica, porém para o tipo de dados imagem. Tipos de dados complexos, e imagens em particular, podem ser indexados através de estruturas métricas ou estruturas espaciais, se um conjunto adequado de características forem deles extraídos. Uma estrutura de indexação deve utilizar o mesmo conjunto de características para todos os seus elementos, por isso, características que se revelam de pouca utilidade discriminatória para um subconjunto dos elementos de um conjunto de dados, mas significativamente relevantes para outro subconjunto, devem ser mantidas no conjunto.

Dessa maneira, explora-se a possibilidade de manter os dados indexados em mais de uma estrutura, sendo que o conjunto de todas as estruturas deve conter todos os dados a serem indexados, mas um dado estará indexado apenas nas estruturas que incluam as características mais relevantes para esse elemento de dado em particular. Ou seja, ocorre uma divisão horizontal nas relações de características, em partes não necessariamente mutuamente exclusivas. Os critérios para essa divisão são outras características, idealmente extraídas com um custo computacional relativamente reduzido, que permitem definir em qual (ou quais) divisões o dado deve ser indexado. Como cada divisão leva em conta apenas um subconjunto de todas as características, o processo como um todo se torna mais eficiente. Essa arquitetura permite que haja opções de otimização para escolha não apenas sobre uso ou não das estruturas de indexação, mas também na maneira como elas são construídas.

Os extratores obtêm características que são utilizadas para um processo de decisão baseada numa expressão condicional sobre seus valores, permitindo que um processo que

seria apenas uma seqüência de extratores seguidos por uma operação de indexação, possa ser controlado, entremeando essa seqüência por operações condicionais. Isso permite a criação de configurações com as quais o projetista possa criar alternativas de indexação a serem exploradas automaticamente pelo otimizador de consultas.

Este projeto modifica o funcionamento interno dos SiRIC de forma a permitir que o processamento dos dados possua um fator semântico determinado pelo usuário e posteriormente processado automaticamente. É apresentado também uma forma de gerenciar domínios das imagens de forma hierárquica, fazendo com que domínios de imagens possam ser especializados em subdomínios.

Embora os conceitos tratados neste projeto sejam, em princípio, válidos para quaisquer domínios de dados complexos, tais como dados multimídia, seqüências temporais e seqüências genéticas, este projeto tem o objetivo de dar suporte a armazenagem e recuperação de imagens, e mais especificamente, imagens médicas, obtidas de exames de pacientes em um ambiente hospitalar.

Assim, este projeto vai ao encontro de prover um ambiente que integre PACS e "Sistema de Informatização Hospitalar" (*Hospital Information System* - HIS) estendendo-os com a habilidade de recuperação baseada no conteúdo das imagens.

1.4 Contexto do Trabalho

O trabalho aqui descrito faz parte de um projeto maior do qual participam os grupos de pesquisa: GBDI (Grupo de Bases de Dados e Imagens) do ICMC-USP (Instituto de Ciências Matemáticas e de Computação) e o CCIFM-HC (Centro de Ciências das Imagens e Física Médica do Hospital das Clínicas) da FMRP-USP (Faculdade de Medicina de Ribeirão Preto).

A Figura 1.1 mostra a arquitetura geral do sistema no qual este projeto está inserido. Esse sistema, denominado CIRCE (*Content Based Image Retrieval Core Engine*) [Araujo et al., 2002, Rosa et al., 2002], tem como principal característica estender as funcionalidades de um SGBDR (Sistema Gerenciador de Banco de Dados Relacional) para prover funcionalidades de busca por conteúdo em imagens. A grande vantagem da proposta CIRCE sobre outros tipos de sistemas de recuperação por conteúdo é que ele possibilita que os sistemas atuais sejam utilizados sem que suas funcionalidades sejam alteradas.

Os módulos principais do CIRCE são: Interpretador de Consultas (IC), Extrator de Parâmetros (XP) e Métodos de Acesso (MA). O módulo IC inclui a extensão da linguagem SQL para suportar imagem como mais um tipo de dado, permitindo que o usuário especifique atributos de relações como sendo desse tipo. O módulo MA [Lopes, 2005] atua como interface do CIRCE com estruturas de acesso, como é o caso da estrutura

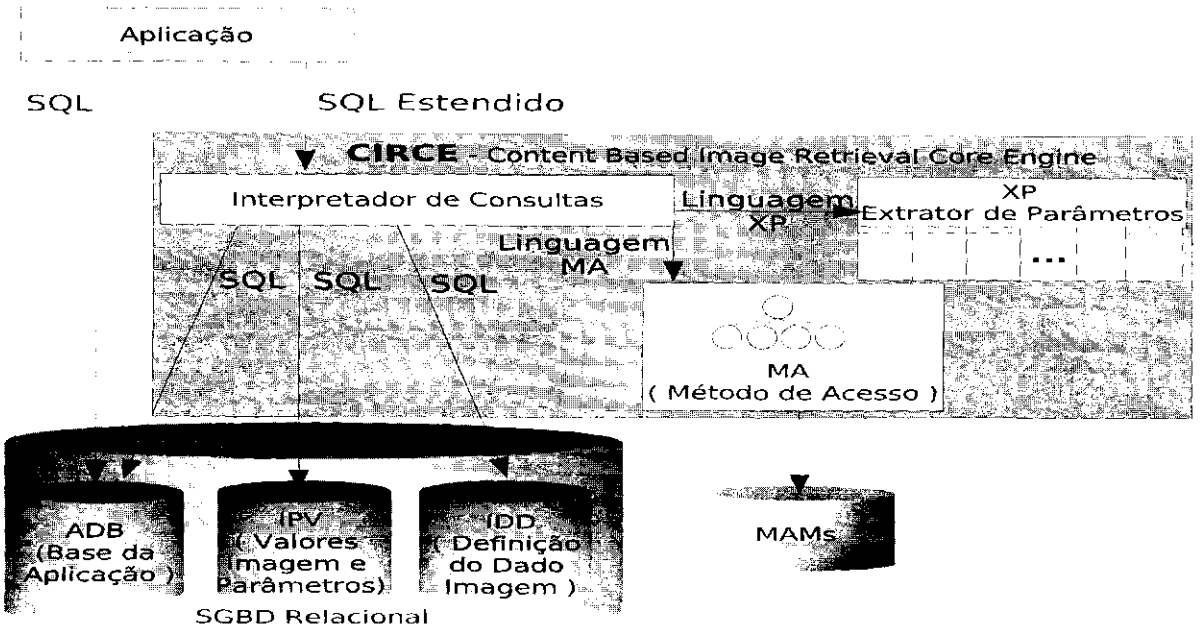


Figura 1.1: Arquitetura do ambiente CIRCE.

Slim-Tree[Traina Jr. et al., 2000]. O módulo XP atua no sentido de obter, das imagens, dados que servirão de parâmetros para os métodos de acesso.

Portanto, no contexto da arquitetura CIRCE, este trabalho está relacionado com a comunicação entre seus diversos módulos pretendendo que a maioria de seu processamento seja executado automaticamente.

1.5 Organização do Trabalho

Este documento está organizado com a seguinte estrutura:

- No Capítulo 2 são apresentadas as principais funcionalidades que um SiRIC deve possuir, com enfoque principal sobre os requisitos de SiRICs da área médica;
- O Capítulo 3 aprofunda a discussão sobre os conceitos e as técnicas utilizadas pelos SiRIC para manipulações dos dados multidimensionais;
- No Capítulo 4, são discutidas as influências e conseqüências da inclusão do tipo de dado imagem como um tipo nativo dos SGBDs Relacionais;
- No Capítulo 5, uma formalização algébrica para o domínio imagem é apresentada, com detalhamento a respeito das propriedades e uso dos operadores criados;
- No Capítulo 6 são abordadas questões sobre a validação da álgebra apresentada no Capítulo 5, bem como os resultados da aplicação dos conceitos sobre um conjunto

de imagens médicas;

- No Capítulo 7 as conclusões são apresentadas, destacando as principais contribuições deste trabalho, bem como as propostas para trabalhos futuros.

Capítulo

2

Sistemas de Recuperação de Imagens por Conteúdo

2.1 Considerações Iniciais

Este capítulo apresenta os aspectos gerais dos Sistemas de Recuperação de Imagens por Conteúdo (SiRICs), mostrando os principais módulos que um sistema desse tipo deve possuir e de que forma esses módulos funcionam e interagem. Apesar dos conceitos apresentados servirem para qualquer SiRIC, o enfoque principal é mostrar todos os aspectos relacionados a SiRICs em ambientes de apoio a medicina, ou seja, são apresentadas particularidades das imagens médicas e suas influências sobre os SiRICs. O embasamento teórico das operações embutidas em tais sistemas são apresentados no próximo capítulo.

Os principais trabalhos que nortearam a estrutura deste capítulo foram [Smeulders et al., 2000], [Antani et al., 2002], [Lehmann et al., 2003a] e [Müller et al., 2004]. Diversos outros trabalhos apresentam de uma forma mais genérica a evolução e os principais temas relacionados com a área de SiRIC, tais como [Marsicoi et al., 1997, Martinez & Marchand, 1998, Aslandogan & Yu, 1999, Jörgensen, 1999, Patella, 1999, Rui et al., 1999, Huang et al., 2000, Koskela et al., 2000, Veltkamp & Tanase, 2000, Ciaccia & Patella, 2001, Vailaya et al., 2001, Bartolini, 2002, Rummukainen et al., 2003, Sebe et al., 2003, Deb & Zhang, 2004, Kherfi et al., 2004].

Historicamente, trabalhos relacionados com a recuperação de imagens baseada em conteúdo começaram a partir de 1990 [Smeulders et al., 2000], e com maior destaque após 1997, sendo possível estabelecer uma relação direta desses trabalhos com a difusão e ba-

rateamento dos dispositivos para manipulação e armazenamento de grande quantidade de informações gráficas. É possível mostrar também que a evolução dos SiRICs ocorreu a partir da própria evolução de sistemas que manipulam e geram imagens, com a disponibilidade de dispositivos especializados como as placas gráficas e monitores de alta resolução. Contudo, é interessante notar que os sistemas que geram e manipulam as imagens continuam evoluindo separadamente dos sistemas que as armazenam. Um exemplo clássico ocorre com os RIS e PACS, pois o primeiro enfoca a manipulação das imagens, enquanto que o segundo se preocupa somente com o armazenamento e disponibilização das mesmas.

Como principal conseqüência, percebe-se que na maioria dos SiRICs os problemas contidos na geração e manipulação das imagens não são considerados como relevantes. Por outro lado, os programas que geram e processam as imagens normalmente não tratam as questões relacionadas com seu armazenamento e recuperação. As influências dessas separações são discutidas nas próximas seções.

Todos os SiRICs tentam minimizar a complexidade inerente da análise computacional dos dados visuais presentes no conjunto de *bits* que representam a imagem. Duas representações binárias bastante diferentes podem representar duas imagens visualmente muito semelhantes, ou seja, apesar dos *pixels* apresentarem valores distintos, as imagens podem conter as mesmas informações visuais. Dessa maneira, independentemente da forma de tratamento, o principal objetivo da manipulação das imagens, nos SiRICs, é gerar um conjunto de dados que sejam mais adequados para serem processados, ou seja, os dados binários devem ser transformados em dados numéricos e/ou textuais contendo informações o mais próximo possível da semântica envolvida na imagem.

O tratamento automático dos SiRICs envolve a aplicação de um modelo de processamento que gera conjuntos de dados que são considerados relevantes para a imagem. Esse modelo de processamento normalmente é aplicado em domínios muito especializados, pois a disparidade entre as características extraídas e sua interpretação semântica normalmente é pequena. Os exemplos mais difundidos desses modelos envolvem o reconhecimento de faces [Yang et al., 2002] e de impressões digitais, sendo que vários modelos geométricos e estatísticos foram criados tentando resolver esses problemas.

No entanto, para domínios mais abertos, esse tratamento é proibitivo, pois a quantidade de características a ser manipulada tende a ser gigantesca, o que torna a abordagem manual através de uma descrição textual mais simples, e é a que foi utilizada nos primeiros protótipos experimentais de SiRICs. Uma descrição textual (ou lingüística) envolve a participação ativa do usuário, gerando um tratamento manual das imagens. Um aspecto negativo é que a participação do usuário gera uma análise pessoal, assim, uma informação importante para um usuário pode não ser para outro. Uma vantagem do tratamento manual é que uma imagem é interpretada pelo seu conteúdo, isto é, através de seus objetos, de seu estilo, ou por uma razão semântica que pode ser muito difícil de ser gerada por

processamentos computacionais.

Esses dois tipos de enfoques, tratar imagens muito genéricas ou tratar imagens muito específicas, é percebido no contexto médico em trabalhos como em [Lehmann et al., 2003a], que faz um tratamento genérico, enquanto que outros trabalhos enfocam tipos específicos de imagens, como o tratamento específico para Raio-X de espinha [Antani et al., 2004b], exames histológicos [Tang et al., 2003] e tomografias de pulmão [Shyu et al., 1999a].

Independentemente do tipo de tratamento que um SiRIC realize sobre as imagens, os módulos e as etapas de tratamento das imagens tendem a ser os mesmos, conforme apresentados na próxima seção.

2.2 Aspectos Gerais de um SiRIC

A Figura 2.1, mostra as etapas de processamento de um SiRIC, destacando o fluxo de dados nas fases de processamento para inclusão e consulta. Para o fluxo da informação em uma operação de inserção (parte superior da figura), tem-se que o conjunto de imagens a ser mantido deve ser armazenado na base de imagens e cada imagem passa por um conjunto de processamentos que incluem a participação interativa do usuário, seguido de cálculos de propriedades locais em cada imagem e, finalmente a aplicação de extratores de características que podem atuar nas imagens individualmente ou no conjunto como um todo. Na fase de extração de características dois fluxos podem ser gerados, um que gera novas imagens a partir da original e outro que trata os vetores de características obtidos. As imagens são armazenadas na base de imagens e os vetores na base de características.

Em uma operação de consulta (representada na parte inferior da Figura 2.1), uma imagem de referência ou centro da consulta, serve de parâmetro para o processamento. Nesse momento, o mesmo processamento realizado na inserção de cada imagem é realizado sobre a imagem de consulta. Após a obtenção das características da imagem, elas são enviadas para o cálculo de similaridade com as características das imagens cadastradas na base. As imagens resultantes do cálculo de similaridade são apresentadas ao usuário, que pode selecionar subconjuntos que servirão de entrada para um novo processo de consulta.

Cada uma das etapas mostradas na figura envolve conceitos que individualmente possuem complexidades e desafios que demandam diversos estudos, assim os próximos tópicos apresentam, de forma sucinta, os principais aspectos relacionados a cada uma dessas etapas.

2.2.1 Aquisição

A fase de aquisição de uma imagem está relacionada com o processo de digitalização da mesma a partir de algum dispositivo de captura. Nesse processo a imagem é gerada em

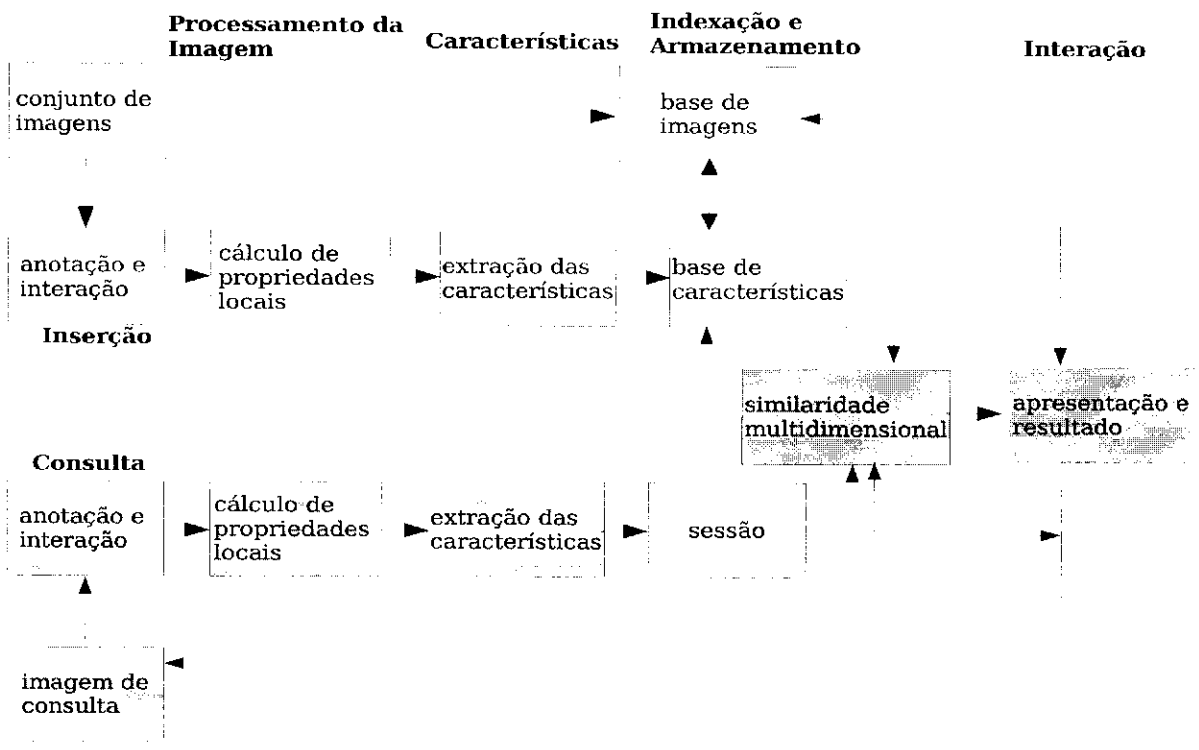


Figura 2.1: Fluxo de Dados pelos módulos de um SiRIC (Adaptado de [Smeulders et al., 2000].

um determinado formato específico, sendo comum o fato de que cada área de aplicação dos SiRICs utiliza formatos próprios. No caso da área médica, em várias modalidades a imagem é codificada em TIFF (*Tagged Image File Format*) em formato de 12 *bits* e embutida dentro de um arquivo que obedece ao padrão DICOM.

Grande parte dos dispositivos realizam, no momento da captura, processamentos com o intuito de diminuir efeitos de ruídos e distorções advindos de fatores externos. Para isso, normalmente são utilizadas as funcionalidades de calibragem desses dispositivos. Contudo a supervisão humana é essencial porque influências, como a movimentação do paciente, são de difícil tratamento e tendem a gerar imagens distorcidas.

A influência da fase de aquisição sobre as outras etapas de processamento é evidente porque métodos de processamento de imagens testados somente com uma base de dados de imagens podem gerar resultados falsos devido à grande variação de propriedades das imagens de diferentes bases [Wirth et al., 2004]. No contexto das imagens médicas isso é mais evidente pela quantidade de dispositivos de aquisição e, pelo fato de que esse processo pode gerar imagens diferentes quando equipamentos distintos e/ou com calibragens diferentes são utilizados. Com isso, informações adicionais sobre o processo de aquisição deveria existir nas imagens de forma a auxiliar os processamentos posteriores aos quais a imagem é submetida.

Outro aspecto, relacionado com o domínio das imagens médicas, é o fato de que

imagens do mesmo objeto podem ser geradas por dispositivos de aquisição variados, ou seja, variando o dispositivo é possível obter imagens diferentes de uma mesma origem, um mesmo pulmão pode ser a origem de uma imagem de Raio-X, ultrassom ou tomografia. Assim, essas várias imagens, apesar de serem de uma mesma fonte, devem ser processadas de formas distintas, sendo que um médico pode desejar fazer análises dessas várias imagens ao mesmo tempo e, um SiRIC ideal deveria tornar essa diferenciação de tratamento transparente ao usuário.

Nesse contexto é preciso entender o processo de aquisição para cada dispositivo, ou seja, de que forma os sinais são captados dos dispositivos e de que forma esses sinais são convertidos em dados binários e posteriormente em *pixels* das imagens. Um exemplo representativo é o processo de aquisição de imagens de Tomografia Computadorizada (CT), no qual a medida principal, feita pelo aparelho, é a atenuação da intensidade dos Raios-X sofrida em cada tecido do corpo. Nessa medida é utilizada como escala a Unidade de Hounsfield (HU), cujos valores variam de -1000 a $+1000$. As imagens são digitalizadas mapeando os valores de HU para escalas de cinza[Webb et al., 2001] que, normalmente, correspondem a valores entre 0 e 2000 para os *pixels* da imagem. Tradicionalmente, os valores mais baixos são mapeados tendendo ao preto e os valores mais altos ao branco. A intensidade de atenuação para cada tecido é conhecida, sendo que o aparelho tem sua calibragem realizada a partir desses valores. Assim, por exemplo, o valor de $-1000HU$ corresponde ao ar, $0HU$ à água e $+10000HU$ aos ossos, gerando na imagem digitalizada, preto para o ar, cinza intermediário para os tecidos moles e branco para os ossos. Outra propriedade dos valores de Hounsfield é que eles ajudam a identificar, na tomografia, além dos tecidos, algumas patologias. Um exemplo é que nódulos em estruturas pulmonares com uma atenuação por volta de $-110HU$ podem ser considerados como um sinal de um câncer benígno, enquanto que coeficientes de atenuação entre -20 e $-40HU$ indicam alta probabilidade de câncer malígno [Sperber, 2001]. Na visualização da imagem, um processo bastante utilizado é o de janelamento, no qual somente os valores de Hounsfield desejados são mostrados, facilitando identificações anatômicas e patológicas.

O fato de que cada dispositivo possui parâmetros específicos para gerar o valor de cada *pixel* em uma imagem, faz com que esses parâmetros influenciem em todo o processo de buscas por conteúdo. Um SiRIC deve prover funcionalidade para essa parametrização. Por exemplo, a relação entre câncer de pulmão e tonalidades de cinza (ou valores de Hounsfield) deve ser vinculada já nas etapas de processamento da imagem.

Além das informações pictóricas, é nesta fase que normalmente são adicionados ao arquivo imagem outros tipos de informações. No caso dos arquivos DICOM, seus cabeçalhos são preenchidos com diversas informações médicas, como nome do paciente, a modalidade do exame, data de aquisição, parâmetros de regulagem dos aparelhos de aquisição, entre outras. É interessante notar que essas informações do cabeçalho DICOM

podem ser utilizadas nas fases posteriores, contudo alguns poucos trabalhos as utilizam na fase de busca [Lehmann et al., 2000, Müller et al., 2000, Güld et al., 2002].

2.2.2 Anotação e Interação

Permitir que o usuário interaja com a imagem a ser inserida é uma das formas encontradas pelos SiRICs para tentar diminuir a grande distância semântica existente entre os dados analisados manualmente e os dados gerados automaticamente. Enquanto o significado de uma imagem depende do contexto de uso (o que é facilmente identificado pelo usuário), os algoritmos de processamento não realizam separação de contexto para processar a imagem.

Informações clínicas importantes não podem ser encontradas utilizando algoritmos que fazem análise global da imagem, pois essas informações são encontradas em variações de escala de cinza nos *pixels* vizinhos, o que pode indicar presença de patologias. Em imagens de CT de pulmão, patologias como efisemas aparecem em regiões de baixa atenuação e com textura diferente das partes normais. Como o objetivo principal das imagens médicas é auxiliar na identificação de informações relevantes para a geração de diagnósticos, a determinação da patologia e do local da mesma é o fator mais importante. Nesse sentido, diversos sistemas disponibilizam ao médico a possibilidade de destacar a região determinante da doença, também chamada de região de interesse (ROI). Em [Shyu et al., 1998] é utilizado um sistema que analisa imagens demarcadas pelo radiologista.

Com a região determinada, algoritmos de segmentação podem ser facilmente aplicados. Contudo, um fator complicador é que a escolha da ROI é uma atividade muito subjetiva, pois essa identificação é influenciada pela experiência e conhecimento do médico que está operando o sistema [Chabat et al., 2000].

Além das ROIs, outro aspecto diferencial da área médica é que a maioria dos exames envolve a utilização de mais de uma imagem (podendo chegar a mais de 100). Esse fator influencia na necessidade de um dispositivo de apresentação que possua a qualidade necessária, bem como uma área de apresentação grande o suficiente para visualização de mais de uma imagem. Por essa razão é comum, em ambientes radiológicos, o uso simultâneo de mais de um monitor.

Além disso, a marcação das imagens que realmente influenciam na determinação do diagnóstico é importante, pois são essas as imagens que norteiam o tratamento do paciente. Essas são algumas das propriedades que são obrigatórias em um RIS. Uma conseqüência da escolha das imagens mais representativas de um exame é que as outras imagens de um mesmo exame podem ser armazenadas em locais diferentes dessas. Imagens menos representativas poderiam ser armazenadas diretamente em dispositivos terciários, enquanto que as mais representativas, que são as imagens mais acessadas, podem ser mantidas na base ativa do sistema. Essa é uma outra funcionalidade que deveria ser

transparente para o usuário de um SiRIC.

2.2.3 Cálculo de Propriedades Locais

Um processo SiRIC não está preocupado em descrever totalmente o conteúdo de uma imagem, ou seja, é suficiente restringir o conjunto de informações naquele que é importante ao usuário [Shyu et al., 1998]. Assim, uma descrição do conteúdo é um primeiro objetivo a ser alcançado. Para descrever o conteúdo, são aplicados algoritmos de processamento de imagens com intuito de obter informações mais específicas, destacando os aspectos relevantes ao domínio da aplicação. Algoritmos de processamento de imagens, como de detecção de bordas, podem ser aplicados juntamente com a segmentação da imagem, de forma a "melhorar" os aspectos relevantes, ou retirar ruídos existentes na imagem [Hellier, 2003]. Em [Yang & Hansell, 1997] é apresentada uma forma de melhorar a identificação de inflamações nos bronquíolos, e nesse processo o autor utiliza técnicas de Wavelets para conseguir o resultado proposto.

Incluído no processo de identificação das propriedades locais, podem vir algoritmos de segmentação que limitam o cálculo somente às regiões resultantes da segmentação. A segmentação pode ser baseada em ROI [Vu et al., 2003] ou automática. A segmentação automática de imagens médicas pode ser feita por algoritmos genéricos de segmentação [Traina et al., 2004], por algoritmos que trabalham com modalidades específicas, como o caso de ultrassom [Hiransakolwong et al., 2003], CTs [Traina et al., 2003] ou ainda mais especificamente para órgão específicos em modalidades diferentes, como em imagens de ressonância magnética cardíaca [Glatard et al., 2004] ou CTs de pulmões [Chabat et al., 2000].

Existem muitos algoritmos para processamento de imagens. A maioria dos trabalhos existentes em SiRICs são classificados em três grandes grupos que tratam especificamente três aspectos importantes das imagens: a forma [Vailaya, 1996, Loncaric, 1998, Petrakis & Milios, 1999, Lee & Kim, 2001, Rantkorpi & Iivarinen, 2004], a cor [Park et al., 1997, Wang et al., 1997, Schettini et al., 2002, Luo & Nascimento, 2003] e a textura [Payne et al., 1999, Marques et al., 2000, Marques et al., 2002, Felipe et al., 2003, Grgic et al., 2003, Glatard et al., 2004]. Alguns trabalhos utilizam somente um desses aspectos, e outros mesclam características obtidas de dois ou mais desses aspectos [Kelly & Cannon, 1994, Huang et al., 1997, Chahir & Chen, 1998, Cinque et al., 1998, Comaniciu et al., 1998b, Tao & Grosky, 1998, Rao et al., 1999, Liapis & Tziritas, 2004, Saykol et al., 2004].

Independente do uso de características (simples ou compostas), muitas vezes é necessário realizar um processo de normalização delas. Por exemplo, para o histograma, as imagens podem possuir tamanhos diferentes ou codificações com diferentes níveis de cinza (uma imagem pode estar codificada em 12 *bits* enquanto outra pode estar em 8 *bits*).

Normalmente, no caso da área médica, imagens de um mesmo tipo de exame possuem as mesmas propriedades. Por exemplo, para exames de CT as imagens usualmente possuem tamanho de 512x512 e a codificação é feita em 12 *bits*.

Nessa fase percebe-se que em domínios vastos é difícil determinar quais tipos de algoritmos devem ser aplicados, tornando-se necessária a intervenção direta do usuário tanto na escolha do algoritmo quanto em seus parâmetros de aplicação. O realce das bordas e/ou a identificação de texturas podem não ser aplicados a todas as imagens, e até, para algumas imagens, esses processamentos podem distorcer informações relevantes. Para o caso de características de cor, o sistema de representação das cores pode alterar o processamento e consequentemente o resultado desejado, como em [Comaniciu et al., 1998a] onde é apresentado o uso do sistema RGB projetado em vetores $L * u * v$, e em [Liapis & Tziritas, 2004], onde o sistema HSV é utilizado para medir a percepção do usuário. Já em [Guldogan et al., 2003] é feita uma análise do efeito da compressão das imagens sobre a consulta baseada em características de cor.

Relacionado às características de forma em imagens médicas, um trabalho interessante é apresentado em [Rautkorpi & Iivarinen, 2004], no qual as características de forma são obtidas sem nenhuma segmentação prévia. Já em [Huang & Lee, 2004], é apresentado um método que trabalha com a relação espacial entre as imagens. O uso de agrupamentos (*clusters*) é mostrado em [Fauqueur & Boujemaa, 2003] para fazer a consulta sobre composição lógica das regiões. Outro exemplo desse tipo de estudo é apresentado em [Antani et al., 2004a], no qual imagens de Raio-X de espinha são utilizadas para avaliar o comportamento de 2 técnicas de extração de características relacionadas com forma: Fourier e aproximação por polígonos. O autor sugere que essas duas técnicas são as mais adequadas para tratar esse tipo de imagem, porque várias especificidades da área médica têm de ser atendidas, como a questão de que formas diferentes do mesmo órgão não necessariamente indicam patologias, pois formas em diferentes anatomias representam informações diferentes. Algumas formas seguem padrões pré-determinados (como pulmões) enquanto outras possuem aspectos arbitrários (como lesões). Dessa maneira, cada forma a ser representada possui suas próprias especificidades. Outro problema é que em vários casos as fronteiras da forma são difíceis de serem identificadas, como no caso de algumas lesões. Nos experimentos apresentados em [Antani et al., 2004a] a técnica de Fourier apresentou menor precisão com um desempenho melhor.

Para texturas, [Traina et al., 2004] faz a segmentação de imagens de Ressonância Magnética (RM) baseada nas texturas mais representativas presentes nas imagens. Também para RM, em [Glatard et al., 2004] filtros de Gabor são utilizados para tratamento de texturas em imagens cardíacas. Já em [Marques et al., 2002] a análise de texturas é utilizada com mamografias com intuito de auxiliar no diagnóstico de lesões de mama. Outro trabalho interessante é apresentado em [Payne et al., 1999], que faz uma breve re-

visão das técnicas de análise de texturas, também destaca que poucos trabalhos fizeram comparações entre essas técnicas e apresenta a comparação de 10 técnicas diferentes. Para validar os resultados das consultas, todas as técnicas foram aplicadas sobre o álbum Brodatz, que é uma coleção que possui 112 texturas repetidas em 1008 imagens. Além disso, vários usuários analisaram as respostas através de percepções próprias. Como resultado, os autores argumentam que os métodos estatísticos (com exceção ao método de Haralick GLCM) são mais adequados. Além disso eles propõem um conjunto de 84 imagens classificadas que podem ser utilizadas para futuras avaliações/validações de técnicas de textura. Outro trabalho que faz comparação entre texturas é apresentado em [Felipe et al., 2003], no qual algumas propostas para trabalhar com características de textura são apresentadas, sendo indicado um conjunto de características que foi o que gereou o melhor resultado no processamento das buscas.

Alguns trabalhos buscam usar características que são invariantes à algumas transformações que possam ocorrer na imagem. Nesse contexto, algumas técnicas tentam minimizar efeitos de mudanças em escala, cor e transformações isométricas, como em [Distasi et al., 2003] que utiliza codificação fractal para isso. Já em [Campo & Traina, 2003], é apresentada uma extensão ao histograma métrico [Traina et al., 2002a] com o intuito de torná-lo invariante às alterações de cor e brilho.

Uma situação mais complexa, e muitas vezes necessária, é a aplicação de algoritmos diferentes em partes diferentes da imagem, o que no caso das imagens médicas é uma funcionalidade interessante porque diversas patologias são identificadas a partir de achados em partes distintas de uma mesma imagem. Um exemplo apresentado em [Chabat et al., 2000] utiliza, em uma imagem de CT de pulmão, a segmentação das veias, seguida da segmentação de áreas de baixa atenuação, seguida da detecção dos brônquios vasculares. Nesse exemplo, o conjunto de operações distintas serviu como base para a identificação adequada da patologia. Um SiRIC deve facilitar esse tipo de manipulação, bem como tratar de maneira uniforme processamentos diferentes, visando facilitar o uso do sistema nesse tipo de contexto.

Outro aspecto a ser destacado é que o formato DICOM em 12 *bits*, da codificação binária da imagem, pode inviabilizar o uso de técnicas como os valores de Haralick, que são gerados a partir da matriz de co-ocorrência. Essa dificuldade ocorre porque o tamanho dessa matriz para imagens de 12 *bits* torna o processamento inviável. Nesse caso a transformação da imagem é necessária e, normalmente ela é convertida para 8 *bits*. Contudo, nenhum trabalho avaliou ainda as conseqüências dessa transformação para o resultado da busca por conteúdo, mesmo considerando que os valores normalizados nessa transformação perde várias informações.

2.2.4 Extração de Características

Um SiRIC normalmente considera que a imagem pertence a um domínio, cujo modelo computacional para interpretá-lo está implementado e embutido em seus módulos. A interpretação de todas as imagens é baseada nesse modelo de processamento, que gera um vetor com dados numéricos sobre a imagem. Assim, a etapa extração de características consiste em obter a partir das informações geradas na etapa anterior um conjunto de dados que seja mais adequado para tratamento computacional, o que se resume em obter conjuntos de dados numéricos e/ou textuais que de alguma forma represente mais adequadamente a imagem ou as informações consideradas relevantes da mesma.

Dessa maneira, a parte da imagem segmentada, a textura e/ou as cores são utilizadas por algoritmos que tentam expressar em números as suas propriedades. A maior dificuldade é conseguir expressar, com valores semelhantes para as propriedades, as imagens consideradas semanticamente semelhantes. Além disso, os valores obtidos da etapa anterior (cálculo das propriedades) podem conter informações redundantes e/ou não relevantes, que para serem retiradas demandam o uso de técnicas especiais, chamadas de técnicas de redução de dimensionalidade. Essas técnicas são variadas e seus resultados são dependentes do contexto de uso, com isso é necessário comparar diversas técnicas e verificar qual atende às necessidades.

Esse processo de testar e comparar diversas técnicas é exaustivo e pode não trazer os ganhos esperados. Por essa razão, muitas vezes técnicas de inteligência artificial e reconhecimento de padrões [El-Naqa et al., 2004] são utilizadas nesta fase do processamento do SiRIC. No trabalho apresentado em [Puuronen et al., 2000] a escolha de características é feita considerando que subconjuntos são relevantes apenas para aquela região do espaço no qual eles se localizam. Assim o cálculo é realizado utilizando um algoritmo de classificação baseado em árvores de decisão e as características presentes no caminho percorrido na árvore são as escolhidas.

Outras informações além das pictóricas podem ser necessárias, principalmente em contextos como os da área médica [Lehmann et al., 2000, Tang et al., 2003]. Por exemplo, informações sobre o diagnóstico, anatomia, posição do paciente e modalidade auxiliam na classificação da imagem. Alguns trabalhos utilizam códigos próprios, como o código IRMA [Lehmann et al., 2003b, Lehmann et al., 2004], que é composto de quatro campos que identificam a modalidade, a posição do paciente, a anatomia e o sistema biológico contidos na imagem. Outra forma de utilizar códigos é a aplicação de códigos padronizados da área médica, como no caso do Código Internacional de Doenças (CID10) [Carro et al., 2003].

Além das questões colocadas neste tópico, existem outros aspectos teóricos que devem ser analisados nesta fase do processamento de um SiRIC. Esses aspectos envolvem questões como a manipulação de dados multidimensionais, reconhecimento de padrões e geometria computacional. Os aspectos teóricos e mais genéricos relacionados com a fase

de extração de características são apresentados com mais detalhes no próximo capítulo.

2.2.5 Base de Características

A base de características normalmente é formada por um conjunto de vetores que são tratados como dados multidimensionais. Assim, se um conjunto de N características puder ser tratado como o conjunto das dimensões em um espaço $N-dimensional$, ele pode ser indexado utilizando um método de acesso (indexação) espacial [Gaede & Günther, 1998, Petrakis, 2002]. Outra alternativa é que seja criada uma função de dissimilaridade (ou de distância) métrica, que permita definir a distância entre dois conjuntos de características, o que permite que o conjunto seja indexado utilizando um método de acesso métrico [Chávez et al., 2001, Traina et al., 2002b]. Na Seção 3.5 são apresentados com detalhes os conceitos envolvidos no uso e acesso de dados multidimensionais.

Além do enfoque multidimensional, outros tipos de características podem ser utilizadas, como em [Zhu et al., 2002], que tenta aplicar teorias de buscas textuais em buscas de imagens fazendo com que palavras-chaves (chamadas de *keyblocks* pelo autor) representem o conteúdo de uma imagem. Dessa maneira, a busca se torna uma busca textual fazendo com que as técnicas de recuperação de informação se aproximem das de buscas por conteúdo em imagem.

Apesar de diversos trabalhos considerarem isoladamente o tratamento de características baseadas em descrições textuais, ou descrições visuais, alguns trabalhos sugerem o uso simultâneo das mesmas [Djeraba et al., 1998, Xu et al., 2000, Besson et al., 2003].

O uso de mais de uma base de características pode ser feito de forma que uma base sirva como dicionário para classificação das outras [Huang et al., 1998, Chang et al., 1998], sendo que as imagens mais representativas tendem a ser inseridas nesse dicionário. Em [Grgic et al., 2003], o dicionário é baseado no uso de centróides (que são calculados usando Wavelets de Gabor) que classificam as imagens, assim os centróides são armazenados em uma base separada que é utilizada em um primeiro passo na consulta. Em [Brodley et al., 1999] são apresentados testes sobre imagens de CT de pulmão, com um método que utiliza um conjunto de características que melhor representa imagens já classificadas e outro conjunto de características para classificar subclasses dessas imagens. Já em [Djeraba, 2003], são utilizados histogramas de cor e descritores de Fourier para textura. Assim, o enfoque é extrair automaticamente a relação entre essas duas características nas diversas imagens. Para facilitar as buscas, as imagens são divididas em repositórios, suas características são extraídas e calcula-se para cada uma a semântica que a discrimina das outras.

Considerando que uma característica é qualquer informação adicional à presente nos *pixels*, para as imagens médicas, a informação contida nos cabeçalhos DICOM pode ser considerada como um conjunto de características. Nesse caso, esse conjunto de caracte-

rísticas indica uma rotulação da imagem e pode ser utilizado na forma de armazenamento, como em [Scott & Shyu, 2003], que faz uma extensão da *k-d tree* (chamada de EBS *k-d tree*) para incluir essas informações de rotulação.

Uma particularidade do cabeçalho DICOM é que as informações nele contidas ficam repetidas em cada uma das imagens, sendo indicado um controle dessa redundância. Em [Power et al., 2004] é apresentado um sistema baseado em malhas (*grid*) que armazena os cabeçalhos das imagens DICOM em bases relacionais. Outro aspecto dos cabeçalhos DICOM é que, se o ambiente possuir restrições de espaço de armazenamento, a imagem DICOM pode ser "desmontada" e somente no momento da disponibilização da mesma, o formato DICOM pode ser montado novamente e a imagem ser enviada como resposta da requisição.

2.2.6 Base de Imagens

Bases de Imagens armazenam as imagens propriamente ditas, isto é, a representação da imagem em formato binário. O uso da base de imagens ocorre logo após a digitalização, com o intuito de armazenamento e disponibilização para outros sistemas. Nesse contexto, o armazenamento ocorre tradicionalmente de duas maneiras: com um SGBD ou com uma hierarquia de diretórios.

Para armazenamento interno no SGBD, as imagens são armazenadas como um conjunto de *bits* através do tipo de dado *Binary Large Object* (BLOB). Vale ressaltar, que para o SGBD não importa qual conjunto de *bits* está sendo armazenado e, normalmente nenhuma validação é feita para verificação do tipo do dado, como o formato correto da imagem. Maiores detalhes a respeito da influência de dados do tipo imagem sobre a tecnologia de SGBD é apresentada no Capítulo 4.

Para o caso do armazenamento em hierarquias de diretórios, cada imagem gera um arquivo que é armazenado em um diretório específico e acessado pelo sistema através de seu nome. No contexto médico, o sistema que faz acesso às imagens é conhecido como Servidor DICOM. Ele é utilizado com o intuito de centralizar e controlar o acesso às imagens geradas pelos diversos dispositivos do ambiente hospitalar.

2.2.7 Cálculo de similaridade

No contexto das aplicações que utilizam imagens, a funcionalidade de encontrar duas imagens totalmente iguais tem um custo de processamento alto e na maioria das vezes não tem sentido prático algum. Por essa razão faz mais sentido realizar comparações entre duas imagens para descobrir quão parecidas elas são, ou seja, o grau de similaridade entre as mesmas. Resumidamente, a similaridade entre imagens é medida através da aplicação de uma função de distância, sobre as características extraídas, que mostra o quão similares

duas imagens são.

A comparação de dados simples, como valores numéricos, é feita medindo-se a distância em unidades entre seus elementos, o mesmo ocorrendo para objetos em um espaço vetorial, em ambos os casos pode-se considerar que um ponto de origem dos dados (eixo de coordenadas) é utilizado. Contudo para dados imagem não existe o conceito de origem dos dados, assim, todas as comparações devem ser realizadas entre as imagens disponíveis. Além disso, considerando que os dados a serem analisados para o cálculo da similaridade são os dados obtidos a partir da etapa de extração de características, então a análise é dependente das características que foram geradas. Por essa razão, os cálculos têm de ser diferentes para características distintas, isto é, funções de distância diferentes devem ser aplicadas em conjuntos de características diferentes.

Uma função de distância tem seu uso mais efetivo quando ocorre especificamente sobre as características em que ela faz uma melhor discriminação. Em [Kokare et al., 2003] são avaliadas diversas funções de distância quando utilizadas com texturas. Já [Shah et al., 2004] faz uma avaliação a respeito da influência do dinamismo dos dados nas buscas por similaridades em imagens. A lógica nebulosa é utilizada nas funções de distância aplicadas por [Zhang & Zhang, 2003], enquanto que [Vasconcelos, 2004a] dá um enfoque de cálculo de probabilidade nas medidas de similaridades entre os vetores de características.

Para um SiRIC, é importante determinar o momento no qual o cálculo de similaridade é realizado. Na arquitetura apresentada na Figura 2.1, a inserção de uma nova imagem e a fase de consulta determinam dois momentos em que o cálculo de similaridade são usados. Para a fase de inserção, o resultado do cálculo de similaridade determina o "local" em que o vetor de característica é inserido.Nesse momento, alguns sistemas aplicam classificadores que podem ser baseados em redes neurais [Sheikholeslami et al., 1998, Lee, 2003], ou até mesmo a utilização de um conjunto desses classificadores [Skrypnik et al., 1999]. As imagens médicas têm a vantagem de possuir informações nos cabeçalhos DICOM, o que facilita o processo de classificação. Contudo conforme mostrado em [Güld et al., 2002], buscas por conteúdo utilizando somente os cabeçalhos DICOM não são confiáveis.

O cálculo da similaridade no momento da consulta obedece aos mesmos passos da inserção da imagem. Contudo, os parâmetros da consulta influenciam de maneira diferente a seqüência de processamento, ou seja, enquanto que na inserção a idéia é determinar qual o local, na consulta a idéia é determinar quais imagens são mais similares dependendo do parâmetro de consulta. Nesse caso, o ambiente pode permitir que o usuário expresse em forma de desenhos (*sketch*) a imagem de consulta [McDonald & Tait, 2003], embora essa técnica seja mais adequada para identificar imagens de objetos de uso cotidiano do que em imagens médicas.

Vale lembrar que as consultas por conteúdo em SiRICs envolvem, na maioria das

vezes, as consultas por vizinhos mais próximos (kNN) ou as consultas por abrangência (Rq). Consultas por vizinhos mais próximos visam encontrar quais são os n vizinhos mais próximos da imagem de consulta. Um exemplo seria, encontre as 5 imagens de Raio-X de cabeça que mais se parecem com a imagem de Raio-X da cabeça do paciente José. Já uma consulta por abrangência busca todas as imagens que estão dentro de um raio r de distância da imagem de consulta. Uma consulta ilustrativa busca todas as imagens de CT que estão à uma distância de 10 unidades da imagem do paciente José. Algumas classificações incluem esses tipos de busca como buscas exatas sobre os dados. Contudo diversos tipos de aplicações não demandam uma resposta muito precisa. Assim, para flexibilizar o resultado e tentar diminuir o processamento alguns trabalhos incluem o uso de lógica nebulosa [Chiu et al., 2003] e outros trabalham com consultas aproximadas[Amato, 2002, Bueno et al., 2005].

2.2.8 Apresentação e Resultado

Alguns SiRICs fornecem ao usuário a facilidade de realização de uma nova consulta restringindo os dados sobre o resultado da consulta anterior, tentando, dessa maneira, facilitar o processo de navegação sobre os dados.

Quanto ao fato das consultas gerarem resultados imprecisos, dois conceitos sobre os dados resultantes são importantes: **falso negativos** e **falso positivos**. Um dado considerado falso negativo é aquele que deveria estar no conjunto de resposta, mas o processo de busca o descartou. Um dado falso positivo é aquele que deveria ser descartado pelo processo de busca, contudo ele foi incluído na resposta.

Para o contexto dos SiRICs, um processo de busca que gere falsos negativos é inadequado, pois dados importantes podem ser descartados. Contudo, um processo de busca que apresente falsos positivos não acarreta muitos problemas porque o usuário pode descartá-lo manualmente, sendo essa uma razão importante para os SiRICs possibilitarem a realimentação de uma consulta a partir do resultado de outra.

Um SiRIC, devido à sua complexidade e manipulação de grande quantidade de dados, tende a ser um sistema multiusuário com vários acessos simultâneos. Assim, a utilização de sessões para cada um dos acessos facilita o gerenciamento e a separação dos recursos utilizados por cada usuário.

Um aspecto importante quanto ao armazenamento de dados relativos à sessão do usuário relaciona-se com as consultas. O gerenciamento de sessões facilita a reutilização de resultados de consultas anteriores e gera dados que possibilitam que o sistema realize suposições com o intuito de apresentar resultados mais relevantes ao usuário. Além de facilitar a navegação, a idéia é permitir que essa navegação gere informações que auxiliem nas próximas buscas [Shyu et al., 1999a]. Várias pesquisas estão sendo realizadas nesse contexto, sendo que o termo mais utilizado para referir a esse tipo de funcionalidade

é realimentação de relevância (*relevance feedback*), ou seja, retorno para o sistema da relevância para o usuário dos resultados que o SiRIC gera.

O principal objetivo de tratar a relevância dos resultados é fazer com que o sistema aprenda, a partir da interação com o usuário. Com isso, pode ser realizado um reajuste automático da consulta utilizando as informações sobre a relevância dos resultados obtidos em consultas anteriores [Rui et al., 1997]. Alguns tipos de sistemas que utilizam a realimentação fazem esse tratamento através da categorização das informações e da realização de inferências utilizando diversas técnicas, como redes neurais, redes bayesianas [Meilhac & Nastar, 1999], composição de atributos [Chua et al., 1999] e uso de relevância negativa [Ashwin et al., 2002]. Já em [Nastar et al., 1998] é feito um refinamento da consulta utilizando cálculos estatísticos cujos parâmetros são baseados nas respostas que o usuário forneceu.

2.3 SiRIC e Semântica

Os módulos e exemplos de SiRICs aqui apresentados mostram os problemas mais usuais que têm de ser resolvidos por esse tipo de sistema. Entretanto, pode-se generalizar que o principal problema dos SiRICs é a incongruência existente entre a informação semântica, com alto nível de abstração do que o usuário deseja [Shyu et al., 1999b], com a informação elementar e de baixo nível gerada pelos algoritmos automáticos [Enser & Sandom, 2003].

Nesse contexto o uso de palavras chave para tratar as informações semânticas se mostra como a forma mais simples. Para o caso de sistemas que realizam processamentos automáticos, a principal vantagem se refere ao tratamento uniforme dados às imagens. Em [Xu et al., 2000] é apresentada uma avaliação do uso de texturas para buscas e conclui que descritores semânticos são necessários, mesmo que sejam simples.

Outra proposta pretende utilizar simultaneamente as duas abordagens [Zhou & Huang, 2002] com o uso de um dicionário de sinônimos especialmente tratado para utilizar palavras-chave encontradas nas imagens. Essas palavras-chave podem ser geradas por processamento de imagens, reconhecimento de voz, entre outros, sendo que nem todas as imagens necessitam dessas palavras chave. No entanto, o problema de que parte das palavras-chave devem ser geradas pelo usuário mantém o processo de entrada de dados trabalhoso e dependente da interpretação do usuário. Além disso, as informações semânticas continuam sendo tratadas explicitamente através de processamento de palavras.

Alguns modelos tentam expressar a semântica através de formas específicas de expressão contextual, como grafos [Huijsmans & Sebe, 2005]. Contudo essa forma é feita independente da manipulação já realizada sobre as imagens e apesar da riqueza semântica alcançada ela adiciona um trabalho extra ao usuário.

Uma outra forma de tratar a semântica da aplicação é utilizada em [Miao et al., 2004], que permite a definição de conceitos semânticos que são mapeados para comandos SQL. Um exemplo citado é que o pedido de uma lesão cerebral "grande", é mapeado para uma consulta cujo predicado é o atributo tamanho maior que 67.

Outra questão importante relacionada com a semântica é a da falta de tratamento dos domínios das imagens. O domínio de uma imagem pode ser classificado como o contexto no qual a imagem é utilizada e processada. O conceito é recursivo, no sentido de que um domínio pode possuir sub-domínios, que também podem ser divididos em outros sub-domínios. O domínio das imagens para o ambiente PACS pode ser inicialmente relacionado com as diversas modalidades médicas e suas respectivas técnicas de aquisição. Um exemplo da falta de tratamento de domínios é que a inclusão de uma imagem de CT, em um conjunto de imagens de RM, é permitida pelo sistema sem que nenhum tipo de validação automática ocorra. Esse tipo de problema pode ser estendido para conjuntos mais específicos de imagens, como é o caso da inserção de uma imagem de pélvis em um conjunto contendo apenas imagens de crânio, ou no caso de imagens com câncer em um conjunto de imagens com órgãos sadios.

No caso do domínio das imagens médicas, o uso de domínios vem sendo realizado de maneira simplista como em [Dy et al., 2003], que usa um enfoque hierárquico que é testado com CTs de pulmão e as consultas são processadas em duas fases. Na primeira fase, é feita a identificação da classe em que a imagem se classifica e, na segunda fase, usa-se características configuradas especificamente para a classe encontrada pela primeira fase. Um problema nesse caso é que o processo não é inteiramente automatizado porque a segunda fase é iniciada com a intervenção do usuário. Além disso, o conceito de domínios normalmente é aplicado recursivamente para imagens, o que faz com que várias etapas de processamento tenham de existir para que todos os domínios sejam cobertos no processamento.

A determinação do domínio a partir do contexto de uso é a principal influência sobre os requisitos semânticos que o SiRIC tem de atender. Nesse sentido, o contexto pode ser dependente de vários fatores, desde o local de utilização até o tipo de usuário do sistema [McDonald et al., 2001, Torres et al., 2003]. Assim, requisitos semânticos do domínio existem em cada processamento realizado no SiRIC, o que é determinante para o funcionamento adequado de todos os módulos citados anteriormente. Em ambientes como dos PACS, diversas formas de aquisição co-existem. Consequentemente, para cada forma de aquisição é preciso tratar aspectos específicos do processamento.

O uso de propriedades específicas leva à necessidade de se criar diversos tipos de configurações de processamento, uma para cada domínio. O problema é que o gerenciamento dessas configurações tende a ficar complexo, visto que o número de domínios é grande. Por exemplo, somente a Ressonância Magnética (RM) possui técnicas variadas de

aquisição, como *spin eco*, ecos de gradiente, recuperação de inversão, entre outras, sendo necessário configurações específicas para cada uma delas.

Para aplicações médicas, a divisão de domínios em hierarquias é comum em diversas áreas, como o Código Internacional de Doenças (CID) e o *Breast Imaging Reporting and Data Systems* (BI-RADS). A criação e organização de uma hierarquia de domínios demanda ao usuário possuir conhecimento a respeito dos aspectos semânticos envolvidos nas imagens. Para criação da hierarquia fornecida pelo BI-RADS foram utilizados como aspectos semânticos informações anatômicas e patológicas.

Devido à esse gerenciamento complexo, os sistemas atuais deixam a cargo do usuário o processo de validação das imagens processadas. No entanto, o tratamento adequado dos domínios da imagem deve ocorrer automaticamente e sua configuração ser feita nas diversas etapas de processamento dos SiRICs.

2.4 Conclusão sobre Sistemas de Recuperação de Imagens

Neste capítulo a maioria dos aspectos relacionados com o funcionamento das etapas de processamento de um SiRIC tradicional são apresentados, tendo como principal enfoque as funcionalidades de cada etapa bem como o fluxo dos dados entre elas. O maior problema dessa arquitetura é que o fluxo dos dados obedece à uma estrutura rígida e cada módulo opera como uma "caixa preta". Como conseqüência, todas as ações realizadas por uma etapa são desprezadas nas outras.

Cada uma das etapas do SiRIC tem seus resultados influenciados pelo resultado do módulo anterior. Os algoritmos de processamento de imagem influenciam os algoritmos de extração de característica que influenciam a classificação e/ou funções de distância utilizadas. Em [Huijsmans & Sebe, 2003] é destacada a necessidade de indexar também informações a respeito dos processos de aquisição e pré-processamento da imagem. Nesse sentido, pode-se afirmar que a escolha do algoritmo a ser aplicado em cada uma das fases deveria ser determinado pela similaridade das imagens, ou seja, um algoritmo deveria ser aplicado somente em imagens que possuem características específicas do domínio ao qual o algoritmo tenha sido criado. Portanto, com esse enfoque, o cálculo de similaridade passa a ser aplicado em cada uma das etapas, podendo ser realizado antes de cada processamento.

A complexidade de gerenciamento e configuração desse novo modelo de processamento está intimamente ligada com a heterogeneidade dos processamentos que ocorrem em cada etapa. Assim, somente um modelo no qual todo processamento é tratado de maneira uniforme consegue diminuir a complexidade dessa arquitetura. A arquitetura apresentada neste trabalho baseia-se na alteração do fluxo de dados de um SiRIC visando diminuir a distância semântica entre o processamento automático e a entrada manual dos

dados. A intenção inicial é permitir que o usuário interfira na maneira como os dados automáticos são gerados, de forma a obter os dados que melhor lhe convier. Encontrada a melhor maneira de processamento, o sistema passa a agir automaticamente obedecendo à configuração escolhida.

No próximo capítulo os aspectos relacionados com o tratamento de dados multidimensionais são apresentados tendo como enfoque suas influências sobre os módulos de um SiRIC. Já as alterações arquiteturais que este trabalho fez sobre os SiRICs, bem como sua formalização são apresentadas no Capítulo 5.

Capítulo

3

Dados Multidimensionais

3.1 Considerações Iniciais

O tratamento de dados multidimensionais envolve aspectos que são estudados por diversas áreas (geometria computacional, reconhecimento de padrões, mineração de dados, visualização e estatística) que se inter-relacionam de várias maneiras. Este capítulo trata dos aspectos relacionados com a influência das propriedades dos dados multidimensionais nas buscas por similaridade em imagens. O foco principal é mostrar que extrair características de uma imagem e organizá-las em vetores faz com que a imagem se torne única e exclusivamente um elemento em um espaço de alta dimensão.

Diferentemente do capítulo anterior, no qual aspectos mais genéricos a um SiRIC na área médica são tratados, neste capítulo é dado um enfoque mais detalhado nas etapas de manipulação dos dados multidimensionais, sem que o enfoque central seja somente a área médica em si.

Apesar de alguns trabalhos colocarem o espaço das imagens como um espaço multidimensional, nenhum estudo usa diretamente a matriz de *pixels* na análise e nas buscas por similaridade. Diversas teorias e algoritmos de transformações de espaço são estudados pela área de geometria computacional [Chazelle, 1994]. Contudo para o contexto deste trabalho, esse enfoque teórico e a análise dessas transformações e manipulações dos espaços não são considerados. Essas transformações são realizadas pelos algoritmos aqui apresentados (extração e seleção de características e funções de distância).

No conjunto de problemas envolvidos com o tratamento de dados multidimensionais, tem sido amplamente reconhecido que a propriedade mais importante é a influência

da dimensionalidade dos dados. Em baixa dimensionalidade, a quantidade de informações tende a ser inexpressiva, e o aumento da dimensionalidade tende a aumentar a riqueza das informações. Entretanto, a complexidade no tratamento aumenta da mesma forma, sendo que a influência da alta dimensionalidade sobre diversos aspectos do tratamento dos dados é tamanha que foi classificada como uma "maldição", cujas conseqüências mais relevantes são:

- algoritmos de manipulação em geral são de ordem quadrática em relação ao número de elementos do conjunto para buscas por vizinhos mais próximos [Indyk, 2000];
- a esparsidade dos dados faz com que a distância entre elementos próximos não seja discriminada;
- muitas dimensões podem representar ruídos que afetam a qualidade dos dados;
- muitas dimensões podem representar informações redundantes gerando processamentos desnecessários.

Para facilitar o entendimento das questões envolvidas com os dados multidimensionais, as quatro principais etapas que determinam o seu gerenciamento adequado são explicadas nos próximos tópicos, são elas: a construção, a manipulação, a análise e o armazenamento.

3.2 Construção

A etapa de construção dos dados multidimensionais está relacionada com a aquisição e preparação dos dados. No esquema apresentado no capítulo anterior, esta fase inclui a aquisição e o cálculo de propriedades locais das imagens. Portanto o resultado desta etapa é normalmente um conjunto de números organizados em vetores.

A fase inicial da construção (a aquisição dos dados) é influenciada pelo contexto da aplicação. Dados de informações genômicas são obtidos de forma distinta de informações da área bancária. Um aspecto interessante é que muitas vezes a mesma origem pode gerar vários tipos de dados, como acontece com aplicações médicas que geram imagens diferentes para a mesma parte do corpo (um Raio-X e um CT de um mesmo tórax).

No contexto de aquisição, normalmente circuitos especializados controlam os ruídos e tentam minimizá-los, sendo que muitas vezes diversas transformações são realizadas de forma a gerar dados mais suscetíveis à análise humana. Um exemplo é a aquisição das imagens de ressonância magnética, na qual os dispositivos captam os sinais (componentes M_x e M_y do sinal de ressonância), e os armazenam usando uma função de fase e tempo. Os dados nesse formato, anterior ao formato da imagem, são denominados de formato *raw*, que no caso da ressonância magnética é também conhecido pertencente ao espaço-k.

Para gerar a imagem final, regras de transformação (Fourier 2D) são aplicadas de forma a gerar o *pixels* da imagem a partir desse espaço-k [Suetens, 2002]. É interessante notar que desde a digitalização, os dados sofrem transformações que são dependentes do contexto ao qual eles pertencem. Na maioria das vezes, as transformações não são padronizadas, cada fabricante de dispositivo (como os de ressonância magnética) possui seu circuito com suas transformações internas próprias.

No caso do dado gerado ser proveniente de diversas fontes, ou seja, ser uma composição de outros dados, é necessário verificar de que forma cada um dos componentes deve ser modificado para que o conjunto deles possa ser analisado. Com isso é necessária uma normalização adequada para que alguns dados não se destaquem mais que outros.

Completada a aquisição, a imagem está pronta para ser manipulada. É interessante notar que a imagem em si já está em um espaço multidimensional, que não é adequado para os tipos de processamentos usuais de um SiRIC. Assim transformações devem ser feitas de forma a expressar as características dos elementos desse espaço na forma de dados mais voltados para o contexto de uso. Por essa razão é necessário aplicar algoritmos de análise de cor, forma e/ou textura, pois são essas características que normalmente são extraídas e analisadas nas imagens.

3.3 Manipulação

Com a maldição da alta dimensionalidade, os vetores numéricos obtidos na construção dos dados multidimensionais precisam ser melhorados para que a análise e armazenamento possam ocorrer de forma mais adequada. Isso ocorre devido à propriedade de que muitas vezes um fenômeno que é aparentemente de alta dimensão pode ser governado por uma quantidade menor de variáveis (chamadas de causas escondidas ou variáveis latentes). Assim, conjuntos dos elementos de um vetor de característica podem estar correlacionados entre si (através de combinação linear e outras funções de dependência) e, um novo conjunto com elementos não correlacionados deve ser achado. Assim, o principal objetivo nesta fase de manipulação é limpar os dados de forma a retirar dados considerados redundantes e/ou irrelevantes.

O estudo da dimensionalidade dos dados complexos levou à definição dos conceitos de dimensão intrínseca e dimensão de imersão. Resumidamente, a dimensão de imersão diz respeito à dimensão do espaço no qual os dados estão inseridos, enquanto que a dimensão intrínseca é a dimensão dos dados independentemente do espaço no qual ele está colocado [Chávez & Navarro, 2000, Traina et al., 2000b]. Uma linha tem dimensão intrínseca 1 independente se está sendo representada em um plano bidimensional ou em um espaço n-dimensional.

Pode-se afirmar que a dimensão intrínseca de um fenômeno é considerada como

sendo o número de variáveis independentes que explicam satisfatoriamente esse fenômeno. Em [Korn et al., 2001] são apresentados testes mostrando que a influência principal ocorre pela dimensionalidade intrínseca e não pela dimensionalidade de imersão dos dados. Exemplos incluem a modelagem de seqüências genômicas, como é o caso de proteínas que podem ser classificadas em famílias dependendo da seqüência de aminoácidos [Williams & Zobel, 2002]. Um modelo de família de proteínas classifica as propriedades de cada uma das famílias, assim novas proteínas podem ser classificadas a partir das propriedades comuns às famílias.

A escolha de atributos em dados reais pode ser facilitada com o uso da dimensão intrínseca, que tende a ser bem menor que a dimensão de imersão [Korn et al., 2001]. O uso de dimensões fractais tem sido uma ferramenta útil na análise da dimensão de dados multi-dimensionais [Belussi & Faloutsos, 1995, Kamel & Faloutsos, 1994], em algoritmos de indexação espacial [Böhm & Kriegel, 2000], previsão de seletividade em domínios espaciais [Traina et al., 1999, Faloutsos et al., 2000] e métricos [Traina et al., 2000a]. Em [Traina et al., 2000b] é apresentado um algoritmo muito rápido, e linear no número de elementos do conjunto de dados, aplicando conceitos da teoria dos fractais. Um algoritmo linear em relação ao número de atributos e de elementos é apresentado em [Sousa et al., 2002], no qual a técnica identifica a existência de grupos de atributos em um conjunto de dados, conseguindo determinar em quais grupos cada atributo está contido.

Além dessas operações, a composição de instâncias dos dados também pode ser realizada, ou seja, um novo vetor de características pode ser criado compondo outros vetores. Em [Ngu et al., 2001] é apresentado um modelo que utiliza a composição de vetores representando cor e textura. [Ooi et al., 2003] ilustra o uso de várias características de duas maneiras: indexadas independentemente ou combinadas em um único vetor com aplicação de pesos. Já em [Vadivel et al., 2004] é feita uma avaliação da variação de peso na combinação de histograma com Wavelet Haar e com Wavelet Daubechie.

Independente da manipulação dos dados multidimensionais, a necessidade de transformação para dimensões menores é na maioria das vezes imprescindível e por essa razão técnicas de redução de dimensionalidade foram desenvolvidas.

3.3.1 Redução de Dimensionalidade

Para efetuar a redução de dimensionalidade existem basicamente dois métodos: extração de características e seleção de características. Basicamente, os algoritmos de extração de características criam novas características a partir de transformações ou combinações do conjunto de características original. Já os algoritmos de seleção de características escolhem um subconjunto do conjunto de características original. Diversos métodos de seleção têm sido estudados, incluindo algoritmos genéticos, seleção seqüencial de propriedades, escolha ponderada de atributos [Aha & Bankert, 1995,

Scherf & Brauer, 1997, Vafaie & Jong, 1993] , uso de técnicas de aprendizado de máquina [Blum & Langley, 1997], através de uso de similaridade [Mitra et al., 2002] e teoria de fractais (já citadas na seção anterior).

Freqüentemente a extração de características precede a seleção, de forma que, inicialmente é feita a extração de características a partir dos dados de entrada, a seguir um algoritmo de seleção de características elimina os atributos menos relevantes segundo um determinado critério, reduzindo a dimensionalidade.

A escolha entre seleção e extração de características depende do domínio de aplicação e do conjunto específico de dados disponível. Em geral a seleção de características reduz o custo de medição de dados, e as características selecionadas mantém a interpretação física original, mantendo as propriedades que possuíam quando foram criadas. Já as características transformadas, geradas por extração de características, podem prover uma habilidade de discriminação melhor que o melhor subconjunto das características originais, mas as novas características (combinações lineares ou não lineares das características originais) podem não possuir um significado físico.

É importante salientar que, se a redução de dimensionalidade for excessiva, a discriminação dos elementos pode diminuir muito. Por isso é importante analisar a variação do comportamento da discriminação com a dimensionalidade, de forma que seja possível estimar a dimensionalidade ideal para determinado classificador e conjunto de dados [Braumer & Shacham, 2000].

Além da tarefa de calcular as características mais relevantes, outra tarefa importante recai sobre o cálculo das características redundantes. Poucos trabalhos enfocam a descoberta de características redundantes, como em [Yu et al., 2004] onde é apresentada uma definição formal de características redundantes, bem como um processo de redução de dimensionalidade no qual o cálculo de características redundantes é feito de forma separada do cálculo de características relevantes.

Uma das abordagens para a redução de dimensionalidade consiste em transformar os dados de maneira que o conjunto resultante possa ser representado de outra maneira que facilite a extração das dimensões mais importantes. Uma das técnicas mais utilizadas para isso é chamada *Singular Value Decomposition* (SVD) [Faloutsos, 1996], que aplica uma transformação global dos dados, de forma que o conjunto inteiro é analisado e então é realizada uma rotação de forma que o primeiro eixo possui a máxima variância possível e os outros eixos são construídos de forma a serem ortogonais aos anteriores. Escolhido o eixo principal, ele é removido, e os restantes são recalculados num processo iterativo, para escolher os próximos eixos mais representativos. Como as transformações aplicadas são lineares, novos dados podem ser submetidos à mesma transformação, e tratados no espaço transformado. No entanto, se novos dados puderem modificar a distribuição e a variância geral do conjunto, essa técnica deixa de ser útil. A SVD é bastante utilizada em

dados estáticos, porém dados dinâmicos tornam a aplicação dessa técnica inviável, pois ela manipula o conjunto todo dos dados. Além disso, como ela opera no espaço transformado, ela não é adequada para aplicações que requerem o tratamento dos dados originais. Uma variação do método SVD é apresentado em [Kanth et al., 1999], que aplica o algoritmo SVD, não de forma global, mas em agrupamentos dos dados, aumentando a quantidade de informação nas dimensões reduzidas e, melhorando a aplicação da técnica em dados dinâmicos.

A técnica *Principal Component Analysis* (PCA) [Ng & Sedighian, 1994] é uma técnica semelhante à SVD, e muito utilizada devido à sua simplicidade conceitual e ao fato de existirem algoritmos relativamente eficientes com complexidade polinomial para seu cálculo. A PCA busca uma representação final de dimensão menor em vetores ortogonais, no qual os vetores de entrada possam ser projetados sem perda de generalidade. A PCA avalia a estutura de variação dos dados e determina em qual direção os dados apresentam alta variação. O primeiro componente principal (ou dimensão) apresenta a maior variação dos dados, o segundo componente o valor subsequente e assim por diante. Com a técnica PCA a maioria das informações do espaço original é condensada em um número menor de dimensões nas quais a variância na distribuição dos dados é a mais alta[Yu, 2002]. A técnica PCA possui diversas propriedades interessantes. Primeiramente, a distância entre 2 pontos p e q no espaço resultante é menor ou igual às suas distâncias no espaço original.Outra propriedade relevante é que, como as primeiras dimensões são as mais importantes a distância de p e q no espaço resultante tende a ser bem próxima da distância no espaço original, mesmo que a dimensão do espaço resultante seja bem menor que a do original. Essas duas propriedades garantem que novos pontos, que não alterem a direção da distribuição dos dados, possam ser mapeados para o novo espaço sem a necessidade de novos cálculos.

Contudo a técnica PCA é eficaz somente para dados que são globalmente correlacionados, propriedade que não é normalmente encontrada em dados reais. Assim, o uso da PCA para dados localmente correlacionados gera uma perda significante de informação o que acarreta no aparecimento de falsos positivos no momento da consulta. Para resolver esse problema o uso de métodos para redução local de dimensionalidade foi proposto. Algoritmos de agrupamentos (*clustering*) foram propostos para descoberta de padrões em espaços de baixa dimensionalidade e, para espaços de alta dimensionalidade eles podem ser utilizados para descoberta de agrupamentos relacionados. Posteriormente a redução de dimensionalidade local é aplicada em cada agrupamento. Para cada agrupamento um índice é montado e um índice global é mantido sobre o conjunto de índices gerados.

Um fato a ser destacado, é que considerando os dados originais, a PCA pode não realizar uma redução de dimensionalidade efetiva, pois os componentes principais podem consistir de uma combinação das variáveis originais. A redução de dimensionalidade ocorre

realmente através de técnicas de regressão que fazem um análise completa de colinearidade das variáveis [Brauner & Shacham, 2000]. Um problema da técnica PCA é realizar uma combinação linear relativamente arbitrária o que acarreta processamentos redundantes.

Outro problema da PCA está relacionado com o tratamento para superfícies não-lineares. Em [Lu et al., 2004] o método Eigenmap Laplaciano é apresentado, mostrando sua utilidade para tratar superfícies não lineares embutidas em espaços de maior dimensionalidade. A relação entre a técnica apresentada e a PCA é que elas utilizam matrizes de pesos diferentes. A técnica PCA utiliza o produto interno das matrizes como uma medida linear de similaridade, enquanto que o método *Eigenmap Laplaciano* usa uma medida não linear de similaridade que preserva a localidade. Uma desvantagem dessa técnica é que ela não produz a matriz de transformação que permite que novos pontos sejam mapeados para o espaço de menor dimensionalidade. Uma proposta que utiliza a combinação de métodos de redução de dimensionalidade linear com não-linear é apresentado em [Ngu et al., 2001], no qual o processo de redução é realizado em duas fases, sendo que na primeira a PCA é aplicada e na segunda uma rede neural é utilizada. Um dos aspectos interessantes é a tentativa de utilizar o conhecimento do usuário na fase de treinamento da rede neural. Assim tem-se a combinação de um processo supervisionado com a posterior aplicação de um processo não-supervisionado.

Além da SVD e da PCA, outras técnicas têm sido utilizadas em domínios de dados específicos, como por exemplo *Adaptative Piecewise Constant Approximation* (APCA) [Keogh et al., 2001b] e *Piecewise Aggregate Approximation* (PAA) [Keogh et al., 2001a] que são voltadas para aplicações temporais. Já outras técnicas são utilizadas para informações textuais, por exemplo, em *Conceptual Indexing* (CI) [Karypis & Han, 2000] os conceito presentes na coleção de dados são utilizados para expressar cada documento. Essa técnica utiliza um algoritmo de agrupamento, para determinar grupos de documentos similares e derivar desses grupos os eixos do espaço multi-dimensional. Já a técnica de *Latent Semantic Indexing* (LSI) [Hull, 1994, Berry et al., 1994] é bastante similar à PCA, porém ao invés trabalhar com a matriz de covariância, trabalha diretamente com a matriz de valores originais. Com isso, não há necessidade de se calcular a matriz de covariância, diminuindo o processamento em relação à PCA.

Outras técnicas, ao invés de transformações lineares, aplicam transformações de domínios dos dados, como por exemplo as transformadas em domínio espectral de Fourier, através dos algoritmos de transformação em domínios discretos como *Wavelets* (WT) [Castañón & Traina, 2002]. Um método importante em domínios métricos é a técnica *FastMap* [Faloutsos & Lin, 1995], que mapeia os dados originais para um domínio espacial, procurando preservar as distâncias entre os objetos no espaço original. Existem ainda variações dessa técnica, adaptadas para domínios de dados específicos [Hristescu & Farach-Colton, 2000]. Em [Hjaltason & Samet, 2003b] o autor argumenta

que o *FastMap* e o método *MetricMap* fazem descartes falsos no processo de mapeamento, por isso é proposto o método *SparseMap* que se propõe a resolver esse problema.

A maioria das técnicas de redução de dimensionalidade atuam de maneira global, isto é, em todo o conjunto de dados. Chakrabarti em [Chakrabarti & Mehrotra, 2000] propõe a redução de dimensionalidade baseada na localização de agrupamentos no conjunto de dados, analisando e indexando cada agrupamento individualmente. O agrupamento de subespaços (*sub-space clustering*) é uma extensão das técnicas de seleção de atributos que tem como objetivo encontrar agrupamentos em subespaços diferentes do mesmo conjunto de dados. Em [Parsons et al., 2004] é apresentado uma revisão detalhada de técnicas de agrupamento de subespaços com a apresentação de uma hierarquia que ilustra a relação entre as diversas técnicas. Um problema a ser resolvido com agrupamento de subespaços, é a retirada de dimensões irrelevantes, o que não é possível com técnicas de extração de características tipo PCA. Além disso, técnicas de extração de características não conseguem separar agrupamentos que se sobrepõem. Outro aspecto importante destacado por Parsons, é que agrupamentos não possuem uma definição formal e também não existem medidas que possam ser utilizadas para comparar os agrupamentos resultantes das aplicações das técnicas.

Todas as técnicas de redução de dimensionalidade são aplicadas no contexto de um SiRIC com o intuito de melhorar e facilitar a discriminação dos dados para que os mesmos possam ser comparados entre si. As comparações de dados multidimensionais envolvem processamentos complexos e conceitos que são relacionados com o cálculo de similaridade dos dados.

3.4 Similaridade

Analisar um dado multidimensional envolve posicioná-lo no espaço ao qual ele está imerso. Dessa maneira, algoritmos de agrupamento e funções de distância são utilizados para fazer o posicionamento adequado do dado. Uma outra forma de enfocar a similaridade é considerar que o dado precisa ser classificado, com isso algoritmos de classificação e técnicas probabilísticas (como teoria de Bayes [Vasconcelos, 2004b]) podem ser utilizados.

Para espaços em que é possível definir coordenadas, como espaços vetoriais, o posicionamento dos elementos pode ser feito em relação à origem das coordenadas. Contudo, os conceitos de coordenadas e sua origem são inválidos para contextos considerados adimensionais, como o espaço das palavras e das imagens. Nesse caso é necessário utilizar o conceito de espaço métrico, que é definido na próxima seção.

3.4.1 Espaço Métrico

Formalmente [Burkhard & Keller, 1973, Ciaccia et al., 1997, Chávez et al., 2001, Arantes, 2005], um espaço métrico é definido pela dupla $\{\mathbb{S}, d()\}$, onde $\mathbb{S}$ representa o conjunto de elementos do domínio e $d : \mathbb{S} \times \mathbb{S} \rightarrow \mathbb{R}^+$ é uma função que deve obedecer às seguintes propriedades:

- **simetria**: $d(s_1, s_2) = d(s_2, s_1)$
- **não negatividade**: $0 < d(s_1, s_2) < \infty$ se $s_1 \neq s_2$ e $d(s_1, s_1) = 0$
- **desigualdade triangular**: $d(s_1, s_2) \leq d(s_1, s_3) + d(s_3, s_2)$

A partir dessas propriedades e das características dos objetos a serem indexados, percebe-se duas maneiras para construção de um espaço métrico:

1. a construção de uma função de distância que seja adequada para o domínio de objetos.
2. a partir de uma função de distância, obter objetos que se adequem à mesma.

A primeira forma é a mais adequada para sistemas reais, contudo o problema de identificar uma função de distância adequada é uma tarefa que não possui métodos e, portanto a função é identificada através de heurísticas dependentes da característica dos dados.

A escolha adequada da função de distância tem outro ponto complicador, que é o fato delas sofrerem efeitos da maldição da dimensionalidade [Katayama & Satoh, 2001]. Isso ocorre devido à esparsidade dos dados nos espaços de alta dimensão, o que faz com que a distância medida pela função tenda a um valor semelhante para todos os elementos do conjunto de dados, isto é, a discriminação dos elementos tende a não ocorrer [Brauner & Shacham, 2000]. Outro exemplo do efeito da alta dimensionalidade é citado por [An et al., 2004], que ilustra que a maldição da dimensionalidade faz com que o volume de um hiperesfera dentro de um hipercubo convirja a zero com o aumento da dimensionalidade. Com isso, consultas por vizinhos mais próximos têm resposta nula mesmo quando o diâmetro da área de busca é o mesmo do tamanho do cubo. Portanto, consultas por vizinhos mais próximos tendem a ser totalmente imprecisas em alta dimensão.

Um subconjunto importante dos espaços métricos é o Espaço Multidimensional, que é o espaço no qual os objetos do domínio $\mathbb{S}$ correspondem a vetores de valores numéricos. Os objetos de um espaço vetorial de dimensão n (ou n-dimensional) são representados por n coordenadas de valores reais $(x_1, ..., x_k)$. Uma propriedade interessante dos espaços vetoriais é que as operações geométricas podem ser aplicadas, como as utilizadas pelas aplicações geográficas.

A Figura 3.1 representa as definições das funções L_1 (Manhatan), L_2 (Euclidiana), L_∞ (Chebychev), que são as funções de distância mais utilizadas para espaços vetoriais. O uso dessa família de funções em dados reais mostrou a necessidade de se aplicar um peso diferente sobre cada uma das dimensões, com o intuito de normalizá-los, gerando o conceito de família L_p. A aplicação das funções da família L_p, na maioria das vezes, é feita como parâmetro de comparação para outras funções distância. Um exemplo para dados temporais é apresentada em [Yi & Faloutsos, 2000].

L_1 ou Manhatan ou city-block :	L_2 ou Euclideana :
$D(x,y) = \sum_d^{i=1} w_i\|x_i - y_i\|$	$D(x,y) = \sqrt{\sum_d^{i=1} w_i(x_i - y_i)^2}$
L_∞ ou Chebychev:	família L_p ou Minkowsky:
$D(x,y) = max_{i=1}^d\|x_i - y_i\|$	$D(x,y) = \sqrt[k]{\sum_m^{i=1} w_i\|x_i - y_i\|^k}$
onde, d é a Dimensão do espaço e w é um peso aplicado a cada dimensão.	

Figura 3.1: Algumas funções de distância da família L_p

Outro aspecto interessante, a respeito das funções da família L_p, é a maneira como elas podem ser visualizadas, isto é, as "formas" geométricas das áreas de abrangência de um determinado raio r. Para um espaço bidimensional, a função L_1 forma um quadrado (de lado $r\sqrt{2}$) que a função L_2 abrange com um círculo(de raio r), que é abrangido pela função L_3, até a função L_∞ que é um quadrado(de lado $2*r$) que abrange todas as outras. Se for realizada uma generalização para um espaço d-dimensional, tem-se que a função L_1 gera um hiper-retângulo d-dimensional, enquanto que a L_2 gera uma hiper-esfera d-dimensional.

Um método para construção de funções de distância que são afetadas minimamente pela maldição da dimensionalidade é apresentado em [Aggarwal, 2001]. Um exemplo da maldição da dimensionalidade sobre as funções de distância é apresentado em [Aggarwal et al., 2001] que mostra o fato de que, em altas dimensionalidades, as funções L_p são mais suscetíveis ao aumento da dimensionalidade com o aumento de p, ou seja, a função L_1 (Manhatan) é mais adequada que a função L_2 (Euclidiana) para espaços de alta dimensão. Uma alternativa a esse problema seria o uso de funções de distâncias fracionárias, permitindo que p seja um número menor que 1 [Aggarwal et al., 2001]. Contudo, nesse caso essas funções de distância não são métricas. Em [Jin et al., 2003] é mostrado o uso da função Mahalanobis para cálculo da similaridade dos dados, sendo destacado que com essa função, a forma elíptica obtida é mais adequada para identificação de agrupamentos que o uso da família L_p.

3.4.2 Consultas por Similaridade

Em relação ao posicionamento dos elementos no espaço é intuitivo imaginar que elementos semelhantes deveriam estar em posições próximas. Assim, as consultas por similaridades

em espaços métricos podem ser feitas usando as funções de distância e a desigualdade triangular para diminuir o número de vezes que a função é calculada.

A similaridade pode ser incluída na classe de problemas de proximidade, cuja definição envolve a noção de distância entre pontos do espaço multidimensional. Algumas subclasses são:

- *Minimum Spanning Tree* (MST): encontrar uma árvore que liga todos os elementos do conjunto de forma que a soma dos comprimentos de seus vértices seja o menor;
- agrupamento (*cluster*): encontrar os agrupamentos do conjunto de dados [Käster et al., 2003];
- vizinhos mais próximos (*nearest neighbor*): encontrar os k vizinhos mais próximos de um determinado elemento;
- abrangência (*range*): encontrar todos os elementos que estão a uma distância r de um determinado elemento.

Com um enfoque mais formal e considerando um conjunto de objetos $S = \{s_1, s_2, s_3, ..., s_n\}$ pertencentes a um domínio $\mathbb{S}$ e uma métrica $d()$, os tipos mais comuns de consultas por similaridade são:

1. **Consulta por Abrangência** (*Range Query - Rq*): uma consulta por abrangência $\sigma_{(Rq(s_q,r))}S$, expressa pelo predicado *range* $Rq(s_q, r)$, recupera objetos que se encontram a uma distância máxima r (raio de busca), a partir do objeto de referência s_q (objeto de busca), onde $s_q \in \mathbb{S}$. Formalmente, pretende-se encontrar o subconjunto $A \subseteq S$ que atenda:

$$Rq(s_q, r) = A = \{a | a \in S, d(s_q, a) \leq r\} \tag{3.1}$$

2. **Consulta aos k-Vizinhos Mais Próximos** (*k-Nearest Neighbor Query - kNN*): uma consulta aos k-vizinhos mais próximos, $\sigma_{(kNN(s_q))}S$, expressa pelo predicado *k-nearest* $kNN(s_q)$, recupera os k objetos mais próximos ao objeto de referência s_q, no qual $s_q \in \mathbb{S}$. Formalmente, pretende-se encontrar o subconjunto $A \subseteq S$ que atenda a:

$$kNN(s_q) = A = \{a | a \in S, \forall s_i \in S - A, d(s_q, a) \leq d(s_q, s_i), |A| = k\} \tag{3.2}$$

A busca por vizinhos mais próximos em alta dimensionalidade é questionável em termos do resultado ser significativo e efetivo [Beyer et al., 1999], pois a distância dos pontos em altas dimensões tende a ser praticamente a mesma. Contudo é importante

salientar que a principal causa desse resultado é que a maioria dos trabalhos utilizam funções de distância indiscriminadamente e sem preocupação com a efetividade e o significado do resultado obtido. Para tentar resolver o problema da não-efetividade dos resultados das buscas, a participação do usuário é considerada por alguns trabalhos como primordial [Aggarwal, 2002b].

Devido à importância das funções de distância no contexto de buscas por similaridade, o próximo tópico apresenta as propriedades mais importantes de algumas funções de distância (adicionais à família da família L_p).

3.4.3 Função de Distância

Conforme já definido anteriormente, uma função de distância é um dos elementos que definem um espaço métrico, sendo ela quem atribui a similaridade (dissimilaridade) entre os objetos. Diversos trabalhos, principalmente na área de Recuperação de Informação (*Information Retrieval* - IR)[1], usam funções de distância não métricas, como [Zhang & Srihari, 2004, Indyk, 2000] que citam o uso de funções não métricas para kNN, entre elas as funções Hausdorf e *Earth Move Distance*. Entretanto, neste trabalho somente as funções de distância métricas são consideradas, principalmente por possibilitarem o uso de métodos de acesso métrico (que são detalhados no próximo tópico).

Apesar do vasto uso da família L_p para tratar da similaridade de imagens, a efetividade de seu resultado é questionável[Hinneburg et al., 2000]. Por essa razão, diversas outras funções são estudadas e normalmente comparadas com as da família L_p. Um exemplo de comparação é citado em [Qian et al., 2004], no qual a distância euclideana e a distância do cosseno são comparadas para dados em alta dimensão, sendo comprovado que elas produzem resultados semelhantes para consultas kNN.

O uso de funções distância para o tratamento de palavras tem como principal representante a função L_{edit}, cuja principal característica é calcular a distância de duas palavras baseada no número de modificações (inserções, alterações e exclusões) que devam ser realizados em uma palavra para que ela se transforme em outra. O principal problema da L_{edit} é que a distância mínima é zero e a máxima é o tamanho da maior palavra do domínio dos dados. Nesse sentido, aplicações nas quais o número de palavras é grande (como o conjunto de palavras da língua portuguesa) a discriminação tende a ser pequena. Outras aplicações demandam a busca por subpalavras, como no caso de bases de seqüências genômicas, nesse caso a aplicação da L_{edit} é inviável, o que torna necessário o uso de outros tipos de funções, como a apresentada em [Kahveci & Singh, 2001].

O comportamento dos dados é o principal fator de escolha de uma função de

[1]Em [Silberschatz et al., 1999] a tradução de *Information Retrieval* é feita como "Reaquisição da Informação", contudo para que não ocorra confusão com outros termos, foi preferido colocar a tradução como Recuperação da Informação e o termo em inglês em seguida.

distância. Por isso alguns trabalhos analisam várias funções de distância utilizadas sobre um mesmo tipo de dado. Um exemplo é mostrado em [Xu et al., 2000], que mostra que a função Bhattacharyya tem um melhor desempenho que a função Mahalanobis para indexação de texturas. Já [Kokare et al., 2003] avalia diversas outras funções quando utilizadas com texturas. [Tang et al., 2002] se preocupa com o relacionamento espacial entre as imagens e para isso utiliza uma matriz para fazer a comparação espacial entre duas texturas. No caso de [Ke et al., 2004] são feitas buscas em subimagens que foram retiradas da imagem original, sendo usado como exemplo a descoberta de possíveis cópias de imagens tentando identificar problemas de direitos autorais. Em [Natsev et al., 2004] decompõe-se a imagem em regiões e computa-se a similaridade através da fração da área coberta pelas regiões que se encaixam entre duas imagens. O uso de regiões também ocorre em [Weber & Mlivoncic, 2003], que apresenta uma técnica de busca baseada em regiões que realiza poucas medidas de similaridade, por volta de 0,5% das imagens. Uma proposta interessante é apresentada em [Aggarwal & Yu, 2000], no qual a estrutura *IGrid Index* é apresentada como tendo seu desempenho melhorado conforme o número de dimensões é aumentado, o argumento principal para esse fato é que a função de distância utilizada é adaptada prevendo a esparsidade dos dados.

Nesse contexto, uma análise interessante é apresentada em [Vasconcelos & Lippman, 2000] com a utilização de um relacionamento hierárquico entre as diversas famílias de funções de distância, sendo que a escolha de uma determinada função é determinada pelas propriedades dos dados. Por exemplo, é citado que assumir que os dados têm comportamento Gaussiano no uso da métrica quadrada Mahalanobis é aceitável somente para imagens com distribuição homogênea.

A participação do usuário na definição e validação da função de distância é explorada em [Aggarwal, 2003], em que é proposto um *framework* para a construção de funções de distância. O enfoque principal se dá sobre a parametrização da família L_p e cosseno. Em [Li et al., 2003] é apresentada uma função de distância que é construída a partir da percepção de similaridade do usuário. Ao invés de utilizar uma função de distância fixa, [Malinchik & Bonabeau, 2004] apresenta um sistema que utiliza conceitos de computação evolutiva com análise exploratória de dados que permite ao usuário interativamente explorar os dados de forma que o sistema adapta a função de distância obedecendo à essas interações. Nesse caso é exemplificado o uso da família L_p para variar o peso sobre as dimensões.

Esse posicionamento dos dados gerados com o uso das funções de distância tem como objetivo principal fazer com que a identificação de uma determinada instância tenha seu custo minimizado. Contudo, para que isso ocorra de forma efetiva, é necessário utilizar em conjunto algum método de acesso que organize os dados e facilite seu armazenamento.

3.5 Armazenamento

Conforme já destacado nos tópicos anteriores, dados multidimensionais necessitam de um tratamento adequado na sua construção, envolvendo técnicas especiais como as de redução de dimensionalidade. Para armazenamento de dados multidimensionais são necessários métodos adequados. Assim, nesta seção, os aspectos gerais dos métodos de acesso multidimensionais são apresentados, com enfoque principal nos métodos utilizados por um SiRIC.

Os métodos de acesso são implementados pelos SGBDs, que têm como característica o fato de conseguirem armazenar grande quantidade de dados e, principalmente, fornecer uma resposta rápida para a recuperação dos mesmos. Nesse sentido, as estruturas de indexação têm papel fundamental, pois é seu uso que garante a rapidez nas consultas. Uma estrutura de indexação implementa as regras definidas por um determinado método de acesso, cujo intuito principal é fazer com que os dados sejam alcançados com menor número possível de acessos no momento da consulta. Os métodos de acesso podem ser generalizados da seguinte maneira: os dados são organizados através de uma política de distribuição que utiliza um determinado parâmetro de comparação para posicionar o dado.

A preocupação principal dos métodos tende a ser a velocidade de consulta, enquanto que o desempenho das outras operações, como inserção, são relegadas a segundo plano (no quesito tempo). Esse tipo de tratamento é percebido pela complexidade das operações de inserção encontradas na maioria dos métodos de acesso.

Outro aspecto determinante na escolha de um método de acesso é o dinamismo dos dados, ou seja, como ele se comporta quando é necessário inserir novos elementos. Alguns métodos tendem a ter seu desempenho deteriorado quando ocorre a inclusão de novos elementos. Outros métodos não permitem a inclusão de novos elementos, sendo necessário reconstruir novamente a distribuição dos elementos indexados. Os métodos que suportam dados dinâmicos sem perda considerável de desempenho são os mais difundidos e utilizados, com a família *B-Tree* sendo a representante mais importante [Bayer & McCreight, 1972].

Os métodos de acesso mais utilizados são variações da estrutura conhecida como árvore. Isso se deve principalmente pela facilidade no tratamento de grande quantidade de dados. Uma árvore tem como propriedade principal ser uma estrutura hierárquica composta de nós, na qual existe um único nó principal (chamado nó raiz) que fornece acesso aos demais nós da estrutura. Nessa hierarquia, cada nó (com exceção do raiz, que não possui ligação) possui somente uma ligação com os nós que estão na hierarquia superior. Dependendo do tipo de árvore, cada nó pode possuir uma ou mais ligações para nós dos níveis inferiores. Outro aspecto importante das árvores, relaciona-se com a

política de distribuição dos dados, que utiliza sempre um parâmetro para determinar qual dos nós inferiores deverá ser acessado, processo conhecido como poda. Convenciona-se que dados à esquerda do parâmetro de distribuição são menores que o mesmo, e os à direita são maiores. Uma árvore que pode possuir mais de dois parâmetros por nó é chamada de árvore multivias.

Os tópicos a seguir apresentam com mais detalhes os diversos métodos de acesso baseados em árvore.

3.5.1 Métodos de Acesso Convencionais

Métodos de Acesso Convencionais são utilizados para manipulação de dados cujo domínio obedeça a relação de ordem total, o que é facilmente verificado para parâmetros de comparação unidimensionais dentro dos domínios dos números, das letras e das palavras.

As árvores que utilizam como parâmetro de indexação um elemento unidimensional têm como representante mais simples a árvore binária, uma árvore na qual cada nó possui ligado no máximo dois nós inferiores (subárvores) que formam conjuntos disjuntos. Um subconjunto das árvores binárias que atendem aos requisitos de dinamismo e número de acesso são denominadas AVL. A diferença central das AVL [Langsam et al., 1996] é tentar manter a árvore balanceada, deixando o número de acessos próximo de uma média para todas as consultas, ou seja, com uma distribuição uniforme. Obviamente, operações e ajustes são necessárias para que essas propriedades sejam mantidas conforme novos elementos são inseridos, o que gera uma complexidade maior no algoritmo de inserção.

Com o intuito de manter as características principais das AVL, e diminuir o número de acessos, foi criada a *B-Tree* [Bayer & McCreight, 1972], uma árvore cuja propriedade principal é ser uma árvore multivias balanceada. Nesse sentido, ao invés de utilizar um parâmetro de comparação por nó, a *B-Tree* utiliza diversos parâmetros em cada nó, sendo que cada parâmetro pode direcionar para uma subárvore. Se cada nó da árvore for considerado uma página em disco, percebe-se que uma *B-Tree* otimiza sensivelmente o número de acessos a disco, mantendo uma estrutura relativamente simples com poucos níveis, sendo esse o principal fator de sua difusão.

Algumas variações, denominadas B^+-*Tree* e B^*-*Tree*, foram criadas com o intuito de melhorar seu desempenho. A B^+-*Tree* cria dois tipos de nós (índices e folhas) cuja principal diferença é que os dados são mantidos somente nos nós folhas. Já a B^*-*Tree*, tem como principal diferença manter a taxa de ocupação dos nós em um nível normalmente alto, acima de 66% da ocupação máxima.

O desempenho de uma *B-Tree* é determinado por sua altura, que por sua vez é determinada pela quantidade de elementos, ordem e pelo grau da árvore. Como os tipos de dados são simples o comportamento da árvore é previsível.

No entanto, quando o parâmetro de comparação é multidimensional ou adimen-

sional, a relação de ordem total não existe, inviabilizando a utilização de métodos como *B-Tree*. Para resolver esse problema, foram desenvolvidos métodos espaciais e métricos.

3.5.2 Métodos de Acesso Espaciais

Métodos de Acesso Espaciais (MAE) [Ahn et al., 2001, Lu, 2002], são aqueles que dependem de propriedades dos elementos multidimensionais. Os MAEs têm como premissa o fato dos dados manipulados pertencerem ao domínio dos dados espaciais ou à um espaço de dimensão n (E^n), em particular os espaços R^n.

Os métodos existentes podem ser classificados em Métodos de Acesso Espaciais Pontuais (MAEP) e Métodos de Acesso Espaciais Não-Pontuais (MAENP) [Yu, 2002, Gaede & Günther, 1998]. Os MAEPs consideram que os dados são pontos em um espaço, enquanto que os MAENPs consideram os dados como regiões, ou seja, regiões de cobertura determinam a distribuição dos elementos na árvore. Dessa maneira, cada elemento é um valor complexo cujas características variam dependendo do domínio dos dados [Gaede & Günther, 1998].

O uso de elementos multidimensionais influencia diversos aspectos operacionais dos métodos, pois operadores e sua respectiva álgebra não podem ser facilmente padronizados. Além disso, o uso dessas estruturas em sistemas de banco de dados pressupõe que as mesmas devam suportar a inclusão e exclusão de dados, bem como uma integração adequada entre memória secundária e terciária.

Um aspecto interessante dos MAEs é que propriedades geométricas podem ser utilizadas para realização de consultas, ou seja, ângulo, projeção e adjacência podem ser utilizados como predicados da consulta.

Diversos MAEPs foram desenvolvidos, sendo que os mais representativos são [Filho et al., 1999, Lu, 2002]:

Point QuadTree : tem como principal característica particionar o espaço em quatro quadrantes (representados por Nordeste, Noroeste, Sudeste e Sudoeste), sendo que cada novo elemento realiza uma nova partição e assim sucessivamente.

k-d-Tree : realiza o particionamento do espaço em hiperplanos $(k-1)$-dimensionais, sendo que esses planos são iso-orientados e suas direções alternam entre as coordenadas.

k-d-B-Tree [Robinson, 1981]: basicamente é uma combinação entre a *k-d-Tree* e a *B-Tree*, ou seja, sua intenção principal é tentar resolver o problema de acesso a disco (com os princípios da *B-Tree*) da *k-d-Tree*.

Hybrid-Tree [Chakrabarti, 1999]: uma variação da *k-d-B-Tree* que tem como principal diferença permitir a geração de subárvores não-disjuntas e a árvore pode não ser

balanceada.

Para os MAENPs, o primeiro e mais importante método de acesso desenvolvido foi o método *R-Tree* (*Rectangle Tree*) [Guttman, 1984], que pode ser visto como uma adaptação da *B-Tree* para indexar dados multidimensionais não pontuais. Desta forma, o método *R-Tree* representa uma árvore multivias balanceada pela altura, no qual os objetos são armazenados apenas nas folhas. Além disso, a interseccão das regiões do espaço abrangidas por duas subárvores de um mesmo nó pode não ser vazia. Ou seja, o particionamento do espaço não gera regiões disjuntas [Yu, 2002]. Diversas variações do método *R-Tree* foram desenvolvidos, como:

R*-Tree [Beckmann et al., 1990]: a otimização no desempenho sobre o método *R-Tree* ocorre devido à modificação das operações de inserção e quebra (*split*) dos nós. Além disso, realiza outra inovação através de um mecanismo de reinserção forçada, que faz com que um determinado elemento seja reinserido na estrutura após uma quebra.

A-Tree (*Approximate Tree*) [Sakurai et al., 2000]: tem como principal característica utilizar um tamanho aproximado para a construção do retângulo de cobertura dos dados.

CUR-Tree (*Cost-based unbalanced R-Tree*)[Ross et al., 2001]: combina um modelo de consulta probabilístico com um modelo de custo para a construção da *R-Tree*, assim, na construção da árvore é levado em conta uma dada distribuição para as consultas e um modelo de custo para sua execução.

SS-Tree (*Spherical Tree*) [White & Jain, 1996]: utiliza esfera e não retângulo como formato da região.

SR-Tree (*Spherical/Rectangle Tree*) [Katayama & Satoh, 1997]: utiliza uma integração de esfera e retângulo como formato de região. O intuito principal é melhorar os problemas que a *SS-Tree* apresenta em espaços de alta dimensão, nos quais o uso de regiões retangulares é mais adequado que esferas.

xS-Tree [Wang & Wang, 2001] apresenta uma extensão da *R-Tree* que utiliza a esparsidade dos dados em alta dimensão e faz a abrangência dos nós baseada no produto cruzado dos quadrados nas altas dimensões. Esse produto é chamado de *xSquare*. A idéia de espaço quase-esparso é trabalhada, ou seja, um espaço quase-esparso é aquele no qual diversas dimensões possuem valores muito pequenos (se comparados com a maioria dos outros valores).

TV-Tree [Lin et al., 1994]: realiza uma melhoria sobre a *R-Tree* para alta dimensão através da redução da dimensionalidade e do uso de uma função de movimentação

(telescópica) das dimensões ativas. É considerada como sendo a primeira a tratar a alta dimensionalidade para dados do tipo imagem e séries-temporais [Yu, 2002].

X-Tree [Berchtold et al., 1996]: introduz o conceito de supernó e altera o algoritmo de quebra, com intuito de diminuir a sobreposição existentes nas subárvores.

Os MAEs resolvem o problema de organizar dados multidimensionais, contudo eles se apresentam inadequados para espaços de alta dimensão. No caso do *PointQuad-Tree*, sua definição faz com que cada ponto particione o espaço em 2^n quadrantes, gerando, na dimensão 10, a cada novo objeto inserido 1024 quadrantes.

3.5.3 Métodos de Acesso Métrico

As conseqüências da alta dimensionalidade nos MAEs e, o fato dos mesmos não suportarem o tratamento de dados adimensionais (como palavras e imagens) geraram o desenvolvimento de Métodos de Acesso Métrico (MAM)[Chávez et al., 2001, Yu, 2002, Hjaltason & Samet, 2003a].

Resumidamente, um MAM funciona de forma a selecionar no conjunto de dados o(s) elementos(s) representante(s) e, a partir dele(s) serem calculadas as distâncias para os outros elementos, ou seja, o parâmetro de comparação passa a ser a distância do elemento para o(s) representante(s). Dessa forma, quando um novo elemento é inserido, sua distância para cada um dos representantes é medida e dependendo do valor o elemento é posicionado em um local (subárvore).

Dentro desse contexto, os MAMs realizam suas comparações (ou buscas) através do cálculo da similaridade entre os elementos. Uma busca por similaridade pode ser definida como aquela na qual a distância entre dois elementos quantifica sua semelhança. Portanto, em uma consulta por similaridade, o descarte (ou poda) de elementos é realizado utilizando a desigualdade triangular, que é uma propriedade de uma função de distância métrica. Assim a quantidade de cálculos de distância é sensivelmente reduzida. Além disso, esse tipo de funcionamento faz com que a métrica funcione como uma caixa preta para o método de acesso, o que permite ao método ser independente da métrica [Zirkelbach, 1999].

Os estudos desenvolvidos por Burkhard e Keller [Burkhard & Keller, 1973] podem ser considerados como as primeiras propostas de se utilizar apenas a distância como parâmetro de indexação de dados em espaços métricos. A partir desse trabalho, diversos MAMs foram desenvolvidos, sendo que os mais representativos são:

VP-Tree (*Vantage-Point Tree*) [Uhlmann, 1991]: é uma árvore binária na qual cada nó armazena um objeto e um raio (que é utilizado como critério de decisão para inserção de um novo dado). Algumas melhorias nesse método são apresentadas em [Fu et al., 2000].

GH-Tree (*Generalized Hyper-plane Tree*) [Uhlmann, 1991]: esse MAM, se diferencia do anterior por usar como critério de decisão apenas a distância. Assim cada nó possui apenas dois elementos (s_1 e s_2) e os elementos mais próximos de s_1 são armazenados à esquerda e os mais próximos de s_2 à direita. Um detalhe importante é que uma escolha adequada dos elementos s_1 e s_2 faz com que a árvore tenda a ser balanceada.

FQ-Tree (*Fixed Queries Tree*) [Baeza-Yates et al., 1994]: uma árvore multivias na qual um único elemento é utilizado como referência para os nós de um mesmo nível da árvore, o que torna o número de elementos de referência igual à altura da árvore.

GNAT (*Geometric Near-Neighbor Access*) [Brin, 1995]: um método variante do GH-Tree que escolhe mais que dois representativos por nó.

MVP-Tree (*Multi-Vantage Point Tree*) [Bozkaya & Özsoyoglu, 1997] [Bozkaya & Özsoyoglu, 1999]: um método variante do *VP-Tree* que utiliza mais que um representativo por nó e armazena o elementos juntamente com sua distância para o representativo.

M-Tree [Ciaccia et al., 1997]: o primeiro MAM dinâmico. O método *M-Tree* pode ser comparado com o método *R-Tree* no sentido de ser uma adaptação do método *B-Tree* para espaços métricos, ou seja, *M-Tree* é uma árvore balanceada pela altura.

QIC-M-Tree [Ciaccia & Patella, 2002] uma extensão do método *M-Tree* que utiliza diversas funções de distância.

Slim-Tree [Traina Jr. et al., 2000]: possui funcionamento semelhante ao método *M-Tree*, com a contribuição adicional de permitir que a taxa de sobreposição dos nós seja medida e que a partir dessa medida a árvore seja otimizada, usando um algoritmo chamado *Slim Down*.

DBM-Tree (*Density-Based Metric Tree*) [Vieira et al., 2004]: minimiza a sobreposição da cobertura dos nós fazendo com que a hierarquia seja mais alta nos locais em que a densidade dos dados é maior, ou seja, faz um desbalanceamento controlado da árvore. É a primeira estrutura que quebra o paradigma de que árvores balanceadas são mais indicadas para indexação em disco. Esse método comprova também que a sobreposição e o custo do cálculo da função de distância são os maiores problemas dos MAMs.

Apesar de alguns espaços métricos não possuirem o conceito de sistemas de coordenadas, uma forma de tratar seus dados é escolher elementos do conjunto de dados como referências para um sistema de coordenadas e indexar os dados utilizando as distâncias a partir desses elementos. Em [Yu et al., 2001, Yu et al., 2002] é mostrada a estrutura

iDistance que particiona os dados e para cada partição escolhe uma instância como referência e faz a indexação da distância dos outros componentes da partição em relação a ele, permitindo que os dados sejam indexados em *B-Tree* comuns. Em [Yu et al., 2004] é mostrado que *iDistance* é mais indicado para consultas por vizinhos mais próximos e outra proposta chamada *iMinMax* é mais indicada para consultas por abrangência.

Uma proposta similar é apresentada em [Santos et al., 2001], no qual o conceito *Omni* é definido utilizando um conjunto de representativos (denominados focos) e o valor das distâncias para esses focos é que são indexadas. Com o conceito *Omini* é possível definir uma família de métodos de acesso, sendo interessante notar que o acesso seqüêncial é mostrado como tendo resultados mais significativos. Em [Digout et al., 2004] são apresentadas duas extensões à classe seqüencial da família *Omni*, chamadas de *OSeq+* e *OSeq**. Sendo que no caso da *OSeq+* a ordenação dos focos é diferenciada e no caso da *OSeq** (que é construída a partir da *OSeq+*) mais focos são incluídos na ordenação e no momento da consulta somente alguns deles são escolhidos para direcionar a consulta.

Uma outra forma de indexação de dados métricos sem a utilização de estruturas de árvore é apresentado em [Weber et al., 1998], que mostra o método VA-File(*Vector Approximation File*) que faz uma compressão dos vetores de dados de forma a melhorar o desempenho da consulta no acesso seqüencial. Em [An et al., 2004] o método VA-File é melhorado com um novo método de redução de dimensionalidade. Já em [Cha, 2004] um enfoque alternativo na indexação e consultas em dados multidimensionais abordando o tema com o uso de um novo índice baseado na técnica de índices Bitmap. Esse índice, denominado *GB-index* (*grid bitmap index*), utiliza cada dimensão de forma independente e realiza operações sobre os *bits* (AND e OR) para os cálculos de similaridade.

Caracterização de Métodos de Acesso

A classificação dos Métodos de Acesso apresentada fornece subsídios para a escolha de quais métodos seriam os mais adequados para a manipulação de grande quantidade de imagens . Nesse sentido, algumas considerações já apresentadas a respeito de um dado imagem podem ser sumarizadas:

1. dados imagens não possuem ordem total para efeitos de comparação.

2. para domínios de imagens, como imagens médicas, não existe o conceito de ponto origem para ser utilizado como referência para cálculos de distância, ou seja, é necessário medir a distância (ou similaridade) entre os elementos disponíveis.

3. a construção de um dado imagem pode ser extremamente cara e na maioria das vezes é impossível (re)construir uma imagem a partir de uma definição de suas características.

Aplicando essas considerações sobre o domínio das imagens médicas, tem-se que, pela consideração 1, não é possível utilizar os Métodos de Acesso Convencionais. As considerações 2 e 3 descartam a possibilidade de uso dos Métodos de Acesso Espaciais, que consideram que o domínio seja espacial ou dimensional. Dessa maneira, a manipulação de imagens se torna mais adequada com o uso de um Método de Acesso Métrico. Contudo outras considerações devem ser feitas para esse tipo de uso:

4. para a maioria das aplicações que envolvem SiRICs a inclusão de novas imagens é uma funcionalidade importante, isto é, existe a necessidade do uso de Métodos de acesso que suportam o dinamismo dos dados, como a *M-Tree*, a *Slim-Tree* e a *DBM-Tree*.

5. a manipulação de um dado imagem normalmente exige grande quantidade de processamento, o que pressupõe a manipulação de características extraídas das imagens. Essas características podem gerar elementos de alta dimensionalidade ($d > 1000$) forçando o uso de alguma técnica de redução de dimensionalidade.

6. para um mesmo conjunto de imagens, características diferentes representam comportamentos diferentes, que são diferentes também para cada tipo de função de distância utilizada.

7. no uso dos MAMs, outro parâmetro que é influenciado pelo comportamento dos dados é a configuração adequada do tamanho da página de disco (como em qualquer método de acesso), que é fortemente influenciada pelo tamanho do vetor de características.

Com essas considerações pode-se concluir que o conhecimento do comportamento dos dados é muito importante para o uso adequado de um MAM, cujo principal objetivo é conseguir velocidade na consulta. Contudo, vale lembrar que para os MAMs dois fatores adicionais a serem considerados são os cálculos de funções de distância e a sobreposição dos nós das subárvores. Nesse sentido, o comportamento dos dados é afetado diretamente pelas propriedades inerentes dos dados (dimensionalidade), bem como pela sua manipulação, ou seja, a função de distância e configurações dos MAMs (como tamanho da página em disco). Um exemplo significativo é que o sucesso obtido pela *DBM-Tree*, com seu desbalanceamento controlado da árvore, pode ser explicado pela possibilidade desse desbalanceamento ser direcionado pelo comportamento específico de cada tipo de dado.

Portanto um ambiente que disponibilize o uso de MAMs deve possibilitar que seus diversos parâmetros sejam ajustados de forma a atender aos requisitos dos tipos de dados a serem tratados. Com isso, a escolha de função de distância, conjunto de características,

tamanho da página em disco e uso simultâneo de diversas instâncias de MAMs têm de ser disponibilizados ao usuário.

3.6 Conclusões sobre Dados Multidimensionais

O principal objetivo deste capítulo é mostrar que as várias fases de processamento dos dados multidimensionais possuem algoritmos com propósitos distintos e de certa forma complementares, mas que diversos aspectos são comuns conforme a discussão a seguir.

O primeiro aspecto geral a ser destacado é que praticamente todos os trabalhos citados tratam propriedades específicas dos dados multidimensionais. Apesar desses esforços serem válidos, eles consideram que o dado disponível foi adequadamente gerado pela etapa anterior. Por exemplo, os dados multidimensionais já são manipulados na fase de construção (aquisição) dos mesmos. Contudo, na quase totalidade dos sistemas, a manipulação em si é considerada como iniciando somente a partir dos dados terem sido adquiridos, que no caso dos SiRICs é a partir do momento que a imagem passa a existir como tal.

Conforme destacado nos tópicos anteriores, todos os tratamentos feitos sobre os dados multidimensionais são dependentes do contexto no momento do processamento. Apesar de ser conhecido esse problema ele é mal avaliado, pois a escolha da técnica a ser utilizada na maioria das vezes se dá utilizando como parâmetro somente o domínio dos dados. O que torna a semântica da aplicação irrelevante, ou seja, se o domínio é temporal os trabalhos consideram suficientes aplicar a técnica temporal mais adequada. Porém, o próprio domínio dos dados normalmente possui divisões relacionadas com a semântica de cada uma delas. Um exemplo é o caso do domínio de imagens, às quais podem ser divididas em imagens médicas, paisagens e faces (entre outros domínios). O problema é que técnicas utilizadas para um desses subdomínios não é válida para outros. Esse fato é mais marcante para o caso das imagens médicas, como ocorre por exemplo com as imagens de Raio-X que têm propriedades diferentes das imagens de TC, conforme já destacado no capítulo anterior.

Além disso, os trabalhos que se preocupam com o contexto dos dados e que tentam aplicar adequadamente os tratamentos específicos do contexto, restringem-se ao aspecto analisado. Ou seja, se estiver sendo analisada a extração de características, como em [Zhou et al., 2003], então somente ela é dependente de contexto, todos os outros processamentos são feitos sem essa preocupação. O mesmo ocorre para questões de agrupamento, como em [Thies et al., 2003] no qual o agrupamento dos dados é feito de forma hierárquica para imagens médicas.

Para a fase de armazenamento, um exemplo é [Yu, 2002] que usa a *TV-Tree* com a idéia de contrair e estender dinamicamente os vetores de características, sendo que os nós

dos níveis mais altos utilizam menos características para realizar a poda e, nos níveis mais baixos mais características são adicionadas no processo de poda. Em [Yang et al., 2003] é apresentado um método que utiliza a redução hierárquica da dimensionalidade, chamado de *Visual Hierarchical Dimension Reduction* (VHDR), que além da idéia de formar hierarquia de agrupamentos, utiliza uma interface visual que permite ao usuário determinar quais as dimensões são mais relevantes para o processamento. Em [Aggarwal, 2002b] apresenta-se um sistema de buscas por similaridades que permite a participação do usuário na definição da melhor projeção a ser realizada no momento da consulta.

Nesse sentido, a utilização do contexto semântico [Santini et al., 2001] ou a configuração de processamento adequada ao domínio dos dados se torna inevitável. Com isso, para que ocorra uma construção adequada do vetor de característica através de sua semântica, o suporte e gerenciamento das hierarquias de domínios deve ser suportado pelo SiRIC nas várias etapas de processamento. O uso de conceito de hierarquias sobre subconjuntos dos dados já se mostrou vantajoso também para o contexto de redução de dimensionalidade, como a técnica apresentada em [Aggarwal, 2002a], na qual o processamento se mostrou linear tanto em questão de quantidade de dados quanto em aumento da dimensionalidade.

Adicionalmente à esses aspectos relacionados com o domínio da imagem a ser indexada, outro aspecto dos SiRICs é que normalmente eles utilizam de forma exclusiva um tipo de processamento, isto é, uma vez determinado (ou escolhido) o tipo de domínio da imagem, ela é processada por um conjunto fixo de algoritmos, com o(s) mesmo(s) extrator(es) de característica(s), a(s) mesma(s) técnica(s) de redução de dimensionalidade, a mesma função de distância e o mesmo MAM. Um dos resultados deste trabalho é a formalização da manipulação uma mesma imagem em mais de uma forma de processamento. Assim, a composição dos processamentos gera o conjunto completo dos dados da imagem processada. Os detalhes dessa abordagem, bem como do uso de hierarquias de domínios dentro um contexto semântico, são apresentados no Capítulo 5.

Capítulo

4

Domínio Imagem e SGBDs

4.1 Considerações Iniciais

Nos dois capítulos anteriores são apresentados os aspectos relacionados com a manipulação de imagens no contexto de buscas por conteúdo. Neste capítulo são discutidos conceitos sobre os requisitos necessários para a extensão de um SGBD Relacional com intuito de possibilitar que o tipo imagem seja adicionado como mais um tipo de dado nativo desse sistema. O objetivo principal é discutir qual suporte deve existir para que ocorra a inclusão de um novo tipo de dado e quais as alterações têm de ser feitas nos módulos de um SGBD.

Pelo fato de consultas em imagens demandarem o uso de buscas por conteúdo, a maioria dos trabalhos envolvendo o uso de imagens enfoca a extensão do SGBD tratando somente a inclusão de consultas por similaridade, que é somente uma parte dos processamentos que ocorrem em um SiRIC. Um exemplo é a álgebra definida por [Nes & Kersten, 1998] que faz buscas baseadas em atributos de formas. Outro caso é apresentado em [Ciaccia et al., 2000], com uma álgebra para consultas por similaridade incluindo imprecisão e preferências do usuário. As preferências do usuário também são apresentadas em [Chomicki, 2003] com operadores para definição e manipulação de preferências do usuário com uma extensão da álgebra relacional. Outra extensão que é apresentada em [Shaft & Ramakrishnan, 1996b, Shaft & Ramakrishnan, 1996a] mostra o sistema PIQ que possibilita a definição de sumários das características usadas para descrever um tipo de imagem, contudo não é feita nenhuma referência aos aspectos de indexação das imagens, apenas questões de armazenamento das características são tratadas e sem nenhum tipo de formalização algébrica ou tratamento das etapas de processamento e

manipulação das imagens.

Um exemplo de inclusão de função de distância é mostrado em [Roddick et al., 2003], sendo que o enfoque principal é a utilização da distância semântica, que é definido pelos autores como a noção relativa ou útil de distância entre conceitos. Contudo a construção de distâncias semânticas é baseada no conceito de grafos e seus arcos. Assim para que uma determinada consulta seja realizada a semântica relacionada tem de ter sido previamente contruída.

Antes de analisar questões internas ao SGBD é interessante perceber de que forma a inclusão do tipo imagem pode alterar a arquitetura tradicional e a forma como o SGBD é acessado externamente.

4.2 Arquitetura

A manipulação de dados culminou em uma arquitetura na qual programas executáveis clientes, construídos a partir de alguma linguagem de programação (C/C++, Java, etc), processam os dados e controlam a interface com o usuário. Contudo a função de armazenamento é mantida por um SGBD (normalmente denominado servidor de dados), que centraliza os dados, fornecendo vários recursos com o intuito de evitar perdas e corrupção dos mesmos.

Uma das grandes vantagens da arquitetura cliente/servidor e centralização dos dados é que a administração do ambiente se torna menos complexa porque atividades como cópias de segurança e políticas de acesso podem ser também centralizadas. Outra questão é que o acesso ao dado é padronizado independentemente do sistemas que o requisita.

Na arquitetura cliente/servidor, o *uso de serviços de rede* é fundamental, pois a transmissão dos dados tem de ocorrer entre os componentes da arquitetura. Porém, a difusão das infra-estruturas de rede ocorreu somente nos últimos anos com a difusão dos protocolos TCP/IP (*Transmission Control Protocol / Internet Protocol*). Esse fato fez com que, em ambientes complexos, diversos ambientes cliente/servidor existissem, com cada cliente possuindo seu servidor de dados. Um exemplo clássico desse contexto é o ambiente hospitalar, no qual diversos tipos de sistemas foram desenvolvidos independentemente. Alguns representantes desses sistemas incluem sistemas de Informação em Radiologia (RIS), ou Hospitalares (HIS) e sistemas de arquivamento e comunicação de imagens (PACS). HIS e RIS manipulam informações médicas usando dados textuais e têm por objetivo controlar todos os aspectos operacionais do contexto do serviço hospitalar e radiológico, respectivamente. O principal problema na implementação desses sistemas é que os esforços são duplicados (na verdade multiplicados) porque ocorrem intersecções em diversas de suas funcionalidades. Mesmo com diversas implementações de interfaces para

troca de dados entre esses sistemas, outros problemas continuam a existir. Um exemplo é a duplicação do armazenamento e o uso de conceitos arquiteturais distintos tornando a administração do ambiente, como um todo, uma tarefa extremamente complexa.

Com a difusão dos recursos de rede e transmissão de dados, a tendência nos ambientes complexos foi de integrar os diversos ambientes disponíveis, o que para os dados representou a utilização de apenas um SGBD para atender os clientes. Até mesmo alguns clientes específicos acabaram se tornando módulos dentro de ambientes mais complexo, como no caso dos RIS que foram sendo implementados como módulos dos HIS[Levine et al., 2003].

Além da centralização dos dados, a centralização do processamento foi transferida ao SGBD com o uso de procedimentos armazenados. Dessa forma, o SGBD passou a ser mais do que um armazenador de dados, enquanto que a tendência para implementação dos clientes foi de incluir somente as funcionalidades relacionadas com a apresentação dos dados. No entanto, essa evolução não ocorreu para o tipo imagem, ou seja, o SGBD não incorporou seu processamento apenas seu armazenamento. Esse contexto está representado na Figura 4.1, que ilustra a forma como os sistemas de um ambiente hospitalar acessam a base de dados. Para os dados textuais um SGBD é usado e para dados do tipo imagem o acesso é feito diretamente na base.

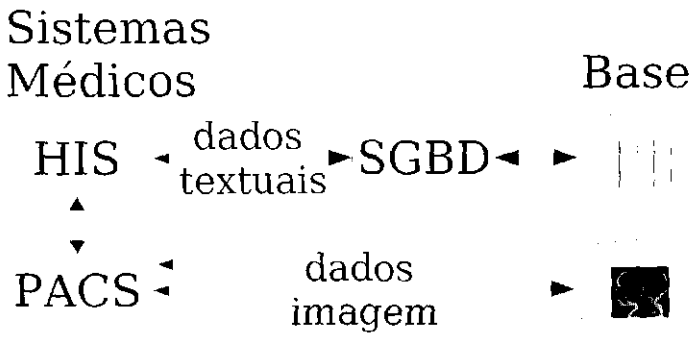


Figura 4.1: Arquitetura Cliente/Servidor sem suporte a imagem pelo SGBD.

Pela Figura 4.1, percebe-se que para sistemas que manipulam dados não nativos o processamento e, muitas vezes o próprio armazenamento desses dados, ficam sob responsabilidade do cliente, como no caso das imagens, que no ambiente hospitalar são manipuladas e armazenadas pelos PACS. Um exemplo de uso dessa arquitetura é apresentada em [Marques et al., 2004], no qual dados textuais são armazenados por SGBDs e as imagens são manipuladas por um Servidor DICOM que as armazena em uma hierarquia de diretórios do sistema de arquivos e, a integração entre o HIS e PACS é simulada ao usuário através de uma interface baseada no ambiente Web.

4.3 SGBDs e Domínio de Dados

A evolução no tratamento dos dados, no contexto dos SGBDs, ocorreu de forma que esses sistemas inicialmente manipulavam apenas dados simples, como números e palavras. Assim, funcionalidades foram implementadas e otimizadas objetivando o armazenamento e a recuperação desses tipos de dados. Entretanto, novos tipos de dados surgiram com o advento de novas aplicações que tratam de dados como as imagens, vídeos, séries temporais, seqüências genômicas, entre outros.

Para definir um tipo de dado é preciso conceituar o domínio dos valores e o conjunto de operadores que manipulam esses valores, ou seja, um tipo de dado é representado pelos operandos, cujos valores devem pertencer ao domínio do dado e, um conjunto de operadores que manipulam esses valores. Os tipos simples são os tipos primitivos que o SGBD manipula diretamente, como no caso dos tipos *inteiro* e *pontoflutuante*. Nesse sentido, definições comuns nos SGBDs Relacionais incluem o domínio dos *inteiros* como sendo os valores possíveis de representar em $64bits$ e seus operadores incluem a soma, multiplicação, subtração e divisão, entre outros. É interessante notar que sub-domínios podem ser definidos para os domínios existentes, para isso novas restrições sobre os valores podem ser definidas. Por exemplo, o domínio *idade* pode ser definido como um sub-domínio do tipo *inteiro* cujos valores podem variar de 0 a 130.

Além da possibilidade de definição de sub-domínios para os tipos simples, diversos SGBDs possuem a funcionalidade de definição de novos tipos a partir dos tipos simples, são os tipos definidos pelo usuário (*User Defined Type* - UDT)[Eisenberg & Melton, 1999, Silberschatz et al., 1999], com novas operações sendo vinculadas na definição das UDT.

O problema é que somente com as UDTs não é possível tratar adequadamente tipos de dados como as imagens. Para que um SGBD forneça à esses novos tipos de dados o mesmo suporte existente aos dados simples, operações específicas têm de ser definidas, sendo que uma operação fundamental é a comparação entre suas instâncias.

4.4 Consultas

Para que um SGBD forneça à esses novos tipos de dados o mesmo suporte existente aos dados simples, operações específicas têm de ser definidas, sendo que uma operação fundamental é a comparação entre suas instâncias. Conforme já colocado nos capítulos anteriores, dados imagem não possuem a relação de ordem total, o que torna sem sentido operações como $<$ or $\geq$, e até a operação $=$ é praticamente inútil. Nos SiRICs, as comparações não são realizadas sobre as imagens em si, mas sobre características retiradas das mesmas e organizadas em vetores. A busca ocorre usando uma imagem como parte do predicado de busca (chamada de centro da consulta) e a similaridade é medida sobre as

características das imagens envolvidas nas comparações. Com isso, o vetor de características da imagem centro da consulta é comparado com os outros vetores da base. Nessas comparações, o valor da similaridade é obtido com o uso de uma função que aplicada sobre as características gera um valor numérico que representa a similaridade entre duas imagens.

Com o intuito de melhorar as consultas, as características são extraídas no momento em que cada imagem está sendo inserida na base e, no momento da consulta, somente o vetor de características da imagem centro da consulta é extraído. No entanto, a comparação de imagens é um processo complexo, existindo diversos aspectos das imagens que podem ser considerados nas comparações, levando a diversos critérios de comparação. Talvez a maior distinção entre comparar tipos complexos (como imagens) e comparar tipos simples (como inteiros) seja que tipos complexos normalmente possuem diferentes aspectos, o que pode levar a diferentes métodos de comparação. Tipos simples possuem poucos aspectos de comparação, com isso a definição de um critério de comparação é simplificado. Por exemplo, mesmo que *idade* e *peso* possuam valores de domínios distintos, eles possuem as mesmas regras de comparação entre si. Não obstante, o tipo imagem necessita que diversos critérios de comparação sejam definidos. É possível comparar duas imagens usando critérios distintos, como cor, forma, área, textura, ou mesmo uma combinação desses e/ou de outros critérios. Por exemplo, o interesse pode existir na comparação de imagens com cores similares ou na distribuição das cores serem semelhantes ou imagens contendo os mesmos objetos, etc. Portanto, conjuntos diferentes de características de cada imagem levam a critérios diferentes de comparação.

Para dados simples, o critério de comparação é informado pelo usuário do SGBD no momento da consulta como parte do predicado, sendo que qualquer valor pertecente ao domínio do atributo utilizado como critério pode ser utilizado. Nesse contexto, para agilizar a velocidade da consulta, atributos muito usados em predicados tendem a ser utilizados para construção de índices, caso contrário a busca na base ocorre de forma seqüencial. Por exemplo, o atributo *Nome* da tabela *Paciente* pode ser indicado para construção de um índice no caso de várias consultas serem realizadas utilizando-o como predicado.

Analogamente, para dados do tipo imagem devem ser criados índices para os critérios de busca mais utilizados. É importante lembrar que esse critério não está relacionado diretamente com o atributo imagem, mas sim com as características retiradas da imagem. Dessa forma, índices adequados para essas características devem ser capazes de manipular dados multidimensionais, como está descrito na Seção 3.5. O mesmo ocorre na escolha de critérios de comparação, ou seja, o critério de comparação é dependente das características que estão sendo manipuladas. Mesmo nos SiRICs, o usuário possui pouca ou nenhuma flexibilidade para determinar o critério de comparação para as imagens, pois normalmente

o conjunto de critérios é fixo e pré-determinado. Um exemplo é o SGBD Oracle, cuja extensão, denominada Intermedia, usa um conjunto fixo de operadores que possibilita a extração e armazenamento de tipos fixos de características das imagens[Ward, 2003].

Além dessas questões de comparação, para que o tipo imagem seja incluído de forma efetiva, as operações sobre instâncias desse tipo também devem ser integradas ao SGBD. Isso significa que as operações que geram as características das imagens também devem ser incluídas nessa integração. Esse enfoque tende a ocorrer principalmente pela definição de operadores considerando somente dados que já estão armazenados na base, desprezando a forma como eles foram gerados, o que simplifica a complexidade do processo de buscas por conteúdo e tende a suportar somente consultas fixas. Um exemplo desse enfoque incompleto é apresentado em [Shaft & Ramakrishnan, 1996a], com o sistema PIQ, que possibilita a definição de sumários das características usadas para descrever um tipo de imagem, não sendo feita nenhuma referência aos aspectos de indexação das imagens. Nesse caso, apenas questões de armazenamento das características são tratadas e nenhum tipo de formalização algébrica bem como tratamento das etapas de processamento e manipulação das imagens são apresentadas. Um enfoque mais formal nesse sentido, é apresentado em [Atnafu et al., 2004] com a definição de uma álgebra que trata as buscas em imagens em SGBDs Relacionais. Essa falta de flexibilidade torna os sistemas impossibilitados de tratarem as especificidades de um domínio imagem em particular. Com isso, os sistemas atuais são genéricos e não são adequados para tratarem imagens de domínios específicos.

Outro problema relacionado com a comparação de imagens é a validação prévia das mesmas, ou seja, imagens não pertencentes ao domínio não deveriam ser processadas, mas com o enfoque dos sistemas atuais, a comparação ocorre e seu resultado gera situações imprevisíveis. A ação de comparar instâncias de domínios diferentes é a mesma que realizar a comparação entre nome e endereço, ou entre distâncias com unidades diferentes($5cm$ e $1km$). Para o caso do tipo imagem seria o mesmo que comparar uma foto de um carro com uma imagem de Raio-X de tórax.

Essa validação deve ocorrer não só no momento da comparação, mas também na etapa de inserção dos dados, de forma que o domínio possa ser restringido, como no caso de um dado *inteiro* que pode ser restringido em um domínio *idade* cujos valores devem ser positivos. Por essa razão que restrições para validação do domínio imagem devem ser integradas também no fluxo de processamento, caso contrário as operações se tornam semanticamente incorretas.

Nas consultas que ocorrem nos SiRICs, normalmente não existe a consideração de que as imagens pertencem a domínios distintos [Kailing et al., 2004], mesmo quando o sistema tenha sido desenvolvido para domínios específicos, como o caso dos que tratam imagens médicas [Müller et al., 2004]. Como nenhum SiRIC possui como funcionalidade a validação de domínio de forma automática e flexível, a validação final se torna um

processo manual, ou seja, de responsabilidade do usuários. Além disso, dentro do contexto de um SGBD, a não validação das imagens é uma barreira a mais para a otimização das operações, já que na preparação das consultas diversas validações poderiam ser realizadas sobre os predicados imagem existentes, o que influenciaria operações como otimização, instanciação de recursos e principalmente o tempo total de processamento.

4.5 Conclusões sobre Imagem e SGBD

As discusões levantadas neste capítulo mostram que para que o tipo imagem seja incorporado em um SGBD é preciso que existam funcionalidades para definição do domínio dos possíveis valores para imagem, bem como do conjunto de operadores que manipulam esses valores. Com isso, o tipo imagem se torna um tipo nativo sob diversos aspectos, sendo que uma das influências mais importantes é que a arquitetura apresentada pela Figura 4.1 se torna obsoleta no sentido de que ela é utilizada em um ambiente por decisão de projeto e não por uma limitação tecnológica. Portanto, a arquitetura adequada para um SGBD, que tenha o tipo imagem como um tipo nativo, é apresentada na Figura 4.2. Nesse contexto todos os dados são armazenados e recuperados somente do SGBD, consequentemente os programas que manipulam imagens podem ser facilmente incorporados como módulos de sistemas mais complexos, como no caso dos PACS, que podem ser implementados como um módulo de um HIS. Dessa forma, todas as funcionalidades dos PACS envolvendo a comunicação e transmissão das imagens continuam sendo implementadas usando alguma linguagem de programação, contudo todos os seus dados são controlados e armazenados pelo SGBD.

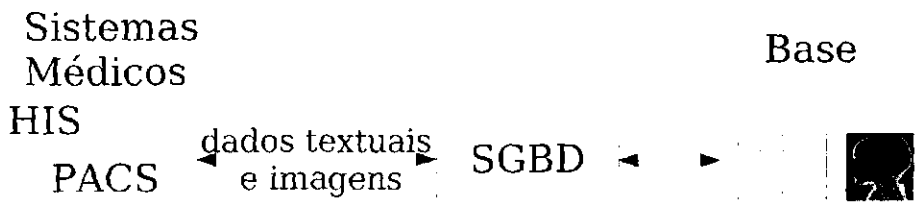


Figura 4.2: Nova Arquitetura Cliente/Servidor com suporte a imagem pelo SGBD.

Sobre as questões internas do SGBD, é interessante existir estruturas de indexação adequadas para o tipo imagem. O que significa, conforme as discussões levantadas na seção 3.5, que é indicado que o SGBD utilize um MAM para indexar o tipo imagem. O problema, é que uma estrutura de indexação precisa ser configurada adequadamente para que seu desempenho seja satisfatório, o que significa que um SGBD tem de possuir funcionalidades que permitam essa configuração e que se possível ela seja feita automaticamente. Contudo, o tratamento automático dos MAMs é complexo porque o comportamento dos dados são previsíveis somente para a dupla ⟨função de distância, domínio ⟩. A alteração de

algum desses elementos torna o comportamento do MAM imprevisível. Nesse sentido, o SGBD deve fornecer flexibilidade suficiente para que a escolha dessa dupla seja realizada facilmente. A possibilidade de inclusão de novos MAMs e de novas funções de distância para atender adequadamente os requisitos dos domínios.

O tratamento efetivo de um domínio definido sobre tipo imagem tem incluir o controle também sobre seus sub-domínios. O problema é que vários sub-domínios podem existir, e se forem criadas restrições totalmente separadas para cada um dos possiveis sub-domínios, então o gerenciamento tende a ser complexo. O mesmo se aplica para os critérios de comparação de um domínio, pois vários critérios tendem a co-existir. Se os vários critérios forem separados de seus sub-domínios, então a tendência é que ocorra um explosão no número de controle desses critérios.

No próximo capítulo é apresentada uma formalização algébrica que contempla os problemas levantados neste e nos dois capítulos anteriores.

Capítulo

5

Álgebra para o Domínio Imagem

5.1 Considerações Iniciais

Neste capítulo é apresentada uma formalização algébrica cuja principal característica é suportar operações de manipulação e de comparação entre instâncias de imagens em SGBD Relacionais usando medidas de similaridade como um modo efetivo para recuperação por similaridade de conteúdo em grande conjuntos de imagens.

A formalização aqui apresentada trata as imagens como um novo tipo de dados que pode ser declarado como o tipo de um atributo que compõe as relações da base. Dessa forma, imagens podem ser tratadas como valores instanciados a partir de domínios, como ocorre com qualquer outro tipo de dado do modelo relacional.

Além das restrições de domínio, é possível a recuperação das imagens utilizando diversos critérios de comparação para um mesmo domínio e, consequentemente, para cada critério é possível utilizar métodos de acesso (índices) específicos e próprios para as características relacionadas com o critério escolhido.

A álgebra aqui definida permite a criação de domínios do tipo imagem específicos ao processamento desejado pelo contexto de uso das imagens. Dessa maneira, para criação de domínios é preciso inicialmente definir o Operando Imagem e os Operadores que o manipulam.

5.2 Operando Imagem

Um **posto de imagens** é um par $\Phi = \langle \mathbb{I}, L \rangle$, no qual $\mathbb{I} = [\ldots I_{-1}, I_0, I_1, I_2, \ldots]$ é um vetor de imagens sem limite de tamanho e, L é um índice de uma das imagens desse vetor, que é chamada de imagem corrente. Cada imagem $I_i | i \leq 0$ é sempre uma imagem sem valor (`null`). Um **posto de imagens** no começo de cada execução de uma expressão sempre tem $L = 1$ e cada imagem $I_i \in \mathbb{I}$ tem atribuído o valor de imagem nula (`null`).

Um **conjunto característica** é representado por $S = \{a_1, a_2, \ldots a_n\}$, onde cada característica a_i é um par $\langle$`nome_característica`, `valor_característica`$\rangle$ descrevendo propriedades de uma imagem.

Um **operando imagem** λ é um par $\lambda = \langle \Phi, S \rangle$, onde Φ é um posto de imagens e S é um conjunto característica.

5.3 Operadores Imagem

A identificação dos componentes básicos envolvidos no processo SiRIC e um modelo formal do processo como um todo são passos importantes no desenvolvimento de sistemas mais flexíveis e mais fáceis de se configurar. O modelo especificado neste capítulo considera que cada processo é composto pelos seguintes componentes básicos:

- processadores de imagem,
- extratores de características,
- construtores de vetores de características,
- funções de distância,
- controles do fluxo de processamento e
- processadores de persistência.

Esses componentes são especificados por três tipos básicos de operadores.

5.3.1 Operadores de Manipulação

Um **processador imagem** representa o componente básico para ambos os processadores de imagens e os extratores de características de um processo CBIR, sendo definido como uma função $\theta(arg \perp s_i) : \lambda \rightarrow \lambda$ que altera qualquer número de imagens $I_i, i > L$ e adiciona qualquer número de novas características em S. As alterações feitas nas imagens em $\mathbb{I}$ e as características adicionadas à S é particular à cada processador θ. Note que arg representa argumentos de entrada que governam a execução do processamento e, s_i indica

qual características devam ser extraídas, O símbolo '$\perp$' é usado para distinguir os dados de entrada dos de saída. Os processadores imagem são definidos pelo usuário usando um módulo de processamento externo escrito em uma linguagem de programação externa.

Existem quatro tipos de processadores imagem pré-definidos, que são: Um processador imagem é chamado de **extrator de característica** quando nenhuma imagem em $\mathbb{I}$ é alterada, mas novas características, extraídas de I_L, são adicionadas em S. Um processador imagem é chamado de **sintetizador imagem** quando somente a imagem I_{L+1} é alterada, sem depender de qualquer imagem I_i existente e, S não é alterado. Um processador imagem é chamado de **operador imagem unário** quando somente a imagem I_{L-1} é alterada, sendo que essas alterações dependem da imagem corrente I_L e, novas características são adicionadas em S. Um processador imagem é chamado de **operador imagem binário** quando somente a imagem I_{L+1} é alterada, com as mudanças dependendo da imagem corrente I_L e da imagem I_{L-1} e, qualquer número de novas características são adicionadas em S.

Processador Imagem:

$$\lambda_1 = \theta(arg \perp s_i)(\lambda_0),$$
$$\lambda_0 = \langle \langle \mathbb{I}_0, L \rangle, S_0 \rangle, \quad \lambda_1 = \langle \langle \mathbb{I}_1, L \rangle, S_1 \rangle$$

`extrator de característica` :
$$s_i = f_s(arg, I_L \in \mathbb{I}_0), \quad S_1 = S_0 \cup \{s_i\}$$

`sintetizador imagem` :
$$I_{L+1} \in \mathbb{I}_1 = f_i(arg)$$

`operador imagem unário` :
$$I_{L+1} \in \mathbb{I}_1 = f_i(arg, I_L \in \mathbb{I}_0),$$
$$s_i = f_s(arg, I_L \in \mathbb{I}_0), \; S_1 = S_0 \cup \{s_i\}$$

`operador imagem binário` :
$$I_{L+1} \in \mathbb{I}_1 \; f_i(arg, I_L \in \mathbb{I}_0, I_{L-1} \in \mathbb{I}_0),$$
$$s_i = f_s(arg, I_L \in \mathbb{I}_0, I_{L-1} \in \mathbb{I}_0), \; S_1 = S_0 \cup \{s_i\}$$

Figura 5.1: Sumário das Classes de Processadores Imagem

A Figura 5.1 apresenta as alterações que cada tipo de processador imagem causa no operando imagem, bem como os dados que eles dependem. Nessa figura, $f_s()$ representa uma função que retorna valores numéricos e, $f_i()$ uma função que retorna uma imagem. Seus argumentos especificam os dados existentes e requeridos para processar cada função.

A execução de um processador imagem $\theta(arg \perp s_i)$, é controlada pelo seu argumento arg. Para garantir a consistência e a repetibilidade em relação à múltiplas comparações das imagens armazenadas na base de dados, esses argumentos devem ser somente valores constantes ou valores de características extraídas previamente durante a execução da expressão de domínio.

Além disso, um processador imagem especial é sempre necessário: o sintetizador `Read()`, que carrega no posto de imagens a primeira imagem a ser processada e a atribui à imagem corrente I_L. Assim, o primeiro processador de uma expressão de domínio normalmente é um sintetizador `Read()`.

5.3.2 Operadores de Controle

Os processadores de controle organizam o fluxo de execução dos outros processadores fazendo a replicação do operando imagem corrente para ser manipulado por um ou mais operadores imagem, conforme descrito a seguir.

O processador de controle Move, representado por $\lambda_0[(n)\lambda_1]$, onde n é um inteiro, replica $\lambda_0 = \langle\langle \mathbb{I}, L\rangle, S\rangle$ into $\lambda_1 = \langle\langle \mathbb{I}, L+n\rangle, S\rangle$. O processador de controle Swap, representado por $\lambda_0[\lambda_1]$, replica $\lambda_0 = \langle\langle \mathbb{I} = \{\ldots I_{L-1}, I_L, I_{L+1}, I_{L+2}, \ldots\}, L\rangle, S\rangle$ em $\lambda_1 = \langle\langle \mathbb{I} = \{\ldots I_{L-1}, I_{L+1}, I_L, I_{L+2}, \ldots\}, L\rangle, S\rangle$.

O processador de controle Fork, denotado por $\lambda[[\lambda]]$ replica λ em qualquer número de operandos iguais ao operando originalλ. Uma instância do operador Fork $\lambda_0[[\lambda_1, \lambda_2, \ldots \lambda_n]]$ cria e executa um ou mais operandos imagem. O resultado λ_p do processador Fork $\lambda_0[[\lambda_1, \lambda_2, \ldots \lambda_n]] \rightarrow S_p$ é o seguinte: o conjunto de características S_p é a união das características extraídas por cada operando imagem $S_p = \bigcup_{i=1}^{n} S_i$, e o posto de imagens $\Phi_p = \langle \mathbb{I}_p, L\rangle$ mantém a mesma imagem corrente $I_{pL} = I_{0L}$ de λ_0 substituindo as imagens subsequentes à imagem corrente de forma que I_{pL+1} é a imagem I_{1L+1} de Φ_1, I_{pL+2} é a imagem I_{2L+1} de Φ_2 e assim sucessivamente, até I_{pL+n} que é a imagem I_{nL+1} de Φ_n.

Uma notação alternativa, que pode ser usada para indicar quantas imagens de cada operando imagem intermediário devam ser transferidas para ao resultado, é preceder cada operando imagem em um fork pelo símbolo (n). Dessa forma, $\lambda_0[[(2)\lambda_1, (0)\lambda_2, \lambda_3]]$ resulta em um posto de imagens $\Phi = \langle \mathbb{I}, L\rangle$ contendo a imagem I_L de λ_0 como a imagem corrente e, as próximas imagens são substituídas de forma que I_{L+1} e I_{L+2} são imagens I_{L+1} e I_{L+2} de Φ_1, nenhuma imagem de Φ_2 é utilizada e I_{L+3} é a imagem I_{L+1} de Φ_3.

O processador de controle ConditionalIf $\lambda?\{\texttt{condicão}\}\{\lambda\}$ replica λ em dois operandos iguais $\langle\lambda_t, \lambda_f\rangle$. O processador de controle ConditionalCase $\lambda?\,[\texttt{enumerator}]\,[\lambda']$ replica λ em qualquer número de operandos imagens iguais. Ambos os processadores de controle condicionais criam dois ou mais operandos imagem, contudo somente um deles é executado, o que depende do valor da condição associada, que é testada sobre as características do conjunto S, e o resultado do controle condicional é o resultado do operador imagem executado.

A condição associada ao processador ConditionalIf é um predicado relacionado com o conjunto de características S que deve resultar em um valor `verdadeiro` ou `falso`. Se o resultado é `verdadeiro`, o primeiro operando imagem λ_t é executado, senão λ_f é

executado. Por exemplo, assumindo que θ_1 é um sintetizador imagem, θ_2 é um extrator de características, θ_3 e θ_4 são ambos operadores imagem unários e $\lambda = \theta_1 : \theta_2?\{c_1\}(\theta_3, \theta_4)$ é uma expressão de domínio utilizando esses operadores. Nesse exemplo, o processador imagem θ_1 produz um operando imagem $\lambda_1 = \langle\langle \mathbb{I}, L\rangle, S\rangle, L = 1$, com $S = \emptyset$ e a imagem $I_1 \in \mathbb{I}$ é a imagem gerada. Assim, o processador imagem θ_2 extrai algumas características dessa imagem e as coloca em S. Após isso, o predicado c_1, do processador de controle, é usado para selecionar se θ_3 ou se θ_4 é executado, e o resultado de toda expressão é o resultado desse operador escolhido.

A condição $[c]$ do processador ConditionalCase é uma expressão relacionada com o conjunto de características S que deve resultar em um valor inteiro, de forma que se $[c] = 1$ então o primeiro operando imagem é executado, se $[c] = 2$ então o segundo é executado, e assim sucessivamente. Se não existe nenhum operando imagem correspondente, então o último é assumido como sendo o padrão (*default*).

Operadores condicionais têm como objetivo permitir que propriedades específicas de um dado domínio imagem sejam utilizadas como parâmetro do procesamento. Um exemplo representativo é o domínio de imagens de CT, no qual existe uma relação entre os valores dos níveis de cinza da imagem e conclusões sobre anatomia e patologia, assim essa relação pode ser vinculada como um parâmetro na condição do processador ConditionalIf.

O operador Projeção $\Pi(S_1)(\lambda_0) \rightarrow \lambda_1$ projeta o conjunto de característica S_0 de λ_0 em um conjunto de característica S_1 de $\lambda_1 | S_1 \subseteq S_0$.

5.3.3 Operadores de Persistência

Existem três processadores de persistência, cujo objetivo é ligar os dados gerados durante o fluxo de processamento ao banco de dados. Esses processadores funcionam da seguinte forma.

O processador de persistência StillImage, representado por Δ_I(`NomeAtributo`), armazena a imagem corrente I_L como um valor do atributo denominado `NomeAtributo`, que é do tipo imagem.

O processador de persistência FeatureVector Δ_S(`NomeAtributo`) armazena o conjunto corrente de características, transformado em um vetor de características, como o valor do atributo denominado `NomeAtributo`, que pode do tipo texto longo ou mesmo binário.

O processador de persistência SimilarityCriterion Δ_X(`NomeCritério`, $df()$) prepara o conjunto de característica como um vetor denominado `NomeCritério`, de forma que ele pode ser usado para comparar ou indexar as imagens armazendas segundo o critério `NomeCritério`, usando a função distância $df()$. Dessa forma, um processador Δ_X atribui o conjunto corrente de características como o parâmetro a ser processado pela função distância, o que define como comparar duas imagens do domínio obedecendo ao

critério correspondente.

A Figura 5.2 mostra os conceitos e notações abordados neste capítulo.

Expressão de Domínio:
$exp(domínio)$
$exp(domínio)\lambda$ ou $exp(dominio)\theta_1 : \theta_2 : \ldots$

Ambiente Imagem:
$\Phi\langle \mathbb{I}, L\rangle$
$\mathbb{I}[\ldots I_{-1}, I_0, I_1, I_2, \ldots]$, imagem corrente: $I_L \in \mathbb{I}$

Conjunto de Características:
$S[a_1, a_2, \ldots a_n]$, $a_i\langle$`nome_característica, valor_característica`$\rangle$

Operando Imagem:
$\lambda\langle \Phi, S\rangle$

Processador Imagem:
$\theta($`args` $\perp$ $\mathtt{s_i}) : \lambda_0 \rightarrow \lambda_1$
`args` são constantes ou características $a_i \in S_0$
$\mathtt{s_i}$ são as características geradas por θ $|S_1 = S_0 \cup \{\mathtt{s_i}\}$
$\theta \in \{$`extrator de característica, sintetizador imagem,`
`operador imagem unário, operador imagem binário`$\}$
sintetizador básico: `Read()`

Processadores de Controle:
Move $\lambda_0 \lfloor (n) \lambda_1 \rfloor$, $\lambda_0 = \langle \langle \mathbb{I}, L\rangle, S\rangle \rightarrow \lambda_1 = \langle \langle \mathbb{I}, L+n\rangle, S\rangle$
Swap $\lambda_0 \dashv \lambda_1 \vdash$,
$\lambda_0 = \langle \langle \mathbb{I} = \{\ldots I_{L-1}, I_L, I_{L+1}, I_{L+2}, \ldots\}, L\rangle, S\rangle \rightarrow$
$\lambda_1 = \langle \langle \mathbb{I} = \{\ldots I_{L-1}, I_{L+1}, I_L, I_{L+2}, \ldots\}, L\rangle, S\rangle$
Fork $\lambda_0 | [\lambda_1, \lambda_2, \ldots] |$
ConditionIf $\lambda_0 ?\, \{$`condition`$\}\, \{\lambda_1, \lambda_2\}$
ConditionCase $\lambda_0 ?\, [$`enumerador`$]\, [\lambda_1, \lambda_2, \ldots]$

Processadores de Persistência:
StillImage $\Delta_I($`NomeAtributo`$)$
FeatureVector $\Delta_S($`NomeAtributo`$)$
SimilarityCriterion $\Delta_X($`NomeCritério`$, df())$

Projeção $\Pi(S_1)(\lambda_0) \rightarrow \lambda_1$, $S_1 \subseteq S_0$

Figura 5.2: Sumário dos Conceitos envolvidos em uma Expressão de Domínio

5.4 Expressão de Domínio

Para expressar as regras que devam ser seguidas quando imagens de um dado domínio são processadas, foi definido o conceito de "expressão de domínio". Uma expressão de

domínio define o pré-processamento, a extração de características, a criação de vetores de características, a função de distância e a forma de persistência dos dados envolvidos no processo de buscas por conteúdo. Uma expressão de domínio *exp(domínio)* considera uma seqüência de processos que devem ser executados sobre imagens de um determinado domínio quando um par delas tem de ser comparadas.

Uma expressão de domínio *exp(domínio)* pode ser simples como um operando imagem λ ou ela pode ser o resultado da execução seqüêncial de diversos processadores imagem, como em $\theta_1 : \theta_2 : \ldots$, no qual cada processador imagem atua sobre o posto de imagens resultante de seu predecessor. O símbolo ':' é usado para expressar a seqÜência dos operandos imagem, de forma que $\theta_1 : \theta_2$ indica que θ_2 é processado depois de θ_1 ser processado, usando o operando imagem resultante do processamento de θ_1.

Com o intuito de facilitar o entendimento dos conceitos e aplicabilidade envolvidos em uma expressão de domínio o exemplo a seguir é detalhado.

Nesse exemplo é assumido que existe um extrator de característica $\theta_1 = Histogram(n \perp h[n])$ e dois operadores imagem unários $\theta_2 = SegmentLung(\perp lung[1])$ e $\theta_3 = TextureShape(n \perp e[n], t[n], v[n])$. Como definido anteriormente, os operadores de extração de características e os operadores imagem analisam a imagem corrente e gera um novo conjunto de características, que são incluídas no conjunto de características S do operando imagem. O operador imagem unário também produz uma nova imagem que é incluída na posição subsequente ao da imagem corrente em I. O extrator de características $Histogram(n \perp h[n])$ analisa a imagem corrente e gera seu histograma de cores $h[n]$ com n *bins*. O operador $SegmentLung(\perp lung[1])$ gera uma nova imagem incluindo somente as partes que contém tecidos de pulmão e inclui uma nova característica que corresponde à área do pulmão. A Figura 5.4(b) mostra um exemplo de um pulmão segmentado gerado por esse operador $SegmentLung(\perp lung[1])$.

O operador $TextureShape(n \perp e[n], t[n], v[n])$ analisa a textura da imagem corrente e cria uma nova imagem particionando a imagem original em regiões que correspondem a n texturas [Balan et al., 2004]. Esse operador também gera n subconjuntos de características com valores estatísticos de energia $e[n]$, entropia $t[n]$ e variância $v[n]$ para as regiões de cada textura. A Figura 5.4(c) mostra um exemplo de um pulmão gerado pelo operador *TextureShape* usando 5 classes, ou seja, $n = 5$.

Esses três operadores podem ser empregados para permitir comparações usando três critérios: histograma de cores, textura e, a proporção do espaço ocupado pelos objetos de cada textura. O processamento para comparar imagens usando esses critérios pode ser expresso pela seguinte expressão de domínio:

Essa expressão é executada da seguinte maneira. Inicialmente o posto de imagens possui um conjunto de imagens nulas e $L = 1$. O sintetizador *Read()* recupera a imagem a ser analisada, atribuindo-a como a imagem corrente I_1. O próximo operador

$Read() : \quad SegmentLung(\perp lung[1]) :$
$[\ ?\{lung[1] > 0\}\{\ \ Histogram(2000 \perp h_l[2000])\ \ \}\ ] :$
$?\{h_l[20-40] > 100\}\{$
$\quad Histogram(256 \perp h_c[256]) : \Pi(h_c[1-256]) : \Delta_X(Cor, L_1()) :$
$\quad TextureShape(5 \perp e[5], t[5], v[5]) :$
$\quad |[\ \Pi(e[1-5], t[1-5]) : \Delta_X(Textura, L_2())\ ,$
$\quad\quad [(2)\ Histogram(5 \perp h_t[5]) : \Pi(h_c[1], h_t[1-5]) : \Delta_X(TexturaObjetos, L_1())\ ]$
$\quad]|$
$\}$

Figura 5.3: Exemplo de expressão de domínio.

$SegmentLung(\perp lung[1])$) gera uma nova imagem I_2 com o pulmão e sua correspondente área. As operações seguintes foram modeladas para serem feitas somente com imagens de pulmão, para isso uma validação é feita pelo primeiro ConditionalIf(?), que testa se o valor lung[1] é maior que 0. Se a área for 0, então a condição é validada como falsa. Do contrário, o operador $Move([\ ])$ é executado para tornar a imagem do pulmão segmentado I_2 como a imagem corrente e, o operador de extração de características $Histogram(2000 \perp h_l[2000])$ gera seu histograma de cores, no qual 2000 bins correspondem ao mapeamento dos valores de HU.

O segundo operador ConditionalIf(?) testa se a imagem possui pixels com valores entre 20 e 40, o que correspondem à alta probabilidade de câncer. Se as características extraídas obedecem à condição, então o processamento da expressão continua com o extrator de características $Histogram(256 \perp h_c[256])$ gerando o histograma normalizado de cores (com 256 bins) e coloca seus valores no vetor h_c.

O próximo operador projeta de $\Pi(h_c[256])$ somente o vetor $h_c[256]$ que é utilizado para indexação pelo operador SimilarityCriterion $\Delta_X(Cor, L_1())$. Como o conjunto de características S nesse momento contém o histograma de cores h_c, ele é usado para comparar imagens relacionadasa com o critério Cor, usando a função distância L_1 (Manhattan).

A imagem corrente continua sendo a imagem original, assim o operador imagem unário $TextureShape(5 \perp e[5], t[5], v[5])$ analisa sua textura para segmentar a imagem considerando as 5 regiões mais distintas. A imagem segmentada é atribuída como I_3, e os a energia, entropia e variância dos pixels das regiões cobertas por cada textura são adicionadas em S, ou seja, os três vetores (de cinco posições) $e[5]$, $t[5]$ e $v[5]$ são adicionados em S.

O próximo operador é o operador fork ($|[\ ]|$) com dois operandos imagem. Os dois caminhos de processamento começam com o conjunto de características com os quatro vetores ($h_c[256], e[5], t[5]$ e $v[5]$) e as três imagens, como ilustrado pela Figura 5.4, ou seja, a imagem original em I_L, a imagem segmentada do pulmão em I_{L+1} e a imagem com as cinco regiões em I_{L+2}. O primeiro caminho de processamento projeta o conjunto de característica com o operador $\Pi(e[1-5], t[1-5])$, que mantém todos os elementos

(de índices 1 à 5) dos vetores c e t e, descarta a variância e o histograma. Após isso, o operador SimilarityCriterion $\Delta_X(Textura, L_2())$ direciona esses dois vetores para serem processados pela função distância L_2 (Euclideana) quando a comparação for realizada tendo a *Textura* como critério.

O segundo caminho de processamento promove a imagem I_3 como a imagem corrente, usando o operador Move ($\lfloor\ \rfloor$) duas vezes, de forma que o extrator de características $Histogram(5 \perp h_t[5])$ opera sobre a imagem segmentada, e o histograma h_t conta o número de pixels cobrindo cada uma das cinco regiões. A próxima projeção $\Pi(h_c[1], h_t[1-5])$ mantém os cinco elementos do histograma h_t e o primeiro elemento de $h_c[1]$ (o histograma de cores) – em imagens médicas, o objeto de análise usualmente está no centro da imagem, assim $h_c[1]$ provavelmente conta a cor da borda da imagem. Esses seis valores são então submetidos à função distância L_1 quando a comparação das imagens é feita utilizando o critério $TexturaObjetos$, como definido pelo operador SimilarityCriterion $\Delta_X(TexturaObjetos, L_1())$.

A Figura 5.4 mostra um exemplo de imagens contidas em um posto de imagens I resultante da execução da expressão de domínio deste exemplo.

(a) (b) (c)

Figura 5.4: Exemplo de um Ambiente Imagem I gerado pela execução de uma expressão de domínio. (a)I_0: imagem original, gerada pelo operador $Read()$. (b)I_2: imagem I_0 segmentada pelo operador $SegmentLung()$. (c)I_1: imagem I_0 segmentada pelo operador $TextureShape()$ usando 5 classes.

5.5 Otimizações na execução das Expressões de Domínio

Algumas das restrições impostas no processamento das expressões de domínio se devem ao objetivo principal delas, que é o processamento de imagens armazenadas por um sistema gerenciador de banco de dados. Nesse contexto, algumas restrições têm como principal objetivo manter a consistência dos dados e permitir otimizações específicas para o processamento de consultas.

Uma operação de busca no banco de dados compara imagens seguindo um dos critérios de comparação associados ao (atributo) domínio no qual a busca deve ser realizada.

Comparar duas imagens requer a extração dos dados associados ao critério pedido. Com isso, a validação pode ser feita executando somente alguns dos processadores da expressão de domínio. A expressão de domínio deve ser validade seguindo o caminho que começa no operador SimilarityCriterion (associado ao critério pedido), segue para trás na expressão de domínio e termina no ponto no qual todos os dados necessários estão disponíveis. A propriedade fundamental a ser garantida é que um dado é obtido somente se a função que o gera tem todos os seus parâmetros de entrada disponíveis.

A partir dos operadores definidos na seção 5.3, pode-se perceber que somente os processadores imagem geram novas características e/ou imagens, como expressado pelas funções $f_i()$ e $f_s()$ na Figura 5.1. Seguindo essas dependências, é possível caminhar de trás para frente na expressão de domínio, obedecendo a um critério de comparação, começando no operador SimilarityCriterion $\Delta_X(\texttt{NomeCritério}, df())$ correspondente para gerar o caminho de execução $Path(NomeCritério)$.

Para ilustrar esse aspecto, considere o exemplo mostrado na seção anterior. A expressão de domínio tem três operadores SimilarityCriterion e, o caminho de execução para cada um pode ser recuperado da seguinte maneira. O operador SimilarityCriterion $\Delta_X(Cor, L_1())$ define os dados necessários para comparar imagens considerando o critério Cor como $\{h_c[256]\}$ - que é todo o conjunto de características já gerados nesse momento. A característica $\{h_c[256]\}$ é gerada pelo operador unário $Histogram(256 \perp h_c[256])$. Como mostrado na Figura 5.1, o conjunto de característica $\{s_i\}$ gerado por esse tipo de operador é $s_i = f_s(arg, I_L \in \mathbb{I}_0)$, assim $h_c[256] = f_s(256, I_L)$. Antes desse operador, a imagem corrente I_L é alterada pelo sintetizador $Read()$, que gera a imagem como uma função de somente um argumento, o que neste operador é o conjunto vazio. Dessa maneira, o caminho de execução necessário para gerar o conjunto de características para comparar imagens considerando o critério Cor é somente $Path(Cor) = Read() : Histogram(256 \perp h_c[256])$.

O operador SimilarityCriterion $\Delta_X(Textura, L_2())$ define os dados necessários para comparar imagens considerando o critério $Textura$ como sendo $\{e[1-5], t[1-5]\}$, por isso, o conteúdo de S depois de executada a projeção e antes de $\Delta_X(Textura, L_2())$. As duas características $\{e[1-5], t[1-5]\}$ são geradas pelo operador imagem unário $TextureShape(5 \perp e[5], t[5], v[5])$, que depende de $(5, I_L)$. Antes desse operador, a imagem corrente I_L é modificada pelo sintetizador $Read()$, que por sua vez não depende de nenhuma outra operação ou parâmetro. Com isso, o caminho de execução para gerar o conjunto de características necessário para comparar imagens considerando o critério $Textura$ é $Path(Textura) = Read() : TextureShape(5 \perp e[5], t[5], v[5])$.

O operador SimilarityCriterion $\Delta_X(TexturaObjetos, L_1())$ necessita das características $h_e[1]$ e $h_t[1-5]$. A característica $h_t[1-5]$ é gerada pelo operador $Histogram(5 \perp h_t[5])$ que depende de $h_h[5] = f_s(5, I_L)$. Essa imagem está disponível depois do ope-

rador Move ter chegado em I_L vindo de I_{L+1}. A imagem I_{L+1} é gerada pelo operador $TextureShape(5 \perp e[5], t[5], v[5])$ que depende somente do sintetizador $Read()$. A característica $h_c[1]$ é gerada pelo operador $Histogram(256 \perp h_c[256])$, que por sua vez depende somente de $Read()$. Com isso, o caminho de execução para o critério $TexturaObjetos$ é $Path(TexturaObjetos) = Read() : Histogram(256 \perp h_c[256]) : TextureShape(5 \perp e[5], t[5], v[5]) : \lfloor \lfloor Histogram(5 \perp h_t[5]) \rfloor \rfloor$.

Dessa forma os três caminhos gerados a partir da expressão de domínio do exemplo da seção anterior são:

- $Path(Cor) = Read() : Histogram(256 \perp h_c[256])$
- $Path(Textura) = Read() : TextureShape(5 \perp e[5], t[5], v[5])$.
- $Path(TexturaObjetos) = Read() : Histogram(256 \perp h_c[256]) : TextureShape(5 \perp e[5], t[5], v[5]) : \lfloor \lfloor Histogram(5 \perp h_t[5]) \rfloor \rfloor$.

Quando uma comparação entre duas imagens desse domínio é requisitada, por exemplo solicitando pelo critério Cor, o caminho de execução correspondente é ativado, e o $Histogram(256 \perp h_c[256])$ de cada imagem é calculado. Com isso, a função distância correspondente é executada, gerando o resultado da comparação. É importante perceber que nem todos os extratores definidos na expressão de domínio precisam ser ativados para cada critério, mas somente para aqueles que realmente influenciam no critério solicitado critério.

Quando imagens são armazenadas em uma base de dados, é importante organizar sua estrutura de forma que as consultas possam ser executadas da maneira mais rápida possível. O objetivo da estrutura de uma base de dados é determinar a melhor configuração para execução de consultas. Algoritmos de processamento de imagens normalmente possuem custos computacionais altos, o que indica a necessidade de pre-processar cada imagem no momento de sua inserção na base de dados. Fazendo com que as características sejam extraídas e armazenadas também no momento da inserção. Contudo, como mostrado no exemplo anterior, não são todas as características extraídas que precisam ser armazenadas, mas somente aquelas necessárias a pelo menos um dos critérios de comparação. Nesse sentido, o processador de persistência FeatureVector Δ_S(`NomeAtributo`) é empregado justamente para indicar essas características que são usadas por algum critério de comparação.

Contudo o uso do processador FeatureVector Δ_S(`NomeAtributo`) pode ser utilizado também para dar maior desempenho a consultas que precisam de características que não foram utilizadas por nenhum critério de comparação. Por exemplo, um usuário poderia estar interessado em saber a variância da textura principal de uma imagem, o que é representada na expressão de domínio como $v[1]$. O operador $\Delta_S(Variância)$ armazena essa característica no atributo mesmo que ela não seja utilizada em algum critério.

O processador de persistencia StillImage Δ_I(`NomeAtributo`) é definido para efeito de completude. Ele permite que imagens calculadas durante o processamento de uma expressão de domínio sejam armazenadas e com isso utilizadas para processamentos diferentes dos definidos nos critérios.

Quando um caminho de execução está extraindo características de uma imagem que está sendo armazenada na base de dados e, um operador $\Delta_S(NomeAtributo)$ ou um $\Delta_I(NomeAtributo)$ é encontrado, as características em S ou a imagem em I_L são armazenadas na mesma relação da imagem, no atributo indicado. Quando um caminho de execução está extraindo as características de uma imagem já armazenada e um operador $\Delta_S(NomeAtributo)$ ou $\Delta_I(NomeAtributo)$ é encontrado, então as características armazenadas ou as imagens processadas são recuperadas e a dependência correspondente é resolvida, impedindo a execução do custoso processador imagem que as gerou.

Portanto, é altamente recomendado (mas não necessário) que cada operador SimilarityCriterion $\Delta_X(NomeCritério.df())$ seja imediatamente precedido por um operador FeatureVector Δ_S(`NomeAtributo`).

Quando um caminho de execução está respondendo à uma consulta mas a imagem não está armazenada na base de dados (como no caso do centro da consulta ser uma nova imagem), então o processador $\Delta_S(NomeAtributo)$ e $\Delta_I(NomeAtributo)$ não são ativados, as dependências não são resolvidas e o processador imagem é executado. Porém, a expressão de domínio ajuda a reduzir o custo do processamento mesmo neste caso, porque ela permite selecionar somente os processadores que efetivamente contribuem para gerar os dados necessários ao critério solicitado.

5.6 Conclusões sobre a Álgebra para o Domínio Imagem

A Álgebra aqui apresentada tem como principal característica formalizar todos os aspectos de um SiRIC com o intuito de incorporar esse processamento em um SGBD relacional. Essa formalização considera que o processamento SiRIC é composto por operações básicas que podem ser agrupadas formando uma expressão de processamento que contempla todas as operações desejadas para um determinado domínio de imagens.

A integração entre o processo SiRIC como um todo e o SGBD é realizada por um grupo de operadores que fazem a persistência dos dados que estão ativos no momento em que a persistência ocorre. Com isso as características e/ou as imagens geradas no processamento podem ser tornadas persistentes na base.

As alterações no fluxo de dados de um SiRIC são ilustradas pelas setas cheias na Figura 5.5, que mostra o fato de que o processamento ocorre sem uma seqüência única de execução, com as diversas etapas podendo ser ativadas automaticamente conforme as

expressões de domínio definidas.

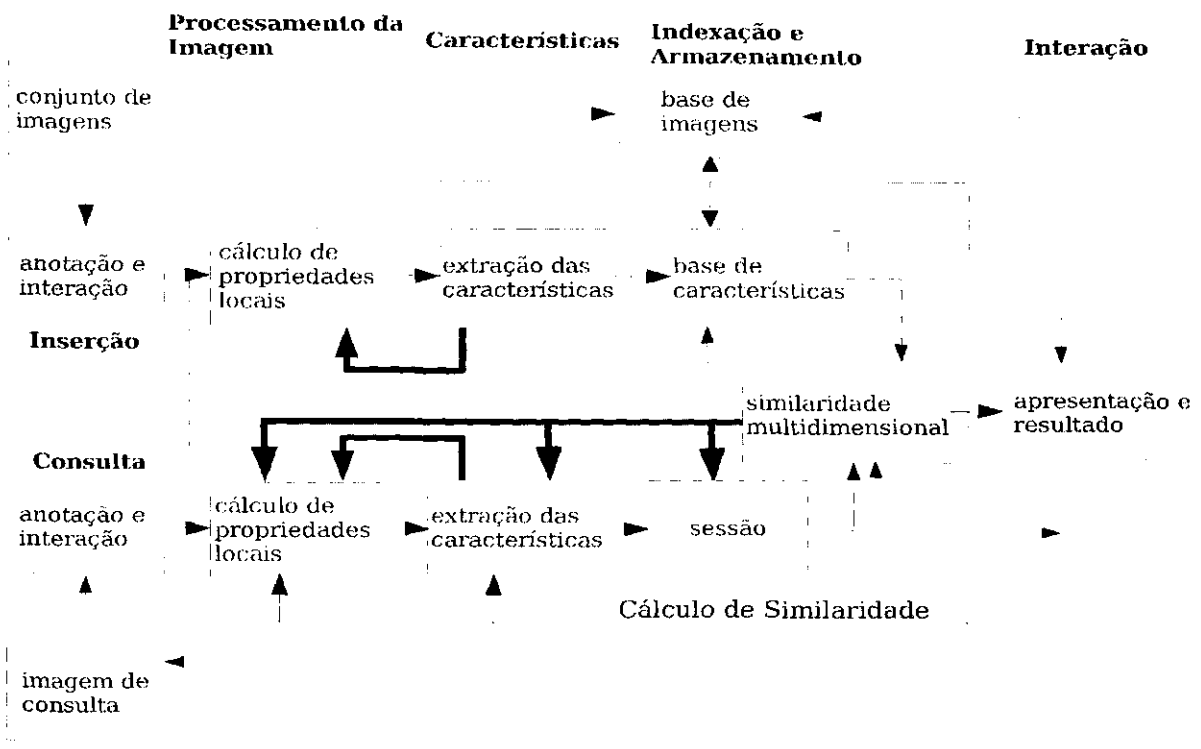


Figura 5.5: Novo fluxo de dados (setas cheias) pelos módulos de um SiRIC.

O cálculo de similaridade permite envolver diversos aspectos da imagem. A proposta aqui apresentada faz o cálculo de similaridade nas várias etapas do SiRIC, ou seja, em cada fase os dados vão sendo processados e filtrados, e com isso o posicionamento dos dados no espaço final vai sendo melhorado no decorrer do processo, de forma que os dados mais semelhantes (ou de mesmo domínio) tendem a seguir o mesmo caminho de processamento na expressão de domínio. A decisão sobre o caminho de processamento a ser ativado é realizada pelos operadores de controle que usam parâmetros condicionais para determinar a próxima ação no processamento. A área destacada na Figura 5.5 ilustra as etapas nas quais as decisões podem ocorrer.

Os operadores de controle determinam automaticamente qual processamento é mais adequado para o domínio, ou seja, um operador de controle verifica se o dado analisado obedece ao comportamento definido para seu domínio. Essa verificação pode ser baseada nas propriedades dos vetores de características ou em qualquer outro parâmetro especificado.

Outra forma de enfocar o(s) caminho(s) de processamento é relacioná-lo(s) com os aspectos semânticos dos domínios, que envolvem propriedades relacionadas aos vetores de características obtidos das imagens. O usuário especialista, que trabalha diretamente com a manipulação das imagens, tem conhecimento sobre o comportamento dos vetores de características em casos específicos como esse. Nesses casos a tendência é que o dado

com comportamento incomum tende a ser posicionado no índice de forma inadequada, ou seja, a análise da patologia é prejudicada pela presença do metal. Esse tipo de situação pode ser contornada com o uso dos operadores definidos na álgebra.

O próximo capítulo analisa com mais detalhes a forma de implementação desta álgebra em um SGBD Relacional, bem como as implicações que cada um de seus operadores gera sobre o processamento SiRIC como um todo.

Capítulo

6

Validação da Formalização do Domínio Imagem

6.1 Considerações Iniciais

No capítulo anterior são apresentadas as principais questões teóricas envolvendo este trabalho. Neste capítulo são abordadas questões relacionadas com a validação desses conceitos, mostrando a sua implementação e os principais resultados gerados pelo seu uso. Inicialmente são abordadas questões relacionadas com os aspectos gerais dos requisitos da formalização e a(s) arquitetura(s) escolhida(s) para sua implementação. Posteriormente são apresentados resultados experimentais sobre o uso individual de cada operador, bem como das expressões de domínio.

6.2 Arquitetura

Com o intuito de facilitar o entendimento dos desafios e os problemas enfrentados na implementação, esta seção é dividida em duas partes: a descrição da implementação envolvendo um SiRIC e da sua incorporação no SGBD. Dessa forma, o próximo tópico trata do ambiente de manipulação das imagens e suas configurações, enquanto que o tópico seguinte trata de como incorporá-lo em um SGBD.

Para que ocorra a implementação dos conceitos envolvidos na formalização, as seguintes questões devem ser respondidas:

1. Como representar domínios no sistema ?
2. Como representar uma expressão de domínio no sistema ?
3. Como gerenciar os domínios e suas respectivas expressões ?
4. Como armazenar as imagens ?
5. Como armazenar os vetores de características ?
6. Como definir os operadores de imagens ?
7. Como definir as funções de distância ?
8. Como expressar e executar as consultas por similaridade ?

As respostas a cada uma dessas questões são apresentadas nas discussões a seguir.

6.2.1 Implementação de um SiRIC

O objetivo principal deste trabalho é incorporar o tipo imagem em um SGBD. Sob a perspectiva de um SiRIC, a maioria de suas funcionalidades não necessitam estar vinculadas a um SGBD. Portanto, a implementação deve possibilitar a execução de cada operação de forma isolada, ou seja, o ambiente SiRIC deve ser um ambiente independente, no sentido de ser um "módulo" que pode ser acoplado em outros ambientes. Uma expressão de domínio indica a seqüência de manipulações que devem ser realizadas sobre uma imagem, sem restringir a maneira como essas manipulações são implementadas.

Para que a funcionalidade de um SiRIC seja embutida em um servidor, não é possível adicioná-las somente no código fonte, porque o ambiente teria de ser (re)compilado para que cada nova funcionalidade pudesse ser usada. Nesse sentido, a definição de uma biblioteca de funções foi descartada.

Duas outras formas de implementação que evitam a (re)compilação do código para inclusão de novos operadores poderiam ser aplicadas, utilizando uma arquitetura de *plugins* ou de componentes. Utilizar uma arquitetura de *plugins* possibilita que novas funcionalidades sejam implementadas obedecendo às definições da arquitetura, com o ambiente incorporando essas novas funcionalidades sem a necessidade de (re)compilação. Contudo, o contexto de um *plugin* normalmente é utilizado para funcionalidades que são ativadas isoladamente, isto é, após acionado, o *plugin* processa a informação e devolve ao servidor o controle sobre ela. Assim, considerando que a execução ocorre sobre uma expressão de domínio, essa (re)chamada ao ambiente torna o processamento mais custoso, pois vários operadores são ativados sucessivamente em uma expressão de domínio. Além disso, os operadores condicionais tornam a seqüência de processamento indefinida até sua execução, pois dependendo da validação realizada, diferentes caminhos de execução podem

ser ativados, sendo que isso é determinado somente em tempo de execução. Por essa razão, optou-se pela implementação do ambiente SiRIC usando uma arquitetura de componentes. Nessa arquitetura, cada funcionalidade é implementada como um componente que se acopla dinamicamente a outro para completar o processamento configurado na expressão de domínio.

Com o intuito de facilitar a configuração do sistema, todos os seus parâmetros são configurados utilizando arquivos XML. A inclusão de novas funcionalidades, bem como a definição de expressões de domínio são feitas nesses arquivos. O que responde à questão 6. O exemplo da Figura 6.1 mostra a expressão de domínio do exemplo da Figura 5.3 representada em XML, fazendo parte do domínio "CT". Indicando a resposta às questões 1, 2 e 3.

```
<domain name="CT">
  <expression name="expressaoA" >
    <imageProcessor name="SegmentLung" type="histogram" arg="lung[1]" />
    <move>
      <conditionalIf cond="lung[1] > 0">
        <then>
          <imageProcessor name="histogram" type="histogram" arg="2000 h[2000]"/>
        </then>
      </conditionalIf>
    </move>
    <conditionalIf cond="h[20-40] > 100">
      <then>
        <imageProcessor name="histogram" type="histogram" arg="256 hc[256]"/>
        <projection type="projection" parameter="hc[i-256]"/>
        <persistence type="similarityCriterion" aspect="Cor" function="L1"/>
        <imageProcessor name="textureShape" type="textureShape" arg="5 d5[1,t[5],v[5]" />
        <fork>
          <operand>
            <projection type="projection" parameter="c[1-5],t[1-5]"/>
            <persistence type="similarityCriterion" aspect="Textura" function="L2"/>
            <move parameter="2">
              <imageProcessor name="histogram" type="histogram" arg="5 ht[5]" />
              <projection type="projection" arg="hc[1],ht[1-5]"/>
              <persistence type="similarityCriterion" aspect="TexturaObjetos" function="L1"/>
            </move>
          </operand>
        </fork>
      </then>
    </conditionalIf>
  </expression>
</domain>
```

Figura 6.1: Exemplo de declaração XML para definição de uma expressão de domínio.

Para a questão de como definir os operadores, criou-se uma interface que todos os operadores devem obedecer. Após sua implementação, o operador é incluído no sistema utilizando um arquivo de configuração, o qual inclui a especificação do código compilado do componente.

Na primeira implementação foi escolhida a inclusão do SiRIC em um servidor Web, de forma que esse ambiente pudesse servir a clientes em plataformas heterogêneas. Com isso, as chamadas são feitas pelos clientes através de requisições HTTP (*Hypertext Transfer Protocol*) enviadas ao servidor, envolvendo também as consultas por similaridade. Essa

arquitetura responde à questão 8.

Para facilitar a organização das imagens manipuladas pelo ambiente, elas são armazenadas em uma hierarquia de diretórios que obedece à organização definida aos domínios e seus respectivos sub-domínios, o que responde à questão 4.

O operador $\Delta_X()$ acessa um servidor de índices métricos, implementado por Lopes[Lopes, 2005], que funciona disponibilizando serviços baseados em duplas $\langle MAM, f \rangle$, assim qualquer conjunto de característica pode ser indexada em um MAM que utiliza uma determinada função de distância(f), respondendo às questões 5 e 7.

6.2.2 Incorporação em um SGBD Relacional

As expressões de domínio foram concebidas para permitir que as buscas por conteúdo em imagens armazenadas em SGBDs Relacionais fossem eficientes. Com isso, as consultas devem ser preferencialmente expressas como uma extensão da linguagem SQL e executada pelo servidor SGBD. No caso do SGBD não possuir suporte às expressões de domínio, uma outra maneira de implementar a expressão de domíno é ter um módulo externo ao SGBD.

A incorporação dos conceitos deste trabalho em um SGBD tem como principal objetivo *tornar transparente* a manipulação das imagens com o intuito de *esconder* do usuário a complexidade das operações sobre as características. Nesse contexto, foi criada uma extensão do dicionário de dados visando armazenar todas as informações relativas ao gerenciamento dos domínios, expressões de domínio, operadores, imagens e suas características.

Nesse contexto, para cada domínio *DomId* definido, uma nova relação é criada na base de imagens, definida pelo seguinte esquema:

$$< DomId >= \{ImgId, Img, Relacão, Atributo, NomeAtr1, NomeAtr2, \ldots\}$$

O atributo *ImgId* identifica a imagem. Quando a aplicação principal define um atributo imagem do tipo "DomId", essa definição é traduzida para um tipo *inteiro* que referencia o atributo *ImgId* da relação *DomId*, o qual armazena de fato a imagem no atributo *Img*. Os atributos *Relação* e *Atributo* armazenam o nome da relação e o nome de seu atributo, respectivamente, que referenciam o atributo *ImgId*. Os atributos *NomeAtr*1, *NomeAtr*2, ... armazenam as características extraídas de *Img*. Com esse enfoque ficam respondidas as questões 1, 3, 4 e 5, levantadas no início deste capítulo.

Quando o protótipo recebe uma consulta em uma declaração XML, o caminho de execução correspondente é recuperado para identificar as características requisitadas do tipo *NomeAtr*, que são recuperadas da relação *DomId* correspondente, chama as funções que extraem as características correspondentes da imagem centro da consulta e, compara

todas as características para responder à consulta requisitada.

Para finalizar a definição do domínio é necessário definir as regras que o regem, com a definição de sua forma de processamento utilizando uma expressão de domínio. Dessa forma as expressões de domínio são armazenadas em uma relação com o seguinte esquema:

$$Expressão = \{\underline{Nome}, Exp.Domínio, ExpressãoSuper\}$$

O atributo *Nome* identifica a expressão. O conteúdo da expressão de domínio é armazenado como uma declaração XML no atributo *Exp*. O atributo *Domínio* indica qual o domínio a que a expressão corresponde. Para o controle de subdomínios é utilizado o atributo *ExpressãoSuper* que referencia qual a expressão de domínio governa o domínio "pai" desse sub-domínio. O que responde às questões 2 e 3. Os detalhes de funcionamento e de gerenciamento de sub-domínios são abordados na Seção 6.3.6.

Os operadores imagem são gerenciados no banco de dados usando procedimentos armazenados que chamam funções externas que implementam os algoritmos de processamento de imagem. O protocolo de passagem de parâmetros para essas funções é definido seguindo a estrutura envolvida no conceito do operador imagem e, os procedimentos armazenados preparam os parâmetros correspondentes obtidos da relação *DomId* correspondente, o que responde à questão 5.

As funções de distância podem ser implementadas de forma semelhante, mas como nossa aplicação sempre usa um conjunto fixo delas, a questão 6 é realmente respondida com a implementação das funções de distância diretamente no código fonte do servidor de índices métricos.

Os Métodos de Acesso normalmente possuem diversos parâmetros que podem ser configurados no momento de sua criação, o que influencia sensivelmente o desempenho das consultas. Por essa razão, uma relação que contempla a ligação entre o critério de comparação e sua instanciação por um MAM, tem o seguinte esquema:

$$Critério = \{\underline{Nome}, Expressão, Caminho, MAM, FunçãoDistância, Configuração\}$$

O atributo *Nome* identifica o critério, enquanto que o atributo *Expressão* referencia a expressão que governa o critério, e o atributo *Caminho* indica o caminho retirado da expressão. Esses caminhos podem ser a declaração XML completa desse caminho ou uma expressão XPath[W3C, 1999] que a represente. A principal razão para usar uma expressão XPath é que não é preciso (re)construir os caminhos de execução quando sua expressão de domínio for alterada, o que é indicado na fase de construção da expressão de caminho. Contudo, isso faz com que o caminho tenha de ser calculado toda vez que ele for utilizado. Já o atributo *MAM* indica a instância e o tipo da MAM utilizado. Já

o atributo *Configuração* armazena os parâmetros utilizados para criação da árvore, que normalmente são: tamanho da página em disco, política de quebra de nós, política de inserção e função de distância utilizada.

A mesma resposta para as questões 1 e 3 a 6 podem ser usadas para implementar o conceito de expressão de domínio em um SGBD. As questões 2 e 7 podem ser respondidas de forma semelhante, mas com a representação de uma expressão de domínio em SQL, como um novo objeto do esquema (com os comandos *CREATE*, *ALTER* e *DROP* correspondentes), e representando as consultas por similaridade com um novo predicado, designado aos atributos imagem.

O exemplo a seguir ilustra o uso desse conjunto de relações para uso efetivo do domínio imagem em um SGBD Relacional. Ele é baseado em um sistema hospitalar que armazena as imagens dos exames de seus pacientes em um banco de dados relacional. Cada tipo de exame gera imagens de domínios distintos. Adicionalmente, é comum que um tipo de exame inclua mais de uma imagem. Por exemplo, um exame de Raio-X possui duas visões frontais dos pulmões. Já um exame de tomografia de cérebro usualmente inclui um vetor com várias imagens axiais da cabeça. Para simplificar o exemplo considere-se que existam apenas esses dois tipos de exames. Então os dados dos pacientes podem ser armazenados em uma relação com o seguinte esquema:

$$Paciente = \{\underline{Nome}, Idade, Peso, TóraxRX1, TóraxRX2, CérebroCT, ...\}$$

Com a definição da tabela *Paciente*, podem ser realizadas consultas por similaridade sobre as imagens armazenadas. Um exemplo é: "recupere os pacientes com os 5 exames $TóraxRX1$ mais similares à imagem de raio-X de tórax contida na variável RX". Se a comparação for feita utilizando o histograma de cor da imagem como critério de comparação, então essa consulta pode ser representada como:

$$\sigma_{(kNN(atributo:TóraxRX1,critério:Cor,k:5,centro:\$RX))} Paciente$$

Essa expressão indica que uma seleção deve ser executada na tabela *Paciente*, procurando pelas $k = 5$ imagens no atributo $TóraxRX1$ que são as mais próximas (kNN) da imagem centro da consulta RX, quando considerado o critério Cor.

Para possibilitar a busca por conteúdo as regras de comparação precisam ser criadas e usadas por imagens do mesmo domínio. Por exemplo, imagens armazenadas nos atributos $TóraxRX1$ e $TóraxRX2$ estão no mesmo domínio, por isso elas podem utilizar as mesmas regras de comparação. Já as imagens contidas no atributo $CérebroTC$ pertencem a outro domínio, com isso elas provavelmente devem utilizar outras regras de comparação.

Para simplificação do exemplo, é considerado que para as imagens de Raio-X não

são definidos processamento diferenciados, ou seja, um domínio não foi criado para expressar sua manipulação. Já, para as imagens de CT é utilizada a expressão de domínio apresentada na Figura 5.3, dessa forma o domínio *Tomografia* e sua(s) expressão(ões) de domínio são criados. Conseqüentemente o conjunto de relações que armazenam as suas informações são criadas e abastecidas na base, conforme ilustrado na Figura 6.3.

O comando SQL para criação da tabela *Paciente* é apresentado na Figura 6.2(a). que é convertido internamente para aquele apresentado na Figura 6.2(b). Analisando a Figura 6.2(b), percebe-se que as imagens são armazenadas em relações como valores de atributos do tipo "StillImage"[13249-5:2001, 2001](atributos *ToraxRX1* e *ToraxRX2*) e, quando um domínio existe, então as imagens são armazenadas na relação respectiva ao seu domínio (atributo Tomografia). A Figura 6.3 ilustra o conjunto total de relações que são utilizadas nesse exemplo.

```
CREATE TABLE Paciente (
   Nome VARCHAR2(50) NOT NULL PRIMARY KEY,
   Idade INTEGER,
   Peso INTEGER,
   ToraxRX1 STILLIMAGE,
   ToraxRx2 STILLIMAGE,
   CerebroCT Tomografia(100) )
```

(a)

```
CREATE TABLE Paciente (
   Nome VARCHAR2(50) NOT NULL PRIMARY KEY,
   Idade INTEGER,
   Peso INTEGER,
   ToraxRX1 STILLIMAGE,
   ToraxRX2 STILLIMAGE,
   CerebroCT INTEGER(100) FOREIGN KEY
        REFERENCES Tomografia
)
```

(b)

Figura 6.2: (a)Comando SQL enviado para o SGBD. (b)Comando SQL convertido com o intuito de suportar o uso do domínio *Tomografia*. (c) Conjunto de tabelas criadas pela definição do domínio *Tomografia* e da relação *Paciente*.

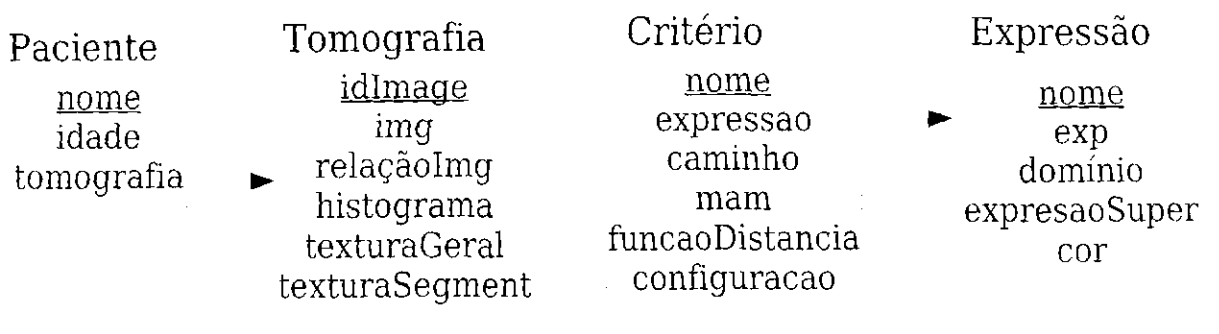


Figura 6.3: Conjunto de tabelas criadas pela definição do domínio *Tomografia* e da relação *Paciente*.

6.2.3 Papéis dos Usuários

Um ambiente que demanda o uso de grandes quantidade de imagens normalmente impoem que diversos profissionais de diversas áreas interagem e trocam informações e conhecimento, como o caso de um ambiente hospitalar. Para cada tipo de imagem é necessário

existir um especialista que analisa todas as suas propriedades. Com a falta de formalização do processamento das imagens, as fronteiras da atuação de cada profissional no ato de manipular as imagens não é bem definida. Consequentemente, o profissional tem de se preocupar com vários aspectos de manipulação, armazenamento e gerenciamento das imagens. Com o uso adequado das expressões de domínio, os usuários do sistema têm explicitados de forma mais adequada os papéis e responsabilidades de cada um.

Os tipos de usuários e suas respectivas funções podem ser categorizados da seguinte forma:

- especialista em imagem - trabalha diretamente com os operadores e com a construção de domínios e suas expressões.
- desenvolvedor de aplicações - trabalha diretamente com os domínios e sub-domínios, sem se preocupar diretamente com o processamento das expressões de cada domínio. Um dado imagem é apenas mais um tipo de dado para ser acessado na base usando as regras de manipulação desse tipo, já definidas pelos especialistas em imagens.
- administrador da base de dados - considera as imagens como mais um tipo de dado sem qualquer diferenciação na manutenção. Monitora o desempenho do ambiente e realiza a manutenção de estatísticas e alocação de discos, memória e processador. Quando necessário, indica a criação de novos índices, que podem ser os índices tradicionais ou os voltados para uso com as imagens.
- programador de algoritmos de processamento de imagens - se preocupa com a instanciação dos algoritmos nos moldes definidos pelas expressões de domínio. Não tem a preocupação de preparação do ambiente para realizar comparações e fazer medidas de resultados.
- usuário final - repassa ao especialista em imagens a necessidade de criação de outros domínios.

As ferramentas para cada tipo de usuário já foram detalhadas no decorrer deste capítulo. A Figura 6.4 mostra a interface criada no protótipo desenvolvido para o usuário final, mostrando o resultado da resposta para uma consulta aos 10 vizinhos mais próximos da imagem de consulta, usando o critério *Cor*.

6.3 Resultados e Discussões

Esta seção apresenta detalhes de uso do ambiente implementado, com enfoque principal sobre a facilidades fornecidas pelo uso das expressões de domínio, bem como algumas vantagens sobre o enfoque tradicional na manipulação das imagens. Inicialmente, o uso

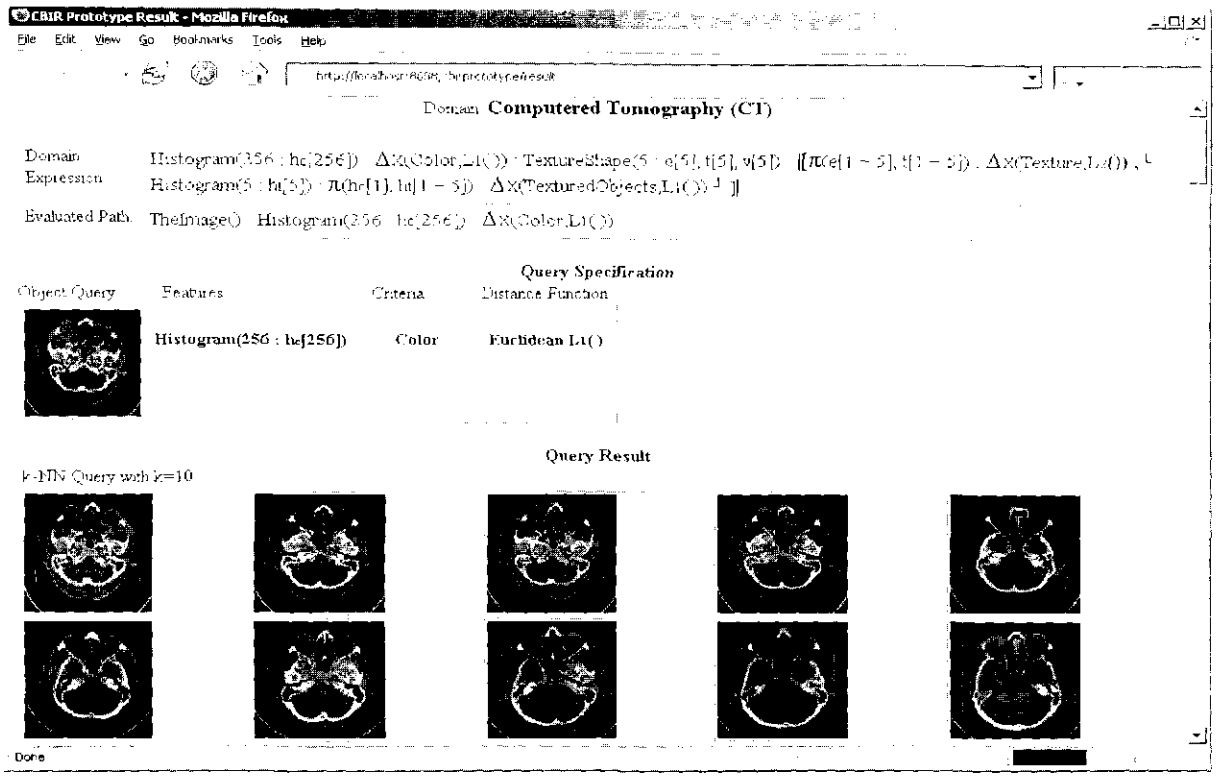


Figura 6.4: Resultado fornecido pelo protótipo para uma consulta $10 - NN$, usando o critério *Cor*.

de cada operador é mostrado e por fim o uso das expressões de domínio incluindo algumas diretivas de construção das mesmas são apresentados.

6.3.1 Estudo de caso

O estudo de caso que norteou os experimentos tem como objetivo mostrar que os recursos de uma expressão de *domínio* para configuração e processamento de propriedades específicas de um domínio imagem são ao mesmo tempo fáceis de utilizar e podem alcançar contextos complexos de processamento.

Os experimentos são ilustrados com o uso de imagens de CT de pulmão. Conforme já detalhado na Seção 2.2.1, imagens de CT são construídas baseadas no mapeamento dos valores de Hounsfield para valores de cinza nos *pixels* da imagem. O objetivo do processamento é aliar os conceitos envolvidos em uma expressão de domínio com o processamento de imagens de CT, tendo como base somente a relação dos valores de Hounsfield para fazer a identificação automatizada da anatomia e patologia de uma imagem.

Nesse contexto, diversas propriedades das imagens de CT foram usadas, entre as principais estão os valores de HU, e seu uso em operações de janelamento e identificação anatômica e patológica. A Figura 6.5 mostra a expressão de domínio utilizada como base dos experimentos.

A Figura 6.6 ilustra o processamento realizado para processar uma imagem segundo

```
(1) θ_pulmãoCT =
(2)   Histogram(2000 ⊥ h_c[2000]) :
(3)   ?{h_c[500 -- 1200] > 20}{
(4)       SegmentHU([W = 811; L = 406] ⊥) :
(5)       ⌊ Histogram(2000 ⊥ h_pulmão[2000]) :
(6)         ?{h_pulmão[500 -- 1200] > 100}{
(7)         Δ_X(Patologia, L_2()) }
(8)     ⌋ }
```

Figura 6.5: Expressão de domínio para processamento de CT de Pulmão.

essa expressão de domínio, usando como referência a imagem indicada como "Imagem Original". Os números indicam os processamentos equivalentes à expressão da Figura 6.5. O processamento inicia-se com a extração do histograma de 16 *bits* da imagem (linha 2 da expressão $\theta_{pulmãoCT}$), que é verificado por um operador condicional, que faz a classificação anatômica da imagem usando como parâmetro as posições do histograma, que indicam a presença de valores correspondentes ao de um pulmão (linha 3 da expressão $\theta_{pulmãoCT}$). A seguir ocorre uma segmentação, também baseada nos valores de HU, o que resulta na adição, no operando imagem, de uma nova imagem contendo apenas o pulmão, que corresponde à uma operação de janelamento com os valores usados como parâmetros (linha 4). Para que essa nova imagem seja processada, o ponteiro da imagem corrente no operando é avançado (com o operador *Move* na linha 5). Finalmente, a possibilidade de encontrar câncer é verificada também pelos valores de HU encontrados na imagem segmentada (com o operador $ConditionalIf$), sendo que as imagens que apresentam essa característica são armazenadas com o critério $Patologia$ e a função de distância L_2 (linha 7). Algumas alterações na expressão de domínio $\theta_{pulmãoCT}$ são apresentadas nas próximas

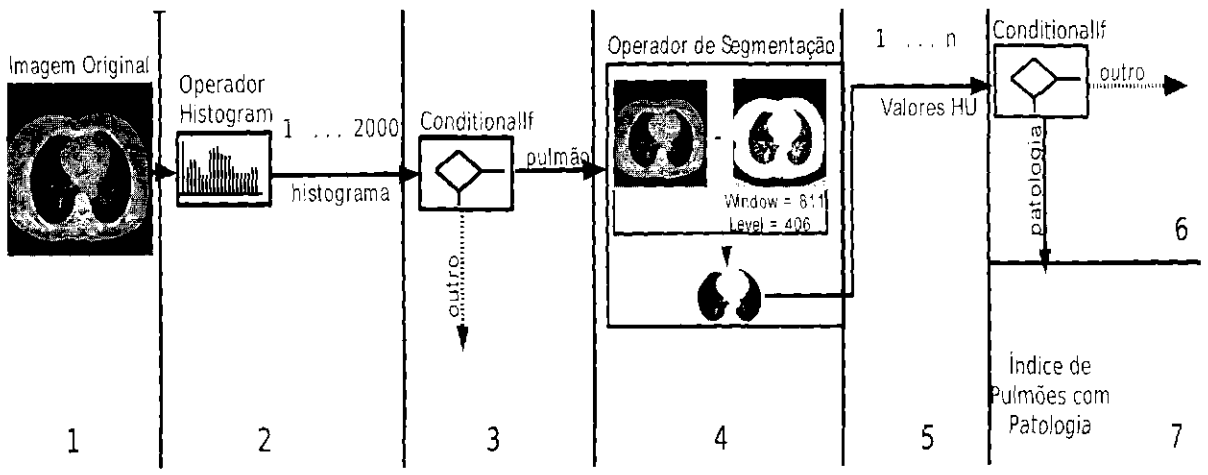


Figura 6.6: Seqüência de processamento usando somente propriedades dos valores de Hounsfield e configurados para execução sobre imagens de CT de pulmão. Os números representam as estapas de processamento equivalentes às linhas da Figura 6.5.

seções, com o intuito de exemplificar o uso dos operadores.

6.3.2 Operadores Imagem

Conforme discutido no capítulo anterior, os operadores de imagem possibilitam que algoritmos de processamento de imagem e de extração de características sejam aplicados sobre o Operando Imagem, podendo adicionar novas imagens e/ou novas características nesse mesmo operando. Com esse enfoque, a maioria dos processos tradicionais envolvendo um ambiente SiRIC, podem ser expressados com os operadores básicos definidos, ou seja, com os operadores *extrator de característica*, *sintetizador imagem*, *operador imagem unário* e *operador imagem binário*

Exemplos de implementação de *extratores de características* incluem qualquer algoritmo que manipule diretamente a imagem corrente do Operando Imagem e adicione novas características obtidas a partir da imagem corrente. Além disso, outras instâncias de operadores desse tipo podem utilizar somente as características presentes no Operando de forma a gerar novas características a partir delas, conforme classificação definida nas Seções 3.2 e 3.3.

Um exemplo de Operador Imagem implementado, que altera a expressão $\theta_{pulmaoCT}$, é o *extrator de característica* representado por $CountHU(valorHU \perp h_u[1])$ que retorna a quantidade de *pixels* cujo valor é o mesmo que o parâmetro $valorHU$. Para o caso da expressão $\theta_{pulmãoCT}$, esse operador é útil para $valorHU$ igual ao do tecido de pulmão. Dessa forma, o operador $Histogram(2000 \perp h_c[2000])$ poderia ser substituído na linha 2, por $CountHU(valorHU \perp h_u[1])$. Isso acarreta a alteração da linha 3 para $?\{h_u[1] > 0\}\{\ \}$, simplificando assim o uso do operador condicional e o próprio tamanho do conjunto de características do Operando Imagem. A expressão $\theta_{pulmãoCT}$ resultante dessas alterações é apresentada na Figura 6.7. Os operadores imagem do tipo *unário* têm como principal

```
(1) θ_pulmãoCT =
(2)   CountHU(HUPulmão ⊥ h_u[1]) :
(3)   ?{h_u[1] > 0}{
(4)      SegmentHU([W == 811; L == 406] ⊥) :
(5)      ⌊ Histogram(2000 ⊥ h_pulmão[2000]) :
(6)        ?{h_pulmão[500 − 1200] > 100}{
(7)        Δ_X(Patologia, I_2()) }
(8)    ⌋ }
```

Figura 6.7: Expressão de domínio resultante da inclusão do operador $CountHU(valorHU \perp hu[1])$ na expressão de domínio $\theta_{pulmãoCT}$ (da Figura 6.5.

característica adicionar uma nova imagem ao Operando, utilizando como parâmetro a imagem corrente, isso é exemplificado pela linha 4 da expressão $\theta_{pulmãoCT}$.

Para exemplificar o uso de um *operador imagem binário*, pode-se considerar que o operador $SegmentHU([w, l] \perp)$ usado na expressão $\theta_{pulmãoCT}$ é uma composição de dois operadores. O funcionamento interno do operando $SegmentHU([w, l] \perp)$ está ilustrado

na Figura 6.6 pelo "Operador de Segmentação". O primeiro é um operador unário que gera uma imagem do janelamento desejado, representado por $Windowing([w, l]\ \perp)$, onde w é o valor do tamanho da janela e l é o valor de sua posição. O segundo operador é o operador imagem binário $Minus(\perp)$ que realiza a operação de subtração entre a imagem I_L e I_{L+1}, com o resultado sendo adicionado ao Operando Imagem. Essa expressão de domínio está ilustrada na Figura 6.8.

```
(1) θ_pulmãoCT =
(2)   CountHU(HUPulmão ⊥ h_u[1]) :
(3)   ?{h_u[1] > 0}{
(4)     Windowing([811, 406], ⊥) :
(5)     Minus(⊥)
(6)     ⌊(2) Histogram(2000 ⊥ h_pulmão[2000]) :
(7)       ?{h_pulmão[500 − 1200] > 100}{
(8)       Δ_X(Patologia, L_2()) }
(9)   ⌋ }
```

Figura 6.8: Expressão de domínio resultante da substituição do operador $SegmentHU()$ pelos operadores $Windowing()$ e $Minus()$ na expressão de domínio $\theta_{pulmãoCT}$ (da Figura 6.7).

6.3.3 Operadores de Controle

Os operadores de controle alteram a forma de funcionamento de um SiRIC, permitindo que mais de uma seqüência de processamento seja definida para uma expressão de domínio. A Figura 6.9 ilustra graficamente um processamento com várias seqüências ocorrendo. Cada operador de controle é detalhado a seguir.

Operador Fork

O operador Fork replica o Operando Imagem para cada seqüência configurada. O importante é que todas as seqüências são ativadas na execução, o que permite definir processamentos paralelos. Um exemplo de processamento paralelo a ser incluído na expressão $\theta_{pulmãoCT}$, é a necessidade de armazenamento de todas as imagens do exame, pois a expressão $\theta_{pulmãoCT}$ contempla apenas o armazenamento das imagens que possuem a patologia indicada. Nesse sentido, uma nova expressão pode ser criada com o uso operador Fork, da seguinte forma. $\theta_{examesCT} = ||[\theta_{pulmãoCT}, \Delta_1(\texttt{ImgCT})]||$

Operadores Condicionais

Os Operadores Condicionais também possibilitam que várias seqüências de processamento sejam configuradas, contudo para cada operador apenas uma das seqüências é ativada no

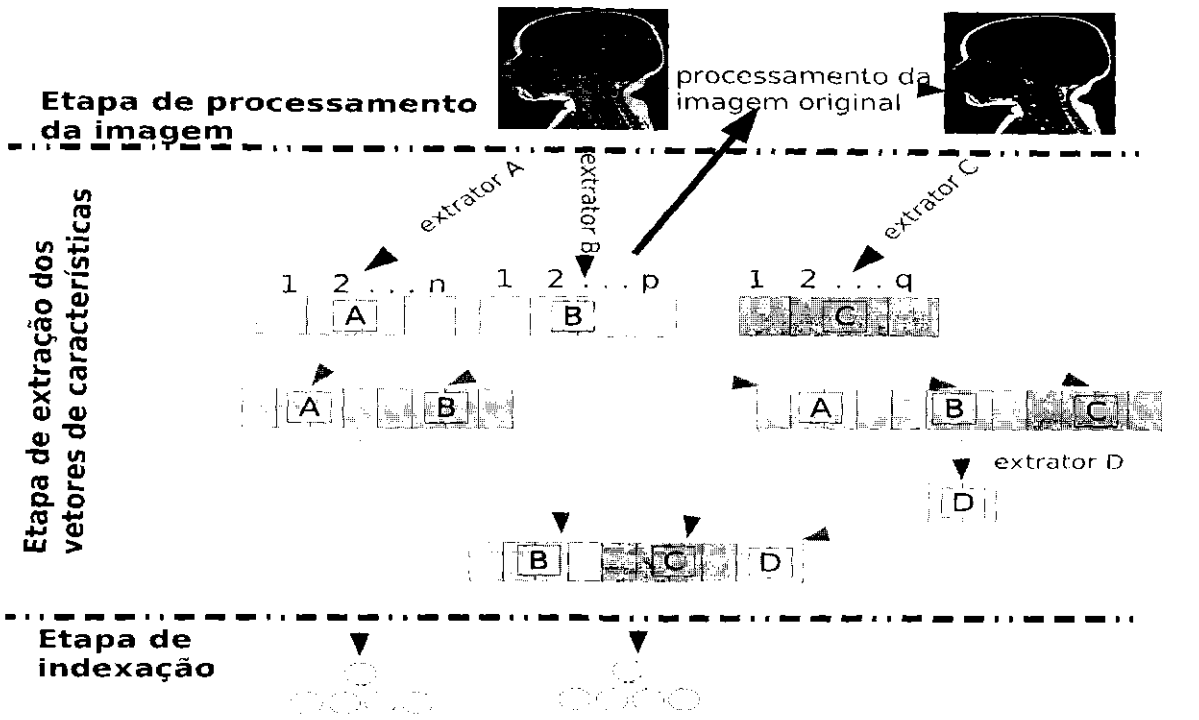


Figura 6.9: Ilustração de várias seqüências de processamento no âmbito de um SiRIC.

momento da execução. É interessante notar que uma expressão de domínio, com vários operadores condicionais, tem um aspecto semelhante à uma árvore de decisão, no sentido de que cada operador condicional atua como um filtro que "classifica" o dado processado em cada etapa. Embora uma árvore de decisão possua regras de construção e manipulação distintas das de uma expressão de domínio, enfocando isoladamente cada operador condicional, pode-se concluir que ele atende as necessidades de um encapsulamento para qualquer algoritmo de classificação de dados.

A forma tradicional dos sistemas de busca por conteúdo para fazer comparações em imagens é que essa ação seja de responsabilidade total da função de distância, ou seja, a similaridade é medida apenas sobre os vetores de características. Adicionalmente, o uso dos operadores condicionais permite que testes sejam realizados com o intuito de direcionar o processamento e fazer com que alguns processamentos sejam realizados somente sobre imagens com propriedades semelhantes. Diferentemente de uma função de distância que compara somente vetores de características, os testes realizados pelos operadores condicionais podem ser feitos sobre outros tipos de propriedades das imagens. Além disso, diversas formas de armazenamento e medidas de similaridade podem ser utilizadas simultaneamente para um mesmo domínio.

O operador *ConditionalIf* está exemplificado na expressão $\theta_{pulmãoCT}$. Para o operador *ConditionalCase*, um exemplo é a criação da expressão $\theta_{exameCT}$, que testa a anatomia do exame, com a seguinte configuração ilustrada na Figura 6.10. A expres-

são $\theta_{exameCT}$, possui um operador *ConditionalCase* que utiliza a função *AnatomiaCT*(), que verifica a anatomia do exame e gera como resultado um valor que indica qual das expressões referenciadas serão utilizadas.

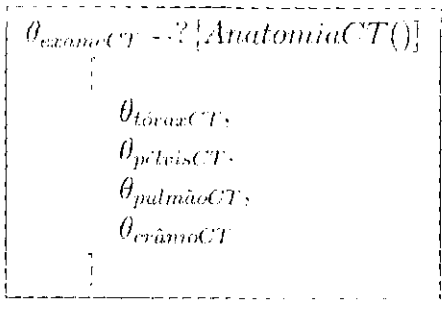


Figura 6.10: Exemplo de uso do operador ConditionalCase, a expressão $\theta_{exameCT}$ usa o operador para determina qual das expressões serão chamadas no momento da execução.

6.3.4 Operador de Projeção

O objetivo do operador de Projeção é permitir o armazenamento das características na expressão. Um exemplo é a alteração da expressão de domínio representada na Figura, pois nela somente as imagens de pulmão estão sendo armazenadas, dessa forma não há necessidade de armazenamento de todo o histograma, mas apenas daquele relacionado com as informações contidas nas características do pulmão segmentado. Com isso, pode ser feita uma modificação na linha 7, e a expressão $\theta_{pulmãoCT}$ resulta em: $\Pi([h_{pulmão}[1 - 2000])\Delta_X(Patologia, L_2())$.

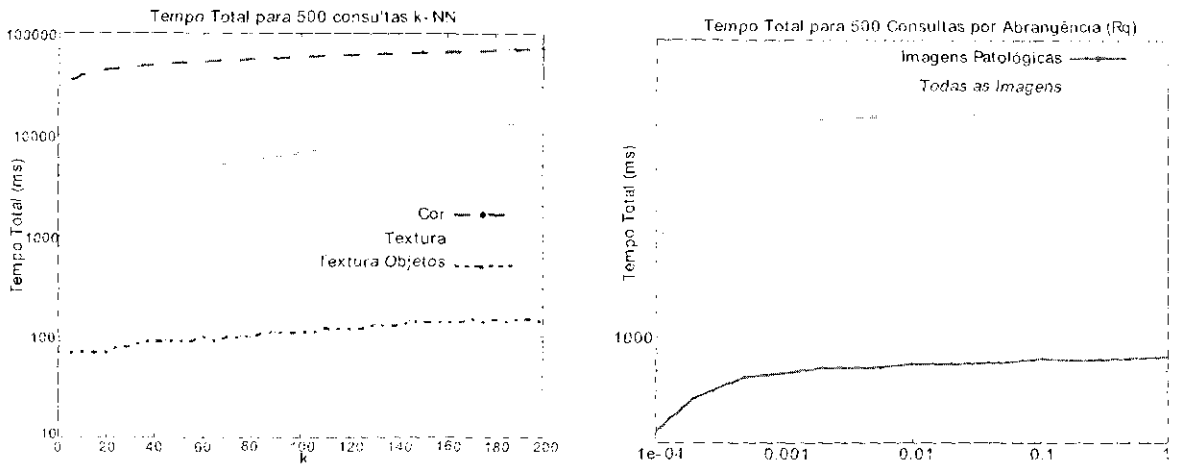
6.3.5 Operadores de Persistência

O operador de persistência Δ_I(`NomeAtributo`), armazena a imagem corrente no atributo `NomeAtributo`. Seu uso no contexto de uma aplicação é diferenciado pela possibilidade de armazenamento das imagens geradas na execução de uma expressão de domínio. Isso porque a imagem original normalmente é armazenada pela própria aplicação no momento de sua aquisição.

O operador de persistência Δ_S(`NomeAtributo`) armazena o conjunto corrente de características no atributo `NomeAtributo`. É interessante notar que as características podem representar extrações, composições e qualquer combinação das características geradas no processamento. Por essa razão, o uso do operador $\Delta_S()$ é indicado para as características que demandam grande quantidade de processamento.

O operador de persistência Δ_X(`NomeCritério`.$MAM.df()$) prepara o conjunto de característica como um vetor denominado `NomeCritério`, de forma que ele pode ser usado para comparar ou indexar as imagens armazendas segundo o critério `NomeCritério`, usando a função distância $df()$ no MAM indicado.

A principal característica dos operadores de persistência é que eles são ortogonais aos outros operadores usados em uma expressão de domínio. Com isso, os operadores de persistência podem ser alocados em qualquer parte de uma expressão de domínio, bastando que as informações estejam disponíveis para serem armazenadas, o que facilita no processo de validação e construção dos processamentos configurados. Além disso, o uso de diversos operadores $\Delta_X()$ para um mesmo domínio acarreta a criação de diversas árvores.



(a) (b)

Figura 6.11: Tempo médio de execução de 500 consultas em uma relação contendo 11.000 imagens de CT. (a) para responder consultas kNN, de acordo com os critérios Cor, $Textura$, e $TexturaObjetos$. (b)Para responder consultas Rq para um índice com imagens filtradas (com patologias) e outro índdice com todas as imagens da base.

Para melhor avaliar a influência do uso de diversas árvores, a Figura 6.11 mostra a média do tempo de execução para 500 consultas kNN com diferentes k, em uma relação contendo 11.000 imagens. Esse conjunto de imagens ocupa aproximadamente $5GBytes$ no disco, com suas características ocupando por volta de $4MBytes$ e indexadas em instâncias do MAM Slim-Tree[Traina Jr. et al., 2000]. Na Figura 6.11(a) é apresentado o tempo total para realização de 500 consultas kNN, com o k variando de 5 a 200, para cada um dos três critérios presentes na exprvessão da Figura 5.3. Conforme ilustrado, o tempo depende da complexidade das características utilizadas na comparação. O critério Cor depende de um conjunto de características com 256 dimensões, sendo mais custoso que os critérios $Textura$ e $TexturaObjetos$, que são baseados em 6 e 10 dimensões, respectivamente.

Outra causa relacionada com a utilização de diversas árvores é a aplicação dos operadores condicionais, que atuam como filtros sobre o processamento, de forma que uma determinada árvore armazena somente as características das imagens que foram processadas pela mesma seqüência de operações. Na Figura 6.11(b) a execução das mesmas

consultas sobre duas árvores são comparadas, sendo ilustrado que o ganho na utilização de árvores sobre dados filtrados é consideravelmente maior.

6.3.6 Gerenciamento de Domínios

O controle das expressões de domínio com a utilização do conceito de domínio facilita seu gerenciamento, pois adiciona um nível de abstração mais adequado. Para realizar efetivamente esse gerenciamento, foi escolhido restringir o controle das expressões de domínio em forma de hierarquias de domínio.

A principal razão de restringir o gerenciamento dos domínios em forma de hierarquia é que dessa forma o gerenciamento é mais intuitivo e se aproxima com os controles de domínio existente, principalmente na área médica. Por exemplo, construir uma hierarquia semelhante à provida pelo BI-RADS facilita a compreensão do tratamento do processamento, bem como do significado semântico de cada domínio. É importante destacar que essa hierarquia é naturalmente construída de forma incremental. Além disso, mesmo com a hierarquia possuindo diversos níveis, a complexidade do gerenciamento não é aumentada.

Uma metodologia adequada para a construção da hierarquia de domínios é o processo de Generalização / Especialização, no qual vários domínios podem ser generalizados em um único, ou um domínio pode ser especializado em vários outros. Nesse aspecto, a hierarquia pode aumentar unindo duas hierarquias separadas em uma só, bem como incluir novas especificidades no processamento já existente estendendo uma hierarquia com outras.

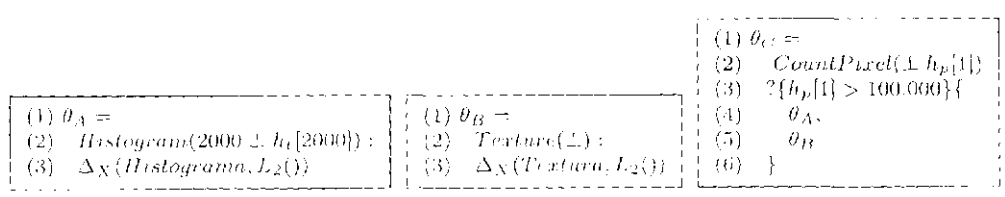


Figura 6.12: Exemplo de generalização de duas expressões de domínio, com a expressão θ_A e θ_B sendo generalizadas pela expressão θ_C.

Um exemplo simples de generalização é mostrado na Figura ??, na qual as expressões de domínio A e B, que são independentes, são generalizados pela expresssão C. A expressão A foi construída com o intuito de indexar os histogramas das imagens completas. Já a expressão B, faz a indexação de dados de textura de regiões de interesse (ROI) previamente selecionadas pelo usuário. Na construção da expressão C, adicionou-se um operador condicional que utiliza como parâmetro a contagem dos *pixels* da imagem, pois uma imagem com menor número de *pixels* é considerada ROI. Assim, o número de *pixels*

é utilizado pelo operador condicional para determinar se o processamento será continuado pelo expressão A ou B. A influência principal, nesse exemplo de generalização, é que o usuário passa a ativar apenas um processamento com um custo computacional adicional muito baixo. Além disso, caso configurado, o operador condicional (da expressão C) pode ativar as expressões A e B ao mesmo tempo. Assim é garantido de forma automatizada que todas as imagens sejam processadas pelas duas expressões. Isso poderia não ocorrer se as expressões A e B fossem independentes, pois o usuário seria o responsável por ativar as duas separadamente. A Figura 6.13 ilustra graficamente o processo de generalização apresentado na Figura ??.

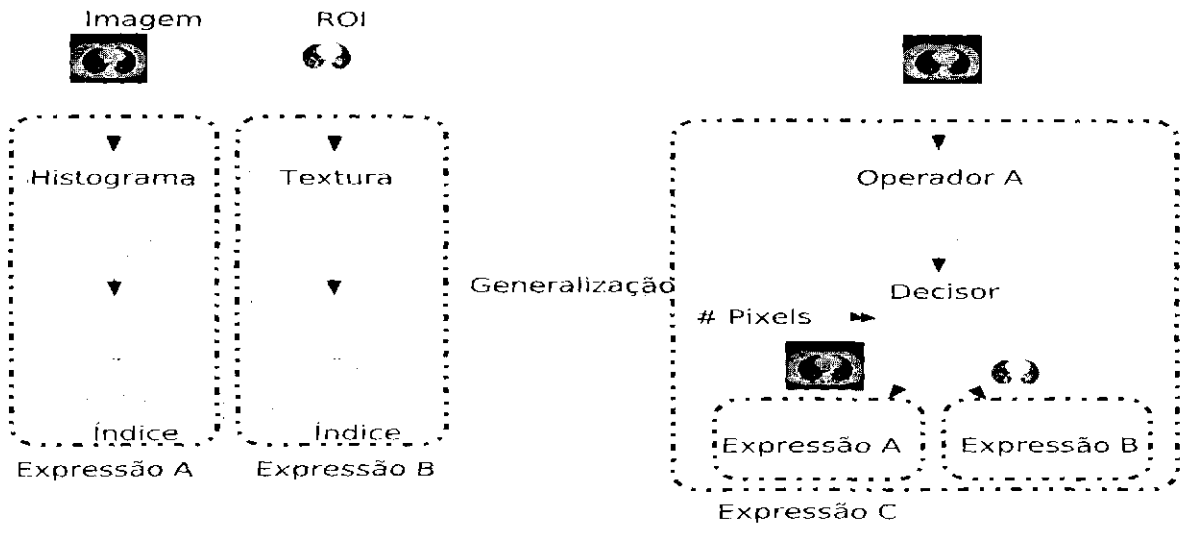


Figura 6.13: Ilustração representando a generalização mostrada na Figura 6.12.

Para o caso de Especialização de domínios, um exemplo é representado na Figura 6.14. A expressão A, é especializada pelas expressões B e C. A principal função da expressão A é utilizar um operador de segmentação e armazenar a imagem segmentada em disco. Para sua especialização optou-se por armazenar, além da imagem, o Histograma (expressão B) ou informações de Textura (expressão C). Para decidir qual expressão processar utiliza-se informações sobre a anatomia da região segmentada.

Nesse exemplo de especialização, novas funcionalidades são facilmente adicionadas ao processamento sem nenhuma alteração das funcionalidades iniciais. Essa é uma tarefa bastante complexa nos sistemas atuais, pois normalmente novas funcionalidades implicam na criação de *novas seqüências* de processamento que são *independentes* uma da outra. Isso faz com que modificações em partes de processamentos comuns tenham de ser em cada uma das seqüências.

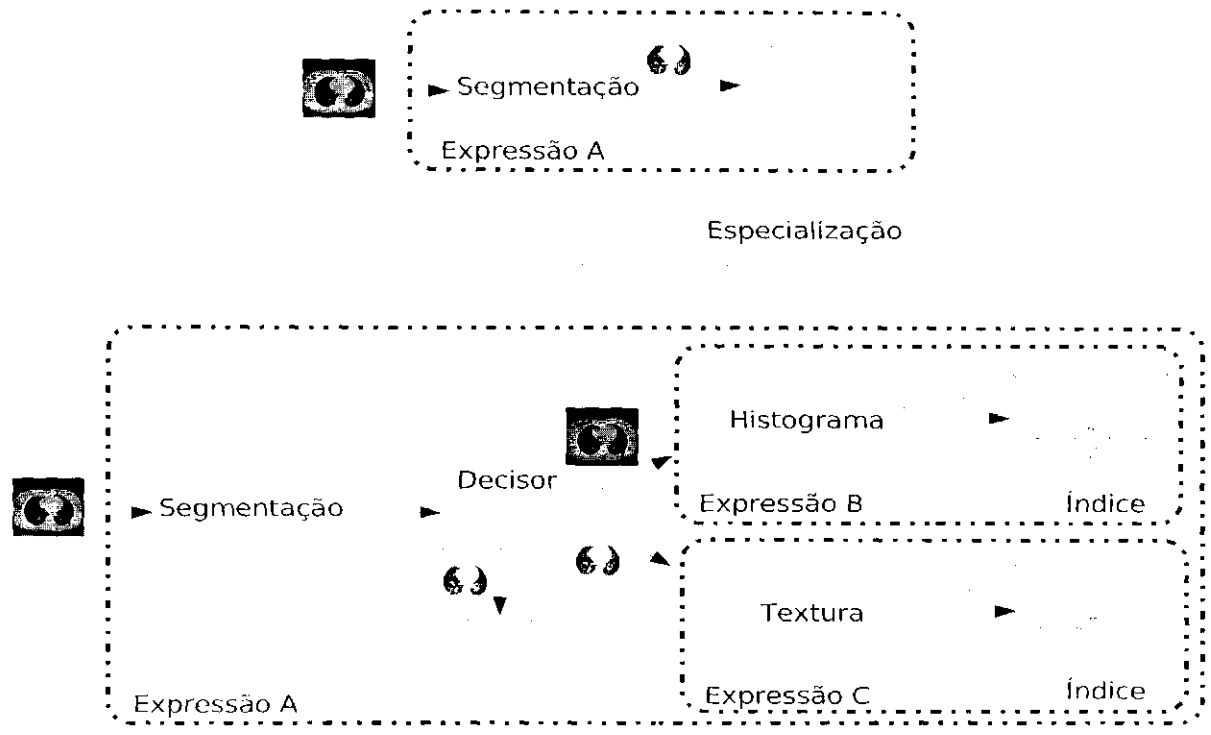


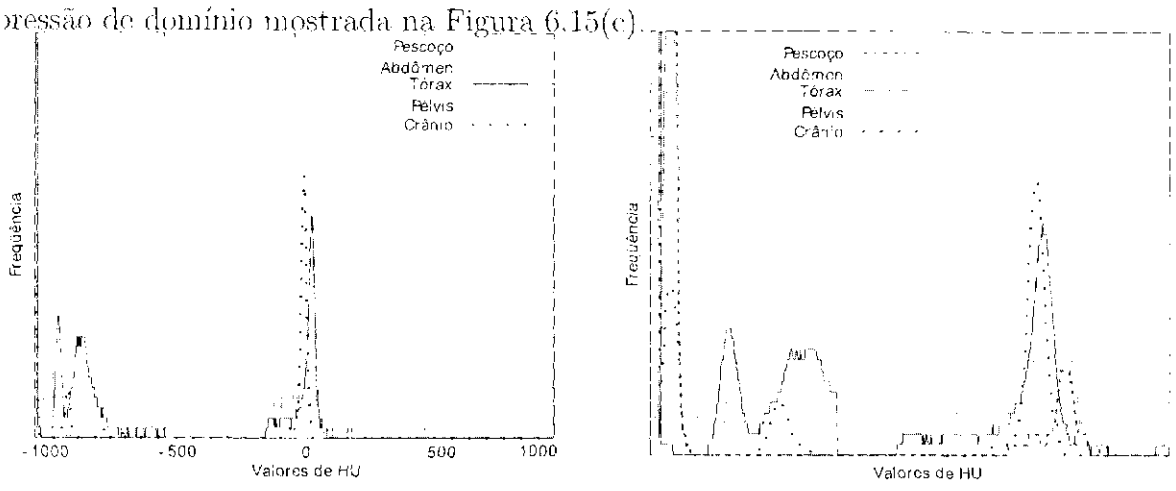
Figura 6.14: Exemplo de especialização de um domínio em outros dois.

6.3.7 Definição e Construção de novos operadores

A extensão de uma expressão de domínio pode ser feita com novos operadores, desde que o modelo de processamento dos dados seja obedecido. Um exemplo de inclusão de um novo operador é o uso do operador condicional em conjunto com o operador de Projeção. Esse operador tem uma ação poderosa e importante no contexto dos SiRICs, pois usando essa dupla pode-se realizar de maneira simples a redução de dimensionalidade sem recorrer a um algoritmo complexo que envolve as técnicas usuais de redução de dimensionalidade (como as descritas na Seção 3.3.1). Nesse contexto um novo operador poderia ser criado englobando essas funcionalidades. Por exemplo, o operador RedutorSemântico teria a seguinte definição: $RedutorSemântico(\lambda_1 \perp [_]) = \lambda_0$. Dessa forma, a condição é um parâmetro que indica o teste a ser realizado, e a saída indica qual características são projetadas se a condição for verdadeira.

A Figura 6.15 mostra um exemplo de contexto de uso do operador *RedutorSemântico*, através do uso da expressão de domínio . A Figura 6.15(a) apresenta o aspecto de vários histogramas de CT de diversas anatomias. Percebe-se pela figura, que os intervalos possuem pouca ou nenhuma influência na diferenciação das imagens, pois seus valores são próximos de zero. No entanto, quando se utilizam somente os valores que realmente influenciam na diferenciação entre as imagens, o processo de comparação se torna mais discriminatório com o número de características reduzido. A Figura

6.15(b) apresenta a projeção das características somente nos intervalos $[-1000, -800]$ e $[-200, 200]$. Nesse contexto, o operador *RedutorSemântico* pode ser utilizado verificando se a imagem é de CT e projetando os intervalos de características indicados, conforme a expressão de domínio mostrada na Figura 6.15(c).



(a) (b)

$$Histogram(2000 \perp h[2000]) : \\ \Pi(h[1-200, 800-1200])$$

(c)

Figura 6.15: Aparência do histograma de CT de várias anatomias. (a)Exemplos de histograma completo para as anatomias indicadas. (b)Parte do histograma com projeção dos intervalos $[-1000, -800]$ e $[-200, 200]$, que são as características que influenciam na diferenciação das imagens. (c)Operadores que realizam as extração do Histograma e a projeção do intervalo equivalente aos mostrados em (b).

6.3.8 Construção das Expressões de Domínio

Uma expressão de domínio é construída com o objetivo de conter todas as regras para manipulação de um domínio imagem. Nesse sentido, um domínio imagem pode ser classificado como o contexto no qual a imagem é utilizada e processada. Esse conceito é recursivo pois um domínio pode possuir sub-domínios, que também podem ser divididos em outros sub-domínios.

A criação de um domínio imagem leva em conta o conhecimento adquirido pelo especialista em imagens, pois, o conjunto de processamento que as imagens do domínio são submetidas e o comportamento das características nesses processamentos são conhecidos pelo especialista. O uso dos operadores definidos em uma expressão de domínio permitem uma grande flexibilidade no tratamento do tipo imagem e suas características. Para facilitar o uso do tipo imagem no contexto das expressões de domínio, os seguintes passos são indicados para a construção adequada de um determinado domínio:

1. determinar em qual contexto as imagens serão manipuladas.
2. determinar o conjunto de características que representam as informações relevantes para o contexto determinado.
3. determinar o conjunto de operadores que geram as características do item anterior.
4. determinar a ordem com que os operadores serão acionados e a forma com que as características serão manipuladas e armazenadas.
5. determinar a forma como a similaridade entre as imagens será medida.
6. determinar a forma de armazenamento das características. Caso seja utilizado um MAM determinar qual é o mais adequado.
7. se a expressão de domíno final encontrada utilizar um número muito grande de operadores, verificar a possibilidade de especializar o domínio em subdomínios, de forma que expressões mais simples sejam utilizadas para os vários domínios.
8. se outro domínio utiliza conjunto semelhante de operadores, verificar se é possível generalizar as partes comuns em outro domínio, com intuito de simplificar a expressão final.

O primeiro passo é importante, pois o contexto de uso da imagem é o principal fator para o direcionamento dos passos seguintes. Uma imagem representando um exame de ultrassonografia de gravidez pode estar contida em uma base de uma clínica médica ou em um álbum de família, ou seja, o tratamento dessa imagem em um contexto é totalmente diferente do outro. Com o uso de uma expressão de domínio, a definição do conjunto de processamento mais adequado para um determinado contexto é facilitada, porque pode ser feita de forma incremental, enfocando um processamento por vez.

O segundo passo envolve escolher as características que melhor representem as informações contidas na imagem. Por exemplo, características de cor e textura podem ser importantes para uma determinada situação, enquanto que em outras as informações de forma podem ser mais adequadas. Para imagens médicas, muitas vezes a contagem de elementos de forma semelhante pode levar ao diagnóstico, como no caso de imagens de células sangüíneas, no qual a contagem de glóbulos podem determinar o diagnóstico final. Nessa etapa o especialista envolvido na manipulação das imagens tem total conhecimento dos possíveis conjuntos de características que melhor representam cada imagem. O uso das expressões de domínio facilita o trabalho do especialista no sentido de que partes da expressão podem ser futuramente alteradas, caso características mais adequadas sejam encontradas posteriormente.

Da mesma forma que no segundo passo, o terceiro passo envolve a escolha dos algoritmos disponíveis que serão utilizados para manipular as imagens. Com isso, aspectos

como tempo de processamento têm de ser considerados, pois um processador imagem pode gerar características expressivas à um custo computacional demasiadamente alto, inviabilizando seu uso com uma base de imagens demasiadamente grande. Com isso, o uso das expressões de domínio se torna importante porque uma expressão de domínio pode ser modificada com a inclusão ou exclusão de operadores virtualmente em qualquer parte, sem que isso afete os outros processamentos já configurados e que não têm relação com as modificações.

Para o quarto passo, a ordem das operações é importante para questões de melhorias no tempo total de processamento, e envolve o uso dos operadores de controle. Uma delas ocorre com o uso do operador Fork, que permite que operações sejam configuradas para serem executadas em paralelo. Outra forma é a utilização dos operadores condicionais e a projeção, que podem evitar que processamentos desnecessários ocorram. Isso afeta também a forma de armazenamento, pois características que não são aproveitadas fazem com que o espaço de armazenamento seja menor.

No quinto passo, é escolhida a forma como as imagens serão efetivamente comparadas, ou seja, como a similaridade das imagens será medida. Assim, a forma tradicional dos sistemas de busca por conteúdo em imagens é que essa ação seja de responsabilidade da função de distância, ou seja, a similaridade é medida apenas sobre os vetores de características. Adicionalmente, o uso dos operadores condicionais permite que testes sejam realizados com o intuito de direcionar o processamento e fazer com que alguns processamentos sejam realizados somente sobre imagens com propriedades semelhantes. Diferentemente de uma função de distância que compara somente vetores de características, os testes realizados pelos operadores condicionais podem ser feitos sobre outros tipos de propriedades das imagens. Além disso, diversas formas de armazenamento e medida de similaridade podem ser utilizadas simultaneamente para um mesmo domínio.

O sexto passo é realizado com o intuito de fazer com que uma expressão de domínio seja executada apenas uma vez para cada imagem, pois o tempo de processamento total tende a ser grande. Dessa maneira, o armazenamento das características evita que elas tenham de ser geradas novamente. Nesse contexto, uma expressão de domínio permite definir qualquer composição do conjunto de características e os operadores de persistência podem ser utilizados em qualquer parte do processamento. Para o caso do operador SimilarityCriterion (Δ_X), seu uso é vinculado a um MAM e a uma função de distância. Sua vantagem reside na facilidade de inclusão em qualquer parte da expressão de domínio, sem afetar o funcionamento dos outros operadores. Já no caso do operador FeatureVector Δ_S, a vantagem do seu uso é que as características por ele armazenadas podem fazer parte de qualquer novo critério de busca, sem que para isso a imagem tenha de ser novamente processada, permitindo que vários critérios de busca possam ser utilizados sobre o mesmo conjunto de características. Dessa forma, um mesmo conjunto pode ser indexado

simultaneamente por MAMs e/ou funções de distância diferentes.

Os passos sétimo e oitavo dizem respeito ao gerenciamento das expressões de domínio e, têm por objetivo facilitar o controle sobre as diversas expressões que podem ser mantidas simultaneamente pelo sistema. O enfoque é utilizar os conceitos de Especialização e Generalização para criação de novas expressões que representem domínios e sub domínios das imagens processadas. Maiores detalhes sobre o gerenciamento efetivo usando esses conceitos podem ser encontrados em [Figueiredo et al., 2004], que apresenta o uso das expressões de domínio para o controle de domínios e sub-domínios de imagens médicas.

6.4 Conclusões sobre o uso das Expressões de Domínio

A implementação e os resultados experimentais apresentados neste capítulo demonstram que o uso das expressões de domínio em ambientes distintos e com grande quantidade de imagens traz vantagens em relação às formas tradicionais de manipulação, armazenamento e recuperação de imagens.

No que concerne aos aspectos de implementação de um SiRIC, tratar os operadores como componentes tem como principal motivação o uso dos operadores condicionais. Isso ocorre, porque uma das complexidades embutidas nos operadores condicionais é que a seqüência exata de processamento de uma determinada imagem é determinada somente em tempo de execução. Nesse aspecto, a quantidade de recursos que o ambiente aloca para uma determinada seqüência tende a ser demasiadamente grande. Conseqüentemente, alocar todos os operadores de uma expressão de domínio influencia muito o tempo total de processamento. Portanto, a implemetação dos operadores em forma de componentes com acoplamento dinâmico torna o ambiente mais propício para o tipo de processamento a que se propõe. Além disso, é contemplado o uso concorrente do sistema, pois operadores instanciados por uma expressão podem ser usados por outra sem que seja preciso desalocar seus recursos e alocá-los novamente.

Ainda no âmbito de um SiRIC, os exemplos de gerenciamento e construção incremental das expressões de domínio mostram que o uso das expressões simplifica a realização de tarefas normalmente complexas, como as relativas à comparação entre algoritmos. Nesse caso, as expressões de domínio consideram algoritmos semelhantes como operadores. Com isso trocar um operador por outro em uma expressão não demanda alterações complexas no gerenciamento do ambiente.

Outra tarefa considerada complexa nos SiRICs tradicionais é o tratamento de questões semânticas no processamento. O uso dos operadores definidos permitem que propriedades específicas do domínio imagem abordado sejam isoladas e embutidas no

processamento. Essa propriedade do processamento automatizado também ser contextualizado torna o tratamento das imagens mais próximo da semântica entendida pelo usuário.

No contexto dos SGBDs, a incorporação efetiva do domínio imagem pode ser observada pelos diversos aspectos embutidos nas expressões de domínio. Um primeiro aspecto está relacionado com a transparência no acesso à imagem, ou seja, aplicações que não utilizam imagens não são afetadas e as que utilizam o fazem sem precisar acessar diretamente as manipulações configuradas nas expressões de domínio.

A tarefa de criação de índices tem o mesmo perfil que para os outros tipos de dados, isto é, os índices para imagens permite que eles sejam criados somente para os critérios com mais acessos. Outra questão importante é o fato dos índices serem transversais aos processamentos realizados sobre as imagens, o que permite que outros índices possam ser criados em qualquer momento. Outro aspecto de transparência no uso dos índices é que os mesmos são acessados pelo critério de comparação definido, o que encapsula toda complexidade inerente ao uso dos MAM.

A influência sobre os usuários dos ambientes SiRIC e SGBD torna o uso das expressões de domínio adequadas para locais com diversos tipos de usuários, pois os papéis e responsabilidades de cada um podem ser mais facilmente definidos.

Capítulo

7

Conclusão

7.1 Considerações Finais

Este trabalho apresenta um conjunto de inovações que tem como principal característica integrar de forma efetiva imagens e SGBDs Relacionais. O ponto central é a definição do conceito de expressões de domínio, o que permite a integração do processo SiRIC em SGBDs, integrando o armazenamento das imagens, a busca por conteúdo e as operações de manipulação que ocorrem sobre elas. O conceito de expressões de domínio é definido com um conjunto pequeno e poderoso de operadores que permite a integração do processamento de características de baixo nível (extraídas automaticamente das imagens) com o processamento semântico desejado pelo usuário.

A especificação do processo SiRIC como uma composição de operadores permite que operações de comparação sejam expressadas de maneira flexível e poderosa. Isso permite a configuração de extratores de características genéricos que podem ser empregados e reutilizados em diversas expressões de domínio. O que leva à definição de como as imagens de um domínio específico podem ser comparadas. A transparência com que isso é feito, permite que essa tarefa seja realizada por especialistas com conhecimento do domínio da aplicação, e com pouco conhecimento da área de processamento de imagens e / ou de banco de dados.

As expressões de domínio permitem que conjuntos de operações complexas sejam construídas de forma incremental. Algoritmos tradicionais dos SiRICs podem ser implementados como instâncias de algum dos operadores definidos de forma que expressões de domínio podem ser criadas contemplando seqüências complexas de operações que podem

ser facilmente alteradas. Além disso, a sintonia de operações específicas ou do processo como um todo podem ser facilmente realizadas. Essa característica permite que os especialistas em imagens restrinjam os problemas de forma mais adequada.

O uso dos operadores condicionais permitem a inclusão de informações de contexto com processamento automatizado. Esse fato possibilita uma aproximação maior dos aspectos semânticos existentes nas imagens com o processamento automatizado. A combinação desse processamento decisório com uma arquitetura baseada em operadores resulta em um ambiente que permite a configuração de processamentos complexos com seqüências variadas execução.

A possibilidade de reuso das expressões de domínio permite que o gerenciamento seja feito na forma de hierarquias que representam a relação dos domínios e seus respectivos sub-domínios. Isso permite também que o processamento como um todo seja abstraído para o controle simples de uso dos domínios e suas respectivas imagens.

A facilidade de separar (indexar) individualmente os diferentes domínios criados na hierarquia. O que permite otimizar o processamento das consultas mais requisitas sem influenciar o processamento usual das imagens. Toda essa gama de inovações abre novas perspectivas que permitem que as funcionalidades de busca por conteúdo nas imagens fiquem mais propensas de serem usadas atendendo aos requisitos existentes nos ambientes complexos e heterogêneo como os PACS.

É possível comparar as técnicas de redução de dimensionalidade com o operador de projeção. As técnicas usuais de redução de dimensionalidade têm como principal característica realizar um determinado processamento sobre todas as características dos vetores. Algumas conseguem realizar processamentos locais, enquanto que outras tratam o conjunto de dados de forma global. No que concerne ao seu uso em sistemas de banco de dados, além da facilidade para tratamento de dados dinâmicos, outra preocupação passa a ser sua escalabilidade. Englobando todos esses fatores percebe-se que o operador de projeção não é uma nova técnica de redução de dimensionalidade, mas sim uma nova maneira para manipulação dos dados na qual a semântica (ou seqüência) da manipulação é o fator mais importante. A semelhança entre as duas abordagens é tratar a escolha de características significativas dos vetores de características como um fator de re-alimentação do processamento. Não obstante, outra diferença importante é o fato de que praticamente todos os métodos para redução de dimensionalidade geram perdas nas informações contidas no vetor original, enquanto que a forma de manipulação aqui proposta permite que o vetor (ou vetores) original seja recuperado. Além disso, o uso das técnicas de redução de dimensionalidade não fica descartado, mas passa a ser uma funcionalidade opcional no conjunto de processamentos.

Para validar a aplicabilidade dos conceitos propostos, foi implementado um protótipo de uma ferramenta SiRIC para responder consultas cujos requisitos envolvem imagens

similares de exames médicos armazenados em uma grande base de dados hospitalar.

Com a arquitetura apresentada é possível substituir os servidores DICOM tradicionais por uma arquitetura aberta que usa como infraestrutura um SGBD Relacional que armazena também os arquivos DICOM. O encapsulamento do servidor DICOM dentro de um SGBD possibilita que programadores trabalhem de forma mais adequada porque as imagens podem ser acessadas da mesma forma que os outros dados e os cabeçalhos DICOM podem ser incluídos como predicados nas consultas.

Outra conseqüência importante é que a divisão das responsabilidades dos usuários do ambiente PACS passsa a ocorrer naturalmente. Cada tipo de usuário se preocupa com a tarefa que mais combina com seus conhecimentos e atividades. O médico define a hierarquia de domínios que deseja trabalhar, o técnico em imagens faz a configuração do sistema e o programador se preocupa somente com os algoritmos a serem implementados. Sob a perspectiva do administrador do sistema, as imagens DICOM se tornam apenas outro tipo de dado sem necessidade de possuir armazenamento e acesso diferenciado.

7.2 Principais Contribuições

As principais contribuições deste trabalho são:

- a apresentação de um modelo formal que cobre todos os aspectos de manipulação das imagens. O que unifica os trabalhos direcionados a imagens que abordam somente questões isoladas. Os Capítulos 2 e 3 discutem os trabalhos mais relevantes desse contexto;
- a identificação e classificação do conjunto mínimo de operações necessárias em um SiRIC. Essas operações foram divididas em:
  - processadores de imagem,
  - extratores de características,
  - construtores de vetores de características,
  - funções de distância,
  - controles do fluxo de processamento e
  - processadores de persistência.
- a inclusão do tipo imagem no conjunto de tipos manipulados internamente por um SGBD Relacional. O que torna transparente as operações internas relacionadas com o tipo imagem, como manipulação, transformação e indexação.

- a inclusão de aspectos semânticos no processamento automatizado permite que grande parte do conhecimento adquirido pelo especialista em imagem influa no conjunto de processamentos realizados.

As contribuições deste trabalho têm influência direta sobre diversas áreas, com intersecções em diversos aspectos. Para facilitar a identificação das contribuições, pode-se dividir as contribuições em três grupos: SGBDs, SiRIC e PACS.

7.2.1 Contribuições para o contexto dos SGBDs

Além da inclusão do tipo imagem como mais um tipo nativos dos SGBDs Relacionais, as seguintes contribuições podem ser destacadas para esse contexto:

- transparência no uso das imagens em relação às aplicações clientes, ou seja, o acesso às imagens é feito da mesma maneira que para os outros tipos de dados nativos (números e pequenas cadeias de caracteres);
- realização de buscas por conteúdo em imagens integradas às consultas tradicionais, com os critérios de comparação das imagens fazendo parte dos predicados das consultas;
- flexibilidade na definição dos índices baseados nos critérios de comparação;
- possibilidade de criação de domínios de imagens com a definição de restrições sobre o tipo imagem;
- encapsulamento dos processamentos com abstrações específicas de
  - funções de distância e conjunto de características abstraídas em critérios de comparação;
  - processamentos sobre as imagens abstraídos em domínios;
  - algoritmos abstraídos em operadores;
  - reuso de domínio abstraído em sub-domínios.

7.2.2 Contribuições para o contexto dos SiRICs

Para a contexto dos SiRIC, além da formalização de seu processamento, as seguintes contribuições podem ser colocadas:

- uniformização das manipulações realizadas sobre as imagens;
- alteração e flexibilização do fluxo de processamento entre as diversas etapas de processamento;

- contextualização dinâmica do processamento, possibilitando que decisões semânticas sejam embutidas no processamento automático;
- criação e validação de domínios e sub-domínios de imagens com gerenciamento hierárquico;
- flexibilização da composição e combinação dos vetores de características;
- realização de processamento paralelo e temporário;
- possibilitar que o tratamento de novos tipos de imagens (incluindo novos formatos de arquivo) sejam incluídos no conjunto de processamento existente sem a necessidade de modificações demasiadas;
- criação de uma plataforma uniforme para testes de novos algoritmos, sejam eles de processamento de imagens, buscas por conteúdo, etc. De forma que o enfoque seja dado especificamente sobre o algoritmo abordado.

7.2.3 Contribuições para o contexto dos PACs

Para o contexto médico, mais especificamente dos PACS, este trabalho inova nos seguintes aspectos:

- fornece uma plataforma aberta que permite a integração efetiva dos PACS com os HIS;
- possibilita a inclusão no processamento de parâmetros que reflitam o conhecimento adquirido pelo especista em imagens;
- uniformiza o tratamento das imagens pertencentes a modalidades diferentes;
- possibilita que informações contidas no cabeçalho DICOM sejam facilmente embutidas no processamento;
- facilita a divisão de responsabilidades entre os diversos tipos de usuários.

7.3 Proposta de Trabalhos Futuros

A formalização apresentada neste trabalho traz uma nova perspectiva para a manipulação de imagens. No entanto, vários aspectos precisam ser complementados para tornar o uso das expressões de domínio mais efetivo. Nesse contexto, a principal demanda para trabalhos futuros envolve o trabalho conjunto dos diversos profissionais que usam imagens. Por exemplo, é preciso a atuação conjunta de profissionais das áreas de medicina, radiologia,

física médica, processamento de imagens e banco de dados. A centralização provida pelas expressões de domínio permitem que esses profissionais usem esse ferramental mantendo as concepções específicas de cada um. Somente com esse trabalho multidisciplinar é possível fazer com que a tecnologia SiRIC evolua de forma efetiva. Um exemplo de trabalho desse tipo é a modelagem, implementação e validação de uma hierarquia de domínios equivalente à hierarquia fornecida pelo BI-RADS.

Outros aspectos mais específicos que podem ser levantados para trabalhos gerados a partir deste são:

- verificar a viabilidade do uso dos operadores na fase de aquisição das imagens. Isso é interessante em contextos como das imagens de ressonância magnética, nos quais é possível vislumbrar que a manipulação direta do espaço-k afeta diretamente a qualidade da imagem;

- implementar operadores de persistência que tenham como parâmetro o local de armazenamento, no sentido de possibilitar o uso de dispositivos terciários e quaternários de armazenamento. Dessa maneira, dispositivos como *jukebox*, CDs e DVDs, podem ser incluídos no processo de busca por conteúdo;

- implementação dos comandos SQL relacionados com a criação dos domínios, das expressões de domínio e critérios de comparação. Essa tarefa está relacionada com a implementação de um interpretador SQL específico que contemple os conceitos envolvidos na formalização;

- implementar interfaces gráficas específicas para cada tipo de usuário. Esse tipo de tarefa é relevante porque possibilita que os aspectos relevantes para cada tipo de usuário sejam destacados individualmente em cada interface. Dessa maneira, as limitações encontradas na infra-estrutura do ambiente podem ser mais facilmente superadas. O importante é que essa infra-estrutura se mantenha centralizada nos conceitos das expressões de domínio.

Referências Bibliográficas

[13249-5:2001, 2001] 13249-5:2001, I. (2001). Information technology - database languages - sql multimedia and application packages part 5: Still image. 77

[Adali et al., 2000] Adali, S., Sapino, M., e Subrahmanian, V. (2000). An algebra for creating and querying multimedia presentations. *Multimedia Systems*, 8:212 230. 2

[Adali et al., 1999] Adali, S., Sapino, M. L., e Subrahmanian, V. S. (1999). A multimedia presentation algebra. In Delis, A., Faloutsos, C., e Ghandeharizadeh, S., editors, *ACM International Conference on Management of Data (SIGMOD'1999)*, pp. 121 132, Philadelphia, Pennsylvania, USA. ACM Press. 2

[Aggarwal, 2001] Aggarwal, C. C. (2001). Re-designing distance functions and distance-based applications for high dimensional data. *SIGMOD Record*, 30(1):13 18. 35

[Aggarwal, 2002a] Aggarwal, C. C. (2002a). Hierarchical subspace sampling: a unified framework for high dimensional data reduction, selectivity estimation and nearest neighbor search. In *ACM International Conference on Management of Data (SIGMOD'2002)*. pp. 452 463, Madison, Wisconsin. ACM Press. 48

[Aggarwal, 2002b] Aggarwal, C. C. (2002b). Towards meaningful high-dimensional nearest neighbor search by human-computer interaction. In *18th International Conference on Data Engineering (ICDE'2002)*, pp. 593- 604, San Jose, CA. IEEE Computer Society. 37, 48

[Aggarwal, 2003] Aggarwal, C. C. (2003). Towards systematic design of distance functions for data mining applications. In *KDD '03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 9 18, Washington, D.C. ACM Press. 38

[Aggarwal et al., 2001] Aggarwal, C. C., Hinneburg, A., e Keim, D. A. (2001). On the surprising behavior of distance metrics in high dimensional space. In den Bussche, J. V. e Vianu, V., editors, *8th International Conference on Database Theory (ICDT'2001)*, v. 1973 of *Lecture Notes in Computer Science*, pp. 420–434, London , UK. 35

[Aggarwal & Yu, 2000] Aggarwal, C. C. e Yu, P. S. (2000). The igrid index: reversing the dimensionality curse for similarity indexing in high dimensional space. In *KDD '00: Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 119–129. ACM Press. 38

[Aha & Bankert, 1995] Aha, D. W. e Bankert, R. L. (1995). A comparative evaluation of sequential feature selection algorithms. In *Proceedings of the Fifth International Workshop on Artificial Intelligence and Statistics*, Ft. Lauderdale, FL. 30

[Ahn et al., 2001] Ahn, H.-K., Mamoulis, N., e Wong, H. M. (2001). A survey on multi-dimensional access methods. Technical Report UU-CS-2001-14, Institute of Infomation and Computing Sciences, Utrecht University, The Netherlands. 41

[Amato, 2002] Amato, G. (2002). *Approximate Similarity Search in Metric Spaces*. PhD thesis, Compute Science Department - University of Dortmund,. 21

[An et al., 2004] An, J., Chen, H., Furuse, K., e Ohbo, N. (2004). Cva file: an index structure for high-dimensional datasets. *Knowledge and Information Systems - Springer Verlag London*. 34, 45

[Antani et al., 2002] Antani, S., Kasturi, R., e Jain, R. (2002). A survey on the use of pattern recognition methods for abstraction, indexing and retrieval of images and video. *Pattern Recognition*, 35(4):945–965. 8

[Antani et al., 2004a] Antani, S., Lee, D. J., Long, L., e Thoma, G. R. (2004a). Evaluation of shape similarity measurement methods for spine x-ray images. *Journal of Visual Communication and Image Representation Special issue: Multimedia Database Management Systems*, 15(3):285–302. 15

[Antani et al., 2004b] Antani, S., Long, L. R., e Thoma, G. (2004b). Content-based image retrieval for large biomedical image archives. In et al., M. F., editor, *11th World Congress on Medical Informatics (MEDINFO) 2004 Imaging Informatics*, pp. 829–833, San Francisco, CA, USA. 10

[Arantes, 2005] Arantes, A. S. (2005). *Consultas por Similaridade Complexas em Gerenciadores Relacionais*. Tese, Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo (ICMC-USP). 34

[Araujo et al., 2002] Araujo, M. R. B., Traina, A. J. M., e Caetano Traina, J. (2002). Extending sql to support image content-based retrieval. In *IASTED International Conference Information Systems and Databases (ISDB 2002)* 6, p., Tokyo, Japan. 5

[Ashwin et al., 2002] Ashwin, T., Gupta, R., e Ghosal, S. (2002). Leveraging non-relevant images to enhance image retrieval performance. In *Multimedia'02*, pp. 331–334, Juan-les-Pins, France. 22

[Aslandogan & Yu, 1999] Aslandogan, Y. A. e Yu, C. T. (1999). Techniques and systems for image and video retrieval. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 11(1):56–63. 8

[Atnafu et al., 2001] Atnafu, S., Brunie, L., e Kosch, H. (2001). Similarity-based algebra for multimedia database systems. In *IEEE*. 2

[Atnafu et al., 2004] Atnafu, S., Chbeir, R., Coquil, D., e Brunie, L. (2004). Integrating similarity-based queries in image dbmss. In *ACM SAC*, pp. 735–739. Nicosia, Cyprus. ACM Press. 54

[Baeza-Yates et al., 1994] Baeza-Yates, R. A., Cunto, W., Manber, U., e Wu, S. (1994). Proximity matching using fixed-queries trees. In *5th Annual Symposium on Combinatorial Pattern Matching (CPM)*, v. 807 of *Lecture Notes in Computer Science*, pp. 198–212, Asilomar, CA. Springer Verlag. 44

[Balan et al., 2004] Balan, A. G. R., Traina, A. J. M., e Traina Jr., C. (2004). Recuperação de imagens por conteúdo utilizando características de forma da imagem segmentada com base em textura. In *IX Congresso Brasileiro de Informática em Saúde (CBIS'2004)*, pp. 1–6(CDROM), Ribeirão Preto, SP, Brazi. 63

[Bartolini, 2002] Bartolini, I. (2002). *Efficient and Effective Similarity Search in Multimedia Databases*. PhD thesis, Dipartimento di Elettronica Informatica e Sistemistica, Università degli Studi di Bologna, Bologna, Italy. 8

[Bayer & McCreight, 1972] Bayer, R. e McCreight, E. M. (1972). Organization and maintenance of large ordered indexes. *Acta Informatica*, 1(3):173–189. 39, 40

[Beckmann et al., 1990] Beckmann, N., Kriegel, H.-P., Schneider, R., e Seeger, B. (1990). The r*-tree: An efficient and robust access method for points and rectangles. In *ACM International Conference on Management of Data (SIGMOD'1990)*, pp. 322–331. 42

[Belussi & Faloutsos, 1995] Belussi, A. e Faloutsos, C. (1995). Estimating the selectivity of spatial queries using the correlation fractal dimension. In Dayal, U., Gray, P. M. D., e Nishio, S., editors, *21th International Conference on Very Large Databases (VLDB'1995)*, pp. 299–310, Zurich, Switzerland. Morgan Kaufmann. 29

[Berchtold et al., 1996] Berchtold, S., Keim, D. A., e Kriegel, H.-P. (1996). The X-tree: An index structure for high-dimensional data. In *Proceedings of the 22nd International Conference on Very Large Data Bases (VLDB)*, pp. 28–39, Bombay, India. Morgan Kaufmann Publishers. 43

[Berry et al., 1994] Berry, M. W., Dumais, S., e O'Brien, G. (1994). Using linear algebra for intelligent information retrieval. *SIAM Review*, 37(4):573–595. 32

[Besson et al., 2003] Besson, L., Costa, A. D., Leclercq, E., e Terrasse, M.-N. (2003). A cbir-framework: using both syntactical and semantical information for image description. In *7th International Database Engineering and Applications Symposium (IDEAS 2003)*, pp. 385–390, Hong Kong, China. IEEE Computer Society. 18

[Beyer et al., 1999] Beyer, K., Godstein, J., Ramakrishnan, R., e Shaft, U. (1999). When is "nearest neighbor"meaningful? In Beeri, C. e Buneman, P., editors, *7th International Conference on Database Theory (ICDT '1999)*, v. 1540 of *Lecture Notes in Computer Science*, pp. 217–235, Jerusalem, Israel. Springer-Verlag GmbH. ISBN 3-540-65452-6. 36

[Bick & Lenzen, 1999] Bick, U. e Lenzen, H. (1999). Pacs: the silent revolution. *European Radiology - Computer applications*, 9:1152–1160. 3

[Bidgood et al., 1998] Bidgood, W. D., Horii, S. C., Prior, F. W., e Syckle, D. E. V. (1998). *Medical Image Databases*, Cap. 2 - Understanding and Using DICOM, The Data Interchange Standard for Biomedical imaging, pp. 25–52. The Kluwer International Series in Engineering and Computer Science. Klower Academic Publishers. 3

[Blum & Langley, 1997] Blum, A. L. e Langley, P. (1997). Selection of relevant features and examples in machine learning. *Artificial Inteligence*, 97:245–271. 30

[Bozkaya & Özsoyoglu, 1997] Bozkaya, T. e Özsoyoglu, Z. M. (1997). Distance-based indexing for high-dimensional metric spaces. In *ACM International Conference on Management of Data (SIGMOD'1997)*, pp. 357–368, Tucson, AZ. ACM Press. 44

[Bozkaya & Özsoyoglu, 1999] Bozkaya, T. e Özsoyoglu, Z. M. (1999). Indexing large metric spaces for similarity search queries. *ACM Transactions on Database Systems (TODS)*, 24(3):361–404. 44

[Brauner & Shacham, 2000] Brauner, N. e Shacham, M. (2000). Considering precision of data in reduction of dimensionality and pca. *Computers & Chemical Engineering*, 24:2603–2611. 30, 32, 34

[Brin, 1995] Brin, S. (1995). Near neighbor search in large metric spaces. In Dayal, U., Gray, P. M. D., e Nishio, S., editors, *21th International Conference on Very Large Databases (VLDB'1995)*, pp. 574–584, Zurich, Switzerland. Morgan Kaufmann. 44

[Brodley et al., 1999] Brodley, C. E., Kak, A. C., Shyu, C. R., Dy, J. G., Broderick, L. S., e Aisen, A. M. (1999). Content-based retrieval from medical image databases: A sinergy of human interaction, machine learning and computer vision. In *Sixteenth National Conference on Artificial Intelligence and Eleventh Conference on Innovative Applications of Artificial Intelligence (AAAI / IAAI'1999)*, pp. 760–767, Orlando, Florida, USA. AAAI Press / The MIT Press. 18

[Bueno et al., 2002] Bueno, J. M., Chino, F., Traina, A. J. M., Traina, Caetano, J., e Marques, P. M. d. A. (2002). How to add content-based image retrieval capability in a pacs. In *IEEE International Conference on Computer Based Medical Systems - CBMS*, pp. 321–326, Maribor, Slovenia. IEEE Computer Society. 3

[Bueno et al., 2005] Bueno, R., Traina Jr., C., e Traina, A. J. M. (2005). Accelerating approximate similarity queries using genetic algorithms. In *20th Annual ACM Symposium on Applied Computing (SAC'2005)*, Santa Fe, New Mexico, USA. ACM Press. 21

[Burkhard & Keller, 1973] Burkhard, W. A. e Keller, R. M. (1973). Some approaches to best-match file searching. *Communications of the ACM*, 16(4):230–236. 34, 43

[Böhm & Kriegel, 2000] Böhm, C. e Kriegel, H.-P. (2000). Dynamically optimizing high-dimensional index structures. In Zaniolo, C., Lockermann, P. C., Scholl, M. H., e Grust, T., editors, *7th International Conference on Extending Database Technology (EDBT)*, v. 1777 of *Lecture Notes in Computer Science*, pp. 36–50, Konstanz, Germany. Springer Verlag. ISBN 3-540-67227-3. 29

[Cabrera, 2002] Cabrera, A. (2002). Defining the role of a pacs technologist. *Journal of Digital Imaging*, 15(1):120–123. 3

[Campo & Traina, 2003] Campo, C. Y. e Traina, A. J. M. (2003). Uma abordagem eficiente para recuperação de imagens médicas. In *Workshop de Informática Médica*. 16

[Carro et al., 2003] Carro, S. A., Scharcanski, J., e de Lima, J. V. (2003). Mediseek: A web based diffusion system for medical visual information. In *WIDM 03*, New Orleans, Louisiana, USA. 17

[Castañón & Traina, 2002] Castañón, C. A. B. e Traina, A. J. M. (2002). Extração de características de imagens médicas baseadas na distribuição de cor dos espaços de wavelets. In *2º Workshop de Informática Médica ? WIM?2002 - Simpósio Brasileiro de Engenharia de Software (SBES) da Sociedade Brasileira de Computação*. 4 pags em CDROM, p., Gramado, RS. 32

[Cha, 2004] Cha, G.-H. (2004). Efficient and flexible bitmap indexing complex similarity queries. In et al., Y. L., editor, *DASFAA*, v. 2973 of *Lecture Notes in Computer Science*, pp. 708–720. Springer-Verlag Berlin Heidelberg. 45

[Chabat et al., 2000] Chabat, F., Hansell, D. M., e Yang, G.-Z. (2000). Computerized decision support in medical imaging - challenges in using image processing and automated feature extraction for improving diagnostic accuracy. *IEEE Engineering in Medicine and Biology*. pp. 89–96. 13, 14, 16

[Chahir & Chen, 1998] Chahir, Y. e Chen, L. (1998). Efficient content-based image retrieval based on color homogeneus objects segmentation and their spatial relationship characterization. In *IEEE International Conference on Multimedia Computing and Systems*. 14

[Chakrabarti, 1999] Chakrabarti, K. (1999). *Supporting Spatial Index Structures as Access Methods in a Database System*. PhD thesis, University of Illinois. 41

[Chakrabarti & Mehrotra, 2000] Chakrabarti, K. e Mehrotra, S. (2000). Local dimensionality reduction: A new approach to indexing high dimensional spaces. In Abbadi, A. E., Brodie, M. L., Chakravarthy, S., Dayal, U., Kamel, N., Schlageter, G., e Whang, K.-Y., editors, *26th International Conference on Very Large Databases (VLDB'2000)*, pp. 89–100, Cairo, Egypt. Morgan Kaufmann. 33

[Chang et al., 1998] Chang, W., Sheikholeslami, G., Wang, J., e Zhang, A. (1998). Data resource selection in distributed visual information systems. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 10(6):926–946. 18

[Chaudhuri et al., 2005] Chaudhuri, S., Ramakrishnan, R., e Weikum, G. (2005). Integrating db and ir technologies: What is the sound of one hand clapping? In *CIDR 2005, Second Biennial Conference on Innovative Data Systems Research*, pp. 1–12, Asilomar, CA, USA. 2

[Chazelle, 1994] Chazelle, B. (1994). Computational geometry: a retrospective. In *STOC '94: Proceedings of the twenty-sixth annual ACM symposium on Theory of computing*, pp. 75–94. ACM Press. 26

[Chiu et al., 2003] Chiu, C.-Y., Lin, H.-C., e Yang, S.-N. (2003). A fuzzy logic cbir system. In *The IEEE International Conference on Fuzzy Systems*, pp. 1171–1176. 21

[Chomicki, 2003] Chomicki, J. (2003). Preference formulas in relational queries. *ACM Transactions on Database Systems (TODS)*, 28(4):427–466. 49

[Chua et al., 1999] Chua, T.-S., Chu, C.-X., e Kankanhalli, M. (1999). Relevance feedback techniques for image retrieval using multiple attributes. In *IEEE International Conference on Multimedia Computing and Systems Volume I*, v. Volume 1, Florence, Italy. 22

[Chávez & Navarro, 2000] Chávez, E. e Navarro, G. (2000). Measuring the dimensionality of general metric spaces. Technical Report TR/DCC-00-1, Dept. of Computer Science, Univ. of Chile. 28

[Chávez et al., 2001] Chávez, E., Navarro, G., Baeza-Yates, R., e Marroquín, J. L. (2001). Searching in metric spaces. *ACM Computing Surveys*, 33(3):273–321. 18, 34, 43

[Ciaccia et al., 2000] Ciaccia, P., Montesi, D., Penzo, W., e Trombetta, A. (2000). Imprecision and user preferences in multimedia queries: A generic algebraic approach. In K.-D. Schewe, B. T., editor, *FoIKS 2000*, v. 1762 of *Lecture Notes in Computer Science*, pp. 50–71, Berlin Heidelberg. Springer-Verlag. 49

[Ciaccia & Patella, 2001] Ciaccia, P. e Patella, M. (2001). Approximate similarity queries: A survey. Technical Report CSITE-08-01, Department of Electronics, Computer Science and Systems - University of Bologna. 8

[Ciaccia & Patella, 2002] Ciaccia, P. e Patella, M. (2002). Searching in metric spaces with user-defined and approximate distances. *ACM Transactions on Database Systems (TODS)*, 27(4):398–437. 44

[Ciaccia et al., 1997] Ciaccia, P., Patella, M., e Zezula, P. (1997). M-tree: An efficient access method for similarity search in metric spaces. In Jarke, M., editor, *23th International Conference on Very Large Databases (VLDB'1997)*, pp. 426–435, Athens, Greece. 34, 44

[Cinque et al., 1998] Cinque, L., Levialdit, S., Olsen, K. A., e A., P. (1998). Color-based image retrieval using spatial-chromatic histograms. In *IEEE International Conference on Multimedia Computing and Systems*. 14

[Comaniciu et al., 1998a] Comaniciu, D., Meer, P., e Foran, D. (1998a). Shape-based image indexing and retrieval for diagnostic pathology. In *Fourteenth International Conference on Pattern Recognition*, v. 1, pp. 902–904, Brisbane, Qld. Australia. 15

[Comaniciu et al., 1998b] Comaniciu, D., Meer, P., Foran, D., e Medl, A. (1998b). Bimodal system for interactive indexing and retrieval of pathology images. In *Fourth IEEE Workshop on Applications of Computer Vision (WACV '98)*, pp. 76 – 81. 14

[Deb & Zhang, 2004] Deb, S. e Zhang, Y. (2004). An overview of content-based image retrieval techniques. In *18th International Conference on Advanced Information Networking and Application (AINA 04)*. IEEE Computer Society. 8

[Digout et al., 2004] Digout, C., Nascimento, M. A., e Coman, A. (2004). Similarity search and dimensionality reduction: Not all dimensions are equally useful. In et al., Y. L., editor, *DASFAA2004*, v. 2973 of *Lecture Notes in Computer Science*, pp. 831–842. Springer- Berlin Heidelberg. 45

[Distasi et al., 2003] Distasi, R., Nappi, M., e Tucci, M. (2003). Fire: Fractal indexing with robust extensions for image databases. *IEEE Transactions on Image Processing (TIP)*, 12(3):373–384. 16

[Djeraba, 2003] Djeraba, C. (2003). Association and content-based retrieval. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 15(1):118–135. 18

[Djeraba et al., 1998] Djeraba, C., Bouet, M., e Briand, H. (1998). Concept-based query in visual information systems. In *IEEE International Forum on Research and Technology Advances in Digital Libraries*, pp. 299 – 308, Santa Barbara, CA USA. 18

[Dy et al., 2003] Dy, J. G., Brodley, C. E., Broderick, L. S., e Aisen, A. M. (2003). Unsupervised feature selection applied to content-based retrieval of lung images. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 25(3):373–378. 23

[Eisenberg & Melton, 1999] Eisenberg, A. e Melton, J. (1999). Sql-1999, formerly known as sql3. *SIGMOD Record*, 28(1):131–138. 52

[Eiter et al., 2000] Eiter, T., Lukasiewicz, T., e Walter, M. (2000). Extension of the relational algebra to probabilistic complex values. In K.-D. Schewe, B. T., editor, *FoIKS 2000*, v. 1762 of *Lecture Notes in Computer Science*, pp. 94–115, Berlin Heidelberg. Springer-Verlag. 2

[El-Naqa et al., 2004] El-Naqa, I., Yang, Y., Galatsanos, N. P., Nishikawa, R. M., e Wernick, M. N. (2004). A similarity learning approach to content-based image retrieval: Application to digital mammography. *IEEE Transactions on Medical Imaging (TMI)*, 23(10):1233–1244. 17

[Enser & Sandom, 2003] Enser, P. e Sandom, C. (2003). Towards a comprehensive survey of the semantic gap in visual image retrieval. In et al., E. M. B., editor, *CIVR 2003*, v. 2728 of *Lecture Notes in Computer Science*, pp. 291–299, Berlin Heidelberg. Springer-Verlag. 22

[Faloutsos, 1996] Faloutsos, C. (1996). *Searching Multimedia Databases by Content*, v. 3 of *The Kluwer International Series on Advances in Database Systems*. Kluwer Academic Publishers, Boston, MA. 30

[Faloutsos & Lin, 1995] Faloutsos, C. e Lin, K.-I. D. (1995). Fastmap: A fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets. In Carey, M. J. e Schneider, D. A., editors, *ACM International Conference on Management of Data (SIGMOD'1995)*, pp. 163–174. San Jose, California. ACM Press. 32

[Faloutsos et al., 2000] Faloutsos. C., Seeger, B., Traina, A. J. M., e Traina, Caetano, J. (2000). Spatial join selectivity using power laws. In *ACM International Conference on Management of Data (SIGMOD'2000)*, pp. 177–188, Dallas, TX. ACM Press. 29

[Fauqueur & Boujemaa, 2003] Fauqueur, J. e Boujemaa, N. (2003). New image retrieval paradigm: Logical, composition of region categories. In *International Conference on Image Processing*, v. 3, pp. III - 601–4 vol.2. 15

[Felipe et al., 2003] Felipe, J. C., Traina, A. J. M., e Traina Jr., C. (2003). Retrieval by content of medical images using texture for tissue identification. In *16th IEEE Symposium on Computer-based Medical Systems (CBMS'2003)*, pp. 26–27, New York. IEEE Computer Society. 14, 16

[Figueiredo et al., 2004] Figueiredo, J. M. d., Traina Jr., C., Traina, A. J. M., e Marques, P. M. d. A. (2004). Flexibilizando buscas por conteúdo em ambientes paes. In *IX Congresso Brasileiro de Informática em Saúde (CBIS'2004)*, pp. 6 p. in CD ROM, Ribeirão Preto, SP, Brazil. Sociedade Brasileira de Informática em Saúde - SBIS. 92

[Filho et al., 1999] Filho, R. F. S., Traina, A. J. M., e Traina, Caetano, J. (1999). Indexando pontos em espaços com altas dimensões através dos métodos de indexação espacial k-d-b-tree, r-tree e tv-tree - uma breve comparação analítica. Relatório Técnico 86, ICMC. 41

[Fu et al., 2000] Fu, A. W.-c., Chan, P. M.-s., Cheung, Y.-L., e Moon, Y. S. (2000). Dynamic vp-tree indexing for n-nearest neighbor search given pair-wise distances. *The VLDB Journal*, 9(2):154–173. 43

[Gaede & Günther, 1998] Gaede, V. e Günther, O. (1998). Multidimensional access methods. *ACM Computing Surveys*, 30(2):170–231. 18, 41

[Gerber & Fernandes, 2004] Gerber, L. e Fernandes, A. A. (2004). An abstract algebra for knowledge discovery in databases. In Benczúr, A., Demetrovics, J., e Gottlob, G., editors, *ADBIS 2004*, v. 3255 of *Lecture Notes in Computer Science*, pp. 83–98, Berlin Heidelberg. Springer-Verlag. 2

[Glatard et al., 2004] Glatard, T., Montagnat, J., e Magnin, I. E. (2004). Texture based medical image indexing and retrieval: Application to cardiac imaging. In *6th ACM SIGMM international workshop on Multimedia information retrieval (MIR '2004)*, pp. 135–142, New York. New York, USA. ACM Press. 14, 15

[Grgic et al., 2003] Grgic, M., Grgic, S., e Ghanbari, M. (2003). Large image database retrieval based on texture features. In *IEEE International Conference on Industrial Technology (ICIT2003)*, v. 2. pp. 959–964, Maribor, Slovenia. IEEE. 14, 18

[Group, 1999] Group, O. G. M. W. (1999). Opengis - simple feature specification for sql. revision 1.1. 2

[Guldogan et al., 2003] Guldogan, E., Guldogan, O., Kiranyaz, S., Caglar, K., e Gabbouj, M. (2003). Compression effects on color and texture based multimedia indexing and retrieval. In *International Conference on Image Processing*, v. 2, pp. II – 9–12 vol.3. 15

[Guttman, 1984] Guttman, A. (1984). R-tree : A dynamic index structure for spatial searching. In *ACM International Conference on Management of Data (SIGMOD'1984)*. pp. 47–57, Boston, MA. ACM PRess. 42

[Güld et al., 2002] Güld, M., Kohnen, M., Keysers, D., Schubert, H., Wein, B., Bredno, J., e Lehmann, T. (2002). Quality of dicom header information for image categorization. In Siegel, E. L. e Huang, H. K., editors, *Medical Imaging 2002: PACS and Integrated Medical Information Systems: Design and Evaluation*, v. 4685 of *Procs SPIE*, pp. 280–287. 13, 20

[Hellier, 2003] Hellier, P. (2003). Consistent intensity correction of mr images. In *IEEE International Conference on Image Processing (ICIP'2003)*, v. 1, pp. I – 1109–12 vol.1, Barcelona, Spain. IEEE. 14

[Hinneburg et al., 2000] Hinneburg, A., Aggarwal, C. C., e Keim, D. A. (2000). What is the nearest neighbor in high dimensional spaces? In El Abbadi, A., Brodie, M. L., Chakravarthy, S., Dayal, U., Kamel, N., Schlageter, G., e Whang, K.-Y., editors, *26th International Conference on Very Large Databases (VLDB'2000)*, pp. 506–515, Cairo - Egypt. Morgan Kaufmann. 37

[Hiransakolwong et al., 2003] Hiransakolwong, N., Hua, K., Vu, K., e Windyga, P. (2003). Segmentation of ultrasound liver images: an automatic approach. In *International Conference on Multimedia and Expo (ICME '03)*, v. 1, pp. I– 573–6 vol.1. 2003 Pages:. 14

[Hjaltason & Samet, 2003a] Hjaltason, G. R. e Samet, H. (2003a). Index-driven similarity search in metric spaces. *ACM Transactions on Database Systems (TODS)*, 28(4):517 580. 43

[Hjaltason & Samet, 2003b] Hjaltason, G. R. e Samet, H. (2003b). Properties of embedding methods for similarity search in metric spaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 25(5):530–549. 32

[Hludov & Meinel, 1999] Hludov, S. e Meinel, C. (1999). Dicom - image compression. In *12 th IEEE Symposium on Computer-Based Medical System (CBMS'1999)*, pp. 282 287, Stamford, Connecticut. IEEE Computer Society. 3

[Hristescu & Farach-Colton, 2000] Hristescu, G. e Farach-Colton, M. (2000). Cofe: A scalable method for feature extraction from complex objects. In Kambayashi, Y., Mohania, M. K., e Tjoa, A. M., editors, *2nd International Conference on Data Warehousing and Knowledge Discovery (DaWaK 2000)*, v. 1874 of *Lecture Notes in Computer Science*, pp. 358 371, Greenwich, U.K. Springer. 32

[Huang et al., 1997] Huang, J., Kumar, S. R., Mitra, M., Zhu, W.-J., e Zabih, R. (1997). Image indexing using color correlograms. In IEEE, editor, *Conference on Computer Vision and Pattern Recognition (CVPR '97)*, Puerto Rico. IEEE. 14

[Huang et al., 1998] Huang, J., Kumar, S. R., e Zabih, R. (1998). An automatic hierarchical image classification scheme. In *MULTIMEDIA '98: Proceedings of the sixth ACM international conference on Multimedia*, pp. 219–228. ACM Press. 18

[Huang & Lee, 2004] Huang, P.-W. e Lee, C.-H. (2004). Image database design based on 9d-spa representation for spatial relations. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 16(12):1486. 15

[Huang et al., 2000] Huang, Q., Puri, A., e Liu, Z. (2000). Multimedia search and retrieval: new concepts, system implementation, and application. *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)*, 10(5):679 692. 8

[Huijsmans & Sebe, 2005] Huijsmans, D. e Sebe, N. (2005). How to complete performance graphs in content-based image retrieval: Add generality and normalize scope. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 27(2):245 251. 22

[Huijsmans & Sebe, 2003] Huijsmans, D. P. e Sebe, N. (2003). Content-based indexing performance: Size normalized precision, recall, generality evaluation. In *International Conference on Image Processing*, v. 3, pp. III– 733–6 vol.2. ISSN: 1522-4880. 24

[Hull, 1994] Hull, D. A. (1994). Improving text retrieval for the routing problem using latent semantic indexing. In *17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, pp. 282–291, Dublin, Ireland. 32

[Indyk, 2000] Indyk, P. (2000). *High-Dimensional Computational Geometry*. PhD thesis, Department of Computer Science of Stanford University. 27, 37

[Jin et al., 2003] Jin, H., Ooi, B. C., Shen, H. T., Yu, C., e Zhou, A. (2003). An adaptive and efficient dimensionality reduction algorithm for high-dimensional indexing. In *19th International Conference on Data Engineering (ICDE'2003)*, pp. 75–86. IEEE. 35

[Jörgensen, 1999] Jörgensen, C. (1999). Access to pictorial material: A review of current research and future prospects. *Computers and the Humanities*, 33:293–318. 8

[Kahveci & Singh, 2001] Kahveci, T. e Singh, A. K. (2001). An efficient index structure for string databases. In *27th International Conference on Very Large Databases (VLDB'2001)*, pp. 351–360, Roma, Italy. Morgan Kaufmann. 37

[Kailing et al., 2004] Kailing, K., Kriegel, H.-P., Schönauer, S., e Seidl, T. (2004). Efficient similarity search for hierarchical data in large databases. In Bertino, E., Christodoulakis, S., Plexousakis, D., Christophides, V., Koubarakis, M., Böhm, K., e Ferrari, E., editors, *9th International Conference on Extending Database Technology (EDBT'2004)*, v. 2992 of *Lecture Notes in Computer Science*, pp. 676–693, Heraklion, Crete, Greece. Springer Verlag. 54

[Kamel & Faloutsos, 1994] Kamel, I. e Faloutsos, C. (1994). Hilbert r-tree: An improved r-tree using fractals. In Bocca, J. B., Jarke, M., e Zaniolo, C., editors, *20th International Conference on Very Large Databases (VLDB'1994)*, pp. 500–509, Santiago del Chile, Chile. Morgan Kaufmann. 29

[Kanth et al., 1999] Kanth, K. V. R., Agrawal, D., El Abbadi, A., e Singh, A. K. (1999). Dimensionality reduction for similarity searching in dynamic databases. *Computer Vision and Image Understanding*, 75(1/2):59–72. Elaine Josiel. 31

[Karypis & Han, 2000] Karypis, G. e Han, E. (2000). Concept indexing - a fast dimensionality reduction algorithm with applications to document retrieval & categorization. Technical Report 00-016, University of Minnesota, Department of Computer Science / Army HPC Research Center Minneapolis;. 32

[Katayama & Satoh, 1997] Katayama, N. e Satoh, S. (1997). The sr-tree: An index structure for high-dimensional nearest neighbor queries. In *ACM International Conference on Management of Data (SIGMOD'1997)*, pp. 369–380, Tucson, Arizona, USA. ACM Press. 42

[Katayama & Satoh, 2001] Katayama, N. e Satoh, S. (2001). Distinctiveness-sensitive nearest neighbor search for efficient similarity retrieval of multimedia information. In *17th International Conference on Data Engineering (ICDE'2001)*, pp. 493–502, Heidelberg, Germany. IEEE Computer Society. 34

[Ke et al., 2004] Ke, Y., Sukthankar, R., e Huston, L. (2004). An efficient parts-based near-duplicate and sub-image retrieval system. In *12th ACM International Conference on Multimedia (Multimedia'2004)*, pp. 869–876, New York, NY, USA. ACM Press. 38

[Kelly & Cannon, 1994] Kelly, P. e Cannon, T. (1994). Candid: comparison algorithm for navigating digital image databases. In *Seventh International Working Conference on Scientific and Statistical Database Management*, pp. 28–30, Charlottesville, VA USA. 14

[Keogh et al., 2001a] Keogh, E., Chakrabarti, K., Pazzani, M. J., e Mehrotra, S. (2001a). Dimensionality reduction for fast similarity search in large time series databases. *Knowledge and Information Systems Journal*, 3(3):263–286. 32

[Keogh et al., 2001b] Keogh, E. J., Chakrabarti, K., Mehrotra, S., e Pazzani, M. J. (2001b). Locally adaptive dimensionality reduction for indexing large time series databases. In *ACM International Conference on Management of Data (SIGMOD'2001)*. 32

[Kherfi et al., 2004] Kherfi, M. L., Ziou, D., e Bernardi, A. (2004). Image retrieval from the world wide web: Issues, techniques, and systems. *ACM Computing Surveys*, 36(1):35–67. 8

[Kiranyaz et al., 2003] Kiranyaz, S., Caglar, K., Guldogan, E., Guldogan, O., e Gabbouj, M. (2003). Muvis: A content-based multimedia indexing and retrieval framework. In *Seventh International Symposium on Signal Processing and its Applications (ISSPA'2003)*, v. 1, pp. 1–8, Paris, France. 2

[Kokare et al., 2003] Kokare, M., Chatterji, B., e Biswas, P. (2003). Comparison of similarity metrics for texture image retrieval. In *TENCON 2003. Conference on Convergent Technologies for Asia-Pacific Region*, v. 2, pp. 571–575. 20, 38

[Korn et al., 2001] Korn, F., Pagel, B.-U., e Faloutsos, C. (2001). On the 'dimensionality curse' and the 'self-similarity blessing'. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 13(1):96 111. 29

[Koskela et al., 2000] Koskela, M., Laaksonen, J., Laakso, S., e Oja, E. (2000). Evaluating the performance of content-based image retrieval systems. In Laurini, R., editor. *VISUAL 2000*, v. 1929 of *Lecture Notes in Computer Science* 430 441, p. Springer-Verlag Berlin Heidelberg. 8

[Käster et al., 2003] Käster, T., VolkerWendt, e Sagerer, G. (2003). Comparing clustering methods for database categorization in image retrieval. In Michaelis, B. e Krell, G., editors, *DAGM 2003*, v. 2781 of *Lecture Notes in Computer Science*, pp. 228–235, Berlin Heidelberg. Springer-Verlag. 36

[Langsam et al., 1996] Langsam, Y., Tenenbaum, A., e Augenstein, M. (1996). *Data Structures using C and C++*. Prentice Hall, 2th edition. 40

[Ledley, 1987] Ledley, R. S. (1987). Medical informatics: a personal view of sowing the seeds. In *Proceedings of ACM conference on History of medical informatics*, pp. 31 41. ACM Press. 2

[Lee & Kim, 2001] Lee, D.-H. e Kim, H.-J. (2001). A fast content-based indexing and retrieval technique by shape information in large image database. *The Journal of Systems and Software*, 56:165 182. 14

[Lee, 2003] Lee, K.-M. (2003). Neural network-generated image retrieval and refinement. In Nürnberger, A. e Detyniecki, M., editors, *AMR 2003*, v. 3094 of *Lecture Notes in Computer Science*, pp. 200 211, Berlin Heidelberg. Springer-Verlag. 20

[Lehmann et al., 2003a] Lehmann, T. M., Guld, M. O., Thies, C., Fischer, B., Keysers, D., Kohnen, M., Schubert, H., e Wein, B. B. (2003a). Content-based image retrieval in medical applications for picture archiving and communication systems. In Huang, H. K. e Ratib, O. M., editors. *Proceedings of SPIE, Medical Imaging 2003: PACS and Integrated Medical Information Systems: Design and Evaluation*, v. 5033, pp. 109 117. 8, 10

[Lehmann et al., 2004] Lehmann, T. M., Güld, M. O., Keysers, D., Deselaers, T., Schubert, H., Wein, B., e Spitzer, K. (2004). Similarity of medical images computed from global feature vectors for content-based retrieval. In Negoita, M. G., Howlett, R. J., e Jain, L. C., editors, *8th International Conference on Knowledge-Based Intelligent Information and Engineering Systems (KES 2004)*, v. 3214 of *Lecture Notes in Computer Science*, pp. 989 995, Wellington, New Zealand. Springer-Verlag. 17

[Lehmann et al., 2003b] Lehmann, T. M., Schubert, H., Keysers, D., Kohnen, M., e Wein, B. B. (2003b). The irma code for unique classification of medical images. In Huang, H. K. e Ratib, O. M., editors, *Proceedings of SPIE , Medical Imaging 2003: PACS and Integrated Medical Information Systems: Design and Evaluation*, v. 5033, 440-451. 17

[Lehmann et al., 2000] Lehmann, T. M., Weinb, B., Dahmenc, J., Brednoa, J., Vogelsangb, F., e Kohnenb, M. (2000). Content-based image retrieval in medical applications: A novel multi-step approach. In Yeung, M. M., Yeo, B.-L., e Bouman, C. A., editors, *Storage and Retrieval for Media Databases 2000*, v. 3972, pp. 312 320. 13, 17

[Levine et al., 2003] Levine, B. A., Mun, S. K., Benson, H. R., e Horii, S. C. (2003). Assessment of the integration of a his/ris with a pacs. *Journal of Digital Imaging*, 16(1):133 140. ISSN 1618-727X. 51

[Li et al., 2003] Li, B., Chang, E., e Wu, Y. (2003). Discovery of a perceptual distance function for measuring image similarity. *Multimedia Systems - Springer-Verlag*, 8:512 - 522. 38

[Liapis & Tziritas, 2004] Liapis, S. e Tziritas, G. (2004). Color and texture image retrieval using chromaticity histograms and wavelet frames. *IEEE Transactions on Multimedia (TM)*, 6(5):676 686. 14, 15

[Lin et al., 1994] Lin, K.-I., Jagadish, H. V., e Faloutsos, C. (1994). The tv-tree: An index structure for high-dimensional data. *VLDB Journal*, 3(4):517-542. 42

[Loncaric, 1998] Loncaric, S. (1998). A survey of shape analysis techniques. *Pattern Recognition*, 31(8):983 1001. 14

[Lopes, 2005] Lopes, O. C. (2005). iris server: An environment for image retrieval by content. Dissertação de mestrado, Instituto de Ciências Matemáticas e de Computação - USP. 5, 74

[Lu, 2002] Lu, G. (2002). Techniques and data structures for efficient multimedia retrieval based on similarity. *IEEE Transactions on Multimedia*, 4(3):372 384. 41

[Lu et al., 2004] Lu, K., He, X., e Zeng, J. (2004). Image retrieval using dimensionality reduction. In Zhang, J., He, J.-H., e Fu, Y., editors, *CIS 2004*, v. 3314 of *Lecture Notes in Computer Science*, pp. 775 781. Springer-Verlag Berlin Heidelberg. 32

[Luo & Nascimento, 2003] Luo, J. e Nascimento, M. A. (2003). Content based sub-image retrieval via hierarchical tree matching. In *MMDB'2003*, New Orleans, Louisiana, USA. ACM. 14

[Malinchik & Bonabeau, 2004] Malinchik, S. e Bonabeau, E. (2004). Exploratory data analysis with interactive evolution. In et al., K. D., editor, *GECCO'2004*, v. 3103 of *Lecture Notes in Computer Science*, pp. 1151 1161, Berlin Heidelberg. Springer-Verlag. 38

[Marques et al., 2004] Marques, P. M. A., Carita, E. C., Benedicto, A. A., e Sanches, P. R. (2004). Integrating ris/pacs: The web-based solution at university hospital of ribeirão preto, brazil. *Journal of Digital Imaging*, 17(3):226 233. 51

[Marques et al., 2002] Marques, P. M. d. A., Honda, M. H., Rodrigues, J. A. H., Santos, R. R. d., Traina, A. J. M., Jr., T., e Caetano (2002). Recuperação de imagens baseada em conteúdo: Uso de atributos de textura para caracterização de microcalcificações mamográficas. *Revista Brasileira de Radiologia*, 35(2):93–98. 14, 15

[Marques et al., 2000] Marques, P. M. d. A., Santos, R. R. d., Traina, A. J. M., Traina, Caetano, J., e Bueno, J. M. (2000). Image retrieval based on texture content. In *World Congress on Medical Physics and Biomedical Engineering (WC'2000)*, Chicago. 14

[Marsicoi et al., 1997] Marsicoi, M. D., Cinque, L., e Levialdi, S. (1997). Indexing pictorial documents by their content: a survey of current techniques. *Image Vision Comput.*, 15(2):119 141. 8

[Martinez & Marchand, 1998] Martinez, J. e Marchand, S. (1998). Towards intelligent retrieval in images databases. In *International Workshop on Multi-Media Database Management Systems (IW-MMDBMS)*, pp. 38 45, Dayton, Ohio. IEEE Computer Society. 8

[McDonald et al., 2001] McDonald, S., Lai, T.-S., e Tait, J. (2001). Evaluating a content based image retrieval system. In *24th ACM International Conference on Research and Development in Information Retrieval (SIGIR'2001)*, pp. 232 240, New Orleans, Louisiana, USA. 23

[McDonald & Tait, 2003] McDonald, S. e Tait, J. (2003). Search strategies in content-based image retrieval. In *26th ACM International Conference on Research and Development in Information Retrieval (SIGIR'2003)*, pp. 80 87, Toronto, Canada. ACM. 20

[Meilhac & Nastar, 1999] Meilhac, C. e Nastar, C. (1999). Relevance feedback and category search in image databases. In *IEEE International Conference on Multimedia Computing and Systems (ICMCS'1999)*, v. 1, Florence, Italy. 22

[Miao et al., 2004] Miao, Y., Wang, Y., e Miao, Y. (2004). The research of semantic content applied to brain ct images. In *17th IEEE Symposium on Computer-Based Medical Systems (CBMS'2004)*. IEEE Computer Society. 23

[Mildenberger et al., 2002] Mildenberger, P., Eichelberg, M., e Martin, E. (2002). Introduction to dicom standard. *European Radiology*, 12:920–927. 3

[Mitra et al., 2002] Mitra, P., Murthy, C. A., e K.Pal, S. (2002). Unsupervised feature selection using feature similarity. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 24(3):301–312. 30

[Müller et al., 2004] Müller, H., Michoux, N., Bandon, D., e Geissbuhler, A. (2004). A review of content-based image retrieval systems in medical applications - clinical benefits and future directions. *International Journal of Medical Informatics*, 73:1 – 23. 8, 54

[Müller et al., 2000] Müller, W., Müller, H., Marchand-Maillet, S., Pun, T., Squire, D. M., Pecenovic, Z., Giess, C., e de Vries, A. P. (2000). Mrml: A communication protocol for content-based image retrieval. In Laurini, R., editor, *VISUAL 2000*, v. 1929 of *Lecture Notes in Computer Science*, pp. 300–311. Springer-Verlag Berlin Heidelberg. 13

[Nastar et al., 1998] Nastar, C., Mitschke, M., e Meilhac, C. (1998). Efficient query refinement for image retrieval. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'1998)*, pp. 547 – 552, Santa Barbara, CA USA. IEEE. 22

[Natsev et al., 2004] Natsev, A., Rastogi, R., e Shim, K. (2004). Walrus: A similarity retrieval algorithm for image databases. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 16(3):301–316. 38

[Nes & Kersten, 1998] Nes, N. e Kersten, M. (1998). The acoi algebra: A query algebra for image retrieval systems. In et al., S. E., editor, *Advances in Databases (BNCOD'1998)*, v. 1405 of *Lecture Notes in Computer Science*, pp. 77 – 88, Berlin Heidelberg. Springer-Verlag. 49

[Ng & Sedighian, 1994] Ng, R. T. e Sedighian, A. (1994). Evaluating multi-dimensional indexing structures for images transformed by principle component analysis. In *SPIE*, pp. 50–61. Pmt. of the SPIE, 2670:, 1994. 31

[Ngu et al., 2001] Ngu, A. H., Sheng, Q. Z., Huynh, D. Q., e Lei, R. (2001). Combining multi-visual features for efficient indexing in a large image database. *The VLDB Journal*, 9:279 – 293. 29, 32

[Ooi et al., 2003] Ooi, B. C., Shen, H., e Xia, C. (2003). Towards efficient image retrieval based on multiple features. In *ICICS-PCM 2003*, pp. 180 –185, Singapore. IEEE. 29

[Park et al., 1997] Park, I. K., Yun, I. D., e Lee, S. U. (1997). Models and algorithms for efficient color image indexing. In *Workshop on Content-Based Access of Image and Video Libraries (CBAIVL'1997)*. IEEE. 14

[Parsons et al., 2004] Parsons, L., Haque, E., e Liu, H. (2004). Subspace clustering for high dimensional data: a review. *SIGKDD Explorations*, 6(1):90–105. 33

[Patella, 1999] Patella, M. (1999). *Similarity Search in Multimedia Databases*. PhD thesis, Dipartimento di Elettronica Informatica e Sistemistica, Università degli Studi di Bologna. 8

[Payne et al., 1999] Payne, J. S., Hepplewhite, L., e Stonham, T. J. (1999). Perceptually based metrics for the evaluation of textural image retrieval methods. In *IEEE International Conference on Multimedia Computing and Systems (ICMCS'1999)*, v. 2, pp. 793–797, Florence, Italy. IEEE Computer Society. 14, 15

[Petrakis, 2002] Petrakis, E. G. (2002). Fast retrieval by spatial structure in image databases. *Journal of Visual Languages and Computing*, 13(5):545–569. 18

[Petrakis & Milios, 1999] Petrakis, E. G. M. e Milios, E. (1999). Efficient retrieval by shape content. In *IEEE International Conference on Multimedia Computing and Systems (ICMCS'1999)*, v. 2, pp. 616–621, Florence, Italy,. IEEE Computer Society. 14

[Power et al., 2004] Power, D., Politou, E., Slaymaker, M., Harris, S., e Simpson, A. (2004). A relational approach to the capture of dicom files for grid-enabled medical imaging databases. In *ACM Symposium on Applied Computing (SAC'2004)*, pp. 272–279. 19

[Puuronen et al., 2000] Puuronen, S., Tsymbal, A., e Skrypnyk, I. (2000). Advanced local feature selection in medical diagnostics. In *13th IEEE Symposium on Computer-Based Medical Systems (CBMS'2000)*, pp. 25–, Houston, TX, USA. IEEE Computer Society. 17

[Qian et al., 2004] Qian, G., Sural, S., Gu, Y., e Pramanik, S. (2004). Similarity between euclidean and cosine angle distance for nearest neighbor queries. In *ACM Symposium on Applied Computing (SAC'2004)*, pp. 1232–1237, Nicosia, Cyprus. ACM Press. 37

[Rao et al., 1999] Rao, A., Srihari, R. K., e Zhang, Z. (1999). Spatial color histograms for content-based image retrieval. In *11th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'1999)*, pp. 183–186, Chicago, Illinois, USA. 14

[Rautkorpi & Iivarinen, 2004] Rautkorpi, R. e Iivarinen, J. (2004). A novel shape feature for image classification and retrieval. In Campilho, A. C. e Kamel, M. S., editors,

International Conference on Image Analysis and Recognition (ICIAR'2004), v. 3211 of *Lecture Notes in Computer Science*, pp. 753–760, Porto, Portugal. Springer-Verlag Berlin Heidelberg. 14, 15

[Robinson, 1981] Robinson, J. T. (1981). The k-d-b-tree: A search structure for large multidimensional dynamic indexes. In Lien, Y. E., editor, *ACM International Conference on Management of Data (SIGMOD'1981)*, pp. 10–18, SIGMOD Conference 1981:. ACM Press. 41

[Roddick et al., 2003] Roddick, J. F., Hornsby, K., e Vries, D. d. (2003). A unifying semantic distance model for determining the similarity of attribute values. In Oudshoorn, M., editor, *Twenty-Sixth Autralasian Computer Science Conference (ACSC'2003) - Conferences in Research and Practice in Information Technology*, v. 16, Adelaide, Australia. Australian Computer Society, Inc. 50

[Rosa et al., 2002] Rosa, N. A., R. F. Santos, F., Bueno, J. M., Traina, A. J. M., e Jr., C. T. (2002). Sistema de recuperação de imagens similares em um hospital universitário. In *VIII Congresso Brasileiro de Informática em Saúde (CBIS'2002)*, Natal - RN. 5

[Ross et al., 2001] Ross, K. A., Sitzmann, I., e Stuckey, P. J. (2001). Cost-based unbalanced r-trees. In *Thirteen International Conference on Scientific and Statistics Database Management*, Fairfax, Virginia - USA. 42

[Rui et al., 1997] Rui, Y., Huang, T., e Mehrotra, S. (1997). Content-based image retrieval with relevance feedback in mars. In *International Conference on Image Processing (ICIP'1997)*, v. Volume 2, Washington, DC. 22

[Rui et al., 1999] Rui, Y., Huang, T. S., e Chang, S.-F. (1999). Image retrieval: Current techniques, promising directions and open issues. *Journal of Visual Communication and Image Representation*, 10:39–62. 8

[Rummukainen et al., 2003] Rummukainen, M., Laaksonen, J., e Koskela, M. (2003). An efficiency comparison of two content-based image retrieval systems, gift and picsom. In et al., E. M. B., editor, *CIVR'2003*, v. 2728 of *Lecture Notes in Computer Science*, pp. 500–510. Springer-Verlag Berlin Heidelberg. 8

[Sakurai et al., 2000] Sakurai, Y., Yoshikawa, M., Uemura, S., e Kojima, H. (2000). The a-tree: An index structure for high-dimensional spaces using relative approximation. In El Abbadi, A., Brodie, M. L., Chakravarthy, S., Dayal, U., Kamel, N., Schlageter, G., e Whang, K.-Y., editors, *26th International Conference on Very Large Databases (VLDB'2000)*, pp. 516–526, Cairo - Egypt. Morgan Kaufmann. 42

[Santini et al., 2001] Santini, S., Gupta, A., e Jain, R. (2001). Emergent semantics through interaction in image databases. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 13(3):337 351. 48

[Santos et al., 2001] Santos, Roberto Figueira, F., Traina, A. J. M., Traina, Caetano, J., e Faloutsos, C. (2001). Similarity search without tears: The omni family of all-purpose access methods. In *International Conference on Data Engineering (ICDE'2001)*, pp. 623–630, Heidelberg, Germany. IEEE Computer Society. 45

[Saykol et al., 2004] Saykol, E., Güdükbay, U., e Özgür Ulusoy (2004). Integrated querying of images by color, shape, and texture content of salient objects. In Yakhno, T., editor, *ADVIS'2004*, v. 3261 of *Lecture Notes in Computer Science*, pp. 363–371, Berlin Heidelberg. Springer-Verlag. 14

[Scherf & Brauer, 1997] Scherf, M. e Brauer, W. (1997). Feature selection by means of a feature weighting approach. Technical report, Technische Universität München, Munich. 30

[Schettini et al., 2002] Schettini, R., Ciocca, G., e Zuffi, S. (2002). *Color Imaging Science: Exploiting Digital Media*, Cap. Indexing and Retrieval in Colour Image Databases, pp. 1 9. J. Wiley. 14

[Scott & Shyu, 2003] Scott, G. J. e Shyu, C.-R. (2003). Ebs k-d tree: An entropy balanced statistical k-d tree for image databases with ground-truth labels. In et al., E. M. B., editor, *CIVR'2003*, v. 2728 of *Lecture Notes in Computer Science*, pp. 467 477. Springer-Verlag Berlin Heidelberg. 19

[Sebe et al., 2003] Sebe, N., Lew, M. S., Zhou, X., Huang, T. S., e Bakker, E. M. (2003). The state of the art in image and video retrieval. In et al., E. M. B., editor, *CIVR 2003*, v. 2728 of *Lecture Notes in Computer Science*, pp. 1 8. Springer-Verlag Berlin Heidelberg. 8

[Shaft & Ramakrishnan, 1996a] Shaft, U. e Ramakrishnan, R. (1996a). Data modeling and feature extraction management in image databases. In C.-C.J.Kuo, editor, *SPIE: Multimedia Storage and Archiving Systems*, v. 2916, pp. 90–102, Boston, MA. 49, 54

[Shaft & Ramakrishnan, 1996b] Shaft, U. e Ramakrishnan, R. (1996b). Data modeling and querying in the piq image dbms. *IEEE Data Engineering Bulletin*, 19(4):28 36. 49

[Shah et al., 2004] Shah, B., Raghavan, V., e Dhatric, P. (2004). Efficient and effective content-based image retrieval using space transformation. In Chen, Y.-P. P., editor, *10th*

International Multimedia Modeling Conference (MMM'2004), pp. 279 284, Brisbane, Australia. IEEE Computer Society. 20

[Sheikholeslami et al., 1998] Sheikholeslami, G., Chatterjee, S., e Zhang, A. (1998). Neuromerge: an approach for merging heterogeneous features in content-based image retrieval systems. In *International Workshop on Multimedia Database Management Systems*, pp. 106 113. 20

[Shyu et al., 1998] Shyu, C., Brodley, C. E., Kak, A., Kosaka, A., A.Aisen, e Broderick, L. (1998). Local versus global features for content-based image retrieval. In *IEEE Workshop of Content-Based Access of Image and Video Libraries*, pp. 30 34., Santa Barbara, CA. 13, 14

[Shyu et al., 1999a] Shyu, C.-R., Brodley, C. E., Kak, A. C., Kosaka, A., Aisen, A. M., e Broderick, L. S. (1999a). Assert: A physician-in-the-loop content-based retrieval system for hrct image databases. *Computer Vision and Image Understanding*, 75(1-2):111–132. 10, 21

[Shyu et al., 1999b] Shyu, C.-R., Kak, A., Brodley, C. E., e Broderick, L. S. (1999b). Testing for human perceptual categories in a physician-in-the-loop cbir system for medical imagery. In *IEEE Workshop of Content-Based Access of Image and Video Libraries*, pp. 102 108, Fort Collins, Colorado, USA. 22

[Silberschatz et al., 1999] Silberschatz, A., Korth, H. F., e Sudarshan, S. (1999). *Sistema de Banco de Dados*. Makron Books, São Paulo, terceira edição edition. 37, 52

[Skrypnik et al., 1999] Skrypnik, I., Terziyan, V. Y., Puuronen, S., e Tsymbal, A. (1999). Learning feature selection for medical databases. In *12th IEEE Symposium on Computer-Based Medical Systems (CBMS'99)*., pp. 53 57, Stamford, CT, USA. IEEE Computer Society. 20

[Smeulders et al., 2000] Smeulders, A. W. M., Worring, M., Santini, S., Gupta, A., e Jain, R. (2000). Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 22(12):1349-1380. iii, 8, 11

[Sousa et al., 2002] Sousa, E. P. M. d., Caetano Traina, J., Traina, A. J. M., e Faloutsos, C. (2002). How to use fractal dimension to find correlations between attributes. In *First Workshop on Fractals and Self-similarity in Data Mining: Issues and Approaches (in conjunction with 8th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining)*, pp. 26–30, Edmonton, Alberta, Canada. 29

[Sperber, 2001] Sperber, M. (2001). Computed tomography of the thorax: Lungs and mediastinum. In Sperber, M., editor, *Radiologic Diagnosis of Chest Disease*, pp. 56–77. Springer Verlag, London, 2 edition. ISBN: 1-8523-3298-0. 12

[Suetens, 2002] Suetens, P. (2002). *Fundamentals of Medical Imaging*. Cambridge University Press. 28

[Tang et al., 2003] Tang, H. L., Hanka, R., e Ip, H. H. S. (2003). Histological image retrieval based on semantic content analysis. *IEEE Transactions on Information Technology in Biomedicine (TITB)*, 7(1):26–36. 10, 17

[Tang et al., 2002] Tang, J., Acton, S. T., e Mukherjee, D. P. (2002). Retrieving similar images in an image database using a relational matrix. In *45th Midwest Symposium on Circuits and Systems (MWSCAS'2002)*, v. 1, pp. I – 667–70. 38

[Tao & Grosky, 1998] Tao, Y. e Grosky, W. I. (1998). Spatial color indexing: A novel approach for content-based image retrieval. In *IEEE International Conference on Multimedia Computing and Systems (ICMCS'1998)*, v. I. 14

[Thies et al., 2003] Thies, C., Malik, A., Keysers, D., Kohnen, M., Fischer, B., e Lehmann, T. M. (2003). Hierarchical feature clustering for content-based retrieval in medical image databases. In Sonka, M. e Fitzpatrick, J. M., editors, *Proceedings of SPIE - Medical Imaging 2003: Image Processing*, v. 5032, pp. 598–608. 47

[Torres et al., 2003] Torres, R. S., Silva, C. G., Medeiros, C. B., e Rocha, H. V. (2003). Visual structures for image browsing. In *12th International Conference on Information and Knowledge Management (CIKM'2003)*, New Orleans, Louisiana, USA. 23

[Traina et al., 2004] Traina, A. J. M., Balan, A. G. R., Bortolotti, L. M., e Jr., C. T. (2004). Content-based image retrieval using approximate shape of objects. In *17th IEEE Symposium on Computer-Based Medical Systems (CBMS'2004)*. IEEE Computer Society. 14, 15

[Traina et al., 2003] Traina, A. J. M., Castañón, C. A. B., e Jr, C. T. (2003). Multiwavemed: A system for medical image retrieval through wavelets transformations. In *16th IEEE Symposium on Computer-based Medical Systems (CBMS'2003)*, pp. 150–155, New York, USA. 14

[Traina et al., 2002a] Traina, A. J. M., Jr., C. T., Bueno, J. M., e Marques, P. M. d. A. (2002a). The metric histogram: A new and efficient approach for content-based image retrieval. In *Sixth IFIP Working Conference on Visual Database Systems*, pp. 297–311, Brisbane, Australia. 16

[Traina et al., 1999] Traina, C., Traina, A. J., e Faloutsos, C. (1999). Distance exponent : a new concept for selectivity estimation in metric trees. Research Paper CMU-CS-99-110, Carnegie Mellon University - School of Computer Science, Pittsburgh-PA USA. 29

[Traina et al., 2000a] Traina, Cactano, J., Traina, A. J. M., e Faloutsos, C. (2000a). Distance exponent: a new concept for selectivity estimation in metric trees. In *International Conference on Data Engineering (ICDE'2000)* 195, p., San Diego - CA. IEEE CS Press. 29

[Traina et al., 2002b] Traina, Cactano, J., Traina, A. J. M., Faloutsos, C., e Seeger, B. (2002b). Fast indexing and visualization of metric datasets using slim-trees. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*. 18

[Traina et al., 2000b] Traina, Cactano, J., Traina, A. J. M., Wu, L., e Faloutsos, C. (2000b). Fast feature selection using fractal dimension. In Medeiros, C. B. e Becker, K., editors, *XV Simpósio Brasileiro de Banco de Dados (SBBD'2000)*, pp. 158–171, João Pessoa - PA - Brazil. 28, 29

[Traina Jr. et al., 2000] Traina Jr., C., Traina, A. J. M., Seeger, B., e Faloutsos, C. (2000). Slim-trees: High performance metric trees minimizing overlap between nodes. In *International Conference on Extending Database Technology (EDBT'2000)*, v. 1777 of *Lecture Notes in Computer Science*, pp. 51–65, Germany. Springer. 6, 44, 85

[Uhlmann, 1991] Uhlmann, J. K. (1991). Satisfying general proximity/similarity queries with metric trees. *Information Processing Letter*, 40(4):175–179. 43, 44

[Vadivel et al., 2004] Vadivel, A., Majumdar, A. K., e Sural, S. (2004). Characteristics of weighted feature vector in content-based image retrieval applications. In *International Conference on Intelligent Sensing and Information Processing*, pp. 127 – 132. 29

[Vafaie & Jong, 1993] Vafaie, H. e Jong, K. A. D. (1993). Robust feature selection algorithms. In *International Conference on Tools with Artificial Inteligence (ICTAI'1993)*, Boston, MA. IEEE Computer Society Press. 30

[Vailaya, 1996] Vailaya, A. (1996). *Shape-based Image Retrieval*. PhD thesis, Michigan State University. 14

[Vailaya et al., 2001] Vailaya, A., Figueiredo, M. A. T., Jain, A. K., e Zhang, H.-J. (2001). Image classification for content-based indexing. *IEEE Transactions on Image Processing (TIP)*, 10(1):117–130. 8

[Vasconcelos, 2004a] Vasconcelos, N. (2004a). Minimum probability of error image retrieval. *IEEE Transactions on Signal Processing (TSP)*, 52(8):2322–2336. 20

[Vasconcelos, 2004b] Vasconcelos, N. (2004b). On the efficient evaluation of probabilistic similarity functions for image retrieval. *IEEE Transactions on Information Theory (TIT)*, 50(7):1482 1496. 33

[Vasconcelos & Lippman, 2000] Vasconcelos, N. e Lippman, A. (2000). A unifying view of image similarity. In *International Conference on Pattern Recognition (ICPR'2000)*, v. 1, pp. 1038 1041, Barcelona, Spain. IEEE. 38

[Veltkamp & Tanase, 2000] Veltkamp, R. C. e Tanase, M. (2000). Content-based image retrieval systems: A survey. Technical report, Department of Computer Science - Utrecht University. 8

[Vieira et al., 2004] Vieira, M. R., Traina Jr., C., Chino, F. J. T., e Traina, A. J. M. (2004). Dbm-tree: A metric access method sensitive to local density data. In *Simpósio Brasileiro de Banco de Dados (SBBD'2004)*, pp. 163 177, Brasília-DF, Brasil. Sociedade Brasileira de Computação. 44

[Vu et al., 2003] Vu, K., Hua, K., e Tavanapong, W. (2003). Image retrieval based on regions of interest. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 15(4):1045 1049. 14

[W3C, 1999] W3C (1999). Xml path language (xpath) - version 1.0 - w3c recommendation. http://www.w3.org/TR/xpath. 75

[Wang & Wang, 2001] Wang, C. e Wang, X. S. (2001). Indexing very high-dimensional sparse and quasi-sparse vectors for similarity searches. *VLDB Journal*, 9(4):344 361. 42

[Wang et al., 1997] Wang, J., Yang, W.-j., e Acharya, R. (1997). Color clustering techniques for color-content-based image retrieval from image databases. In *International Conference on Multimedia Computing and Systems*, pp. 442 449, Ottawa, Ontario, Canada. IEEE Computer Society. 14

[Ward, 2003] Ward, R. (2003). *Oracle interMedia User's Guide, 10g Release 1 (10.1)*. Oracle Corporation. 54

[Webb et al., 2001] Webb, W. R., Müller, N. L., e Naidich, D. P. (2001). *High Resolution CT of the Lung*. Lippincott Williams & Wilkins, Philadelphia, PA, USA, 3 edition. 12

[Weber & Mlivoncic, 2003] Weber, R. e Mlivoncic, M. (2003). Efficient regionbased image retrieval. In *International Conference on Information and Knowledge Management (CIKM'2003)*, pp. 69 76, New Orleans, Louisiana, USA. ACM. 38

[Weber et al., 1998] Weber, R., Schek, H.-J., e Blott, S. (1998). A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. In Gupta, A., Shmueli, O., e Widom, J., editors, *24th International Conference on Very Large Databases (VLDB'1998)*, pp. 194–205, New York City. Morgan Kaufmann. 45

[White & Jain, 1996] White, D. A. e Jain, R. (1996). Similarity indexing with the ss-tree. In Su, S. Y. W., editor, *International Conference on Data Engineering (ICDE'1996)*, pp. 516–523, New Orleans, Louisiana. IEEE Press. 42

[Williams & Zobel, 2002] Williams, H. E. e Zobel, J. (2002). Indexing and retrieval for genomic databases. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 14(1):63–78. 29

[Wirth et al., 2004] Wirth, M., Lyon, J., Fraschini, M., e Nikitenko, D. (2004). The effect of mammogram databases on algorithm performance. In *17th IEEE Symposium on Computer-Based Medical Systems (CBMS'2004)*, pp. 15–20, Bethesda, MD, USA. IEEE Computer Society. 11

[Xu et al., 2000] Xu, K., Georgescu, B., Comaniciu, D., e Meer, P. (2000). Performance analysis in content-based retrieval with textures. In *15th International Conference on Pattern Recognition*, v. 4, pp. 275–278, Barcelona, Spain. 18, 22, 38

[Yang & Hansell, 1997] Yang, G.-Z. e Hansell, D. M. (1997). Ct image enhancement with wavelet analysis for the detection of small airways disease. *IEEE Transactions on Medical Imaging (TMI)*, 16(6):953–961. 14

[Yang et al., 2003] Yang, J., Ward, M., Rundensteiner, E., e Huang, S. (2003). Visual hierarchical dimension reduction for exploration of high dimensional datasets. In Bonneau, G.-P., Hahmann, S., e Hansen, C. D., editors, *Joint EUROGRAPHICS - IEEE TCVG Symposium on Visualization*, pp. 19–29. 48

[Yang et al., 2002] Yang, M.-H., Kriegman, D. J., e Ahuja, N. (2002). Detecting faces in images: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI'2002)*, 24(1):34–58. 9

[Yi & Faloutsos, 2000] Yi, B.-K. e Faloutsos, C. (2000). Fast time sequence indexing for arbitrary lp norms. In *26th International Conference on Very Large Databases (VLDB'2000)*, pp. 385–394, Cairo, Egypt. Morgan Kaufmann. 35

[Yu, 2002] Yu, C. (2002). *High-Dimensional Indexing: Transformational Approaches to High-Dimensional Range and Similarity Searches*, v. 2341 of *Lecture Notes in Computer Science*. Springer-Verlag GmbH. ISSN: 0302-9743. 31, 41, 42, 43, 47

[Yu et al., 2004] Yu, C., Bressan, S., Ooi, B. C., e Tan, K.-L. (2004). Querying high-dimensional data in single-dimensional space. *VLDB Journal*, 13(2):105 119. 30, 45

[Yu et al., 2001] Yu, C., Ooi, B. C., Tan, K.-L., e Jagadish, H. V. (2001). Indexing the distance: An efficient method to knn processing. In Apers, P. M. G., Atzeni, P., Ceri, S., Paraboschi, S., Ramamohanarao, K., e Snodgrass, R. T., editors, *27th International Conference on Very Large Databases (VLDB'2001)*, pp. 421–430, Roma, Italy. Morgan Kaufmann. 44

[Yu et al., 2002] Yu, H., Li, M., Zhang, H.-J., e Feng, J. (2002). Color texture moments for content-based image retrieval. In *International Conference on Image Processing*, v. 3, pp. 929 932. 44

[Zhang & Srihari, 2004] Zhang, B. e Srihari, S. N. (2004). Fast k-nearest neighbor classification using cluster-based trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI'2004)*, 26(4):525–528. 37

[Zhang & Zhang, 2003] Zhang, R. e Zhang, Z. M. (2003). Addressing cbir efficiency, effectiveness, and retrieval subjectivity simultaneously. In *MIR '03: Proceedings of the 5th ACM SIGMM international workshop on Multimedia information retrieval*, pp. 71 78, Berkeley, California, USA. ACM Press. 20

[Zhou et al., 2003] Zhou, X. S., Comaniciu, D., e Krishnan, A. (2003). Conditional feature sensitivity: a unifying view on active recognition and feature selection. In *9th IEEE International Conference on Computer Vision (ICCV'2003)*, v. 2, pp. 1502 1509. 47

[Zhou & Huang, 2002] Zhou, X. S. e Huang, T. S. (2002). Unifying keywords and visual contents in image retrieval. *IEEE Multimedia*, pp. 23–33. 22

[Zhu et al., 2002] Zhu, L., Rao, A., e Zhang, A. (2002). Theory of keyblock-based image retrieval. *ACM Transactions on Information Systems (TOIS'2002)*, 20(2):224257. 18

[Zirkelbach, 1999] Zirkelbach, C. (1999). Similarity indexing by means of a metric. In IEEE, editor, *10th International Workshop on Database & Expert Systems Applications*, Florence, Italy. 43