SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito: 20.01.2005

Assinatura: *Ana Paula Jampaio Treina*

iRIS Servidor: Sistema servidor para recuperação de imagens por conteúdo

Odnei Cuesta Lopes

Orientador: *Profa. Dra. Agma Juci Machado Traina*

Dissertação apresentada ao Instituto de Ciências Matemáticas e de Computação - ICMC-USP, como parte dos requisitos para obtenção do título de Mestre em Ciências da Computação e Matemática Computacional.

USP – São Carlos
Janeiro/2005

Aluno: *Odnei Cuesta Lopes*

A Comissão Julgadora:

Profa. Dra. Agma Juci Machado Traina

Profa. Dra. Roseli de Deus Lopes

Prof. Dr. Adilson Gonzaga

“A suprema felicidade da vida é a convicção de ser amado por aquilo que você é, ou melhor, apesar daquilo que você é.”

Victor Hugo

Dedico este trabalho a meus pais, que estão sempre presentes em minha vida, nos piores e nos melhores momentos, a um certo alguém especial, que nem sempre está presente, mas que entrou em minha vida e no meu coração e nunca mais vai partir e aos amigos, minha segunda família.

Odnei Cuesta Lopes

AGRADECIMENTOS

Agradeço a Deus, por ter me dado a oportunidade de ter uma família maravilhosa, por ter me concedido recursos que me possibilitassem batalhar e conseguir alcançar sempre um lugar melhor.

A meus pais, Odney e Palmyra, meus irmãos, Cassia e Ricardo, que empregaram seus esforços para me apoiar e me mostraram que o mundo é feito de vários tipos de pessoas e que todas elas têm sua importância.

A Agma e a Josiane, por terem me dado a oportunidade de desenvolver este trabalho e terem iluminado os caminhos quando a sombra da dúvida se fazia presente.

A todo o pessoal do Grupo de Bases de dados e Imagens, que me apoiaram durante todo o mestrado, tanto na discussão de recursos técnicos da área de computação quanto no lado humano com suas experiências de vida.

E finalmente ao CNPq, pelo auxílio financeiro concedido no último ano deste trabalho.

Sumário

Lista de figuras	iv
Lista de tabelas	vi
Lista de símbolos	vii
Lista de abreviaturas	viii
Resumo	ix
Abstract	x
1 Introdução	1
1.1 Definição do problema	2
1.2 Organização do documento	3
2 Sistemas de armazenamento e comunicação de imagens	4
2.1 PACS Tradicionais	4
2.1.1 Sistema de armazenamento de dados e imagens	6
2.2 PACS com técnicas de CBIR	6
2.2.1 Sistema de armazenamento de dados e imagens	7
2.3 PACS/cbPACS via Internet	8
2.3.1 Segurança	10
2.4 Trabalhos correlatos e outras considerações	10
2.5 Considerações finais	12
3 Tecnologias de suporte	14
3.1 Padrão DICOM	14
3.1.1 Definições propostas pelo padrão DICOM	15
3.1.2 Funcionamento do DICOM	17

3.1.3	Objetivos da versão 3.0 e vantagens de utilização do padrão	17
3.2	Consulta por similaridade	18
3.2.1	Estruturas de índice	18
3.2.2	Definições	19
3.2.3	Extratores de características	20
3.2.4	Características visuais	21
3.2.5	Comparando imagens	24
4	O sistema iRIS	25
4.1	Funcionalidades e objetivos do iRIS	25
4.2	Arquitetura	26
4.3	Servidor de indexação	26
4.3.1	Servidor	27
4.3.2	Cliente	33
4.3.3	Portabilidade	34
4.4	Servidor iRIS - Web	34
4.4.1	Desenvolvimento Web	34
4.4.2	O padrão de projeto MVC	34
4.4.3	Tecnologia Java	35
4.4.4	Estruturação do Servidor iRIS	36
4.5	Base de dados	38
4.5.1	Processamento de uma consulta	38
4.5.2	Esquema inicial	39
4.5.3	Projeto CIRCE	40
4.5.4	Esquema final	43
4.5.5	Origem das imagens	44
4.5.6	População da base e o Sistema de Alimentação de Dados e Imagens - SADI	45
4.6	Considerações finais	46
5	Resultados	48
5.1	A base computacional	49
5.2	A plataforma de desenvolvimento e testes	49
5.3	O sistema em funcionamento	50
5.4	Considerações finais	51
6	Conclusões	56
6.1	Análise dos resultados	56
6.2	Propostas de trabalhos futuros	58

Referências Bibliográficas	58
Glossário	64

Lista de Figuras

2.1	Infraestrutura de um PACS	5
2.2	Arquitetura de um PACS tradicional	6
2.3	SADI de um PACS tradicional, explicitando o processo de alimentação da base de dados	7
2.4	Arquitetura genérica de um cbPACS	8
2.5	SADI de um cbPACS	9
4.1	Arquitetura do iRIS, onde destaca-se o servidor de indexação em relação à figura 2.4	27
4.2	Esquema UML de políticas de estruturas de indexação	29
4.3	Esquema UML dos objetos indexáveis	29
4.4	Esquema UML de funções distância	30
4.5	Esquema UML de um <i>plugin</i>	30
4.6	Esquema UML de um Tipo de estrutura de indexação e de suas instâncias .	31
4.7	Divisão em camadas segundo o padrão MVC	35
4.8	Estrutura do iRIS segundo o padrão MVC	37
4.9	Organização do módulo SRDI	39
4.10	Processamento de uma consulta	40
4.11	Diagrama entidade relacionamento da base de dados	41
4.12	Arquitetura geral do CIRCE (adaptada de [Araujo et al., 2002])	42
4.13	Arquitetura detalhada do iRIS	46
4.14	Arquitetura detalhada do iRIS utilizando o CIRCE	47
5.1	Exemplo de um exame completo destacando a imagem utilizada como base para a consultas por similaridades ilustradas nas figuras 5.3, 5.4, 5.5 e 5.6 . .	50
5.2	Exemplo dos parâmetros de uma consulta por similaridade	51
5.3	Exemplo de resposta a uma busca por abrangência com raio 0.0015 utilizando textura	52

5.4	Exemplo de resposta a uma busca por abrangência com raio 0.01315 utilizando histograma	53
5.5	Exemplo de resposta a uma busca pelos k vizinhos mais próximos utilizando textura com $k = 6$	54
5.6	Exemplo de resposta a uma busca pelos k vizinhos mais próximos utilizando histograma com $k = 6$	55

Lista de Tabelas

3.1	Taxonomia sobre espaços de características extraídas de imagens	21
5.1	Tempo de construção da estrutura de indexação de toda a base de imagens .	49

Lista de símbolos

D	Domínio ou universo de objetos possíveis ou válidos.
$d()$	Função de distância métrica. $d(O_i, O_j)$, nos quais $O_i, O_j \in D$, representa a distância (similaridade) entre O_i e O_j .
k	Quantidade de objetos que se deseja recuperar em uma consulta de k -vizinhos mais próximos (<i>k-Nearest Neighbors Query</i>).
$kNNQ()$	Consulta aos k -vizinhos mais próximos (<i>k-Nearest Neighbors Query</i>).
$kNNQ(o_q, k)$	Representa uma consulta aos k -vizinhos mais próximos, cujo o centro é dado por o_q e a quantidade de vizinhos que se deseja recuperar é dado por k .
$\mathbb{M}$	Espaço métrico definido pelo par $\langle \mathcal{D}, d() \rangle$.
O	Conjunto (ou base, ou coleção) de objetos.
o_i	Elemento de O . Quando indicado, o_i é usado de forma restrita, representando um objeto de um subconjunto $o_i \in O \subseteq \mathcal{D}$.
o_q	Caso especial de o_i , no qual o objeto é tido como objeto de busca ou de referência em uma consulta.
r_q	Raio de busca em uma consulta por abrangência (<i>Range Query</i>).
$RQ()$	Consulta por abrangência (<i>Range Query</i>).
$RQ(o_q, r_q)$	Representa uma consulta por abrangência cujo o centro é dado por o_q e o raio por r_q .
Convenção:	
<i>Itálico</i>	O tipo itálico é usado para indicar palavras e abreviaturas de termos em língua estrangeira.
Negrito	O tipo negrito é usado para destacar a definição de um termo.

Lista de abreviaturas

ACR-NEMA	<i>American College of Radiology - National Electrical Manufactures Association.</i>
CAD	<i>Computer Aided Diagnosis</i>
CBIR	<i>Content Based Image Retrieval</i>
cbPACS	Sistema de Armazenamento e Comunicação de Imagens baseado em conteúdo (<i>content-based Picture Archiving and Communication System</i>).
CCIFM	Centro de Ciências de Imagens e Física Médica.
CIRCE	<i>Content Image Retrieval Core Engine.</i>
CT	Tomografia computadorizada (<i>Computadorized Tomography</i>).
DNA	Ácido Desoxirribonucléico. (<i>Deoxyribonucleic Acid</i>).
DICOM	<i>Digital Imaging and COmmunication in Medicine.</i>
DICOM-SR	<i>Digital Imaging and COmmunication in Medicine - Structured Report.</i>
GBdI	Grupo de Base de dados e Imagens.
ICMC	Instituto de Ciências Matemáticas e de Computação.
iRIS	Internet-based Retrieval of Images by Similarity.
JDBC	<i>Java Data Base Connectivity</i> - Tecnologia de acesso genérico à base de dados em Java.
k NNQ	Consulta aos k -Vizinhos mais próximos (<i>k-Nearest Neighbors Query</i>).
MAM	<i>Metric Access Method</i> - Método de Acesso Métrico.
MVC	<i>Model View Controler</i> - Padrão de projeto Modelo Visão Controlador.
OID	<i>Object IDentification</i> (código de identificação do objetos).
PACS	Sistema de armazenamento e comunicação de imagens (<i>Picture Archiving and Communication System</i>).
RQ	Consulta por Abrangência. (<i>Range Query</i>).
RM	Ressonância Magnética.
SAM	<i>Spatial Access Method</i> - Método de Acesso Espacial.
SGBD	Sistema de Gerenciamento de Bases de Dados.
SIH	Sistema de Informação Hospitalar.
USP	Universidade de São Paulo.
WWW	Teia mundial (<i>World Wide Web</i>).

LOPES, O. C. *iRIS Servidor: Sistema servidor para recuperação de imagens por conteúdo* São Carlos, 2004. 64 p. Monografia de Qualificação de Mestrado - Instituto de Ciências Matemáticas e de Computação - ICMC/USP.

RESUMO

A computação tem sido uma ferramenta de desenvolvimento para outras ciências e os desafios encontrados no desenvolvimento de ferramentas computacionais ajudam a desenvolver a própria computação. Uma das áreas que tem se beneficiado sobremaneira da utilização de sistemas e equipamentos computacionais é a medicina. O diagnóstico baseado em imagens e o rápido surgimento de novas modalidades de exames têm ajudado a medicina a se tornar cada vez mais precisa e eficiente, tanto no diagnóstico como na evolução dos tratamentos. No entanto, as possibilidades até então exploradas nos PACS (*Picture Archiving and Communication Systems*) não têm utilizado eficientemente os recursos empregados na sua implementação. Dois fatores que tornam estes sistemas inadequados são a baixa acessibilidade dos mesmos e a grande subjetividade na associação de palavras-chave às imagens que são empregadas nas buscas subseqüentes. Este trabalho propõe a disponibilização de acesso ao sistema PACS pela Internet de modo a aumentar a acessibilidade do mesmo, visto que, um médico que não se encontra no hospital pode consultar os exames de seu paciente, os exames podem ser utilizados em salas de aulas de faculdades associadas a hospitais universitários, pode-se formar juntas médicas virtuais para a análise de um exame, entre outras possibilidades. Este trabalho é suportado por técnicas de recuperação de imagens baseadas em seu conteúdo para a resolução do problema associado à subjetividade, ou seja, utilizam-se extratores de características e estruturas de indexação métricas baseadas nas assinaturas geradas pelos extratores para recuperar as imagens similares.

ABSTRACT

The computer science area has been a great support on the development of many scientific fields, and, on the other hand, the demand of those fields gives motivation to the continuous growth of the computer science area itself. The medical field is one of the main areas to be benefited from such symbiotic iteration. This scenario contributes to make medical image analysis and diagnosis more precise and efficient, leading to better treatment of the patients. The Picture Archiving and Communication Systems (PACS) can have their resources better exploited to improve further the medical diagnosis area. Two points restrict PACS functionality: the low accessibility from abroad and high subjectivity on associating keywords to images to perform queries on them. This work proposes to show how to build an Internet-based PACS, thus increasing PACS accessibility (first point). Therefore, a physician outside of the hospital (who has granted access to it) can browse on data/images from patients, the exams can be viewed during classes in School Hospitals, specialists can be virtually gathered and make diagnosis and discuss it even being in different places, among other possibilities. This work is also supported by content-based image retrieval techniques aiming to diminish the subjectivity problem (second point). That is, features extracted from images and metric access methods are employed to process similarity queries, what can be a great aid in diagnosis based on images.

Introdução

A integração entre as áreas de pesquisa científica tem sido um dos grandes motores no desenvolvimento da ciência. Particularmente, a área de computação tem contribuído sobremaneira para o crescimento e aprimoramento de quase todas as demais áreas. Por outro lado, a computação tem sido beneficiada pelo lançamento de novos desafios oriundos de seus parceiros científicos, o que tem também impulsionado o seu próprio crescimento.

A medicina tem sido uma das áreas que mais tem se beneficiado da interação com a computação, seja através dos chamados Sistemas de Informação Hospitalares (SIH), seja através de equipamentos computacionais que suportam a coleta de dados e exames de pacientes. Porém, atualmente existe uma demanda bastante forte para armazenar e consultar dados complexos, por exemplo, imagens coletadas em exames médicos, tais como raios-X, tomografia (CT ou RM), ultra-som, seqüências de DNA, etc. Com a introdução dos sistemas PACS (*Picture Archiving and Communication Systems*) [Cao 2000, Furuie 1999, Siegel 1999, Marsh 1997], cresceu o interesse por integrar num só sistema todas as informações dos pacientes (textos, imagens, gráficos, dados temporais, etc), e não apenas as informações textuais como é feito nos SIH. A integração de imagens médicas com outras informações hospitalares exige a utilização de técnicas otimizadas que não comprometam a performance do sistema PACS [Stasiu 2001]. A importância dos PACS vem do fato de possibilitar a integração de todas as informações dos pacientes de um hospital ou centro médico, permitindo associar e recuperar rapidamente os exames médicos em si, e não apenas as descrições e laudos realizados. Atualmente, as pesquisas relacionadas a sistemas PACS têm voltado a atenção para disponibilizar acesso à informação hospitalar através da Web, permitindo realizar consultas sobre os exames médicos em situações emergenciais quando o especialista não se encontra no ambiente hospitalar [Feng 2002, Feng 2001, Kitney 2001, Pereira 2001, Zhang 2001].

Os PACS comerciais permitem recuperar as imagens que foram associadas aos pacientes através de chaves textuais. Isto é, atendem consultas do tipo: “Retorne os exames de Raios-X de tórax efetuados sobre o paciente José da Silva” - neste caso a busca é realizada através da chave especificada pelo nome do paciente. É interessante observar que tais consultas não são as mais desejadas em um sistema de suporte a diagnósticos, o qual visa auxiliar o médico a encontrar imagens e efetuar cruzamentos com diagnósticos previamente realizados, muitas vezes conhecendo-se apenas a imagem que se tem em mãos. Consultas tais como: “Retorne as 5 imagens mais semelhantes à imagem de Raio-X de tórax armazenada no arquivo ‘Img123.pac’ ”, utilizam como chave de busca uma imagem, neste caso a imagem correspondente ao Raio-X de tórax indicado. Assim, a partir de uma imagem, deve-se pesquisar na base de imagens e encontrar as que são mais semelhantes (mais próximas) segundo critérios específicos.

O tipo de dado imagem mantém informação inerentemente volumosa e complexa, demandando todo um ambiente para processamento de imagens a fim de possibilitar que o sistema de gerenciamento de bases de dados (SGBD) integre imagens aos demais dados do sistema, permitindo não apenas a armazenagem, mas também a recuperação baseada nas imagens em si, e não em textos ou outros dados do objeto aos quais elas estejam vinculadas. O suporte à recuperação de imagens baseada em seu conteúdo tem sido estudado dentro do Grupo de Bases de Dados e Imagens (GBdI) do Instituto de Ciências Matemáticas e de Computação (ICMC) - Universidade de São Paulo (USP) [Bueno 2002, Bueno 2002b, Rosa 2002, Traina 1998]. Esses trabalhos têm sido orientados no sentido de permitir tratar imagens como dados complexos, suportando as mesmas funcionalidades como os SGBDs tradicionais tratam dados simples, como números e cadeias de caracteres curtas. Essas funcionalidades incluem a armazenagem, recuperação e manipulação de imagens.

Já o Centro de Ciências de Imagens e Física Médica (CCIFM) do Hospital das Clínicas da USP de Ribeirão Preto vem buscando o desenvolvimento de um SIH apoiado em um PACS, que atenda às necessidades do hospital. O GBdI e o CCIFM vêm juntando esforços para o desenvolvimento de um sistema PACS que incorpore características de recuperação de imagens por conteúdo propiciando um auxílio efetivo à recuperação de imagens e informações de pacientes de forma rápida e descentralizada. É certo que a parceria entre os dois grupos de pesquisa, que já vêm atuando juntos, será solidificada e contribuirá para a melhoria do estado da arte em sistemas PACS, resultando também em um sistema real para utilização em ambientes clínicos.

1.1 Definição do problema

A construção de um PACS com técnicas de CBIR (*Content-Based Image Retrieval*), integrado com o SIH disponibiliza, além de buscas baseadas em conteúdo para o auxílio a

diagnósticos, um sistema completo e centralizado de informações do hospital.

Adicionando a este sistema o acesso através da Internet, mais especificamente através da Web, ganha-se um significativo aumento de acessibilidade e, com isso, justificam-se os recursos empregados em tecnologia de informação na área de saúde.

A contribuição dos desafios lançados pelo desenvolvimento de tal sistema tornará a arquitetura dos PACS mais madura e completa, possibilitando ainda, a interconexão de trabalhos realizados pelo GBdI, tais como extratores de características, funções distância e o *framework* de desenvolvimento de árvores métricas (Arboretum - www.gbdi.icmc.usp.br/arboretum).

Finalmente, propõe-se através do desenvolvimento deste protótipo, mostrar que é factível a utilização deste sistema em um ambiente de produção, por meio da obtenção de uma interface com o usuário amigável e intuitiva, tempos de resposta curtos e acesso descentralizado, encorajando e estimulando desta maneira sua adoção em hospitais escola, faculdades de medicina e mesmo em hospitais comerciais.

1.2 Organização do documento

No capítulo 2 são descritos PACS representativos encontrados na literatura, suas características e arquitetura. Segue-se, capítulo 3, detalhando algumas das tecnologias utilizadas como suporte a construção de um PACS com CBIR.

Os desafios e soluções encontrados na construção do protótipo são elucidados durante o capítulo 4. Apresentam-se os resultados da construção e teste do sistema no capítulo 5.

E finalmente, no capítulo 6, infere-se sobre os resultados e os objetivos propostos, demonstrando o cumprimento da proposta deste trabalho, apresentado ainda alguns aspectos complementares recentes como proposta de trabalhos futuros.

Sistemas de armazenamento e comunicação de imagens

Os sistemas de armazenamento e comunicação de imagens (*Picture Archiving and Communication Systems - PACS*) têm sido de grande importância em ambientes que utilizam-se da análise de imagens para suporte à tomada de decisões, como o suporte ao diagnóstico baseado nas imagens geradas a partir de exames médicos. Este capítulo discorre sobre os PACS tradicionais, passando pela complementação destes sistemas com técnicas de CBIR, dando origem ao cbPACS (*content-based Picture Archiving and Communication System*) [Bucno 2002], apresentando ainda alguns PACS que utilizam a Internet e as questões relacionadas a este tipo de sistema.

2.1 PACS Tradicionais

Segundo Günter [Gunther 1999a] um PACS completo deve executar as seguintes funções sobre imagens usando tecnologia digital:

- aquisição;
- comunicação (transferência);
- armazenamento;
- exibição;
- e processamento.

No entanto, os sistemas implementados geralmente não suportam todas as características acima, gerando um novo tipo de sistema que é mais adequado ao ambiente em que se encaixa por possuir somente as características que são indispensáveis. Mesmo porque as funções básicas de um PACS são somente armazenamento e comunicação de imagens, sendo que as outras características citadas por Gunter são geralmente complementações adicionadas pelos fabricantes de aparelhos de aquisição de imagens em seus sistemas proprietários.

A infra-estrutura associada a um PACS pode ser observada na figura 2.1.

Figura 2.1: Infraestrutura de um PACS

Nestes sistemas, a busca por uma imagem segue um processo de busca por palavras associadas à imagem no momento de seu armazenamento. Deste modo, se a associação das palavras às imagens não for realizada pelo mesmo usuário que efetua as buscas, ou se o usuário modificar seu modo de observação em algum momento durante o ciclo de vida das imagens armazenadas, o processo de recuperação ficará comprometido.

Na figura 2.2 observa-se a arquitetura de um PACS tradicional. O processo de armazenamento de dados e imagens no Sistema Gerenciador de Banco de Dados (SGBD) será detalhado em 2.1.1. A partir do momento em que as imagens e os dados já se encontram no SGBD é possível a realização de buscas. Em uma busca, o Cliente envia os parâmetros (palavras que descrevem as imagens a serem recuperadas) que especificam a consulta ao Servidor PACS e este efetua a consulta no SGBD retornando os dados e/ou imagens ao Cliente.

Figura 2.2: Arquitetura de um PACS tradicional

2.1.1 Sistema de armazenamento de dados e imagens

O Sistema de Armazenamento de Dados e Imagens (SADI) de um PACS tradicional pode ser observado na figura 2.3. O usuário faz a inserção dos dados do paciente, se estes dados ainda não estiverem armazenados no SGBD. A partir de então, quando houver imagens de exames daquele paciente a serem armazenadas, o usuário deve associar palavras que descrevam tais imagens. Estas palavras serão comparadas com as palavras utilizadas como parâmetro para a consulta durante a recuperação.

2.2 PACS com técnicas de CBIR

Como foi visto na seção 2.1, a consulta por palavras chaves cria problemas pela subjetividade inerente à associação das mesmas pelos usuários. Dessa forma, buscou-se uma alternativa na qual pode-se realizar uma consulta fornecendo como parâmetro não mais palavras, mas uma imagem. Esta imagem pode estar tanto armazenada no SGBD, quanto num arquivo local na máquina do usuário que está realizando a consulta. Para tanto, utiliza-se o conceito de consulta por conteúdo ou similaridade.

O *cbPACS* (*content-based Picture Archiving and Communication System*) [Bueno 2002] utiliza estruturas de indexação e extratores de características das imagens para efetuar a recuperação das imagens mais similares à imagem fornecida como parâmetro para a consulta. O termo *cbPACS* será utilizado como sinônimo de PACS empregando técnicas de CBIR no restante deste documento. O funcionamento da consulta por similaridade é

Figura 2.3: SADI de um PACS tradicional, explicitando o processo de alimentação da base de dados

explicado com mais detalhes na seção 3.2.

Na figura 2.4 observa-se a arquitetura de um cbPACS. A imagem que serve de base à consulta será submetida ao extrator de características para a obtenção do vetor de características que a descreve (somente se a imagem for uma imagem local da máquina do usuário, ou se a imagem - estando armazenada no SGBD - não possuir um vetor de características para o extrator selecionado). A seguir, este vetor de características será enviado ao servidor cbPACS, juntamente com outros parâmetros necessários à consulta, para que as imagens sejam recuperadas. O servidor se encarrega de consultar a estrutura de índice adequada ao extrator de características utilizado e de recuperar os identificadores das imagens (*Object IDentifiers* - *OIDs*) que satisfazem tal consulta. Após a recuperação dos OIDs o servidor obtém do SGBD as imagens e as envia ao Cliente.

2.2.1 Sistema de armazenamento de dados e imagens

O processo de armazenamento dos dados e imagens em um cbPACS é diferente do processo de armazenamento de um PACS tradicional. Apesar de os dados do paciente serem inseridos no SGBD pelo usuário da mesma forma, a associação de informações para a recuperação das imagens é realizada de modo automático, eliminando a subjetividade associada às palavras-chave, substituindo-as por vetores de características que são obtidos através de extratores de características conforme descreve a figura 2.5. Assim, o processo de inserção e busca será realizado sempre da mesma maneira tornando a recuperação das imagens mais precisa.

Figura 2.4: Arquitetura genérica de um cbPACS

2.3 PACS/cbPACS via Internet

A disponibilização de PACS/cbPACS na Internet tem como objetivo tornar estes sistemas acessíveis de qualquer parte do mundo e, logo, aumentar tanto a sua utilização, quanto sua utilidade. O aumento da utilização é decorrente diretamente do aumento da acessibilidade. No entanto, o aumento da utilidade pode ser analisado por um exemplo: suponha-se um cbPACS para auxílio ao diagnóstico médico. Se em algum momento o médico não se encontrar no hospital, ele não poderia fazer uma consulta ao exame de determinado paciente e portanto teria que se deslocar até o hospital. Em um caso de emergência, o tempo de deslocamento pode ser inaceitável, incorrendo em risco de vida para o paciente. Se o sistema estiver disponível para acesso via Internet, basta que o médico acesse remotamente tal sistema e instrua qualquer outro médico, presente no hospital, como agir até que seja possível a ele chegar ao hospital.

Vários PACS de suporte a diagnóstico médico vêm sendo desenvolvidos e implantados em várias partes do mundo. Os primeiros sistemas eram PACS tradicionais como o SINFIM [Pereira 2001, Pereira 2002, Pereira 2003], mas com o decorrer do tempo surgiram os que

Figura 2.5: SADI de um cbPACS

utilizam consultas baseadas no conteúdo das imagens, como é o caso do sistema descrito por Zhuang [Zhuang 2001], do IBMAS descrito por Zhang [Zhang 2001] e do sistema proposto por Feng [Feng 2000a, Feng 2000b, Feng 2001, Feng 2002].

Um sistema protótipo de cbPACS também foi desenvolvido por Long [Long 2003]. Este sistema tem características muito interessantes, como a segmentação das imagens de forma manual e auxiliada por computador, quatro métodos para a extração de vetores de características e ainda a busca de imagens por similaridade baseadas em um esboço da forma procurada.

Todos os sistemas citados acima foram idealizados para estarem acessíveis pela Internet. É um fator comum no ciclo de desenvolvimento de sistemas baseados na Internet é o aspecto da segurança, visto que, informações sigilosas a respeito de pacientes estarão disponíveis.

Li [Li 2003] apresenta um sistema muito similar ao que pretende-se desenvolver neste projeto de pesquisa utilizando-se Java, JSP, JavaBeans e EnterpriseJavaBeans (EJB). A importância de modularização fica evidente pelo uso de JavaBeans, que são altamente reutilizáveis. A tecnologia EJB e a construção intuitiva de componentes levou Wong [Wong 2003] a escolher a tecnologia Java para implementar seu sistema protótipo.

Nikolaidou [Nikolaidou 2003] desenvolveu um sistema de Biblioteca Digital pela Internet, para utilizar-se das imagens geradas por um hospital universitário para ensinar às novas turmas.

Na sub-seção 2.3.1 discorre-se sobre os procedimentos observados nos trabalhos estudados.

2.3.1 Segurança

A disponibilização de acesso aos dados providos por um PACS na Internet tem vários aspectos positivos como visto em 2.3. No entanto, esta característica implica o uso de mecanismos de segurança, tanto para restringir os acessos (identificação e autenticação dos usuários) como em relação à comunicação segura (transmissões encriptadas) [Pereira 2001].

O SINFIM (Sistema de INformación de Imágenes Médicas) proposto por Pereira ([Pereira 2001, Pereira 2002, Pereira 2003]) provê segurança, definindo que dados são para acesso público e quais têm acesso restrito, e complementa a segurança por meio da utilização de um protocolo seguro (Secure Socket Layer - SSL). Definiu-se também perfis de usuários no SGBD com privilégios distintos (convidado, médico, radiologista e operadores do sistema).

No E-HEALTH [Feng 2002], utilizou-se uma solução similar, com o emprego de autenticação de usuários e comunicação protegida por meio protocolo seguro. Já no IBMAS [Zhang 2001] o controle de acesso só é realizado pela autenticação de usuários.

2.4 Trabalhos correlatos e outras considerações

Traina [Traina 2003] descreve como foi realizada a integração de recuperação baseada em conteúdo em um sistema de um hospital universitário.

Há vários outros aspectos envolvidos na construção de um PACS, como o formato das imagens a serem manipuladas e transferidas no sistema, formatos e padrões de intercâmbio de dados já utilizados na área médica e questões relativas ao modo de armazenamento empregado.

Uma das questões mais importantes nos sistemas que trabalham com imagens é o formato que as imagens seguirão. Apesar dos avanços recentes em velocidade de transmissão e capacidade armazenamento dos sistemas, os PACS necessitam utilizar-se de imagens comprimidas para acelerar o tempo de transmissão e minimizar o espaço em disco requerido para o armazenamento, de modo a reduzir o custo de implementação desses sistemas [Sung 2002]. Em particular, o padrão JPEG foi a primeira tecnologia de compressão utilizada no padrão DICOM. O padrão JPEG inclui técnicas de compressão com ou sem perdas e ambas têm sido utilizadas em aplicações para imagens médicas. No entanto, a tecnologia de compressão de imagens também tem evoluído significativamente. O JPEG2000 provê taxas de compressão mais altas que o JPEG com menos distorção e degradação da imagem [Sung 2002]. O JPEG2000 também provê características muito interessantes para a área de imagens médicas, tais como codificação especial de Regiões de Interesse, capacidade inerente de múltiplas re-

soluções e decodificação progressiva. Desse modo, a ACR-NEMA adotou recentemente o JPEG2000 como um dos esquemas de compressão utilizados pelo padrão DICOM, o que deve levá-lo a ser um esquema de compressão amplamente utilizado em imagens médicas nos próximos anos. No entanto, o uso de compressão implica em descompressão para a visualização. Agarwal [Agarwal 2003] propôs um sistema de decodificação do JPEG2000 em tempo adequado para ser utilizado em um PACS. Um decodificador rápido é essencial nestes sistemas, e especialmente se for utilizada recuperação por similaridade, onde não apenas uma, mas uma série de imagens serão retornadas como resposta a determinada consulta. Segundo Gur et al. *apud* [Sung 2002] para os usuários destes sistemas, se o tempo necessário para exibir uma imagem em tela cheia exceder alguns segundos os usuários considerarão o sistema ineficiente.

Um dos padrões utilizados na área médica é definido pelo padrão DICOM (*Digital Imaging and COmmunications in Medicine*). Este padrão será detalhado na seção 3.1 no capítulo 3. No entanto, destaca-se aqui a grande abrangência do padrão e a forte evolução que o mesmo tem sofrido. Por exemplo, Bortoluzzi [Bortoluzzi 2003] criou um sistema para edição e gerenciamento de relatórios médicos baseando-se na especificação de relatórios estruturados do padrão DICOM (do inglês *DICOM-Structured Report* - DICOM-SR), com a intenção de distribuir estes relatórios entre sistemas diferentes (inclusive PACS), desde que estes sejam compatíveis com o DICOM-SR. Os relatórios estruturados possuem uma série de vantagens como facilitar a busca, tradução e comparação de relatórios.

O padrão DICOM-SR possui regras que definem como os documentos estruturados (que contém informações de saúde) devem ser compostos, armazenados e transmitidos. Esses documentos podem conter referências para outras instâncias DICOM como imagens, formas de onda, arquivos de áudio, outros relatórios estruturados além de referências para regiões de interesse contidas em imagens e formas de onda. O padrão ainda faz uso de uma terminologia controlada, como forma de evitar a ambigüidade das linguagens naturais, para facilitar o entendimento automático de conteúdo, buscas e internacionalização [Bortoluzzi 2003].

Outra questão relevante para PACS é o volume de informações geradas pelos equipamentos de aquisição de imagens. Furuie [Furuie 2002] analisou um hospital cardiovascular de médio porte (500 leitos) e chegou à conclusão que são gerados em torno de 20TB em um período de 10 anos. Assim, deve-se considerar seriamente os custos relativos à capacidade de armazenagem necessária para tais sistemas. Para resolver este problema, Furuie propôs uma arquitetura que combina modos *online*, *nearline* e *offline* em um fluxo assíncrono de dados. Este fluxo é controlado por um sistema gerenciador com base em informações sobre os dados armazenadas em um banco de dados. O sistema *online* é baseado em um sistema de discos magnéticos com RAID-5, e possui tempo de acesso clinicamente aceitável (menos de 3 segundos) utilizando-se uma infraestrutura de rede rápida (*fast Ethernet* e *Gigabit Ethernet*). Para o sistema em *nearline*, utilizou-se uma *jukebox* de DLT (*Digital Linear Tapes*)

com 48 *slots* e 4 *drives*. Estes múltiplos *drives* são necessários para permitir operações de armazenagem e leitura simultâneas, melhorando assim o tempo de resposta aos clientes. E finalmente, o sistema *offline* é simplesmente a organização das fitas, que saem do sistema *nearline*, em estantes. O sistema de gerenciamento é realizado com base em uma tabela armazenada em um SGBD, que possui os seguintes campos:

ONLINE_STATUS - Booleano: indica se o DICOM está *online* ou não;

ONLINE_PATH - String: indica o caminho do diretório onde o exame em DICOM se encontra ou *null* (caso o exame não esteja no sistema *online*);

NEARLINE_LABEL - String: identificação da mídia *nearline* (identificação da fita) ou *null* (caso o exame não esteja no sistema *nearline*);

OFFLINE_LABEL - String: identificação da mídia *offline* (identificação da fita) ou *null* (caso o exame não esteja no sistema *offline*).

Para cada exame DICOM é gerada uma entrada nesta tabela que possui os campos de controle acima. O fluxo é conduzido de modo que sempre haja uma margem de segurança vazia no sistema *online*. Quando esta margem é ultrapassada os exames mais antigos no sistema são gravados em uma fita (toda uma fita é preenchida) e o espaço do *online* é liberado. O mesmo esquema é utilizado para a retirada de uma fita do sistema *nearline* e inserção no sistema *offline*.

Para a recuperação de um exame, o tempo de acesso varia de alguns segundos, se o mesmo estiver *online*, a 3 minutos se estiver *nearline*. Os exames recuperados do sistema *nearline* são marcados com vida útil de 2 dias no sistema *online*. Para o sistema *offline* a recuperação envolve um operador humano e o carregamento da fita no sistema *nearline* deve ser agendado. Os exames são colocados em *nearline* e cabe ao usuário que realiza a requisição acessar o exame em *nearline* para trazê-lo para *online* como é realizado para as imagens que já se encontram em *nearline*.

Tanto Furuie [Furuie 2002] quanto Li [Li 2003] optaram por armazenar os exames DICOM fora da base de dados, armazenando somente o caminho do arquivo DICOM na tabela que registra os exames. Esta abordagem pode levar a inconsistências provenientes da falta de controle do SGBD sobre os arquivos do Sistema Operacional.

2.5 Considerações finais

A disponibilização de acesso a um PACS na Internet abre grandes possibilidades de uso, visto que, por exemplo, uma junta médica pode ser formada para analisar determinado caso e os médicos não precisam estar reunidos fisicamente para emitir seus pareceres. Há ainda

a possibilidade de um determinado exame poder ser analisado pelo médico que acompanha seu paciente, sem que este esteja no hospital em um caso de emergência. O exame pode ainda, ser laudado por um especialista mesmo que este não esteja presente no hospital naquele momento. Por fim, os dados estarão acessíveis para as faculdades de medicina que se encontram associadas ao hospital e assim, estes dados podem ser utilizados em aulas expositivas como no sistema criado por Nikolaidou [Nikolaidou 2003].

No entanto, os dados e os exames dos pacientes precisam estar seguros, logo, é necessário que se garanta o controle de acesso e o sigilo durante a transmissão destes dados. Para isso, faz-se necessário o controle de acesso com a criação de usuários autenticados mediante senha e o uso de um protocolo de transmissão seguro.

Outra questão importante é a eliminação da subjetividade do método de recuperação dos PACS tradicionais e isto pode ser resolvido através da recuperação baseada em conteúdo por meio de um PACS que inclua tais recursos.

Tecnologias de suporte

Neste capítulo discute-se as principais tecnologias que suportam os sistemas cbPACS. A seção 3.1 discorre sobre o padrão DICOM e formato de arquivo empregado para a comunicação de exames médicos. Na seção seguinte, apresenta-se o conceito de busca por similaridade, estruturas de indexação e extratores de características.

3.1 Padrão DICOM

O padrão *Digital Imaging and COmmunications in Medicine* (DICOM) foi definido por um comitê de trabalho composto por membros do *American College of Radiology* (ACR) e do *National Electrical Manufacturers Association* (NEMA). Os trabalhos tiveram início em 1983 com a formação do comitê e a primeira versão do padrão foi lançada em 1985 sob o nome de ACR-NEMA 300-1985, e logo após, em 1988, surgiu a segunda versão (ACR-NEMA 300-1988). No entanto, em 1993, uma terceira versão, já sob o nome de DICOM 3.0, foi lançada com alterações de conteúdo e em alguns novos processos, especialmente no protocolo de comunicação em rede. Foram discutidos e modificados também alguns dos problemas das versões anteriores.

O DICOM foi criado com o intuito de padronizar o formato de geração e comunicação de imagens médicas, visto que, havia grande variedade de padrões proprietários que dificultavam sobremaneira a integração dos equipamentos de geração de imagens médicas aos sistemas hospitalares. Atualmente, o DICOM é utilizado pelos mais diversos equipamentos de geração de imagens médicas. No entanto, ainda há uma série de versões diferentes e os sistemas têm que se adaptar a estas versões.

Inicialmente o padrão deveria tornar transparente a comunicação das imagens médicas

sem que fosse necessário levar em consideração o fabricante do aparelho que a gerou. Deveria ainda permitir a criação de uma base de dados de informações de diagnósticos que pudesse ser alimentada por um grande número de aparelhos diferentes e distribuídos geograficamente, além de, tornar possível o desenvolvimento de PACS para que, através destes, fosse possível a interconexão com os mais diversos sistemas hospitalares já implantados nos hospitais.

Apesar de o padrão estar bem completo, podem ocorrer mudanças devido a grande evolução por que passa a área de imagens médicas, como o surgimento e a evolução de diversos aparelhos e áreas da medicina. Estas mudanças devem ser facilmente incorporadas ao padrão porque ele apresenta estrutura em multi-partes e deste modo pode-se adicionar partes sem modificar as que já estão implantadas.

Para os sistemas de informação as principais partes do padrão são:

Objeto de informação: que especifica como os atributos que definem o objeto armazenado no padrão DICOM estão organizados; e

Especificação das classes de serviços: que contém as especificações das classes de serviços providas pelo objeto armazenado segundo o padrão, sendo que as classes de serviço são:

- *Certification Service Class*
- *Storage Service Class*
- *Query/Retrieve Service Class*
- *Study Content Notification Service Class*
- *Patient Management Service Class*
- *Study Management Service Class*
- *Results Management Service Class*
- *Print Management Service Class*

Estas partes podem ser vistas como os atributos (informações do objeto) e os métodos (classes de serviços) especificados por uma classe (Teoria de orientação a objetos) e instanciados por um determinado objeto. Assim, um exame estruturado pelo padrão DICOM é um objeto instanciado segundo uma classe DICOM.

3.1.1 Definições propostas pelo padrão DICOM

- **Atributo (*Attribute*):** é uma propriedade que compõe o Objeto de Informação, como um atributo de um objeto em orientação a objetos. Cada atributo é composto por um nome e um valor;

- Comando (*Command*): corresponde a uma chamada de método sobre um objeto, mas é conduzida por uma chamada sobre rede, sendo equivalente a uma chamada de método ou procedimento remota como *Remote Method Invocation* (RMI) de Java ou uma chamada *Remote Procedure Call* (RPC);
- Elemento de comando (*Command Element*): corresponde a um parâmetro de comando, ou seja, é como um ponteiro para o valor de determinado parâmetro de um dado comando;
- Fluxo de comandos (*Command Stream*): é um conjunto de comandos acionado sobre um exame no padrão DICOM;
- Descrição de compatibilidade (*Conformance Statement*): é uma descrição formal de como uma implementação específica do DICOM se adequa ao padrão, ou seja, especifica que Classes de serviço, Objetos de informação e protocolos de comunicação determinada implementação suporta;
- Dicionário de dados (*Data dictionary*): registra como os elementos de dados são compostos, nomeados, que valores assumem e sua semântica;
- Elemento de dado (*Data element*): é uma unidade de informação definida segundo uma única entrada no Dicionário de dados;
- Conjunto de dados (*Data set*): as informações que são trocadas no padrão DICOM devem estar estruturadas segundo a composição de um Objeto de informação. Isto é, um aglomerado de valores que representam atributos de um determinado Objeto de informação. Este aglomerado é denominado Conjunto de dados;
- Fluxo de dados (*Data stream*): é uma seqüência de dados codificados segundo o padrão DICOM e que obedecem aos dados definidos no Dicionário de dados;
- Objeto de informação (*Information object*): é a abstração de uma entidade do mundo real (*CT Image*, *Study*, etc.) que é manipulada segundo comandos DICOM;
- Classe de objetos de informação (*Information object class*): é a declaração de uma classe (Teoria de orientação a objetos) para um objeto de informação;
- Instância de objeto de informação (*Information Object Instance*): é uma instância (ocorrência) para uma definição especificada por uma Classe de objetos de informação que possui valores associados aos atributos especificados pela Classe;
- Mensagem (*Message*): é utilizada para a cooperação de duas aplicações DICOM e é especificada por um Fluxo de comandos e opcionalmente também por um Fluxo de dados;

- Classe de serviço (*Service class*): é uma descrição estruturada de um serviço provido por um objeto de informação.

3.1.2 Funcionamento do DICOM

Nota-se claramente pela análise do padrão DICOM que ele especifica como devem ser criadas Classes (Teoria de orientação a objetos) para o armazenamento, manipulação e comunicação de exames médicos gerados a partir de aparelhos de geração de imagens médicas.

O Objeto de informação e as classes de serviço especificam, respectivamente, como o objeto está estruturado e quais as classes de métodos estarão disponíveis para a manipulação dos objetos.

As demais partes do padrão especificam a codificação das informações, questões relativas a protocolos de transmissão em rede e outras questões mais específicas que fogem do domínio e interesse deste trabalho e são detalhadas em Bueno [Bueno 2002].

3.1.3 Objetivos da versão 3.0 e vantagens de utilização do padrão

Os objetivos principais do padrão são:

- Representar a semântica dos comandos e dados associados, de modo que, os equipamentos possam interagir através de um protocolo comum;
- Especificar como a implementação se adequa ao padrão, e assim, tornar claro os detalhes de possíveis divergências de implementação em equipamentos distintos para que se possa resolver problemas de interconexão;
- Tornar as operações em rede mais fáceis sem a necessidade de um mecanismo de interface de rede;
- Utilizar sempre que possível padrões internacionais para que não seja necessário reconstruir algo que já foi especificado e/ou implementado;
- Ser flexível para a inserção de novos serviços de suporte a futuras aplicações em imagens médicas, pelo uso de estruturação em camadas.

Com a inserção do padrão DICOM, pôde-se resolver o problema da grande diversidade de formatos das imagens médicas geradas pelos equipamentos, propiciando assim, a criação de vários PACS que utilizam-se do padrão para obter o que se julga necessário de cada exame.

3.2 Consulta por similaridade

O conceito de consulta por similaridade considera quanto dois dados são semelhantes entre si. Para medir esta semelhança utiliza-se de uma função de distância ($d()$) adequada ao domínio dos dados. Esta função recebe dois objetos (O_i e O_j) pertencentes ao mesmo domínio (D) e retorna o grau de dissimilaridade destes objetos. Se O_i e O_j forem idênticos, a função $d()$ retorna zero, se eles não forem idênticos, o valor será um inteiro positivo que será tão grande quanto mais distantes estiverem O_i e O_j .

Para suportar uma consulta por similaridade utiliza-se estruturas de índice que podem tornar este processo mais ágil.

3.2.1 Estruturas de índice

Os SGBDs comerciais não suportam consulta por similaridade sobre dados complexos como imagens. No entanto, várias estruturas de indexação para dados complexos já foram propostas. Elas se dividem basicamente em dois tipos:

- Estruturas espaciais;
- Estruturas métricas.

As *estruturas espaciais* (*Spatial Access Method* - *SAM*) são a primeira abordagem para indexação de dados multidimensionais com poucas dimensões e número de dimensões definido. Esta área teve seu início com o surgimento das R-Trees [Guttman 1984] seguindo-se com variações e adaptações [Sellis 1987, Beckmann 1990, Hellerstein 1995, Papadias 1995].

Como as estruturas espaciais não obtiveram sucesso com dados de alta dimensionalidade e adimensionais [Ciaccia 1997, Ciaccia 1998, Ciaccia 1999], sugeriram as estruturas métricas (*Metric Access Method* - *MAM*).

A primeira estrutura de indexação utilizando somente função distância para comparar os dados foi proposta por Burkhard [Burkhard 1973] e motivou vários outros trabalhos de desenvolvimento de estruturas, como as árvores “*fixed query*” [Bacza-Yates 1994], a *mvp-tree* (*multi-vantage point tree*) [Bozkaya 1997, Bozkaya 1999], a *vp-tree* [Uhlmann 1991] (*vantage-point* [Chiuch 1994]), a GNAT [Brin 1995], a M-Tree [Ciaccia 1997] e a Slim-Tree [TrainaJr 2000]. Tais estruturas têm se mostrado promissoras em sistemas de recuperação de dados baseados no conteúdo dos próprios dados [Bozkaya 1999]. Além disso, os MAM tem suporte nativo para consultas por similaridade e também para dados adimensionais e de alta dimensionalidade. Desta forma, utilizar um MAM para indexar imagens, ou mais propriamente, as características que foram extraídas das imagens suportando consultas por similaridade tem-se mostrado bastante apropriado [Bueno 2002]. A estrutura de indexação é construída segundo o MAM, utilizando-se das características previamente extraídas das

imagens e a função distância (para estabelecer um parâmetro de comparação entre as características). Este procedimento deve corresponder à comparação entre as imagens originais [Traina Jr 2000, Traina 2001].

Visto que, as características da imagem são a base para a construção da estrutura de indexação, pode-se perguntar como resolver o problema de duas imagens que possuem determinada característica igual. No entanto, este problema possui pelo menos três soluções factíveis. A primeira é a utilização de um filtro para a seleção de um segundo conjunto de características a ser utilizado em caso de empate. A segunda solução é o pior caso, em que se compara as imagens pixel a pixel para diferenciá-las, o que devido a alterações, como por exemplo, escala e rotação poderá levar a resultados incorretos. E, finalmente, uma terceira solução é utilizar o OID que estará armazenado na estrutura de indexação juntamente com o conjunto de características da imagem para diferenciá-las, pois o OID é a chave primária da imagem no SGBD e, portanto, um identificador único.

3.2.2 Definições

Um *espaço métrico* é definido pelo par $M = (D, d())$, onde D é o domínio do conjunto de características dos objetos (utilizado como chave na construção da estrutura de indexação) e d é uma função de distância métrica adequada ao domínio D .

Para que $d()$ seja uma função métrica ela deve possuir as seguintes características:

Simetria: $\forall O_i, O_j \in D \Rightarrow d(O_i, O_j) = d(O_j, O_i)$;

Não negatividade: $\forall O_i, O_j \in D | O_i \neq O_j \Rightarrow 0 < d(O_i, O_j) < \infty$ e $d(O_i, O_i) = 0$;

Desigualdade triangular: $\forall O_i, O_j, O_k \in D \Rightarrow d(O_i, O_j) \leq d(O_i, O_k) + d(O_k, O_j)$.

Existem basicamente dois tipos básicos de consulta por similaridade: consulta por abrangência (*range query*) e consulta pelos k vizinhos mais próximos (*k nearest-neighbors query*) [Traina 2001].

Consulta por abrangência

Dados um conjunto de objetos $O = \{O_1, O_2, \dots, O_n\}$ com $O_1, O_2, \dots, O_n \in D$, uma função distância métrica $d()$, um objeto de consulta $Q \in D$ e uma abrangência máxima de busca $r(Q)$, a consulta por abrangência:

$$range(Q, r(Q)) = \{O_i | O_i \in O \text{ e } d(O_i, Q) \leq r(Q)\}$$

seleciona todos os objetos O_i do conjunto O , tal que a distância de O_i para Q seja menor ou igual a abrangência da consulta ($r(Q)$). Exemplo: "Encontre as estrelas que estão até 10 anos-luz de distância do Sol". Nessa consulta, o objeto foco da consulta é o Sol, o domínio D é o conjunto de estrelas do Universo, e a abrangência ou raio de busca (distância máxima)

é 10 anos-luz. A distância utilizada é a medida astronômica que mede o espaço em anos-luz [Traina 2001].

Consulta pelos k vizinhos mais próximos

Dados um conjunto de objetos $O = \{O_1, O_2, \dots, O_n\}$ com $O_1, O_2, \dots, O_n \in D$, uma função distância métrica $d()$, um objeto de consulta $Q \in D$ e um inteiro $k \geq 1$, a consulta pelos k vizinhos mais próximos (k-nearest neighbor - k-NN):

$$k - NN(Q) = \{A_i | A_i \in A, A \subseteq O, |A| = k \text{ e } \forall A_i \in A, O_i \in O - A, d(Q, A_i) \leq d(Q, O_i)\}$$

seleciona os k objetos do conjunto O que estão mais próximos de Q .

3.2.3 Extratores de características

As imagens possuem diversas características, dentre elas atributos de forma, textura e histograma de cores.

O processo de extração de características é uma das fases de reconhecimento de padrões, que segundo Castelman [Castelman 1996] possui três etapas.

A primeira etapa é a *segmentação da imagem*, que tem como objetivo o isolamento de cada objeto que pertence à imagem. Em seguida, extrai-se as características dos objetos identificados na fase anterior. A *extração de características* busca a composição de um vetor de valores que descreva o objeto. Este vetor, terá uma dimensão n , e identificará o objeto em um espaço n -dimensional, este espaço é chamado de espaço de características. É importante que as características selecionadas para compor o vetor que identifica o objeto sejam o mais invariantes possível com relação a variações de escala, translação e rotação, visto que, estas transformações não modificam o objeto transformando-o em outro, mas apenas modificam seu tamanho e posição. A terceira e última etapa, é a *classificação da imagem*. Nesta etapa, cada objeto pertencente à imagem é classificado em um dos grupos de objetos pré-definidos que se espera que estejam presentes na imagem.

Uma taxonomia proposta por Brown [Brown 1992], sugere que há cinco espaços de características (ver tabela 3.1).

Para se utilizar das características extraídas das imagens como chave para a construção da estrutura de indexação deve-se utilizar as características dos espaços de bordas ou de alto nível, pois estes são mais efetivos para a seleção como assinatura das imagens. Os espaços estatísticos representam características que definem o comportamento global das imagens e são mais adequados para os primeiros passos de seleção e eliminação de candidatos. Já os espaços de intensidade bruta são adequados para as comparações de igualdade entre imagens por representar a imagem em nível de pixels, e por isso não são utilizados em consultas por similaridade.

Espaço de características	Atributos
Intensidade real ou pura	Baseia-se nas intensidades dos pixels.
Bordas	Estrutura intrínseca, menos sensível a ruídos. Inclui bordas propriamente ditas, contornos e superfícies.
Características salientes	Estrutura intrínseca, posicionamento preciso. Inclui pontos de curvatura máxima locais sobre linhas de contorno, centros de janelas com variancia máxima local, centros de massa de regiões bem delimitadas, intersecção de linhas e descritores de Fourier.
Características estatísticas	Utiliza toda a informação presente na imagem, bons resultados para transformações rígidas, previsível em relação ao espalhamento espacial. Inclui momentos invariantes, eixos principais e centróides.
Características de alto nível	Utilizam relações e informações de alto nível, bons resultados para <i>matching</i> local e impreciso. Inclui características estruturais (grafos de configurações de sub-padrões), sintáticas (gramáticas compostas a partir de padrões) e redes semânticas (regiões de cena e suas relações). Podem criar várias técnicas de assinatura para indexar imagens.
Encaixe em modelos pré-definidos	Características intrínsecas acuradas, apresenta ruído em conjuntos de somente uma imagem. Inclui atlas anatômicos, mapas geográficos e modelos de objetos.

Tabela 3.1: Taxonomia sobre espaços de características extraídas de imagens

3.2.4 Características visuais

As características visuais tais como: cor, forma e textura, são naturalmente candidatas a serem utilizadas para a indexação de imagens, visto que, os seres humanos utilizam-se das mesmas como forma básica de comparação entre imagens.

Cor

As cores podem ser representadas por vários modelos, como o RGB (*red, green, blue*), que é o modelo mais simples, pois realiza o mapeamento direto das características do dispositivo de exibição, ou por modelos mais complexos como o HSI (*hue, saturation, intensity*) que se aproxima mais do modo de percepção de cores dos humanos.

Um modelo de cor é, basicamente composto de um sistema de coordenadas de três dimensões e um subespaço contido neste sistema, no qual cada cor pode ser representada por um ponto. Os modelos de cores foram propostos para padronizar a representação de cores.

No modelo RGB, as cores são representadas por uma mistura de quantidades específicas de vermelho, verde e azul. Utilizando-se 24 bits (8 bits para R, 8 para G e 8 para B) para a representação de 1 pixel em RGB, consegue-se representar aproximadamente 16,7

milhões de cores diferentes.

A identificação de uma imagem através da característica cor é realizada, geralmente, pela construção de um histograma em que são calculados para cada cor o número de pixels da imagem com tal cor. Deste modo, quando a representação da imagem permite um grande número de cores, as imagens tem que passar por um processo de re-quantização de cores para que o processamento da imagem possa ser realizado de forma mais factível e eficiente.

Para comparar computacionalmente duas imagens segundo suas cores, utiliza-se os histogramas de cores das imagens e a comparação é realizada pelo somatório das diferenças absolutas ou quadráticas sobre o número de pixels de cada cor. Este processo é computacionalmente simples e barato, invariante a pequenas alterações na imagem (movimentações) e diferencia bastante bem objetos distintos. Deste modo, este método de comparação tem sido amplamente estudado e empregado tanto em sistemas comerciais como acadêmicos.

No entanto, este método apresenta alguns problemas:

Ambiguidade: Quando leva-se em consideração a imagem como um todo, mesmo sendo diferentes elas podem possuir histogramas de cores muito semelhantes. Deste modo, este método deve ser empregado em conjunto com outros para elevar a acuracidade da diferenciação das imagens.

Número de cores: Frequentemente o número de cores pode ultrapassar 100 níveis. A indexação de histogramas com esta profundidade de níveis deve contemplar pontos 100-dimensionais, o que se torna um grave problema para as estruturas espaciais de indexação pois causa um colapso na maioria destas estruturas. Este problema se deve à chamada “maldição da alta dimensionalidade” [Hinneburg 1999a, Pagel 2000] segundo a qual, a partir de determinada dimensão k , as estruturas de indexação espaciais apresentam desempenho pior do que a simples busca seqüencial.

Textura

Textura é um padrão visual composto por um grande número de elementos arranjados de forma equânime com densidades variadas. Cada elemento de textura é uma região de intensidade uniforme com forma simples que se repete dentro de um intervalo.

A análise de uma textura pode ser de dois tipos:

Estatística: onde a análise é realizada por janelas ou intervalos. Este tipo de análise é aplicada a texturas pequenas e não muito regulares [Tomita 1990]. Tomando uma janela da textura, analisa-se a variação de intensidade. Exemplos de análise estatística são: contraste (pele de zebra - grande contraste, pele de elefante - baixo contraste), granularidade (tamanho dos elementos) e direcionalidade (estampa de padrão xadrez em um tecido em oposição um padrão liso). Um modo de manipular estes padrões é

através do espectro de Fourier, com o qual consegue-se gerar uma assinatura mediante a aplicação da transformada sobre a janela. O agrupamento das janelas é realizado por comparação de suas assinaturas, assim, se duas janelas possuem assinaturas próximas ou similares elas podem ser agrupadas.

Estrutural: análise dos elementos da textura. Aplica-se a texturas que possuem elementos que seguem alguma organização espacial bem comportada e previsível. Com a análise estrutural obtém-se os elementos da textura da imagem, seus formatos e estima-se regras de posicionamento. Estas regras descrevem como os elementos são em relação aos demais elementos, indicam vizinhanças (conectividade dos elementos), a densidade (número de elementos por unidade espacial) e sua homogeneidade ou regularidade.

As texturas são tratadas de modo diferente do das cores, pois as cores levam em consideração pixels, enquanto que a textura é tratada em janelas ou regiões. Depois de segmentada por textura, a imagem estará dividida em regiões de texturas uniformes envolvidas por retângulos (*bounding boxes*), os quais podem ser utilizados para construir uma estrutura de indexação tipo R-Tree. No entanto, a textura apresenta o mesmo problema que o histograma de cores, pois uma imagem pode possuir várias texturas e gerar elementos de alta dimensionalidade incorrendo no problema da “maldição da alta dimensionalidade”. Outro problema ainda é a ambigüidade, visto que, duas ou mais imagens podem apresentar texturas bastante similares o que torna difícil a diferenciação das mesmas.

Um exemplo de utilização de textura para a recuperação de imagens é descrito por Felipe [Felipe et al. 2003]. Ele desenvolveu um método para a geração de assinatura de uma imagem baseada na textura de tecidos do corpo humano, ao qual chamou de descritor de gradiente. O descritor de gradiente mede a suavidade da transição através dos pixels da imagem. Além do método, foi desenvolvida também uma ferramenta de testes para a validação do método e através de *precision vs. recall* detectou-se que o método alcança resultados superiores a 90% de precisão para pesquisas que envolvam até de 20% da base de dados.

Forma

O atributo visual mais difícil de ser tratado pelos sistemas de recuperação de imagens baseadas no seu conteúdo é a forma. Esta dificuldade se relaciona ao problema de segmentar objetos de interesse presentes na imagem. Assim, a recuperação por forma é tipicamente limitada aos poucos objetos mais bem discriminados na imagem.

Para utilizar esta característica, faz-se necessário um pré-processamento para selecionar os objetos que estão presentes na imagem que será indexada. Este pré-processamento

faz uso de algoritmos que são dependentes do domínio da aplicação das imagens, por exemplo, em imagens que possuam objetos tais como tumores cerebrais e lesões de pele, utiliza-se algoritmos diferentes dos que se utilizaria para detectar objetos como carros e aviões. Uma das diferenças dos algoritmos do exemplo acima é que no primeiro caso, não pode-se utilizar algoritmos que encontrem formas pré-definidas, enquanto que no segundo, os objetos possuem formas baseadas em geometria e neste caso, algoritmos para busca de modelos pré-definidos obtêm sucesso no reconhecimento das informações morfológicas. Destaca-se ainda, a necessidade da utilização de algoritmos para a eliminação de ruídos na etapa de pré-processamento de imagens mais complexas.

Após a etapa de pré-processamento, tendo os objetos sido encontrados, utiliza-se algoritmos de perseguição de contornos para a determinação de suas bordas. No entanto, para cenas complexas, a determinação de bordas é um processo complicado, visto que, oclusões parciais, ruídos e sombras tornam o processo mais complicado e custoso.

A representação da característica forma para os objetos presentes nas imagens é realizada através de vetores, mas estes podem possuir dimensões distintas. Deste modo, para realizar a comparação destes vetores é necessário o emprego de uma função de dissimilaridade, o que torna tal característica potencialmente “indexável” por uma estrutura de indexação métrica. Outra solução para este problema é a utilização de formas mais simples para aproximar as formas encontradas, como é o caso da triangularização e da aproximação por retângulos de contorno (*bounding boxes*), que podem ser utilizados para representar formas irregulares. Esta segunda solução possui espaço de armazenagem menor e modo de comparação mais simples que a primeira. Mas em última instância faz-se necessária a manipulação do objeto irregular em si.

Long [Long 2003] descreve um sistema que utiliza a recuperação baseada em conteúdo segundo a característica forma. Nesse sistema, o usuário pode entrar com um esboço da forma procurada ou com uma imagem de exemplo, e o sistema retorna as imagens mais similares.

3.2.5 Comparando imagens

Utilizando-se as características extraídas das imagens e uma margem de tolerância é possível compará-las e dizer o quão similares duas imagens são.

Os vários procedimentos de extração de características devem ser escolhidos conforme as variações presentes nas imagens e a modalidade ou domínio ao qual a imagem pertence (por exemplo Raios-X, ToRM, etc), da área considerada na imagem (tórax, crânio, membros, abdome, etc), da patologia considerada (tumores, fraturas, lesões, etc), sendo necessário o estudo da aplicação de cada extrator de que se disponha para os determinados domínios de imagens que se queira indexar.

O sistema iRIS

As várias tecnologias de suporte à implementação de um cbPACS motivaram a intensa modularização no desenvolvimento do sistema iRIS.

Nas seções seguintes, expõe-se as funcionalidades e metas deste projeto, a arquitetura do iRIS e como cada componente do sistema foi desenvolvido. Em seguida, questões relativas à estruturação e a população da Base de dados são discutidas e elucidadas. Encerra-se este capítulo explicitando como se dá o relacionamento de todos os componentes do sistema.

4.1 Funcionalidades e objetivos do iRIS

- Construir um protótipo totalmente funcional de um PACS integrado com CBIR com acesso através da Web, facilitando assim, o acesso ao sistema aumentando e difundindo com isso sua utilização;
- Integrar os vários trabalhos do Grupo de Bases de Dados e Imagens, tais como as estruturas de indexação do *framework* Arboretum, alguns extratores de características e funções distância;
- Gerar um sistema facilmente extensível, através de modularização, facilitando assim, o acoplamento de novos trabalhos desenvolvidos no grupo;
- Mostrar que é possível possuir um sistema centralizado no hospital, que inclua tanto informações administrativas e de controle de processos (presentes nos Sistemas de Informações Hospitalares - SIH), quanto as informações sobre exames e suas imagens (que tradicionalmente ficam isoladas em sistemas do setor de radiologia [Seckman 2004]);

- Melhorar o ensino de técnicas e análise de exames, através da utilização do sistema em salas de aulas e laboratórios de faculdades de medicina e hospitais escolas;
- Auxiliar o diagnóstico de enfermidades (Computer Aided Diagnosis - CAD) através da busca de imagens similares, aproveitando o conhecimento já empregado em exames anteriores, através da comparação de laudos e imagens já armazenadas no sistema.

4.2 Arquitetura

A arquitetura do iRIS encontra-se ilustrada na figura 4.1. Segundo a arquitetura cliente/servidor o cliente deve ser enxuto e leve e o servidor robusto. Este propósito foi alcançado, visto que, o cliente nesse sistema é apenas o navegador Web, com a capacidade de rodar aplicativos java (*applets*) para a extração de características local.

Nota-se ainda, a alta modularidade do sistema que tem por objetivo, não somente a independência e o baixo acoplamento entre os módulos visando a manutenibilidade do sistema, mas também a possibilidade de integração, fácil e sem a necessidade de parada do sistema, de novas estruturas de indexação, extratores de características e funções distância.

Por fim, observa-se a possibilidade de distribuição de carga no sistema, através da execução dos servidores em máquinas distintas, tornando o sistema escalonável para diversas tipos de demanda.

Cada um dos componentes do sistema será detalhado nas seções que se seguem.

4.3 Servidor de indexação

Os SGBDs atuais possibilitam a criação de estruturas de indexação para seus tipos de dados nativos. Estas estruturas de indexação são utilizadas com a finalidade de recuperar mais rapidamente as informações que estão vinculadas as chaves utilizadas na indexação. No entanto, imagens não são tipos nativos do SGBD e ainda não há uma álgebra que possibilite o tratamento de imagens como tal.

Para resolver este problema são utilizadas estruturas de indexação especiais, baseadas em funções distância e características extraídas das imagens (vide capítulo 3 seção 3.2). O GBdI, possui um framework denominado Arboretum, que especifica uma interface para o desenvolvimento de estruturas de indexação métricas, que são utilizadas para recuperar os objetos mais similares a um dado objeto de consulta (centro da consulta).

O servidor de indexação tem por objetivo possibilitar o acesso a várias estruturas de indexação, baseadas no Arboretum, de forma centralizada, isto é, o servidor de indexação funciona como uma extensão do SGBD, fornecendo o serviço de recuperação baseada em conteúdo.

Figura 4.1: Arquitetura do iRIS, onde destaca-se o servidor de indexação em relação à figura 2.4

Nas sub-seções que se seguem são explicitados os desafios que foram encontrados na construção do servidor de indexação e como cada um deles foi resolvido.

4.3.1 Servidor

O servidor foi construído de forma a lidar com tipos de estruturas de indexação e suas instâncias de forma centralizada e concorrente.

Para isso, cada tipo de estrutura foi idealizado como sendo um *plugin* para o servidor, que o capacite a lidar com este tipo de estrutura. A seguir são expostas as questões referentes ao desenvolvimento dos *plugins* e a concorrência no acesso aos tipos e instâncias das estruturas de indexação.

Plugins

A área de extração e tipos de características de imagens e de funções distância, para medir a dissimilaridade dessas características, está em plena efervescência, visto que, para cada

domínio de imagens e para cada aplicação determinada combinação característica/função distância pode apresentar resultados mais acurados do que outra.

Para que fosse possível adicionar e retirar essas combinações do sistema de maneira fácil, sem necessidade de parada e sem influenciar o sistema como um todo, foi incluída uma estrutura de *plugins* no servidor.

A interface e a estrutura dos *plugins*, foi baseada na atual estrutura do *framework* de desenvolvimento de árvores métricas - Arboretum. Desse modo, cada *plugin* representa um tipo de estrutura de indexação para o servidor. O tipo é definido pela trinca {Árvore, Tipo de característica, Função distância}. Para políticas de árvores pode-se citar a Slimtree [TrainaJr 2000] ou a DBMTree [Vieira et al., 2004]. Como exemplo de tipo de característica têm-se histograma de cor, histograma métrico [Bueno 2002], medidas de textura baseadas em matrizes de co-ocorrência [Félice et al. 2003]. Já no âmbito de funções distância, a distância euclidiana para histogramas ou a diferença de áreas definidas pelos histogramas métricos [Bueno 2002] são algumas das possibilidades.

Definido determinado *plugin*, ou seja, elegendo uma trinca compatível, é gerada uma biblioteca de vínculo dinâmico (*Dynamic Link Library* (dll) no Windows ou *Shared Objects* (so) no Linux) que pode ser carregada em tempo de execução no servidor, não afetando em nada os outros tipos já em execução. Para que este processo fosse possível foi definida uma interface que todos os *plugins* devem seguir.

A partir do momento em que um determinado tipo (*plugin*) é inserido no servidor de indexação passa-se a ter a possibilidade de instanciação desse tipo, ou seja, a criação de uma estrutura de indexação baseada na trinca definida pelo tipo.

As interfaces

Como dito acima, o próprio Arboretum define os requisitos mínimos necessários para a recuperação de objetos por similaridade. A estrutura de indexação métrica é definida pelas classes `stMetricAccessMethod` e `stMetricTree` (figura 4.2), as características segundo a classe `stObject` (figura 4.3) e finalmente as funções distâncias segundo a classe `stMetricEvaluator` (figura 4.4).

As classes `stMetricAccessMethod` e `stMetricTree` possuem, como observa-se na figura 4.2, parâmetros (*templates*) que são classes filhas das classes `stObject` e `stMetricEvaluator`, ou seja, possuem a mesma interface definida no topo da hierarquia, o que tornou possível utilizá-las sempre da mesma maneira, não importando que tipo de especialização ocorre hierarquia abaixo.

A estrutura de *plugins* também necessita de uma interface comum a todos eles (figura 4.5) e foi definida através da utilização de funções para criar uma estrutura de indexação (`constructIndex`), destruí-la (`destroyIndex`), para criar um novo objeto do tipo da característica sendo indexada (`newObject`), e duas funções de informações, uma para gerencia-

Figura 4.2: Esquema UML de políticas de estruturas de indexação

Figura 4.3: Esquema UML dos objetos indexáveis

mento do tipo definido pela trinca (information) e outra para acesso a dados estatísticos da estrutura de indexação (statistics).

Na figura 4.5, observa-se que o método information retorna um objeto da classe

Figura 4.4: Esquema UML de funções distância

Figura 4.5: Esquema UML de um *plugin*

IndexMetaInformation. Este objeto representa o tipo de estrutura de indexação para o servidor, e todas as operações de manipulação de tipo são realizadas através dele. Sua representação pode ser observada na figura 4.6, onde encontra-se representada também a estrutura das instâncias do tipo.

Figura 4.6: Esquema UML de um Tipo de estrutura de indexação e de suas instâncias

Concorrência

Tanto o gerenciamento dos *plugins*, quanto o de tipos e instâncias das estruturas de indexação, é realizado sobre uma área de memória compartilhada pelos vários fluxos de execução (*threads*). O fluxo de execução principal dispara mais um fluxo de execução específico para atender cada cliente que se conecta ao servidor de indexação.

Essa área de memória compartilhada sofre condições de corrida, e portanto, precisou ser protegida por mecanismos de sincronização para que não haja corrompimento ou inconsistência dos dados lá armazenados.

O esquema de controle de concorrência deste projeto foi baseado no uso de semáforos e regiões de exclusão mútua, que somente possibilitam uma resolução não tão justa, mas completamente segura, do problema de acesso concorrente. Dessa maneira, a prioridade de acesso é dos fluxos que realizam somente a leitura da região compartilhada. Para resolver o problema de uma forma mais justa, é necessário o emprego de monitores, que incluem as requisições em filas de acesso e, assim, o acesso é conferido por ordem de requisição. No entanto, a maioria dos clientes do servidor não realizará escrita nas áreas compartilhadas, visto que, a consulta por similaridade é a operação mais freqüentemente realizada e esta não faz uso de escrita. Os fluxos que escrevem na área compartilhada são disparados somente para inclusão de novos tipos de estruturas ou inserção de objetos em instâncias de estruturas. Além disso, as regiões de exclusão mútua são pequenas e, conseqüentemente, de rápida execução. Sendo assim, o emprego de monitores torna-se uma questão apenas de tratamento igualitário entre os processos, não influenciando de maneira significativa o desempenho do sistema. O tratamento de vários fluxos de execução (*Multithreads*) e o controle de condições

de corrida sobre a memória compartilhada foi realizado com o uso da biblioteca *pthread* (*POSIX Threads*) que possui versões para Linux e Windows.

Comunicação em rede

A comunicação em redes de computadores já é uma área bem madura da computação. O emprego de sockets TCP/IP fornece comunicação ponta-a-ponta segura e conectada, de modo que, o usuário pode utilizá-los como um canal de comunicação sem grandes preocupações além de enviar e receber informações através do canal. Desse modo, a construção de sistemas segundo a arquitetura cliente/servidor pode se resumir a empregar essa estrutura de comunicação para transmitir informações de acordo com um protocolo que deve ser definido especificamente para facilitar a comunicação.

O protocolo

O servidor de indexação utiliza-se de um protocolo estruturado da seguinte maneira:

1. Comandos para o servidor

- Tamanho da mensagem (em bytes);
- Código do comando (definido pelo protocolo);
- Pares {tamanho do parâmetro - em bytes, valor do parâmetro} até o fim da mensagem.

2. Respostas para o cliente

- Tamanho da mensagem (em bytes);
- Pares {tamanho do parâmetro - em bytes, valor do parâmetro} até o fim da mensagem.

3. Respostas para o cliente (Erros)

- Tamanho da mensagem (em bytes);
- Tamanho do código de erro (em bytes);
- Código de erro.

Observa-se da estrutura do protocolo acima, que as respostas de erro têm a mesma estrutura das respostas com conteúdo útil, no entanto, o código de erro será sempre um número negativo, o que torna possível sua identificação.

As ações definidas pelo protocolo são identificadas por códigos (número inteiros). São elas:

- Carregar um determinado tipo de estrutura (*plugin*) (LOADINDEXPLUGIN)
- Descarregar um determinado tipo de estrutura (*plugin*) (UNLOADINDEXPLUGIN)
- Listar os tipos de estruturas (*plugin*) carregados e utilizáveis (LISTINDEXESTYPES)
- Criar uma instância um determinado tipo de estrutura ou carregá-la a partir de um arquivo (CREATEINDEX)
- Descarregar uma determinada instância de estrutura (DESTROYINDEX)
- Listar as instâncias carregadas de determinado tipo de estrutura (LISTINDEXES)
- Inserir um objeto em uma determinada instância de estrutura (INSERTINTOINDEX)
- Realizar uma busca por abrangência em uma determinada instância de estrutura (RANGEQUERY)
- Realizar uma busca pelos k vizinhos mais próximos em uma determinada instância de estrutura (KNNQUERY)
- Executar algum método específico da política da árvore para otimizar a estrutura em uma determinada instância de estrutura (INDEXOPTIMIZE)
- Executar algum método específico da política da árvore para exibir algumas estatísticas de uma determinada instância de estrutura (STATISTICS)

Estes comandos cobrem toda a funcionalidade genérica das estruturas de indexação métricas para a busca por similaridade e o gerenciamento de tipos e instâncias de estruturas de indexação necessárias ao servidor. Os códigos de erro foram definidos da mesma maneira.

4.3.2 Cliente

O cliente para o servidor de indexação foi construído como uma classe, visto que, ele deve ser utilizado por outras aplicações somente para realizar as buscas por similaridade, gerenciar os tipos de estruturas de indexação e as instâncias dos mesmos. Foram desenvolvidos dois clientes para o servidor de indexação, um escrito na linguagem ANSI C++ (como o servidor) para ser utilizado nos outros projetos do grupo de pesquisa, e outro utilizando-se a Tecnologia Java, para utilização do Servidor iRIS na Web.

4.3.3 Portabilidade

A portabilidade de código do servidor e dos clientes foi garantida utilizando-se somente código ANSI e ISO C/C++ e bibliotecas que possuem versão tanto para Windows como para Linux. Foram realizados testes em ambas plataformas e não foi necessário alterar nenhuma linha de código. Os testes foram realizados utilizando-se o compilador gcc no Linux e Mingw (uma versão reduzida do GNU gcc para plataforma Windows). Para o cliente Java, não foi necessário nem a compilação do código entre plataformas, visto que, Java é por definição multiplataforma.

4.4 Servidor iRIS - Web

4.4.1 Desenvolvimento Web

A construção do servidor Web demonstrou ser um desafio interessante e motivador, visto que, o desenvolvimento Web é relativamente recente e, portanto, ainda surgem novos modelos de desenvolvimento mais adequados à problemática de desenvolvimento mais adequadas as várias facetas presentes nesse tipo de projeto. Nas subseções seguintes são explicitadas algumas questões relativas à técnica de desenvolvimento empregada, é apresentada a solução presente na tecnologia Java para este tipo de aplicação. E, por fim, é apresentada a estruturação do servidor iRIS.

4.4.2 O padrão de projeto MVC

O padrão de projeto *Model View Controller* (MVC) surgiu no início da programação orientada a objetos, com a linguagem SmallTalk. Este padrão de projeto apregoa o desenvolvimento tão independente quanto possível das três estruturas presentes em aplicações: Modelo, Controle e Visão (ver figura 4.7). Deste modo, o Modelo deve lidar com questões relativas à lógica do negócio. A Visão deve se ocupar somente da interface com o usuário. Por fim, o Controlador deve realizar a tarefa intermediária de receber as requisições do usuário através da Visão e distribuir a realização dessa requisição entre os vários componentes da camada de Modelo, enviando o resultado à camada de Visão, e novamente ao usuário.

Este modelo é extremamente adequado ao desenvolvimento de aplicações em grandes negócios, visto que, tanto as aplicações tradicionais, quanto as aplicações Web podem se utilizar dos mesmos componentes da camada de Modelo, realizando reutilização de código e componentes já desenvolvidos, contribuindo para a maturidade dos mesmos. O Controlador também pode tratar dos dois tipos de aplicação, somente com a adição de comportamentos diferenciados, conforme o tipo de requisição. Finalmente, a camada de Visão é a única que deverá ser específica para cada tipo de aplicação.

Figura 4.7: Divisão em camadas segundo o padrão MVC

4.4.3 Tecnologia Java

A plataforma Java tem, desde o seu início, suportado amplamente o desenvolvimento de aplicações tanto em redes locais como sobre a Internet. Como exemplo pode-se citar a idéia de *applets* que são aplicações que executam no cliente, mas ela mesma vem para o cliente através de uma requisição sobre a rede.

No aspecto de páginas dinâmicas para a Web, houve uma abordagem inicial da plataforma que era baseada em *servlets*. Os *servlets* são programas java que respondem às chamadas segundo o protocolo HTTP (*HyperText Transport Protocol*) e que executam sobre um *container* (um servidor que fornece suporte para todos os *servlets* - Exemplo Catalina/Tomcat (<http://jakarta.apache.org/tomcat/>, última visita em 13 de dezembro de 2004)). No entanto, esta primeira abordagem gerava sistemas difíceis de manter, visto que, o conteúdo estático e dinâmico estavam misturados dentro do programa escrito em Java. Surgiu assim, a tecnologia JSP (*Java Server Pages*), segundo a qual escrevem-se as páginas dinâmicas com algumas *tags* especiais que fornecem todas as funcionalidades necessárias à criação de conteúdo dinâmico. Essa solução ainda apresentava um grande inconveniente que era a necessidade de amplos trechos de código misturados as *tags* HTML e JSP.

Para solucionar o problema de conteúdo visual (que é altamente volátil em páginas Web e deve ser criado e mantido por *Web designers*) misturado com código Java (que deve ser criado e mantido por desenvolvedores de software) surgiu a idéia de bibliotecas de *tags* (*taglibs*) em conjunto com o uso do padrão de projeto MVC. As bibliotecas de *tags* fornecem ao desenvolvedor de software o poder de criar *tags* que possuem uma funcionalidade específica, estendendo assim as *tags* de uso dos *Web designers* que, por sua vez, não precisam se preocupar com códigos Java. Deste modo, os papéis no desenvolvimento deste tipo de aplicação tornaram-se bem definidos e estanques.

A aplicação do padrão de projeto MVC também contribui enormemente para a maturidade e isolamento de módulos de sistemas Web baseados na tecnologia Java, visto que, as Visões ficam a cargo do *Web designer* e são escritas em JSP, sem necessidade de uso de código. O Modelo de negócio, ou componentes do negócio, são escritos em java, idealmente através da escrita de componentes *Java Beans* e/ou *Java Enterprise Beans* que fornecem uma ampla gama de funcionalidades, como persistência e gerenciamento de transições. Estes componentes são reutilizados por todos os softwares existentes no negócio e, assim, tornam-se maduros e estáveis rapidamente contribuindo para a manutenibilidade do sistema. Outra questão é a escalabilidade que pode ser alcançada com o emprego de tal tecnologia, pois através do uso de *caches* de objetos o sistema pode suportar uma grande carga sem perda de desempenho.

A camada intermediária, ou seja, o Controlador, é escrita como um *servlet* e fará a composição de uso dos componentes de negócio (*beans*) para atender às requisições realizadas pelo usuário do sistema.

4.4.4 Estruturação do Servidor iRIS

O servidor iRIS foi estruturado de maneira que sua manutenibilidade fosse o mais simples possível, sem que isto afetasse sua funcionalidade. Foi empregado o padrão MVC e a tecnologia Java no seu desenvolvimento, e assim alcançou-se a estrutura ilustrada na figura 4.8.

Pode-se notar claramente que as Visões (interface com o usuário) foram todas escritas em JSP. Na Camada de Controle, encontra-se o Controlador (`controler.java`), escrito em Java, que realiza as requisições feitas pelo usuário do sistema, através das Visões, fazendo uso dos objetos do negócio, que estão na Camada de Modelo. Estes componentes podem ser divididos em duas categorias, os que representam informações presentes na base de dados e os de manipulação da base de dados. Na primeira categoria encontram-se as classes `Exam`, `Patient`, `SimilarImage`, `SimilarImagesVector` e `iRISUser`. A segunda categoria é composta pelas classes `DBControler`, `IndexInstance` e `IndexInstancesVector`, que são utilizados para recuperar objetos da primeira categoria.

Como dito nas subseções 4.3.1 e 4.3.2, as buscas por similaridade são realizadas por

Figura 4.8: Estrutura do iRIS segundo o padrão MVC

intermédio de um servidor de estruturas de indexação através de um cliente para o mesmo. As consultas são feitas a partir do componente **DBControler** que encapsula todas as questões referentes ao acesso à Base de dados. Esse acesso será tratado com mais detalhes na seção 4.5.

4.5 Base de dados

O acesso à Base de dados foi idealizado de modo que um único componente o realiza. Este componente (**DBController**) faz desde consultas sobre dados primitivos do SGBD (por exemplo: números e *strings*) como as consultas por similaridade sobre imagens. Tais consultas são efetuadas através de chamadas para métodos desse componente, que por sua vez, utiliza-se de outros dois tipos de componentes. Um deles corresponde às classes necessárias à utilização de JDBC (*Java Data Base Connectivity*), ou seja, classes genéricas que são *drivers* para o acesso ao SGBD utilizando a linguagem Java. O outro é uma classe cliente para o servidor de indexação (apresentado na seção 4.3.1) através da qual se realiza a composição dos requisitos necessários a busca por similaridade sobre as imagens.

O Sistema de Recuperação de Dados e Imagens (SRDI) (ver figura 4.9) proposto originalmente foi modificado, de maneira que, foi introduzido o cliente e o servidor de estruturas de indexação no lugar antes ocupado por apenas uma estrutura. Desse modo, é possível inserir e criar novos tipos de estruturas com o sistema em execução, como foi descrito na seção sobre o servidor de estruturas de indexação (4.3.1), alcançando-se assim, tanta flexibilidade quanto possível em relação às estruturas, métricas e características utilizadas para uma busca por similaridade.

Como pode-se notar o SRDI corresponde ao componente **DBController** e utiliza a base de dados criada e alimentada pelo SADI, e também às estruturas de indexação criadas sobre tal base. O servidor de estruturas de indexação provê serviços para a manipulação das estruturas e para a busca sobre as mesmas, retornando os OIDs utilizados posteriormente para recuperar as imagens mais similares da Base de dados.

4.5.1 Processamento de uma consulta

Uma consulta por similaridade é realizada conforme os passos que podem ser observados na Figura 4.10. Nos passos 1 e 2, o usuário forneceu uma imagem para consulta por similaridade que não se encontra armazenada no SGBD. Esta imagem é submetida ao extrator de características. O extrator devolve o vetor de características ao cliente, que submete a consulta de similaridade ao servidor (passo 3). A consulta é encaminhada ao servidor de indexação (passo 4) que faz a pesquisa por similaridade no índice selecionado (parâmetro da consulta) e recupera os OIDs das imagens do resultado (passo 5). A seguir, o SRDI requisita as imagens pelos OIDs ao SGBD (Passo 6), que responde com os *thumbnails* das imagens (passo 7). Os *thumbnails*, juntamente com estatísticas de similaridade são retornados ao servidor que por sua vez retorna estes resultados ao cliente (passo 8). No entanto, se a imagem já estiver armazenada no SGBD, torna-se desnecessário submetê-la ao extrator de características. Neste caso, o processo de busca é realizado a partir do passo 3. Se a consulta

Figura 4.9: Organização do módulo SRDI

for sobre dados nativos do SGBD, o cliente encaminha a requisição diretamente ao servidor (passo 3), que reenvia a requisição ao SGBD (6), obtendo os resultados do SGBD (7), o servidor responde ao cliente (8).

4.5.2 Esquema inicial

O esquema inicial da base de dados pode ser visto na figura 4.11. Nota-se que há dados tanto de pacientes quanto dados de exames e imagens. Esse esquema foi proposto levando em conta a possibilidade de integração desse sistema com o SIH, de modo a manter todas as informações centralizadas e acessíveis de qualquer ponto dentro do hospital ou através da Internet.

O mapeamento para o modelo relacional ficaria:

Paciente(chave_paciente, nome)

Exame(chave_exame, tipo, laudo, data, chave_imagem)

Imagem(chave_imagem, descricao, thumbnail, classe, binario, patologicaS/N, janelaCorte_xSuperiorEsquerdo, janelaCorte_ySuperiorEsquerdo, janelaCorte_altura, janelaCorte_comprimento, chave_exame)

Figura 4.10: Processamento de uma consulta

Caracteristica(assinatura, chave.imagem)

Onde os atributos, `binario` e `thumbnail` em `Imagem` e `assinatura` em `Caracteristica` seriam modelados como BLOB (*Binary Large Object*). As várias características extraídas das imagens ficariam concatenadas dentro de `assinatura`.

No entanto, o mapeamento para o modelo relacional foi realizado de modo diferente, para que esta mesma base de dados e mesmo o iRIS servidor pudessem fazer uso de um projeto denominado CIRCE (*Content Image Retrieval Core Engine*) que será detalhado na subseção 4.5.3.

4.5.3 Projeto CIRCE

A arquitetura básica de uma base de dados relacional pode ser vista como sendo constituída por três bases de dados lógicas:

- A base da aplicação, que efetivamente armazena os dados da aplicação;
- A meta-base, que armazena a descrição da estrutura da base de dados da aplicação;

Figura 4.11: Diagrama entidade relacionamento da base de dados

- E a base de estatísticas, que armazena dados obtidos pelo gerenciador sobre os dados da aplicação, que são utilizados para auxiliar no processo de otimização das consultas.

A base de estatísticas e a meta-base são usualmente chamadas de dicionário de dados da aplicação, e são dependentes de cada SGBD, mas independentes das aplicações. Cada base de dados segue uma modelagem independente das demais, e essa independência garante que SGBDs com diferentes recursos possam ser utilizados para armazenar os dados de uma aplicação.

O projeto CIRCE tem como objetivo a incorporação de um novo tipo de dados nativo aos SGBDs. Este tipo de dados (STILLIMAGE) possibilitaria o tratamento de imagens pelos SGBDs de forma automática e similar aos tipos já suportados atualmente pelos SGBDs.

A arquitetura do CIRCE (figura 4.12) utiliza mais duas bases de dados lógicas para manter a independência entre os vários recursos envolvidos (suporte aos dados tradicionais, suporte a imagens, e as características específicas de cada gerenciador. As duas bases de dados adicionais correspondem a uma meta-base de imagens (*Image Data Definition - IDD*) e uma base operacional para imagens (*Images and Parameter Values - IPV*). A meta-base

de imagens armazena a estrutura de como as imagens são utilizadas pela aplicação. Por exemplo, quais são os atributos do tipo imagem armazenados em cada relação, e como são montados os vetores de características de cada atributo do tipo imagem em cada relação, bem como as respectivas funções de comparação entre imagens. A base operacional de imagens armazena os dados sobre as imagens - como, por exemplo, os valores das características extraídas de cada imagem - que permitem a execução de consultas envolvendo similaridade de conteúdo de imagem. A separação entre essas bases, seguindo a estrutura tradicional do modelo relacional, permite manter uma compatibilidade estrita da extensão para imagens com o modelo original, ao mesmo tempo em que garante a independência do suporte a imagens com as aplicações, mesmo com aquelas já implantadas sem esse suporte.

Figura 4.12: Arquitetura geral do CIRCE (adaptada de [Araujo et al., 2002])

Essa abordagem também garante que os resultados obtidos pelo projeto CIRCE sejam genéricos e possam ser utilizados com qualquer gerenciador ou sistema operacional onde seja necessário o suporte a imagens, incluindo o suporte a aplicações específicas, como é o caso do suporte a imagens de exames médicos.

Em um gerenciador tradicional, a meta-base e a base de estatísticas são tratadas exclusivamente pelo SGBD, de maneira que apenas a base da aplicação está acessível para a aplicação. Deste modo, o CIRCE pode ser visto como uma camada entre a aplicação e o SGBD, que trata das duas bases de imagens, de maneira que a aplicação continua tendo

acesso somente a sua base de aplicação. Por outro lado, o CIRCE manipula o mesmo SGBD utilizado pela aplicação para armazenar ambas as bases de imagens (meta-base de imagens e a base operacional), requerendo desse gerenciador apenas a capacidade de armazenar imagens como atributos do tipo “BLOB”.

Dessa maneira, embora a implementação atual do CIRCE esteja sendo feita utilizando o SGBD Oracle 9i, ela é conceitualmente independente de um gerenciador em particular. Isso permite também que, uma vez validado o suporte a imagens, esses conceitos possam vir a ser, no futuro, incorporados internamente aos SGBDs em geral, sem afetar os demais recursos que estes já possuem.

4.5.4 Esquema final

Supondo que o CIRCE estivesse pronto e utilizando-se suas funcionalidades, os atributos, do esquema exposto na figura 4.11, **binario** e **thumbnail** em **Imagem** passariam a ser do novo tipo nativo do SGBD **STILLIMAGE** e portanto o CIRCE manipularia o mapeamento para o modelo relacional da seguinte forma:

- Para cada atributo do tipo **STILLIMAGE** o CIRCE substitui tipo do atributo da tabela original para um inteiro, criando uma outra tabela de nome **ipv\$nomedataoriginal_nomedoatributo**, com dois atributos um de nome **id** (que conterá o valor do inteiro que foi armazenado na tabela original no lugar do atributo **STILLIMAGE**) e outro com o nome do atributo do tipo **STILLIMAGE**;
- Além disso, para cada atributo **STILLIMAGE** indexado é adicionada na tabela **ipv\$nomedataoriginal_nomedoatributo** mais um atributo para cada característica extraída do atributo de tipo **STILLIMAGE**. Estes atributos que, representam as características extraídas, devem possuir o nome da característica e este nome deve ser o mesmo utilizado para nome de característica nos *plugins*, do servidor de indexação, capacitados a indexar tal característica.

Assim o mapeamento para o modelo relacional ficou:

Paciente(chave_paciente, nome)

Exame(chave_exame, tipo, laudo, data, chave_imagem)

Imagem(chave_imagem, descricao, thumbnail, classe, binario, patologicaS/N, janelaCorte_xSuperiorEsquerdo, janelaCorte_ySuperiorEsquerdo, janelaCorte_altura, janelaCorte_comprimento, chave_exame)

IPV\$Imagem_binario(id, binario, histograma, textura)

IPV\$thumbnail(id, thumbnail)

Onde os atributos, `binario`, `histograma` e `textura` em `IPV$Imagem_binario` e `thumbnail` em `IPV$Imagem_thumbnail` seriam modelados como BLOB, supondo que tenham sido extraídas as características textura e histograma para as imagens.

4.5.5 Origem das imagens

As imagens utilizadas para a alimentação da base de dados são provenientes de exames de mamografia realizados no Hospital das Clínicas da Universidade de São Paulo atrelado ao Campus de Ribeirão Preto.

Foram identificados 6 tipos de cortes:

- Corte coronal direito - CCD;
- Corte coronal esquerdo - CCE;
- Corte axial direito - AXID;
- Corte axial esquerdo - AXIE;
- Corte oblíquo direito - OBLD;
- Corte oblíquo esquerdo - OBLE;

Cada paciente possuía um diretório, com nome correspondente ao código do paciente no cadastro do hospital. Cada arquivo dentro de um determinado diretório era uma imagem, identificada pelo código do paciente concatenado com o nome do corte (sigla - vide listagem acima) e uma data. Assim, um exame é identificado por um código de paciente e uma determinada data.

Os exames, por sua vez, possuíam um laudo, que estava registrado, juntamente com as outras informações já citadas, em uma planilha, onde cada linha descrevia um exame.

O total de imagens, pertencentes a exames laudados, que foi utilizado para compor a base de dados é de 1274 imagens. Todas estas imagens estavam armazenadas no formato TIFF e foram obtidas através da digitalização pelo uso de *scanner* dos filmes originais. As dimensões das imagens variavam em torno de 2000 por 2000 pixels o que gerou arquivos de cerca de 11 MB para cada imagem. Ressalta-se ainda, que todas as imagens tiveram suas regiões de interesse demarcadas manualmente, gerando, para cada uma delas, um arquivo de mesmo nome com extensão `.crop`, contendo as informações: valor no eixo y do canto superior esquerdo, valor no eixo x do canto superior esquerdo, comprimento (em relação a x) e altura (em relação a y). Por meio deste processo retiraram-se todas as bordas e outras deformidades que pudessem influenciar na boa qualidade das imagens. Foram encontradas

algumas imagens muito alteradas (com interferências - como riscos no filme, ou imagens editadas) e estas foram descartadas.

4.5.6 População da base e o Sistema de Alimentação de Dados e Imagens - SADI

O Sistema de Armazenamento de Dados e Imagens (SADI) é responsável pela alimentação da base de dados e da atualização/criação das estruturas de indexação associadas a cada extrator de características disponibilizado no servidor de indexação.

O SADI não foi implementado durante este projeto, visto que, não houve tempo hábil (como previsto na monografia de qualificação) e devido às diversas complicações de adequação às versões do padrão DICOM envolvidas no processo de aquisição das imagens.

No entanto, surgiram várias questões interessantes, discutidas entre o GBdI e o CCIFM em relação a como os exames DICOM serão mantidos e armazenados e em relação à alimentação da base de imagens a partir destes exames. Sugeriu-se a criação de um servidor para armazenamento dos exames DICOM que não houvessem sido laudados ainda e a partir do processo de atribuição de laudos, determinadas imagens marcadas como relevantes no exame DICOM seriam introduzidas na base associadas com os respectivos laudos. O exame em DICOM permaneceria à disposição por determinado período *online* no servidor DICOM, e depois de transcorrido tal período este exame estaria disponível somente *offline* em meio removível (CDs ou DVDs).

Para o processo de alimentação da base de imagens, o SADI deve permitir a inserção de dados de pacientes novos e a inserção de imagens de exames destes pacientes, que seriam associadas através da chave do Paciente (inserida também no exame em DICOM). A extração das características das imagens segundo os extratores disponíveis no servidor e a atualização da estrutura de indexação também fica a cargo do SADI e pode ser realizada tanto no momento da inserção na base, como posteriormente no caso da disponibilização de novos extratores de características e novas estruturas de indexação.

Visto que, o SADI não foi implementado durante a execução deste projeto, a população da base de dados teste foi realizada através da confecção de um programa que utilizou-se da estrutura de armazenamento e dos dados já discutidos na sub-seção 4.5.5 para a alimentação do esquema proposto (figura 4.11), gerando automaticamente o nome do paciente e deixando vazias as informações sobre se a imagem é patológica ou não e sua descrição (Dados ausentes e não dedutíveis).

4.6 Considerações finais

A arquitetura detalhada de todo o iRIS Servidor (conforme ilustrado na figura 4.13) integra todos os componentes desenvolvidos durante este projeto e descritos ao longo deste capítulo.

Figura 4.13: Arquitetura detalhada do iRIS

Nota-se, a partir do arquitetura detalhada que, a interface com o SGBD foi totalmente encapsulada pelo Sistema de Recuperação de Dados e Imagens, o que possibilita que o iRIS utilize-se do CIRCE (figura 4.14), tão logo este esteja completamente funcional, com poucas alterações no sistema.

Deste modo, cumpriu-se os quatro primeiros objetivos propostos por este projeto para o iRIS (seção 4.1).

Figura 4.14: Arquitetura detalhada do iRIS utilizando o CIRCE

Resultados

É natural para o médico lembrar-se de casos marcantes que ele tratou e diagnosticou. Então, ao se deparar com situações similares em exames atuais, surge a necessidade de recuperar as informações desses casos. A recuperação de imagens por conteúdo auxiliará o processo de diagnóstico, pois associado às imagens similares recuperadas da base de dados, estará o laudo e todo o conhecimento já empregado para o diagnóstico de tais imagens.

Através da disponibilização de um sistema com estas características na Web, torna-se mais rápido e fácil o acesso às informações de exames, dados dos pacientes e seu histórico hospitalar, visto que, o médico não precisará mais se deslocar até o setor de Radiologia e, nem mesmo necessitará estar presente no hospital consultar estas informações de forma integrada.

Adicionalmente, ganha-se também uma interface única de acesso aos dados, facilitando o aprendizado da utilização da tecnologia para o acesso às imagens e informações, difundindo assim a utilização do sistema, justificando com isso os recursos empregados na aquisição de equipamentos e desenvolvimento desse sistema.

Algumas consequências da aumento da acessibilidade de tais dados são a formação de juntas médicas virtuais, sem o dispêndio de recursos com o deslocamento físico e de tempo por parte dos médicos, além da utilização desse sistema em hospitais escola e faculdades de medicina, como um recurso didático prático no ensino de técnicas de análise de exames e diagnóstico de enfermidades. Cumpre-se assim, os dois últimos objetivos propostos para o sistema (seção 4.1).

Por outro lado, os desafios gerados pelas diversas áreas em que a computação auxilia, também contribuem para o desenvolvimento e lançam outros desafios para a computação. Neste âmbito, destaca-se, na área de bases de dados e processamento de imagens, a adequação

dos sistemas computacionais para suportar a manipulação, processamento, armazenamento e recuperação de imagens em tempo factível e cada vez com mais acuracidade.

5.1 A base computacional

O emprego das estruturas de indexação desenvolvidas no Grupo de Bases de dados e Imagens (GBdI), segundo o *framework* Arboretum, e dos extratores de características e funções distância foi de fundamental importância, possibilitando a recuperação de imagens baseadas em conteúdo.

As imagens (recuperadas da base de dados) foram submetidas aos extratores de características, que geraram os vetores de características utilizados para indexar e recuperar as imagens, sem que se incorresse em um processo subjetivo, como o processo de atribuição de palavras-chave dos PACS tradicionais, visto que um extrator sempre gera o mesmo vetor de características para uma determinada imagem.

A partir das características extraídas e armazenadas no SGBD, foram criadas as estruturas de indexação (SlimTree utilizando histograma de 256 tons de cinza e utilizando textura), no servidor de indexação. O tempo de construção de cada estrutura pode ser observado na tabela 5.1.

Estrutura de indexação	Característica	Tempo (em segundos)
SlimTree	Histograma	7
	Textura	8

Tabela 5.1: Tempo de construção da estrutura de indexação de toda a base de imagens

5.2 A plataforma de desenvolvimento e testes

O servidor Web, juntamente com o SGBD foram instalados e executaram em um Athlon XP 2.2 Ghz, com 512 MB de memória RAM e 80 GB de disco (divididos em 2 discos de 40GB). A base de dados foi criada em uma partição separada, gerenciada pelo próprio SGBD, reduzindo o *overhead* causado pelo sistema operacional no gerenciamento da base de dados. Como já foi dito anteriormente, foi utilizado o *container* Tomcat 5 (embutido no Ambiente de desenvolvimento NetBeans 4.0 - www.netbeans.org) e SGBD Oracle 9i, rodando sobre Windows 2000 Server.

5.3 O sistema em funcionamento

Após a alimentação da base de dados, com as informações e imagens obtidas do Hospital das Clínicas, extraídas as características e criadas as estruturas de indexação o sistema já possui todas os dados necessários para seu funcionamento.

Nas figuras 5.3, 5.4, 5.5 e 5.6, encontram-se ilustradas as respostas do sistema para buscas por similaridade propostas a partir da imagem em destaque no exame exibido na figura 5.1.

Figura 5.1: Exemplo de um exame completo destacando a imagem utilizada como base para a consultas por similaridades ilustradas nas figuras 5.3, 5.4, 5.5 e 5.6

Pode-se notar na figura 5.2 como são especificadas as consultas, após a seleção de uma imagem. O usuário deve realizar a opção pelo tipo da consulta através dos botões de seleção, então a respectiva caixa de texto será selecionada (valor do raio ou de k, para consulta por abrangência ou pelos k vizinhos mais próximos, respectivamente). Finalmente, através da caixa de rolagem, é selecionada a estrutura de indexação que será utilizada na busca.

RIS Similarity query

Login
Logout
Search
by exam
by patient data
by image

Range 0.00144
k 6
☒ Range Query
☐ k-NN Query
Index Index exam
submit query

Figura 5.2: Exemplo dos parâmetros de uma consulta por similaridade

5.4 Considerações finais

Destaca-se, a partir das figuras do sistema em funcionamento exibidas neste capítulo, que o sistema é fácil de utilizar e intuitivo, possibilitando ainda, como pode ser observado nas figuras que ilustram exemplos de respostas a consultas por similaridade (figuras 5.3, 5.4, 5.5 e 5.6) que é possível o acesso às informações presentes nos exames a que as imagens exibidas estão associadas, através do botão *View exam* e, assim, recupera-se todo o conhecimento médico empregado no diagnóstico de tal exame através do laudo associado.

Figura 5.3: Exemplo de resposta a uma busca por abrangência com raio 0.0015 utilizando textura

Figura 5.4: Exemplo de resposta a uma busca por abrangência com raio 0.01315 utilizando histograma

Figura 5.5: Exemplo de resposta a uma busca pelos k vizinhos mais próximos utilizando textura com $k = 6$

Figura 5.6: Exemplo de resposta a uma busca pelos k vizinhos mais próximos utilizando histograma com $k = 6$

Conclusões

Um dos fatores primordiais que influenciam na utilização de um sistema computacional é o tempo de resposta ao usuário. No entanto, a qualidade da resposta é tão ou mais importante que o tempo.

6.1 Análise dos resultados

Através dos resultados expostos no capítulo 5 pode-se concluir que o sistema cumpre o que foi proposto, auxiliando no processo de diagnóstico através da reutilização do conhecimento já empregado.

Durante a fase de testes, notou-se que os tempos de respostas ao usuário (médico) às buscas por similaridade são extremamente satisfatórios (poucos segundos), não causando desestímulo, especialmente levando em consideração que, no processo de renderização de uma página Web, as imagens que compõem a página vão aparecendo uma a uma, levando o usuário a ir analisando as que já foram exibidas e deixando-o interessado nas imagens que ainda estão por ser exibidas.

Por outro lado, a qualidade da resposta, ou seja, quão similar as imagens realmente serão à imagem de consulta, é uma questão subjetiva e de solução muito específica, que depende das características extraídas das imagens.

A questão da subjetividade está relacionada ao usuário, o médico, pois, mesmo entre os especialistas é comum haver divergências em relação ao que, de fato, encontra-se ilustrado em uma imagem e mesmo em relação a um exame completo. Desta maneira, a recuperação de imagens similares por conteúdo de forma automatizada, não substitui de forma alguma o especialista, atuando apenas como um filtro ou como uma pré-seleção das imagens, facilitando

e auxiliando o trabalho do especialista.

Ressalta-se ainda, o quão específicas são as soluções de extração de características. Essa questão se deve ao fato de que as características geralmente empregadas na recuperação baseada em conteúdo serem geralmente as características visuais: cor, forma e textura. Esse tipo de característica tem valor semântico muito baixo e os métodos empregados na sua determinação e extração possuem domínios extremamente específicos. Ou seja, são limitados a determinados tipos de imagem, configuração de componentes dentro da imagem (por exemplo, existe ou não um tumor) ou ainda limitadas ao tamanho da imagem.

Neste ponto acrescenta-se também a questão do tempo de resposta, visto que, um processo de extração de característica computacionalmente caro pode inviabilizar a utilização desta característica, pois se somente para a criação do vetor de característica for consumido um tempo demasiadamente longo, ainda restará todo o processo de busca e montagem da página de resposta, fazendo com que o usuário deixe de utilizar o sistema, pretextando que ele é muito lento.

No entanto, complementar a esse aspecto é o desenvolvimento das funções distância, pois as estruturas de indexação fazem amplo uso destas para medir a dissimilaridade e assim encontrar de modo mais eficiente e rápido as imagens que respondem a determinada busca.

Todos estes pontos, indicam que a área de processamento de imagens, extração de características e funções de distância é de fundamental importância. Deste modo, a qualidade da resposta está diretamente relacionada a quão bem determinada característica descreve a imagem segundo sua categoria e as características esperadas de tal imagem (configurações dos elementos da imagem, enfermidades e características alteradas na imagem por estas enfermidades).

Além das questões de acuracidade e tempo de resposta do sistema está o aspecto da facilidade de utilização do mesmo. A disseminação do acesso a Internet e com ele da utilização de sistemas computacionais através da Web, torna-se natural a questão de acesso por meio de *hiperlinks* e botões aos resultados esperados de uma requisição do usuário. Os formulários html encerram as necessidades de entrada de dados do usuário com campos de texto, botões de escolha múltipla, caixas de rolagem e botões de seleção. Desta maneira, empregando a Web como base para este sistema de recuperação de imagens por conteúdo, conseguiu-se fazer com que a utilização deste sistema, através dos componentes já citados - que são tão familiares a usuários da Internet, seja fácil e intuitiva. As entradas de usuário foram todas exemplificadas, ao lado da própria entrada, trazendo ao usuário a intuição do formato e do que é esperado dele naquela entrada.

Assim, conclui-se que o desenvolvimento deste sistema levou em conta desde a construção de uma interface amigável e intuitiva com o usuário até a arquitetura interna com questões como modularização, integração com outros projetos em desenvolvimento no Grupo de Bases de dados e Imagens e capacidade de adição e remoção fácil de estruturas de in-

dexação, funções distância e características de maneira a se adequar a efervescência e as especificidades de domínio da área de processamento de imagens e geração/avaliação de vetores de características.

6.2 Propostas de trabalhos futuros

Conforme foi explicitado na seção anterior, a área de processamento de imagens tem estado em constante evolução para gerar descritores que identifiquem cada vez mais precisamente as imagens.

Figueiredo em [Figueiredo 2004] propõe uma arquitetura *dataflow* para pré-processamento das imagens médicas, de modo que, através do emprego de uma série de decisores e componentes de processamento de imagens, uma imagem é processada de maneira específica, dependendo do resultado da etapa anterior de processamento. Determina-se assim a categoria da imagem e, a partir de então, são aplicados componentes de processamento mais adequados a tal imagem. Consegue-se dessa maneira, incutir mais semântica a recuperação da imagens por conteúdo, pois pode-se indexá-las em estruturas diferentes e mais adequadas.

Observando por outro lado o processo de recuperação baseado em conteúdo, destacam-se as técnicas de realimentação de relevância (*Relevance Feedback*) que propõem que o sistema aprenda com as buscas já realizadas, tornando-se mais acurado nas buscas subseqüentes através da interação com o usuário. Este processo corresponde a marcação dos componentes mais relevantes do conjunto resposta e indicação ao sistema de tais componentes. Existem várias técnicas de implementação do ajuste do sistema como a modificação do objeto central da busca, o uso de pesos para adequação da função distância, abordagens mistas das duas anteriores como utilizado por Li [Li 2004]. A substituição de funções distância por técnicas de inteligência artificial, com aprendizado através de realimentação de relevância foi utilizada por El-Naqa [El-Naqa 2004]. E ainda, abordagens híbridas de CBIR com anotações textuais, utilizando modelos estatísticos, e alterações de pesos foi utilizada por Brahmi [Brahmi 2004].

Estas idéias podem ser incorporadas ao sistema já desenvolvido, facultando o desenvolvimento de trabalhos futuros nestas áreas.

Referências Bibliográficas

- [Agarwal 2003] A. Agarwal, A. H. Rowberg, Y. Kim, “Fast JPEG2000 decoder and its use in medical imaging”, in IEEE Transactions on Information Technology in Biomedicine, vol. 7, No. 3, 2003, pp. 184-190.
- [Araujo et al., 2002] M. R. B. Araujo, C. Traina Jr, A. J. M. Traina, J. M. Bueno, H. L. Razente, “Extending relational databases to support content-based retrieval of medical images”, in proceedings of the 15th IEEE International Conference on Computer-Based Medical Systems, 2002, p. 303-308.
- [Baeza-Yates 1994] R. A. Baeza-Yates, W. Cunto, U. Manber, S. Wu, “Proximity Matching Using Fixed-Queries Trees”, in the 5th Annual Symposium on Combinatorial Pattern Matching (CPM), Asilomar, CA, 1994.
- [Beckmann 1990] N. Beckmann, H. P. Kriegel, R. Schneider, B. Seeger, “The R*-Tree: An Efficient and Robust Access Method for Points and Rectangles”, in proceedings of ACM International Conference on Data Management (SIGMOD), 1990, pp. 322-331.
- [Bortoluzzi 2003] M. K. Bortoluzzi, A. Von Wangenheim, K. Maximini, “A clinical report management system based upon the DICOM structured report standard”, in proceedings of the 16th IEEE Symposium on Computer-Based Medical Systems, 2003, pp. 183-188.
- [Bozkaya 1999] T. Bozkaya, Z. M. Özsoyoglu, “Indexing Large Metric Spaces for Similarity Search Queries”, in ACM Transactions on Database Systems (TODS), vol. 24, 1999, pp. 361-404.
- [Bozkaya 1997] T. Bozkaya, Z. M. Özsoyoglu, “Distance-Based Indexing for High-Dimensional Metric Spaces”, in proceedings of ACM International Conference on Data Management (SIGMOD), Tucson, AZ, 1997.
- [Brahmi 2004] D. Brahmi, D. Ziou, “Improving CBIR systems by integrating semantic features”, in proceedings of the 1st IEEE Canadian Conference on Computer and Robot Vision, 2004.
- [Brin 1995] S. Brin, “Near neighbor search in large metric spaces”, in proceedings of International Conference on Very Large Databases (VLDB), Zurich, Switzerland, 1995.
- [Brown 1992] L. G. Brown, “A Survey of Image Registration Techniques”, in ACM Computing Surveys, vol. 24, 1992, pp. 325-376.

- [Bueno 2002] J. M. Bueno, "Suporte a Recuperação de Imagens Médicas Baseada em Conteúdo Através de Histogramas Métricos", Tese de Doutorado apresentada ao ICMC-USP, fevereiro de 2002.
- [Bueno 2002b] J. M. Bueno, F.J.T. Chino, A. J. M. Traina, C. Traina, Jr., "How to Add Content-based Image Retrieval Facilities in a PACS", in proceedings of the 15th IEEE International Conference on Computer-Based Medical Systems, 2002, pp. 321-326.
- [Burkhard 1973] W. A. Burkhard, R. M. Keller, "Some Approaches to Best-Match File Searching", in proceedings of CACM, vol. 16, 1973, pp. 230-236.
- [Cao 2000] X. Cao, H. K. Huang, "Current Status and Future Advances of Digital Radiography and PACS", IEEE Engineering in Medicine and Biology Magazine, vol. 9, pp. 80-88.
- [Castelman 1996] K. R. Castelman, "Digital Image Processing", Prentice-Hall Inc., 1996.
- [Chiueh 1994] T. C. Chiueh, "Content-Based Image Indexing", in International Conference on Very Large Databases (VLDB), Santiago de Chile, Chile, 1994.
- [Ciaccia 1999] P. Ciaccia, A. Nanni, M. Patella, "A Query-sensitive Cost Model for Similarity Queries with M-Tree", in proceedings of the 10th Australasian Database Conference (ADC'99), Auckland, New Zealand, 1999, pp. 65-76.
- [Ciaccia 1998] P. Ciaccia, M. Patella, "Bulk Loading the M-Tree", in proceedings of the 9th Australasian Database Conference (ADC'98), Perth, Australia, 1998, pp. 15-26.
- [Ciaccia 1997] P. Ciaccia, M. Patella, P. Zezula, "M-Tree: An Efficient Access Method for Similarity Search in Metric Spaces", in proceedings of International Conference on Very Large Databases (VLDB), Athens, Greece, 1997, pp. 426-435.
- [El-Naqa 2004] I. El-Naqa, Y. Yang, N. P. Galatsanos, R. M. Nishikawa, M. N. Wernick, "A Similarity Learning Approach to Content-Based Image Retrieval: Application to Digital Mammography", in IEEE Transactions on Medical Imaging, vol. 23, NO. 10, October 2004, pp. 1233-1244.
- [Felipe et al. 2003] J. C. Felipe, A. J. M. Traina, C. Traina Jr., "Retrieval by content of medical images using texture for tissue identification", in proceedings of the 16th IEEE Symposium on Computer-Based Medical Systems, 2003, pp. 175-180.
- [Feng 2002] D. D. Feng, T. W. Cai, J. Kim, S. Eberl, "Content Access and Distribution of Multimedia Medical Data in E-Health", in proceedings of 2002 IEEE International Conference on Multimedia and Expo, 2002, vol. 2, pp 341-344.
- [Feng 2001] D. D. Feng, W. Cai, J. Kim, Y. S. Lim, "Medical Image Data Retrieval and Manipulation through the WWW", in proceedings of 2001 International Symposium on Intelligent Multimedia, Video, and Speech Processing, 2001, pp 381-384.
- [Feng 2000a] D. D. Feng, Z. Chen, X. Yu, "A Telemedicine System over the Internet", in Conferences in Research and Practice in Information Technology, Vol 2, P. Eades and J. Jin, Eds. 2000, pp 113-115.

- [Feng 2000b] D. D. Feng, J. Kim, T. W. Cai, "A Web Based Medical Image Data Processing and Management System", in *Conferences in Research and Practice in Information Technology*, Vol 2, P. Eades and J. Jin, Eds. 2000, pp 89-91.
- [Figueiredo 2004] J. M Figueiredo, C. Traina Jr., A. J. M. Traina, "Including Conditional Operators in Content-Based Image Retrieval in Large Sets of Medical Exams", in *proceedings of the 17th IEEE Symposium on Computer-Based Medical Systems*, 2004.
- [Furuie 2002] S. S. Furuie, M. Yamaguti, M. A. Gutierrez, "A flexible storage architecture for large PACS", in *Computers in Cardiology*, 2002, pp. 405-408.
- [Furuie 1999] S. S. Furuie, N. B. Bertozzo, J. C. B. Figueiredo, M. Yamaguti, "Archiving and Retrieving Long-Term Cineangiographic Images in a PACS", in *Computers in Cardiology*, 1999, pp. 435-438.
- [Gunther 1999a] G. Gunther, R.A. Bauman, "Large-Scale PAC Systems", in *Filmless Radiology*, E. L. Siegel, R.M. Kolodner, Eds. New York City, NY:Springer Verlag, 1999a, pp. 21
- [Guttman 1984] A. Guttman, "R-Trees: A Dynamic Index Structure for Spatial Searchig", in *Proceeding of 1984 ACM SIGMOD International Conference on Management of Data*, 1984, pp. 47-57.
- [Hellerstein 1995] J. M. Hellerstein, J. F. Naughton, A. Pfeffer, "Generalized Search Trees for Database Systems", in *International Conference on Very Large Databases (VLDB)*, Zurich, Switzerland, 1995.
- [Hinneburg 1999a] A. Hinneburg, D. A. Keim, "Optimal Grid-Clustering: Towards Breaking the Curse of Dimensionality in High-dimensional Clustering", in *International Conference on Very Large Databases (VLDB)*, 1999.
- [Kitney 2001] R. I. Kitney, S. Claesen, J. Halls, "A Comprehensive Web-Based Patient Information Environment", in *23rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, vol. 4, 2001, pp. 3584-3585.
- [Li 2003] K. Y. Li, X. X. Liu, X. F. Yu, X. D. Yang, J. M. Hu, "The complete solution of PACS based on B/S mode", in *proceedings of the 2nd International Conference on Machine Learning and Cybernetics*, 2003, pp. 847-852.
- [Li 2004] B. Li, S. Yuan, "A Novel Relevance Feedback Method in Content-Base Image Retrieval", in *proceedings of the IEEE International Conference on Information Technology: Coding and Computing*, 2004.
- [Long 2003] L. R. Long, S. K. Antani, G. R. Thoma, "A prototype content-based image retrieval system for spine X-rays", in *proceedings of the 16th IEEE Symposium on Computer-Based Medical Systems*, 2003, pp. 156-162.
- [Marsh 1997] A. Marsh, "EUROMED - The Creation of a Telemedical Information Society", presented at *10th IEEE Symposium on Computer Based Medical Systems*, Maribor, Slovenia, 1997.

- [Nikolaidou 2003] M. Nikolaidou, D. Anagnostopoulos, M. Hatzopoulos, "Using a medical digital library for education purposes", in proceedings of the 16th IEEE Symposium on Computer-Based Medical Systems, 2003, pp. 22-27.
- [Pagel 2000] B. U. Pagel, F. Korn, C. Faloutsos, "Deflating the Dimensionality Curse Using Multiple Fractal Dimensions", in 16th International Conference on Data Engineering (ICDE2000), San Diego, CA, 2000.
- [Papadias 1995] D. Papadias, Y. Theodoridis, T. K. Sellis, M. J. Egenhofer, "Topological Relations in the World of Minimum Bounding Rectangles: A Study with R-Trees", in ACM International Conference on Data Management (SIGMOD), San Jose, CA, 1995.
- [Pereira 2003] J. Pereira, A. Castro, B. Arcay, A. Pazos, A. Gomez, "Framework of a Systyem for the Management and Assistance in Analysis of Medical Images", in proceedings of the 4th Annual IEEE Conference on Information Technology Applications in Biomedicine, 2003, pp. 86-89.
- [Pereira 2002] J. Pereira, A. Castro, D. Ronda, B. Arcay, A. Pazos, "Development of a System for Access to and Exploitation of Medical Images", in proceedings of the 15th IEEE Symposium on Computer-Based Medical Systems, 2002, p. 6.
- [Pereira 2001] J. Pereira, A. Lamelo, J. M. Vázquez-Naya, M. Fernández, J. M. López-Gestal, J. Teijero, A. Pazos, "Design and Implementation of a DICOM PACS with Secure Access via Internet", in proceedings of the 23rd Annual EMBS International Conference, 2001, pp. 3724-3727.
- [Rosa 2002] N. A. Rosa, "Uma abordagem prática e eficiente de consultas por similaridade para suporte a diagnóstico por imagens", Dissertação de mestrado, Departamento de Ciências de Computação. São Carlos: Universidade de São Paulo, pp. 146.
- [Seckman 2004] C. A. Seckman, C. Romano, S. Rosenfeld, "Standard Web-Based Clinical Radiology Images and Reports at the Point of Care", in proceedings of the 17th IEEE Symposium on Computer-Based Medical Systems, 2004, pp.73-78.
- [Sellis 1987] T. Sellis, N. Roussopoulos, C. Faloutsos, "The R+-Tree: A Dynamic Index for Multi-Dimentional Objetcs", in International Conference on Very Large Databases (VLDB), Brighton, England, 1987.
- [Siegel 1999] E. L. Siegel, "Current State of the Art and Future Trends", in Filmless Radiology, E. L. Siegel and R. M. Kolodner, Eds. New York City, NY: Springer Verlag, 1999, pp. 3-20.
- [Stasiu 2001] R. K. Stasiu, G. L. Bichinho, "Components proposal for medical images and HIS", in proceedings of the 14th IEEE Symposium on Computer-Based Medical Systems, 2001, pp.73-78.
- [Sung 2002] M. M. Sung, H. J. Kim, E. K. Kim, J. Y. Kwak, J. K Yoo, H. S Yoo, "Clinical evaluation of JPEG2000 compression for digital mammography", in IEEE Transactions on Nuclear Science, Vol. 49, No. 3, 2002, pp. 827-832.
- [Tomita 1990] F. Tomita, T. Saburo, "Computer Analysis of Visual Textures", Kluwer, 1990.

- [Traina 2003] A. J. M. Traina, N. A. Rosa, C. Traina Jr, "Integrating images to patient electronic medical records through content-based retrieval techniques", in proceedings of the 16th IEEE Symposium on Computer-Based Medical Systems, 2003, pp. 163-168.
- [Traina 2001] A. J. M. Traina, "Suporte à Visualização de Consultas por Similaridade em Imagens Médicas através de Estrutura de Indexação Métrica", Tese de livre docência apresentada ao ICMC-USP, setembro de 2001.
- [Traina 1998] C. Traina, Jr., A. J. M. Traina, R. R. d. Santos, E. Y. Senzako, "Support to Content-Based Image Query in Object-Oriented Databases", presented at ACM.
- [TrainaJr 2000] C. Traina Jr., A. J. M. Traina, B. Seeger, C. Faloutsos, "Slim-Trees: High Performance Metric Trees Minimizing Overlap Between Nodes", in International Conference on Extending Database Technology, Konstanz, Germany, 2000.
- [Uhlmann 1991] J. K. Uhlman, "Satisfying General Proximity/Similarity Queries with Metric Trees", in Information Processing Letter, vol. 40, 1991, pp. 175-179.
- [Vieira et al., 2004] M. R. Vieira, C. Traina Jr., F. J. T. Chino, A. J. M. Traina, "DBM-Tree: A Metric Access Method Sensitive to Local Density" in proceedings of Brazilian Symposium on Databases, 2004, p. 163-177.
- [Wong 2003] S. T. Wong, D. Tjandra, H. Wang, W. Shen, "Workflow-enabled distributed component-based information architecture for digital medical imaging enterprises", in IEEE Transactions on Information Technology in Biomedicine, vol. 7, No. 3, 2003, pp. 171-183.
- [Zhang 2001] Z. M. Zhang, A. Krol, G. Pu, "IBMAS: An Internet-Based Medical Archive System", in proceedings of the 14th IEEE Symposium on Computer-Based Medical Systems, 2001, pp.541-546.
- [Zhuang 2001] Y. Zhuang, Q. Li, R. W. H. Lau, "Web-Based Image Retrieval: a Hybrid Approach", in proceedings on Computer Graphics International 2001, 2001, pp. 62-69.

Glossário

Arboretum	<i>Framework</i> de desenvolvimento de estruturas métricas criado pelo GBdI (www.gbdi.icmc.usp.br/arboretum).
BLOB	<i>Binary Large Object</i> - Tipo de dado utilizado em SGBDs para armazenar objetos binários grandes.
DICOM Lib	Biblioteca de manipulação de imagens no formato DICOM desenvolvida pelo GBdI.
EJB	<i>Enterprise Java Beans</i> - Tecnologia de construção de componentes de <i>software</i> para encapsulamento da lógica corporativa desenvolvido pela <i>Sun Microsystems</i> .
<i>fast Ethernet</i>	Tecnologia de redes de computadores.
<i>Gigabit Ethernet</i>	Tecnologia de redes de computadores.
Java	Linguagem de programação.
JavaBeans	Componentes de <i>software</i> escritos em Java.
JSP	<i>Java Server Pages</i> - Tecnologia de construção de páginas Web dinâmicas da <i>Sun Microsystems</i> .
JPEG	Padrão de codificação de imagens.
JPEG2000	Padrão de codificação de imagens.
<i>jukebox</i>	Sistema de armazenamento e/ou leitura de múltiplos dispositivos de armazenamento (exemplo fitas ou CDs).
RAID-5	Arranjo de discos magnéticos.
SSL	<i>Secure Socket Layer</i> - Protocolo seguro de comunicação em rede.