#### UNIVERSIDADE DE SÃO PAULO INSTITUTO DE CIÊNCIAS MATEMÁTICAS E DE COMPUTAÇÃO DEPARTAMENTO DE CIÊNCIAS DE COMPUTAÇÃO E ESTATÍSTICA

#### Suporte à Recuperação de Imagens Médicas Baseada em Conteúdo através de Histogramas Métricos

JOSIANE M. BUENO

Orientação: Profa Dra Agma Juci Machado Traina

Tese apresentada ao Instituto de Ciências Matemáticas e de Computação, como parte dos requisitos para a obtenção do título de Doutor em Ciências — Área de Ciências de Computação e Matemática Computacional.

Trabalho realizado dentro do projeto de pesquisa apoiado pela FAPESP, Processo Nº 98/03209-6

USP - São Carlos

Novembro de 2001

Eu já nasci amando-o.

Ele me ensinou a dar meus primeiros passos,

A virar cambalhotas,

A empinar pipa

E a andar de bicicleta.

Foi ele quem me apresentou o mar.

Ciumento, ele aterrorisava meus namorados com o seu olhar bravo.

Algumas vezes me deixava maluca com seu jeitão autoritário.

Outras vezes me fazia rir com seu humor de moleque.

Nos momentos mais importantes da minha vida,

Lá estava ele com uma máquina fotográfica e um sorriso orgulhoso.

Nas horas tristes, com apenas um olhar, ele me dizia

Que poderia ir à lua pra me fazer feliz.

E com o mesmo olhar ele me fazia reconhecer as minhas falhas

E desejar ser uma pessoa cada vez melhor.

Ele era a base sólida a quem eu recorria quando a vida ficava complicada.

Mas ele se foi.

E às vezes bate uma saudade que não tem tamanho.

E então eu me consolo

Pois ele me ensinou a ser o que eu sou hoje

E me deixou muita coisa

Incluindo a minha vida.

Seu nome era José, Zé, Bueno, Neguinho, Eno

Ou, para mim, simplesmente

Pai.

## **Agradecimentos**

A Deus por estar sempre comigo.

A minha família, principalmente à Dona Marla, pelo amor incondicional e por sempre acreditar na minha capacidade.

Ao Juvenal por manter meus pés no chão e a cabeça nas nuvens.

A Agma pela orientação e pelo enorme suporte profissional e pessoal.

A Renata pelo ombro (ou melhor, ouvido) amigo e pela ajuda na revisão desta tese.

A Arlete e a Vera pelo companheirismo.

Ao Caetano, Camilo, Enzo, Fábio e Humberto pela paciência e pelo apoio intelectual e técnico sem o qual eu estaria perdida entre "bugs" e "pontos de interrogação".

A todos do GBDI pela contribuição crítica a este trabalho e pelos bolos e festas.

Ao Prof. Dr. Paulo M. Marques pelo suporte ao desenvolvimento do projeto mini-PACS.

Aos funcionários do ICMC pela atenção e apoio administrativo.

A todos pela grande amizade que demonstraram.

A FAPESP pelo suporte financeiro que possibilitou o desenvolvimento deste trabalho.

## Sumário

LISTA	DE FIGURAS	V
LISTA	DE TABELAS	IX
RESUMO		X
ABSTI	RACT	XI
CAPÍT	TULO 1. INTRODUÇÃO	1
1.2.	Motivação	3
1.3.	Objetivos	4
1.4.	Organização da Tese	4
CAPÍT	TULO 2. SISTEMAS DE ARQUIVAMENTO E COMUNICAÇÃO DE	IMAGENS6
2.1.	Introdução	6
2.2.	ARMAZENAMENTO DE IMAGENS	7
2.3.	SISTEMAS REAIS	10
2.4.	Integração com Sistemas de Gerenciamento	10
2.5.	PACS É FINANCEIRAMENTE VIÁVEL?	12
2.6.	Segurança dos Dados	13
2.7.	Conclusões	13
CAPÍT	ULO 3. ESTRUTURAS DE INDEXAÇÃO PARA ESPAÇOS MÉTRI	COS15
3.1.	Introdução	15
3.2.	Definições	17
3.3.	Estado da Arte em Estruturas Métricas	18
3.4.	Estrutura de Indexação para Dados Métricos Slim-tree	21
3.4	4.1. Construindo a Slim-tree	23
3.4	4.2. Problema de Sobreposição	25
3.4	4.3. Comparando Árvores Diferentes Construídas sobre o Mesmo Conj	unto de
Do	udos	30

3.4	4.4. Visualização dos Dados na Slim-tree	31
3.5.	Aplicações	37
3.6.	Conclusões	38
CAPÍT	TULO 4. IMAGENS E SUAS CARACTERÍSTICAS INERENTES	39
4.1.	Introdução	39
4.2.	Extração de Características de Imagens	40
4.2	2.1. Atributos Visuais para Indexação e Recuperação de Imagens Baseada o	em seu
Ca	onteúdo	42
4.3.	REGISTROS DE IMAGENS	47
4.4.	Conclusões.	49
CAPÍT	TULO 5. O SISTEMA MINI-PACS PROPOSTO	50
5.1.	Introdução	50
5.2.	MÓDULOS OPERACIONAIS	52
5.2	2.1. Sistema de Armazenamento de Dados e Imagens (SADI)	53
5.2	2.2. Sistema de Gerenciamento de Estruturas Métricas (SGEM)	54
5.2	2.3. Sistema de Recuperação de Imagens por Similaridade (SRIS)	55
5.2	2.4 Categorias de Usuários	59
5.3.	Extração de Características: Histogramas Métricos	60
5.3	3.1. Área de Interesse (Cropping)	62
5.3	3.2. Histogramas Métricos	64
5.4.	Extenção da Slim-tree para Armazenar Imagens	68
5.4	4.1. Função de Distância Proposta: por Diferença de Área	69
5.5.	Conclusões	73
CAPÍT	TULO 6. RESULTADOS	74
6.1.	Introdução	74
6.2.	Experimentos	74
6.3.	Medidas de Precisão e Revocação	77
6.4.	COMPARAÇÃO DE TEMPO	79
6.5.	Conclusões	81
CAPÍT	TULO 7. CONCLUSÕES GERAIS	82
7 1	CONCIDED A CÕES CED AIS E CONTRIBUIGÕES	92

7.2.	Trabalhos Futuros	84
7.2	.1 – Ampliação do Conjunto de Extratores de Características de Imagens	85
7.2	2.2 – Avaliação por Radiologistas dos Resultados de Recuperação de Imagens po	r
His	stogramas Métricos	85
7.2	.3 – Geração de Assinaturas de Imagens	85
7.2	.4 – Execução de Macros para a Realização de Consultas	86
7.2	5. – Desenvolvimento do Servidor de Web (SW)	86
7.2	.6 – Cruzamento de Informações e Mineração nos Dados do Mini-PACS	87
REFER	RÊNCIAS BIBLIOGRÁFICAS	88
	D I. AVALIAÇÃO DE PERFORMANCE NA RECUPERAÇÃO DE	
INFOR	MAÇÕES	97
I.1.	Precisão e Revocação ( <i>Precision and Recall</i> )	99
I.1.	1 Reduções de Valores Únicos	105
I.1.	2. Adequação de Medidas de Precisão e Revocação	108
I.2.	MEDIDAS ALTERNATIVAS	108
ANEX(	O II. FUNÇÕES DE DISTÂNCIA	112
APÊND	DICE A. DESENVOLVIMENTO DE APLICAÇÕES PARA A WWW	115
A.1.	A LINGUAGEM JAVA	120
A.2.	BANCO DE DADOS NA WWW	123
A.3.	RMI – REMOTE METHOD INVOCATION	125
A.4.	ARQUITETURA DO GERENCIADOR DE REQUISIÇÕES DE OBJETOS COMUNS	126
A.5.	CORBA NA WWW	131
A.6.	CORBA E JAVA	133
A.7.	Conclusões	134
APÊND	DICE B. PADRÕES DE COMUNICAÇÃO DE DADOS NA MEDICINA	136
B.1.	CORBAMED	136
B.2.	Padrão DICOM	137
B.2	2.1. Objetivos do Padrão DICOM 3.0	139
B.2	2.2. Definições Utilizadas pelo DICOM	140
$R \subset$	23 Servicos do nadrão DICOM	111

B.2.4.	Suporte para rede DICOM	147
B.2.5.	Vantagens do padrão DICOM	148

# Lista de Figuras

FIGURA 1 – INFRAESTRUTURA DE UM PACS EM AMBIENTE HOSPITALAR
FIGURA 2 – ESQUEMA MOSTRANDO O FLUXO DE DADOS MÉDICOS AO REDOR DO PRONTUÁRIO
DO PACIENTE11
FIGURA 3 – PARTICIONAMENTO DE UMA VP-TREE NO NÍVEL DA RAIZ COM <i>FANOUT</i> IGUAL A 3. AS
TRÊS REGIÕES SÃO ROTULADAS (1,2,3) E APRESENTADAS EM TONS DE CINZA DISTINTOS 19
FIGURA 4 – (A) REPRESENTAÇÃO GRÁFICA DA SLIM-TREE APRESENTANDO OS NÓS INTERNOS
(INDEXNODES) E OS NÓS FOLHAS (LEAFNODES); (B) UM EXEMPLO DE UMA SLIM-TREE
ARMAZENANDO 7 PALAVRAS UTILIZANDO A DISTÂNCIA L <sub>EDIT</sub>
FIGURA 5 – SLIM-TREE ARMAZENANDO 17 OBJETOS REPRESENTADA GRAFICAMENTE POR RAIOS
DE COBERTURA E POR UMA ESTRUTURA DE ÁRVORE CONVENCIONAL
FIGURA 6 – MECANISMO DE QUEBRA DE NÓS UTILIZANDO O ALGORITMO MST. (A) NÓ ANTES DA
QUEBRA. (B) CONSTRUÇÃO DA ÁRVORE DE CAMINHO MÍNIMO (MST). (C) DOIS NÓS
GERADOS PELA QUEBRA DO NÓ EM (A)
FIGURA 7 – DUAS ÁRVORES ARMAZENANDO O MESMO CONJUNTO DE DADOS, PORÉM COM
NÚMERO DE NÓS E ABSOLUTE FAT-FACTORS DIFERENTES. OS NÓS RAÍZES SÃO
DESENHADOS COM LINHAS TRACEJADAS E COR MAIS CLARA
FIGURA 8 – FUNCIONAMENTO DO ALGORITMO SLIM-DOWN
FIGURA 9 – INDEXANDO O CONJUNTO DE SIERPINSKY COM 9850 PONTOS, E UTILIZANDO O
ALGORITMO DE QUEBRA DE NÓS ALEATÓRIO. APRESENTA OS NÓS ACIMA DAS FOLHAS. (A)
ANTES DE APLICAR O ALGORITMO SLIM-DOWN, COM FAT-FACTOR=0,03. (B) DEPOIS DE
APLICAR O ALGORITMO SLIM-DOWN, COM FAT-FACTOR=0,01
FIGURA 10 – CONJUNTOS DE DADOS UTILIZADOS: (A) TRIÂNGULO DE SIERPINSKY. (B)
MONTGOMERY. (C) CONJUNTO DE PALAVRAS EM PORTUGUÊS. (D) CONJUNTO DE PALAVRAS
EM INGLÊS
FIGURA 11 – SLIM-TREE INDEXANDO UM SUB-CONJUNTO DE 21.473 PALAVRAS DO DICIONÁRIO
DA LÍNGUA PORTUGUESA. (A) VISUALIZAÇÃO TRIDIMENSIONAL DO CONJUNTO DE
PALAVRAS INDEXADAS. (B) VISUALIZAÇÃO TRIDIMENSIONAL DAS PALAVRAS E DOS NÓS
DA ÁRVORE DO NÍVEL IMEDIATAMENTE SUPERIOR A ELAS34
FIGURA 12 – VISUALIZAÇÃO DO CONJUNTO MONTGOMERY_COUNTY INDEXADO NA SLIM-TREE.
(A) APENAS OS OBJETOS DO CONJUNTO. (B) OS OBJETOS E OS NÓS IMEDIATAMENTE ACIMA
DELES
FIGURA 13 – UMA VISÃO TRIDIMENSIONAL DO CONJUNTO DE PALAVRAS <i>ENGLISH</i> . (A) AS
PALAVRAS E SUA DISPOSIÇÃO NO ESPAÇO. (B) O MAPEAMENTO DO CONJUNTO DE
PALAVRAS, OS PIVÔS UTILIZADOS PELO ALGORITMO <i>FASTMAP</i> , E OS CÍRCULOS
REPRESENTANDO OS NÓS ACIMA DAS FOI HAS NA SLIM-TREE ÁRVORE CONSTRUÍDA COM O

ALGORITMO DE QUEBRA DE NOS MINMAX, PELA SELEÇAO DO RAIO MINIMO E COM	60
PALAVRAS POR NÓ.	36
FIGURA 14 – FORMATOS DE NÓS AO UTILIZAR AS DISTÂNCIAS $L_1$ (MANHATAN), $L_2$ (EUCLIE	oIAN) L∞
(INFINITY).	37
FIGURA 15 – PROCESSO DE EXTRAÇÃO DE CARACTERÍSTICAS DE UMA IMAGEM	40
FIGURA 16 – NÍVEIS DE ABSTRAÇÃO PARA O PROCESSO DE RECONHECIMENTO DE OBJET	OS EM
IMAGENS.	42
FIGURA 17 – IMAGEM QUANTIZADA EM 256 NÍVEIS DE CINZA E SEU HISTOGRAMA ASSOCI	IADO 43
FIGURA 18 – MESMO HISTOGRAMA DE CORES (DOIS NÍVEIS DE CINZA) ASSOCIADO A QUA	ATRO
IMAGENS DISTINTAS.	44
FIGURA 19 – EXEMPLIFICAÇÃO DE QUATRO ELEMENTOS DE TEXTURA	45
FIGURA 20 – INFRAESTRUTURA DO MINI-PACS	51
FIGURA 21 – MÓDULOS OPERACIONAIS DO MINI-PACS.	53
FIGURA 22 – SISTEMA DE ARMAZENAMENTO DE DADOS E IMAGENS.	54
FIGURA 23 – SISTEMA DE GERENCIAMENTO DE ESTRUTURAS MÉTRICAS.	55
FIGURA 24 – SISTEMA DE RECUPERAÇÃO DE IMAGENS POR SIMILARIDADE	57
FIGURA 25 – TELA PARA A DEFINIÇÃO DE CONSULTA A IMAGENS MÉDICAS POR SIMILAR	IDADE.
AO CONFIRMAR OS PARÂMETROS DA CONSULTA, OS RESULTADOS APARECEM COM	10
MOSTRADO NA FIGURA 26.	58
FIGURA 26 – TELA DE RESULTADOS DE CONSULTA BASEADA EM CONTEÚDO DE IMAGEN	S. O
BOTÃO FULL VIEW AND DETAILS LEVA A UM VISUALIZADOR GENÉRICO CONTENDO	
FUNÇÕES DO SPI.	58
FIGURA 27 – UMA IMAGEM E SEU HISTOGRAMA NORMALIZADO.	61
FIGURA 28 – RESULTADO DO <i>CROPPING</i> AUTOMÁTICO.	62
FIGURA 29 – ALGORITMO PARA ELIMINAÇÃO DO FUNDO DA IMAGEM ( <i>CROPPING</i> )	63
FIGURA 30 – HISTOGRAMA NORMALIZADO COM PONTOS DE CONTROLE $< B_K, H_K >$ QUE DEF	INEM OS
BUCKETS CORRESPONDENTES AO SEU HISTOGRAMA MÉTRICO	64
FIGURA 31 – IDENTIFICAÇÃO DA JANELA DE INTERESSE.	65
FIGURA 32 – SENTIDO ENTRE PARES DE BINS E OS PONTOS DE CONTROLE ENCONTRADOS	S PELO
ALGORITMO.	66
FIGURA 33 – HISTOGRAMA DA IMAGEM APRESENTADA NA FIGURA ANTERIOR, COM SEUS	S
PONTOS DE CONTROLE.	68
FIGURA 34 – (A) IMAGEM ORIGINAL, (B) IMAGEM MAIS SEMELHANTE E (C) IMAGEM A ME	NOS
SEMELHANTE. OS HISTOGRAMAS APRESENTAM A DENSIDADE DE PIXELS, EM	
PORCENTAGEM, PARA 256 NÍVEIS DE CINZA DAS IMAGENS.	69
FIGURA 35 – DISTÂNCIA ENTRE DOIS HISTOGRAMAS MÉTRICOS CALCULANDO A ÁREA E	NTRE
ELES. (A) DOIS HISTOGRAMAS MÉTRICOS A E B, E OS PONTOS USADOS PARA ESPECI	FICAR
OS PASSOS DO ALGORITMO. (B) PRIMEIRO PASSO DO ALGORITMO, EXEMPLIFICAND	O
QUANDO OS DOIS HM SE INTESECTAM. (C) SEGUNDO PASSO DO ALGORITMO. (D) TE	RCEIRO
PASSO DO ALGORITMO.	71

FIGURA 36 – VISUALIZAÇÃO DOS HISTOGRAMAS DE 256 DIMENSOES NA SLIM-TREE. NESTE T	
FOI UTILIZADA A DISTÂNCIA MANHATHAN (L1)	
FIGURA 37 – VISUALIZAÇÃO DOS HISTOGRAMAS MÉTRICOS NA SLIM-TREE. NESTE TESTE FO	
UTILIZADA A DISTÂNCIA POR DIFERENÇA DE ÁREA ENTRE HISTOGRAMAS (DM)	76
FIGURA 38 – BANCO DE DADOS <i>RMCRÂNIO500</i> : PLOTAGENS DE PRECISÃO E REVOCAÇÃO AO	
RESPONDER CONSULTAS POR VIZINHOS MAIS PRÓXIMOS (K-NN) ONDE O NÚMERO K É I	OADO
COMO UMA PORCENTAGEM DO TAMANHO DO BANCO DE DADOS. (A) 15%, (B) 10%, (C) 59	%, (D)
2%, (E) 1%, (F) 0.5%	
FIGURA 39 – BANCO DE DADOS <i>RMVARIADOS4247</i> : PLOTAGENS DE PRECISÃO E REVOCAÇÃO A	
RESPONDER CONSULTAS POR VIZINHOS MAIS PRÓXIMOS (K-NN), EM QUE K É DADO CO	MO
UMA PORCENTAGEM DO TAMANHO DO BANCO DE DADOS: (A) 15%, (B) 10%, (C) 5%, (D) 2	2%,
(E) 1%	79
FIGURA 40 – TEMPO TOTAL PARA RESPONDER 50 CONSULTAS POR VIZINHOS MAIS PRÓXIMO	S
PARA CADA PORÇÃO DO CONJUNTO DE DADOS: 0,5; 1; 2; 5; 10 E 15%, UTILIZANDO OS	
HITOGRAMAS NORMALIZADOS E MÉTRICOS. (A) CONJUNTO DE DADOS RMCRÂNIO500; (I	
CONJUNTO DE DADOS RMVARIADOS4247	80
FIGURA 41 – TAXA DE TRANSIÇÃO PARA A RADIOLOGIA SEM FILME NOS ESTADOS UNIDOS	
[SIEGEL_1999]	
FIGURA 42 – PRECISÃO E REVOCAÇÃO PARA UM DADO EXEMPLO DE PEDIDO A INFORMAÇÃO	
FIGURA 43 – PRECISÃO EM 11 NÍVEIS DE REVOCAÇÃO PADRÕES	102
FIGURA 44 – PRECISÃO INTERPOLADA NOS 11 NÍVEIS PADRÕES DE REVOCAÇÃO RELATIVA A	
$R_Q = \{D_3, D_{56}, D_{129}\}.$	
FIGURA 45 – CÁLCULOS DA MÉDIA DE PRECISÃO VERSUS REVOCAÇÃO PARA DOIS ALGORITI	MOS
DISTINTOS DE RECUPERAÇÃO	104
FIGURA 46 – UM HISTOGRAMA DE PRECISÃO PARA DEZ CONSULTAS HIPOTÉTICAS	107
FIGURA 47 – TAXAS DE COBERTURA E DE ORIGINALIDADE PARA UM DADO PEDIDO A	
INFORMAÇÃO	110
FIGURA 48 – EQUAÇÕES DE FUNÇÕES DE DISTÂNCIA MAIS COMUNS ( $X \in Y$ SÃO VETORES DE $X \in Y$	M
VALORES DE ATRIBUTO) [WILSON_1997]	114
FIGURA 49 – UMA REDE VIRTUAL PRIVADA – INTRANET. NESTA CONFIGURAÇÃO, EMBORA O	)
SERVIDOR WEB ESTEJA DESPROTEGIDO, OS DADOS DA INTRANET PODEM SER DIVULGA	ADOS
NA INTERNET DE MODO SEGURO. AQUI O FIREWALL É CONFIGURADO PARA IGNORAR O	)
TRÁFIGO DA BASE DADOS, DEIXANDO A SEGURANÇA PARA O SISTEMA DE	
GERENCIAMENTO DE BANCO DE DADOS	118
FIGURA 50 – ACESSO A PÁGINAS HTML.	119
FIGURA 51 – COMO FUNCIONA O ACESSO A PROGRAMAS CGI	119
FIGURA 52 – COMO FUNCIONA PROGRAMAS JAVA NA WWW	122
FIGURA 53 – COMUNICAÇÃO DE DADOS VIA JDBC.	
FIGURA 54 – ARQUITETURA PARA DESENVOLVIMENTO DE APLICAÇÃO JAVA EM 3 CAMADAS	126
FIGURA 55 – AROUITETURA DE GERENCIAMENTO DE OBJETOS.	126

FIGURA 56 – COMMON OBJECT REQUEST BROKER ARCHITECTURE (CORBA)	129
FIGURA 57 – ARQUITETURA JAVA/CORBA PARA A WWW	131
FIGURA 58 – EVOLUÇÃO DAS TECNOLOGIAS PARA A WWW.	135
FIGURA 59 – ESTRUTURAS PRINCIPAIS DO MODELO DE INFORMAÇÕES DO DICOM	142
FIGURA 60 – PARTES CORRENTES DO PADRÃO DICOM E PARTES PROPOSTAS PARA EXTE	NSÃO DO
PADRÃO	145

## Lista de Tabelas

TABELA 1 – CAPACIDADE DE ARMAZENAMENTO NECESSÁRIA PARA IMAGENS DIGITAIS	
[RATIB_1997]	8
TABELA 2 – FORNECEDORES DE SISTEMAS PARA MANIPULAÇÃO DE IMAGENS DIGITAIS	
[MEIRE_1996]	12
TABELA 3 – ALGORITMO DE REORGANIZAÇÃO DE ÁRVORES MÉTRICA: SLIM-DOWN	
[TRAINAJR_2000]	28
TABELA 4 – TAXONOMIA SOBRE ESPAÇOS DE CARACTERÍSTICAS EXTRAÍDAS DAS IMAGEN	S
[BROWN_1992]	41
TABELA 5 – ALGORITMO PARA CÁLCULO DE DISTÂNCIA ENTRE HISTOGRAMAS MÉTRICOS	70
TABELA 6 – TEMPO PARA CONSTRUIR A SLIM-TREE UTILIZANDO HISTOGRAMAS MÉTRICOS	E
NORMALIZADOS A PARTIR DOS CONJUNTOS DE DADOS RMCRÂNIO500 E RMVARIADOS42	247 79
TABELA 7 – ANALOGIA ENTRE A CONSTRUÇÃO DE UMA SENTENÇA E UMA CLASSE DICOM	
SOP	143
TABELA 8 – TABELA DO RESUMO DAS PARTES DO PADRÃO DICOM 3.0.	145

### Resumo

Os grandes centros médicos e hospitais de todo o mundo têm procurado integrar as informações de seus pacientes incluindo os exames de imagens efetuados (tomografía computadorizada, tomografía por ressonância magnética, ultrasson, medicina nuclear, etc.). Um sistema que integra as imagens junto às informações tradicionais é chamado de Sistema de Arquivamento e Comunicação de Imagens (*Picture Archive and Communication System* - PACS). Os sistemas PACS comerciais associam as imagens de exames às informações de pacientes através de chaves de consultas textuais e numéricas, não suportando consultas baseadas no conteúdo pictórico das imagens. Entretanto, muitas vezes o médico gostaria de recuperar as imagens armazenadas que fossem semelhantes (similares) a uma determinada imagem de consulta. Por exemplo, seja a consulta: "encontre as 10 imagens mais semelhantes à imagem Raio-X-tórax do Jõao da Silva". Ao responder a consultas desse tipo, o sistema permite que o médico relembre casos ocorridos anteriormente. Além disso, o conhecimento já gerado de exames e tratamentos anteriores pode ser recuperado mais rapidamente do que utilizando apenas a memória humana ou um sistema não automático de recuperação de informações. Um sistema com a capacidade de recuperar imagens utilizando o seu conteúdo pictórico é uma ferramenta valiosa para o auxílio ao diagnóstico médico.

Esta tese apresenta a arquitetura de um PACS atualmente em desenvolvimento. Este sistema está sendo denominado mini-PACS. Tal sistema necessita da integração de três sistemas, a saber:

- Um Sistema de Processamento de Imagens (SPI), o qual é responsável pela leitura e pré-processamento das imagens que são recebidas de diferentes dispositivos e possuem diferentes formatos. O SPI extrai as características relevantes das imagens, as quais serão utilizadas para a sua indexação e recuperação por conteúdo.
- Um Sistema de Gerenciamento de Bases de Dados e Imagens (SGBDI), que permite a armazenagem e a recuperação de imagens baseada em seu conteúdo. O SGBDI utiliza uma estrutura métrica, a Slim-tree, que indexa as imagens através das características extraídas pelo SPI e possibilita responder consultas por similaridade.
- Um Servidor de Web (SW), que disponibiliza o acesso às informações através da *internet*. A construção do Servidor de Web encontra-se fora do escopo do desenvolvimento desta tese. Porém, testes iniciais sobre a transferência e comunicação de imagens utilizando um servidor e aplicativos Java foram desenvolvidos para avaliar o comportamento do sistema.

Entre as principais contribuições deste trabalho encontra-se um novo método de extração de características de imagens chamado *histograma métrico*. Os histogramas métricos permitem comparar imagens de diferentes tamanhos e mapeadas em faixas de quantização diferentes (se a alteração de brilho for linear). O tempo de resposta às consultas por similaridade utilizando histogramas métricos é, em média, 5 vezes menor do que o tempo de resposta utilizando histogramas tradicionais. Para permitir a indexação das imagens utilizando a Slim-tree foi necessário desenvolver uma nova função de distância métrica. Tal função de distância utiliza a diferença entre as áreas das curvas do histograma métrico. A construção da árvore de indexação utilizando os histogramas métricos chega a ser 10 vezes mais rápida do que com os histogramas tradicionais.

As inovações e aperfeiçoamentos oriundos deste trabalho estão sendo integrados ao mini-PACS. Este sistema vem sendo desenvolvido de forma conjunta entre o Grupo de Bases de Dados e Imagens (GDBI) do Instituto de Ciências Matemáticas e de Computação da USP e o Centro de Ciências de Imagens e Física Médica (CCIFM) da Faculdade de Medicina de Ribeirão Preto da USP.

### **Abstract**

Medical centers and large hospitals are seeking a way to integrate the information of patient, including the images obtained from exams such as computerized tomography, magnetic resonance imaging, nuclear imaging and so on, in a complete system. A system like this is called *Picture Archive and Communication System* - PACS. Traditional PACS associate images to patients through textual and/or numerical keys, taking advantage of a data base management system and do not allow to retrieve images based on their content. However, many times the physician would like to retrieve images without the help of textual or numerical keys. That is, the physician would like to get the images which are similar to a given image. Exemplifying, a common question could be: "find the 10 most similar images to the X-Ray-Thorax of John Doe". Answering questions like this would help the physician to recall previous study cases. Moreover, the knowledge generated beforehand could be retrieved faster than only using human memory or not automatic techniques. An information system with capability for retrieval images by their content would be a valuable tool in order to help medical diagnosis.

This thesis presents the architecture of a PACS under development in functional modules. This system is named mini-PACS. To have the whole mini-PACS running it is necessary to design and to integrate the three following systems:

- an Image Processing System (IPS), which is responsible since for reading the images coming from different devices, until extracting the useful characteristics from the images regarding their domain.
- an Image Data Base Management System (IDBMS), which stores and retrieves the images based on their content. To do so, the IDBMS uses a metric access method (Slim-tree) which index the images through their characteristics extracted by the IPS. The Slim-tree was developed aiming to answer similarity queries.
- a Web Server (WS), which allows to make all the information available on the web. The web server is out of the scope of this thesis. Only preliminary tests on communicating images through Java server and applets were done in order to evaluate the behavior of such programs in the Web.

The main contributions of this thesis include the development of a new image feature extractor called the *metric histogram*. The metric histograms allow comparing images which different sizes and even which different gray level ranges. Moreover, the speed up for retrieving images is about 5 times when comparing to the traditional histograms. A new metric distance function to compare two metric histograms was developed and integrated to the Slim-tree. This new metric is based on the difference of areas between two metric histograms. The speed up to build the index over the images was impressive: going up to 10 times faster.

These new improvements are being integrated to the mini-PACS under joined development between the Image Data Base Group of Instituto de Ciências Matemáticas e de Computação of USP and Centro de Ciências de Imagens e Física Médica of Faculdade de Medicina de Ribeirão Preto of USP.

# Capítulo 1

### Introdução

A computação é uma ciência que tem evoluído de forma exponencial nos últimos anos. Tal evolução tem também contribuído para o crescimento de muitas outras áreas da ciência, através de um suporte efetivo a simulações e experimentos, tornando mais rápido e barato testar hipóteses, o que de forma tradicional poderia levar muito mais tempo e esforço. Essa interação entre áreas acontece com, por exemplo, a engenharia, física, genética, bioquímica, entre outras. Particularmente a medicina é uma das áreas que mais tem sido beneficiada dessa simbiose com a computação. A contribuição da computação para a medicina vem desde o gerenciamento das informações usuais de pacientes em hospitais e consultórios até a utilização de ferramentas sofisticadas de processamento de imagens, que possibilitam a recuperação de informações que seria custosa de ser feita manualmente [Duncan\_2000] [Smeuders\_2000]. Outro aspecto importante é a simulação de procedimentos que seriam arriscados de serem realizados sem preparação prévia, o que pode ser feito utilizando modelos obtidos por reconstrução tridimensional a partir de imagens apoiando procedimentos cirúrgicos [Dev\_1999] [Gross 1998].

Os recursos de tratamento de imagens com auxílio de computador possibilitam obter informações mais precisas sobre a estrutura anatômica e a patologia, ajudando médicos a diagnosticar doenças que até então não poderiam ser detectadas por procedimentos tradicionais [Dev 1999] [Rhodes 1996]. Além disso, os médicos precisam destes sistemas não somente para sua própria compreensão e diagnóstico, mas também para compartilhar os resultados com seus colegas [Sclabassi 1996], instituições de pesquisas e o público em geral [Mehuys 1997]. Um aspecto interessante é a utilização de sistemas multimídia que possibilitam a integração de dados obtidos por várias modalidades de exames (tomografia computadorizada - TC, ressonância magnética ToRM, ultrasson), além de áudio animações [Zhang 2001].

Com o objetivo de organizar e divulgar informações médicas relevantes ao bom funcionamento de um centro de aquisição de imagens médicas, estão se desenvolvendo cada vez mais os chamados sistemas de arquivamento e comunicação de imagens (*Picture Archiving and Communication Systems* - PACS) [Stasiu\_2001] [Marsh\_1997] [Alcocer\_1996] [Gennip\_1992]. Aplicados às imagens médicas, esses sistemas devem organizar todas as informações relevantes (textuais e pictóricas) dos pacientes em análise [Crudele\_1997]. Tais sistemas são um grande impulso para o auxílio ao diagnóstico médico, pois possibilitam a recuperação e o cruzamento de informações de forma muito mais eficiente e conforme a especificação do usuário, neste caso o médico.

Os sistemas PACS tradicionais baseiam-se na recuperação de imagens a partir de atributos textuais previamente associados às mesmas. Porém, a recuperação de imagens baseada em suas próprias informações pictóricas traz um potencial de auxílio muito maior ao diagnóstico. Com esse recurso torna-se possível, por exemplo, solicitar de um banco de imagens todas as imagens que possuem um determinado objeto (que é representado por uma imagem de referência) [Tchounikine 1997], ou mesmo solicitar imagens que sejam semelhantes entre si. Esta potencialidade tem sido estudada e aperfeiçoada pelo grupo de Bases de Dados e Imagens (GBDI) do Instituto de Ciências Matemáticas e de Computação - ICMC-USP, cujos resultados têm sido utilizados para desenvolver um sistema para recuperação de imagens baseada em conteúdo (content-based image retrieval) [TrainaJr 1998B] [TrainaJr 1997A] [TrainaJr 1997B] [Santos 1996] [Senzako 1996], o qual vem sendo incorporado a um protótipo de um sistema mini-PACS. O desenvolvimento desse sistema mini-PACS vem sendo realizado através de pesquisa comum entre o GBDI e o e o Centro de Ciências das Imagens e Física Médica do Hospital das Clínicas da Faculdade de Medicina de Ribeirão Preto (CCIFM-HC- FMRP) - USP. Nesse sistema, as consultas realizadas são processadas por um gerenciador de banco de dados que efetua a busca e a recuperação das imagens baseadas em um sistema de extração de características de imagens. O trabalho apresentado nessa tese é parte de um projeto maior, que visa o desenvolvimento de um sistema PACS completo, com tecnologia totalmente dominada pelas instituições participantes. Para validação dos resultados teóricos, foram desenvolvidos módulos que permitem o funcionamento básico do sistema de tratamento global das imagens, possibilitando uma visão completa do sistema mini-PACS. Como o CCIFM é um centro que gera imagens por vários modos de aquisição, decidiu-se optar inicialmente por um domínio de aplicação: imagens de Tomografia por Ressonância Magnética (ToRM). Futuramente, será possível expandir o mini-PACS para tratar outras modalidades.

#### 1.2. Motivação

A organização e disponibilização rápida e descentralizada da informação de pacientes incluindo seus exames são os principais anseios da comunidade médica em geral. Foi nessa linha de pensamento que a comunidade médica sugeriu um protocolo de comunicação de dados, o DICOM para possibilitar o intercâmbio de dados em formato padrão entre equipamentos de geração de exames médicos. Este protocolo é descrito no Apêndice B desta tese. Porém, a organização desses dados para facilitar a recuperação e disseminação rápida foi o que motivou as propostas iniciais dos sistemas PACS. Os sistemas PACS também têm sido colocados como sistemas que permitem a manipulação e visualização de imagens médicas sem utilização de filmes, o que barateia o preço da manutenção desses exames em torno de 25% [Siegel 1999]. Nos hospitais escola, como é o caso do Hospital das Clínicas de Ribeirão Preto - USP, a necessidade de intercâmbio de informações é ainda maior, pois o caminho entre as imagens desde a coleta até o laudo final é ainda mais longo. Isso se deve ao fato do exame ser coletado inicialmente pelo técnico em radiologia, indo para o residente que está aprendendo a laudar, depois para o professor responsável que tem que validar o laudo feito pelo residente ou pelos residentes se mais do que um deles for ouvido, e somente depois será transferido para o prontuário do paciente. Logo, a agilidade de tráfego de informações é uma das preocupações nesses ambientes.

Tradicionalmente os sistemas PACS associam as imagens ao prontuário de pacientes utilizando chaves textuais. A proposta desta tese é possibilitar a construção de um sistema PACS que permita a recuperação de imagens pelo seu conteúdo pictórico, mantendo a utilização de chaves textuais como um filtro de consultas adicional. Dessa forma, os médicos podem solicitar consultas em que se buscam imagens semelhantes a uma imagem dada, mantendo um grau de semelhança previamente especificado. Tais consultas permitem a construção de aplicativos para suporte a diagnósticos, possibilitando que o médico faça cruzamentos entre tratamentos realizados anteriormente em pacientes cujos exames por imagens sejam parecidos. Observe que

tais aplicativos são um suporte ao diagnóstico, porém não visam o desenvolvimento de um sistema especialista que possa prescindir de um profissional da área médica.

#### 1.3. Objetivos

O objetivo direto deste projeto é o de fornecer ferramentas e técnicas para auxiliar o profissional da área médica no processo de diagnóstico através de imagens. Isso pode ser feito com a inclusão das técnicas desenvolvidas aqui a um sistema PACS, que permitiria tratar imagens de forma natural, como é feito com dados tradicionais como números e textos, nos gerenciadores de dados comerciais.

Para que o sistema PACS possa manipular imagens de forma natural é preciso construir um arcabouço para manipulação de imagens que permita seu armazenamento e recuperação de forma rápida e eficiente, e que possa responder ao tipo de consultas que seriam solicitadas pelo médico. Tais consultas são chamadas de consultas por similaridade, por exemplo: "Dada a imagem de Raio-X de tórax de J. Silva, quais são as 5 imagens presentes no banco de dados mais parecidas com ela". Este é um exemplo de consulta que é respondida pelo sistema desenvolvido e apresentado nesta tese. Para tanto, é necessário o desenvolvimento de um sistema de processamento de imagens que possibilite a manipulação e a apresentação das imagens dos exames, além de módulos de tratamento de protocolos de imagem e de um servidor de WWW para permitir a descentralização dos dados.

#### 1.4. Organização da Tese

A tese está organizada da seguinte forma:

No Capítulo 1 encontra-se a introdução geral ao trabalho proposto, bem como a motivação para o seu desenvolvimento e os objetivos que foram seguidos.

No Capítulo 2 é apresentado um levantamento sobre os sistemas PACS, incluindo um modelo de arquitetura tradicional presente na literatura da área. São, também, enfatizados os pontos principais de tais sistemas e o que é esperado deles pela comunidade da área.

O Capítulo 3 traz uma descrição das estruturas de indexação de dados, focalizando nos métodos de acesso métricos (MAMs) que são os mais indicados para suportar consultas por

similaridade, como as que o sistema proposto deve responder. Define-se também as consultas por similaridade mais utilizadas e apresenta-se o MAM Slim-tree que foi utilizado como base para o projeto desenvolvido nessa tese.

O Capítulo 4 aborda resumidamente as principais técnicas de extração de características encontradas na literatura, preparando para a apresentação da técnica desenvolvida neste trabalho que é um aperfeiçoamento dos histogramas tradicionais.

O Capítulo 5 descreve o sistema mini-PACS que está sendo desenvolvido, bem como sua arquitetura geral. A forma de gerenciamento do sistema é discutida, a qual constitui-se de três níveis de usuários. As principais contribuições inovadoras deste trabalho, que consiste dos histogramas métricos e da função de distância desenvolvida sobre ele são aqui apresentados.

O Capítulo 6 apresenta os resultados dos experimentos realizados sobre a recuperação das imagens via histogramas métricos versus histogramas tradicionais.

As conclusões gerais e linhas de futuras pesquisas são delineadas no Capítulo 7 dessa tese.

Material adicional de auxílio à leitura e compreensão da tese encontram-se nos Anexos e Apêndices ao final da tese. O Anexo I apresenta as principais medidas utilizadas para qualificar métodos de recuperação de informação, com ênfase na medida de *precision/recall*. O Anexo II traz uma descrição suscinta das principais funções de distância encontradas na literatura. O Apêndice A mostra o ferramental utilizado para desenvolvimento de aplicativos para a WWW. O Apêndice B apresenta um resumo do protocolo DICOM, o qual foi utilizado para a captura das imagens apresentadas nessa tese.

# Capítulo 2

## Sistemas de Arquivamento e Comunicação de Imagens

#### 2.1. Introdução

O desenvolvimento de novas técnicas de obtenção de imagens digitais e o aumento de modalidades em imagens médicas que geram imagens na forma digital tem levado ao desenvolvimento de sistemas de gerenciamento de imagens digitais [Ratib\_1997]. Porém em sistemas atuais a transferência e disseminação eletrônica das imagens tornam-se uma necessidade. Os sistemas de arquivamento e comunicação de imagem (*Picture Archiving and Communication Systems* - PACS) preenchem essa lacuna, ao cuidar da captura, armazenamento, recuperação, distribuição e exibição de imagens em múltiplos *sites* [Meire\_1996] [Siegel\_1999]. O objetivo do PACS é ter informações sobre exames em formato digital necessitando da utilização de técnicas de compressão de imagens sem perdas. Os sistemas PACS devem permitir a transmissão rápida e o armazenamento organizado para as imagens digitais, tanto em termos de disponibilidade do exame em curto prazo como também no aspecto de multivisualização. Os cirurgiões devem ser capazes de tomar decisões diagnósticas logo após um exame, bem como acessar simultaneamente as imagens para discutir as seqüências de imagens nas salas de ensino, ou nas salas de consulta [Alcocer\_1996] [Siegel\_1999] [Lundberg\_1999] [Furuie\_1999]. A Figura 1 mostra a infraestrura necessária a um PACS.

Os primeiros projetos de PACS foram gerados levando-se em consideração um sistema centralizado onde imagens e dados são armazenados em um banco de arquivamento central e distribuídos quando requisitados a estações de trabalho clientes. Com o desenvolvimento de tecnologias de rede e sistemas de comunicação e transferência eficiente de informações tornouse possível projetar bancos de arquivamento distribuídos onde as imagens e os dados podem ser

armazenados em diferentes lugares de uma rede e ainda ser acessível em qualquer parte desta rede. Estes projetos têm evoluído para uma arquitetura aberta com equipamentos de diversas marcas com padrões industriais para interconexão de dispositivos de imagens e outros componentes PACS, assim permitindo uma abordagem mais realística em um ambiente clínico real onde equipamentos diferentes de aquisição de imagens são raramente adquiridos de um único vendedor [Ratib\_1997].

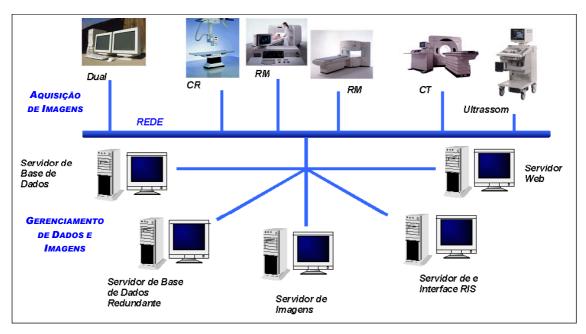


Figura 1 - Infraestrutura de um PACS em ambiente hospitalar.

#### 2.2. Armazenamento de Imagens

Um dos primeiros objetivos de um PACS é coletar imagens radiológicas na forma digital para arquivamento e comunicação. Embora haja um número crescente de imagens radiológicas que são adquiridas diretamente na forma digital (TC, ToRM, Medicina Nuclear etc), muitos fabricantes não fornecem um modo fácil de acesso às imagens em sua forma digital original. Somente recentemente vêm surgindo alguns padrões industriais para permitir a transferência de imagens dos dispositivos de aquisição para ambientes PACS [Stasiu\_2001].

Muitos dos projetos PACS tinham sérias dificuldades em relação à extração de imagens dos dispositivos de aquisição. Interfaces customizadas tinham que ser desenvolvidas para

dispositivos diferentes para se conectar a equipamentos que pudessem fazer parte de um PACS. Estes desenvolvimentos podiam somente ser realizados com uma contribuição dos fabricantes, o que era muito dispendioso, porque soluções técnicas individuais são sempre mais caras do que as soluções mais genéricas pois estas podem ser reusadas em configurações diferentes. Pelas mesmas razões é muito dificil integrar equipamento de aquisição de imagens de fabricantes diferentes. Alguns projetos PACS ficaram limitados a equipamentos de um único fabricante [Ratib 1997].

O armazenamento de imagens radiológicas na forma digital não é um problema trivial. Quase todos os sistemas de radiologia produzem imagens em diferentes formatos e os arquivos de imagem digital são freqüentemente grandes. Felizmente, a capacidade dos sistemas de armazenamento de dados vem aumentando cada vez mais, e o custo está diminuindo consideravelmente. Porém, a capacidade necessária para que um departamento de radiologia guarde todas as suas imagens *online* é ainda tecnologicamente e financeiramente inviável [Ratib\_1997]. A Tabela 1 mostra a capacidade de armazenamento necessária para imagens digitais.

**Tabela 1 -** Capacidade de armazenamento necessária para imagens digitais [Ratib\_1997].

CAPACIDADE DE ARMAZENAMENTO NECESSÁRIA PARA GUARDAR IMAGENS DIGITAIS	
Imagem digital	Tamanho do arquivo de imagem
Ultra-sonografia	250 Kb
Tomograma de TC ou ToRM	1 Mb
Mamograma digital	4 Mb
Raios-X de tórax de alta resolução	16 Mb
Exame completo de TC ou ToRM	100 Mb
Imagens digitais de uma semana (sem compressão)	100.000 Mb (100 Gb)
Imagens digitais de um ano (sem compressão)	5.000.000 Mb (5 Tb)
Floppy disk de alta densidade	1,44 Mb
Hard disk típico de PC	40 Gb
Compact Disk Gravável	650Mb

Para serem clinicamente aceitáveis, as imagens de raios-X devem ter uma resolução muito alta. Devem ser adquiridas e armazenadas em matrizes da ordem de 2000 x 2000 pixels no

mínimo, com uma faixa dinâmica de 8 a 12 bits por pixel. Isto representa entre 4 a 8 Mbytes por imagem. Modalidades digitais tais como TC ou ToRM geram imagens com matrizes menores (em torno de 256x256 ou 512x512 com uma faixa dinâmica de 12 a 15 bits por pixel), mas há dificuldades com relação ao grande número de imagens geradas para cada exame. Um exame pode gerar entre 20 e 100 imagens, e em algumas situações até mais [Ratib\_1997].

O rápido desenvolvimento de discos óticos de alta capacidade que podem ser tratados em grande quantidade por dispositivos chamados de *juke-boxes* oferece uma solução prática para o problema de armazenamento. O problema em se utilizar discos óticos é a lentidão para se ler e gravar dados. Assim, o acesso a imagens de um grande número de discos óticos pode necessitar de constantes trocas de discos nos *juke-boxes* que é uma operação mecânica muito lenta. Além disso, o projeto de arquivos de discos óticos para PACS requer uma grande capacidade de *caching* em disco magnético para manter um rápido acesso das imagens mais recentes e mais usadas. Somente uma combinação bem feita de espaço de armazenamento em disco magnético e disco ótico permitirá uma performance adequada das tarefas de arquivamento e recuperação dos sistemas PACS [Ratib 1997].

Os sistemas de visualização são um obstáculo adicional. Os radiologistas estão acostumados a examinar oito ou mais filmes ao mesmo tempo e freqüentemente com vários tipos de exames de um paciente. Reproduzir estas condições com imagens digitais de alta resolução fica muito caro em termos computacionais. A maioria dos sistemas usa somente uma ou duas telas em cada estação. Além disso, o tratamento de imagens requer do radiologista procedimentos muito diferentes daqueles que está habituado [Meire\_1996] [Siegel\_1999].

Em projetos PACS é necessária a utilização de vários tipos de estações de trabalho. Para os primeiros diagnósticos o radiologista precisa de uma estação de trabalho que forneça a melhor performance possível com resolução altíssima. Para revisão e conferências clínicas não há necessidade da mesma alta performance. Além disso, um terceiro tipo de estação pode ser considerado para processamento e análise de imagens. Esta estação deveria fornecer ferramentas mais específicas para análise de certos tipos de imagens e fornecer uma interface excelente para permitir fácil acesso a estas ferramentas por usuários que não estão acostumados a lidar com computadores [Ratib\_1997].

#### 2.3. Sistemas Reais

Para implementar um sistema eficiente e de baixo custo é necessário levar em consideração alguns procedimentos:

- Armazenar somente alguns exames, tais como ultra-som, TC ou ToRM, permitindo que o
  radiologista se acostume a tratar imagens eletrônicas antes de se sujeitar a um ambiente
  totalmente digital, embora as economias associadas à ausência de filmes não sejam
  completamente realizadas.
- Guardar imagens online somente por um período limitado. Este compromisso é satisfatório se poucos pacientes retornam para exames periódicos, mas pode ser um problema sério em um centro com muitos pacientes em tratamento; exatamente o tipo de instituição que pode se benefiar da utilização de um PACS.
- Comprimir arquivos de imagens, o que é sempre praticado. Com software moderno é possível comprimir um arquivo de imagem radiológica em média 2,5:1 com nenhuma perda de detalhe. Uma maior compressão resultaria em perda de dados. Os médicos resistem à idéia de utilizar técnicas de compressão que provoquem perdas na qualidade da imagem, ou mesmo de informação, o que é justificável.

Na prática ao menos duas destas opções são seguidas em muitos departamentos digitais. Muitos sistemas PACS armazenam todas as imagens do departamento e tem um *hot file*, com imagens comprimidas sem perda de informação e de poucas semanas, que pode ser acessado rapidamente. Quando uma imagem não é acessada durante um período de tempo é então comprimida e passada para o arquivo principal. Esta imagem pode ainda ser recuperada, porém mais lentamente. Após um período adicional de inatividade as imagens podem ser passadas para um dispositivo de armazenamento *offline* e podem ser recuperadas somente perante requisição antecipada [Meire\_1996] [Seigel\_1999].

#### 2.4. Integração com Sistemas de Gerenciamento

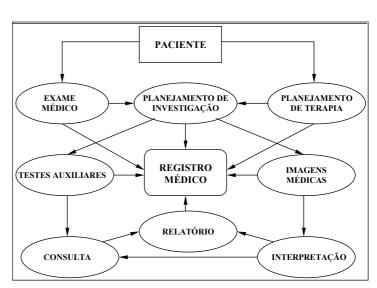
Muitos departamentos de radiologia de grande porte agora utilizam sistemas computadorizados de gerenciamento. Estes sistemas envolvem todos os detalhes de histórico

radiológico, anotações, relatórios, entre outros dados de um paciente, e incluem um índice principal. Se um PACS é acrescentado a tal sistema é essencial que os dois sejam completamente integrados. Surpreendentemente, isto geralmente não é feito e, assim, é necessário procurar imagens e relatórios relevantes em dois sistemas separados. Qualquer sistema novo de PACS deve incluir um sistema de gerenciamento departamental completamente funcional [Siegel 1999].

O rápido desenvolvimento de sistemas de informação computadorizados e médicos permite o acesso a uma variedade de documentos em diferentes formas. Em particular, um grande número de dados médicos é gerado na forma de conjuntos estáticos de imagens. Investigações clínicas contam com uma variedade de modalidades e os clínicos enfrentam a

dificil tarefa de integrar grandes quantidades de dados obtidos por diferentes modalidades devem correlacionar o que pode ser observado nestas imagens e comparar as imagens a dados clínicos e auxiliares. Estações multimodais capazes de tratar todos estes dados em uma plataforma conveniente e prática logo se tornarão ferramentas necessárias para a prática médica. Para torná-las viáveis é necessário gerenciamento um adequado das imagens obtidas a partir

de diferentes modalidades.



**Figura 2 -** Esquema mostrando o fluxo de dados médicos ao redor do prontuário do paciente.

O conteúdo de um prontuário médico consiste de dados fornecidos tanto por técnicas de investigação diferentes quanto por notas, relatórios e observações, como mostra a Figura 2. O prontuário médico também contém observações cronológicas e planos de tratamento. A completude do prontuário médico que deveria conter todos os eventos médicos e terapêuticos do tempo de vida de um paciente é essencial para o atendimento adequado do paciente. Um prontuário computadorizado de um paciente deve oferecer esta completude a fim de preencher as

necessidades dos médicos. Deve tratar dados multimídias obtidos de diferentes fontes (texto, sons, sinais fisiológicos e imagens), mas deve também oferecer uma ligação coerente entre todos estes dados. As ferramentas oferecidas para os usuários navegarem através destes dados e os visualizarem de um modo conveniente são tão importantes quanto a qualidade dos próprios dados. Os usuários clínicos devem ser capazes de acessar estes dados de modo natural e conveniente [Ratib 1997].

A Tabela 2 mostra alguns fornecedores e seus produtos para manipulação de imagens digitais.

FORNECEDORES DE SISTEMAS DE IMAGEM DIGITAL	
Companhia	PRODUTO
Kodak	Sistema de câmera digital
Occulab	Câmera digital
ImageNet	Câmera digital
Philips	Sistema PACS
Siemens	Sistema PACS
SIMIS	Sistema PACS

Sistema PACS

**Tabela 2 -** Fornecedores de sistemas para manipulação de imagens digitais [Meire 1996].

#### 2.5. PACS é Financeiramente Viável?

**AMS** 

O custo de um sistema completo PACS em um grande hospital, dependendo de sua complexidade, é da ordem de milhões de dólares, e o custo de manutenção anual é alto. As principais vantagens são velocidade, eficiência e confiabilidade na recuperação de imagens do arquivo e em comunicação. Se um cirurgião quer ver imagens é só chamá-las de um console e será possível obtê-las mais rápido e de forma mais confiável que os serviços atuais de armazenamento e recuperação de raios-X. As imagens podem ser acessadas de qualquer lugar, mesmo em um hospital remoto se este estiver conectado em rede (teleradiologia) [Seigel 1999].

#### 2.6. Segurança dos Dados

É importante observar que um departamento sem papéis e sem filmes será totalmente incapacitado se o sistema computacional "quebrar". Um sistema bem projetado deve ser capaz de operar, mesmo de modo precário, durante as falhas no suprimento de energia ou nos principais componentes do sistema. O ideal seria que todos os arquivos fossem duplicados, de preferência todos os discos rígidos do computador, o que é tecnicamente possível, mas financeiramente inviável atualmente. Deveriam ser feitas cópias de segurança de todos os novos dados diários em uma mídia removível. Se estas regras fossem seguidas, muitos desastres poderiam ser evitados [Seigel\_1999].

#### 2.7. Conclusões

Os sistemas PACS têm mudado os pré-requisitos para o trabalho cooperativo, uma vez que suportam diagnósticos cooperativos de imagens, mesmo estando a equipe médica em diferentes lugares [Lundberg 1999] [Cao 2000].

Futuramente, todos os PACS se tornarão diretamente integrados com todas as empresas de tratamento de saúde, resultando na disponibilidade dos prontuários médicos eletrônicos dos pacientes, bem como uma vasta biblioteca de imagens e texto. Haverá acesso instantâneo a qualquer imagem do sistema de saúde a qualquer momento, com uma melhor segurança das imagens e uma qualidade mais alta e mais imagens de diagnósticos, e uma nova geração de ferramentas para os radiologistas, ferramentas essas que irão permitir aos radiologistas melhorar a qualidade das imagens existentes e combinar múltiplas imagens de uma ou mais modalidades em uma única imagem ou estudo para melhorar a exatidão do diagnóstico. Finalmente, um novo conjunto de características de suporte a decisão estarão disponíveis que usará informações clínicas do prontuário eletrônico médico em conjunto com as imagens do banco de dados e irá combiná-las com as informações clínicas e imagens associadas com um novo estudo para ajudar a encontrar ou mesmo sugerir o diagnóstico [Siegel 1999] [Cao 2000].

Devido ao desenvolvimento rápido da base tecnológica para a maioria dos componentes PACS, os componentes se tornam obsoletos em poucos anos. Existem, contudo, diferenças. Por exemplo, investimentos em redes tais como Ethernet ou FDDI não serão perdidos se a ATM se

tornar a tecnologia dominante, pois eles irão, ao contrário, ser integrados via *gateway*. As estações de trabalho podem se tornar ultrapassadas, porém permanecerão utilizáveis se possuírem uma interface DICOM de acordo. O problema mais crucial irá surgir no armazenamento, onde a tecnologia não é ainda estável e existe uma competição entre diferentes métodos, formatos de dados e densidades de armazenamento. Em poucos anos as instalações podem encarar alternativas desagradáveis de ou copiar seu arquivo de dados em uma nova mídia (cara e que consome tempo), ou manter equipamentos ultrapassados e ineficientes a alto custo.

# Capítulo 3

## Estruturas de Indexação para Espaços Métricos

#### 3.1. Introdução

O processo de organização de informação efetuado pelos sistemas de gerenciamento de banco de dados (SGBDs) utiliza como base as estruturas de indexação ou métodos de acesso para acelerar a busca aos dados. Logo, as estruturas de indexação são ferramentas fundamentais que habilitam os sistemas gerenciadores de dados a eficientemente armazenar e recuperar informações de um grande volume de dados. As primeiras estruturas de indexação foram propostas para suportar dados espaciais em espaços de poucas dimensões. Essas estruturas, que são amplamente divulgadas na literatura são chamadas de Métodos de Acesso Espaciais - SAM ("Spatial Access Methods"). Essa área iniciou-se com o trabalho pioneiro sobre as R-Trees [Guttman\_84] e prosseguiu com numerosas variações e adaptações [Sellis\_1987] [Beckmann\_1990] [Hellerstein\_1995] [Papadias\_1995] (ou veja [Gaede\_1998] para um apanhado geral sobre estruturas espaciais). No entanto, para dados em espaços de altas dimensões ou mesmo adimensionais, nenhuma dessas estruturas mostrou-se adequada [Ciaccia\_1997B] [Ciaccia\_1998] [Ciaccia\_1999].

As imagens em geral são indexadas e recuperadas a partir de informações textuais associadas a elas. Porém, o processo de recuperação de imagens utilizando apenas o seu conteúdo pictórico demanda extrair as características inerentes da imagem. Através dessas características, a imagem será indexada e posteriormente recuperada. Note que o conjunto de características extraídas de uma imagem varia em número e especificidade dependendo do domínio de aplicação envolvido. Além disso, geralmente o que o médico deseja é buscar as imagens que são semelhantes a alguma imagem conhecida. Visando suportar tais tipos de consultas por similaridade, as estruturas de dados para espaços métricos (que englobam tanto

dados espaciais com dimensão definida quanto dados adimensionais) foram propostas. Busca por similaridade é aquela em que se consideram quão semelhantes dois dados são entre si. A similaridade entre os dados é definida por uma função distância, ou função de "dissimilaridade"  $d(O_i, O_j)$ , que retorna zero se ambos os objetos  $O_i$  e  $O_j$  forem idênticos, e um valor positivo que aumenta quanto maior for a distância (ou dissimilaridade) entre os objetos.

A primeira proposta de desenvolvimento de estruturas de indexação de dados utilizando apenas as distâncias entre eles foi apresentada por Burkhard e Keller [Burkhard\_1973], para operar em espaços métricos (tal como definidos na subseção seguinte). A partir desse trabalho pioneiro, outras estruturas foram desenvolvidas, a saber: as árvores "fixed query" [Baeza-Yates\_1994], a mvp-tree (multi-vantage point tree) [Bozkaya\_1997] [Bozkaya\_1999], a vp-tree [Uhlmann\_1991] (vantage-point [Chiueh\_1994]), a GNAT [Brin\_1995], a M-tree [Ciaccia\_1997B] e recentemente a Slim-tree [TrainaJr\_2000]. Estruturas métricas utilizam apenas medidas de distância entre pares de objetos para organizá-los na base de dados. Tais estruturas têm se mostrado promissoras em sistemas de recuperação de dados baseados no conteúdo dos próprios dados [Bozkaya\_1999].

As estruturas métricas, ou os Métodos de Acesso Métricos - MAM suportam naturalmente consultas por proximidade ou similaridade além de se mostrarem eficientes para dados de dimensões altas. Dessa forma, utilizar um MAM para indexar imagens, ou mais propriamente as características que foram extraídas das imagens suportando busca por similaridade tem-se mostrado bastante apropriado. Tais características podem ser atributos de forma, textura, histograma de cores, resultados de transformações como, por exemplo, *singular value decomposition*, ou Karhunen-Loeve [Faloutsos\_1996]. Utilizando as características extraídas previamente da imagem, o MAM constrói a estrutura de índices calculando as distâncias entre elas, procedimento que deveria corresponder à comparação entre as imagens originais [TrainaJr 2000] [Traina 2001].

O que acontece quando duas imagens distintas produzem dois conjuntos de características iguais? Outro nível de comparação deve ser realizado, isto é, um filtro subsequente utilizando outro conjunto de características deveria ser ativado para completar a seleção apropriadamente, ou, no pior caso, seria realizada uma comparação direta entre as imagens (pixel a pixel). Como o identificador único, chamado *oid*, da imagem é também armazenado junto com o vetor de

características na estrutura de indexação, esse problema pode ser facilmente corrigido [TrainaJr\_2000] [Traina\_2001].

#### 3.2. Definições

Formalmente um **espaço métrico** é um par M=(D, d), onde D é o domínio de características de objetos, que são as chaves de indexação, e d() é uma função distância métrica que satisfaz as seguintes propriedades:

- 1. Simetria: d(x,y) = d(y,x);
- 2. Não negatividade:  $0 < d(x,y) < \infty$ ,  $x \neq y e d(x,x) = 0$ ;
- 3. Designaldade triangular:  $d(x,y) \le d(x,z) + d(z,y)$ .

onde: x, y e z são objetos pertencentes a D.

Basicamente, há dois tipos básicos de consultas por similaridade: consulta por abrangência (range queries) e consulta pelos vizinhos mais próximos (nearest neighbors query) [Traina\_2001].

**Definição 3.1.1 (Abrangência ou** *range query*): Dado um conjunto de objetos  $O = \{O_1, O_2, ..., O_n\}$  de um domínio D, uma função distância métrica  $d(\cdot)$ , um objeto de consulta de consulta  $Q \in D$  e uma distância de busca máxima r(Q), a consulta por abrangência:

 $range(Q, r(Q)) = \{ O_i \mid O_i \in O \text{ e } d(O_i, Q) \le r(Q) \}$  seleciona todos os objetos  $O_i$  do conjunto O.

Um exemplo deste tipo de consulta seria: "*Encontre as estrelas que estão até 10 anos-luz de distância do Sol*". Nesse caso, o objeto de consulta é 'Sol', o domínio *D* é o conjunto de estrelas do Universo, e o 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].

**Definição 3.1.2** (k Vizinhos mais próximos, k-nearest neighbor ou k-NN do inglês): Dado um conjunto de objetos  $O=\{O_1,O_2,...,O_n\}$  de um domínio D, uma função distância métrica  $d(\cdot)$ , um objeto de consulta  $Q \in D$  e um inteiro  $k \ge 1$ , a consulta pelos k vizinhos mais próximos:

k-NN(Q)={ $A_i \mid A_i \in A, A \subseteq O, \mid A \mid = k \ e \ \forall A_i \in A, O_i \in O$ - $A, d(Q,A_i) \leq d(Q,O_i)$ } seleciona os k objetos da estrutura que estão mais próximos de Q.

Utilizando o exemplo anterior, a consulta "Selecione as 5 estrelas mais próximas do Sol" busca os k=5 objetos 'estrelas' mais próximos ao mesmo objeto 'Sol', considerando o mesmo domínio de objetos [Traina 2001].

#### 3.3. Estado da Arte em Estruturas Métricas

O primeiro trabalho voltado ao tratamento de dados métricos descrito na literatura é o de Burkhard e Keller [Burkhard\_1973], através da utilização de três técnicas diferentes para solucionar o problema de encontrar o melhor par (*matching*) entre palavras dada uma consulta textual em um arquivo ASCII. Para a comparação das palavras, eles utilizaram uma distância métrica que retornava apenas valores inteiros.

A primeira técnica proposta em [Burkhard\_1973] é uma estrutura de decomposição hierárquica não binária. No início um objeto-chave é selecionado arbitrariamente. Calcula-se a distância dos demais objetos para a chave escolhida e agrupa-se os objetos que encontram-se a mesma distância. Como a distância é discreta, esse agrupamento é sempre possível.

A segunda técnica descrita em [Burkhard\_1973] decompõe o espaço de dados em um número fixo de conjunto de chaves. Para cada conjunto de chaves, uma chave *central* é selecionada arbitrariamente, bem como um *raio* de cobertura, o qual especifica a distância máxima entre os objetos desse conjunto e a chave central. Os conjuntos de chaves são subdivididos consecutivamente da mesma forma gerando uma estrutura de árvore multivias.

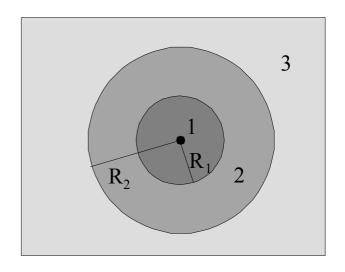
Já a terceira técnica apresentada em [Burkhard\_1973] é bastante similar à segunda, onde se coloca uma restrição adicional para o 'diâmetro' (a distância máxima entre quaisquer dois objetos de um grupo) de cada um dos grupos seja menor que uma dada distância k, onde k é diferente para todos os níveis da árvore. Os grupos que satisfazem esse critério são chamados de 'clique'. Esta técnica busca encontrar o clique maximal em cada nível, guardando os objetos centrais dos nós para direcionar e minimizar (podar) a busca. Observe que objetos chave podem aparecer em mais do que um clique, e a idéia é selecionar a chave central que aparece no maior número de cliques.

Shasha e Wang [Shasha\_1990] sugerem armazenar previamente as distâncias entre objetos para acelerar consultas por similaridade. Em objetos complexos, como imagens, o

cálculo de distâncias é computacionalmente caro, e essa é uma boa abordagem. Essa técnica é efetiva apenas quando o tamanho do conjunto de dados é pequeno.

Em Uhlmann [Uhlmann 1991] propôs duas estruturas de índice hierárquicas para consultas por similaridade. A primeira dela é a vp-tree (vantage-point tree). A vp-tree particiona

o espaço em áreas esféricas em volta de um vantage-point, que é definido a cada nível da árvore. A Figura 3 ilustra esse conceito. Essa técnica é similar à primeira apresentada em [Burkhard 1973]. As distâncias entre os objetos e o vantage-point de cada nó são computadas e guardadas. Através da mediana do intervalo de distâncias, os objetos são particionados em dois grupos. O primeiro grupo armazena os objetos cujas distâncias para o vantage-point é menor ou igual à Figura 3 - Particionamento de uma vp-tree no nível da raiz mediana e o segundo grupo armazena os apresentadas em tons de cinza distintos.



com fanout igual a 3. As três regiões são rotuladas (1,2,3) e

demais objetos. O primeiro grupo é colocado à esquerda do vantage-point e o segundo grupo à direita. Para cada grupo aplica-se a técnica recursivamente, cujo resultado final será uma árvore binária. A vp-tree pode ser generalizada em uma árvore com fanout (número de saídas) maior do que dois.

Yiannilos em [Yiannilos 1993] apresentou resultados analíticos para a vp-tree além de sugerir técnicas mais elaboradas na seleção dos vantage-points. Um algoritmo bastante interessante para possibilitar consultas de nearest neighbor em vp-trees foi proposto por Chiueh em [Chiueh 1994].

A estrutura gh-tree (generalized hyperplane tree) foi também proposta por Uhlmann [Uhlmann 1991]. Essa estrutura é construída da raiz para as folhas da seguinte forma: selecionase dois objetos chave do conjunto de dados e os demais objetos são associados a eles pela sua distância. Isso é, os objetos são associados ao objeto-chave mais próximo. Cada conjunto é tratado da mesma forma recursivamente, gerando uma árvore. Nesse caso, não se pode generalizar o número de saídas para mais do que dois (o que pode ser feito na vp-tree). Se os dois objetos chave são selecionados apropriadamente, a árvore tende a ser balanceada.

Utilizando a idéia de particionar o espaço de dados através de um objeto de referência, a FQ-tree (*Fixed Queries tree*) foi proposta em [Baeza-Yates\_1994]. A grande diferença entre a FQ-tree e a vp-tree é que a primeira usa o mesmo objeto de referência para todos os nós internos do mesmo nível. Assim, o número dos objetos de referência é igual à altura da árvore.

Visando responder consultas de *nearest neigbor*, Brin propôs a estrutura GNAT (*Geometric Near-Neighbo Access Tree*) [Brin\_1995]. No nível mais alto da árvore, seleciona-se k objetos de quebra. Os demais objetos do conjunto de dados são associados a um dos k objetos selecionados (àquele que estiver mais próximo). Para cada objeto de quebra, as distâncias mínima e máxima dos objetos associados a ele são armazenadas. A árvore é então construída recursivamente para cada grupo obtido num nível. O número de objetos de quebra k é parametrizado e é escolhido para cada conjunto de dados, dependendo da sua cardinalidade.

Com a proposta de ampliação do número de objetos chave para cada nó de uma árvore, a mvp-tree (*multi-vantage point tree*) foi proposta em [Bozkaya\_1997]. Dessa forma a mvp-tree armazena um número maior de distâncias pré-calculadas entre objetos do conjunto de dados e dos objetos chaves, conseguindo diminuir entre 20% e 80% o número de distâncias a serem calculadas para responder consultas por abrangência (*range queries*), quando comparada com a vp-tree. Tal medida de comparação foi obtida através da comparação da mvp-tree com dois *vantage-points* por nó. Observe que a mvp-tree armazena todas as distâncias entre os objetos que se encontram na folha da árvore com todos os *vantage points* que estão no caminho até ele desde a raiz.

Apesar de terem sido concebidas para dar suporte a consultas por similaridade, todas as estruturas métricas descritas até aqui são estáticas, não permitindo inserções ou exclusões posteriores à construção da árvore. Isso é uma desvantagem, dado que novas imagens e dados devem ser adicionados ao sistema conforme os exames de pacientes vão sendo realizados.

A primeira estrutura métrica dinâmica apresentada na literatura foi a M-tree, proposta por Ciaccia, Patella e Zezula [Ciaccia\_1997B]. A M-tree foi desenvolvida utilizando uma técnica de construção de baixo para cima (*bottom-up*), que ao mesmo tempo mantém a árvore balanceada e ainda possibilita novas inserções após a construção da árvore. A M-tree possui dois tipos

diferentes de nós, os internos e as folhas. Os nós internos armazenam o objeto centro deste nó, bem como a distância dele para seu nó pai (para o nó raiz da árvore não existe naturalmente essa distância), o raio de cobertura da região indexada por essa subárvore, e um vetor de ponteiros para suas subárvores. Os nós folhas (que armazenam os objetos) contêm seus identificadores de objetos (*Olds*), bem como as características dos objetos que estão sendo utilizadas na indexação do conjunto de dados. A busca de objetos na M-tree é acelerada pelo processo de poda que utiliza o raio de cobertura de subárvores, a distância entre os objetos e os que são centro dos nós, bem como do objeto centro do nó pai. Essa poda de subárvores é otimizada pela utilização da desigualdade triangular, uma propriedade dos espaços métricos, como visto na seção 3.2.

Em [Santos\_2001] foi proposta uma nova família de MAMs denominada Omni-family que utiliza estruturas de índices já presentes em gerenciadores de dados comerciais. A técnica Omni baseia-se na utilização de objetos especiais chamados **foci** que propiciam a diminuição de cálculo de distância entre os objetos quando se estiver realizando consultas por similaridade. Obteve-se nesse trabalho reduções nos cálculos de distância de até 10 vezes durante o processamento de consultas aos *k* vizinhos mais próximos.

#### 3.4. Estrutura de Indexação para Dados Métricos Slim-tree

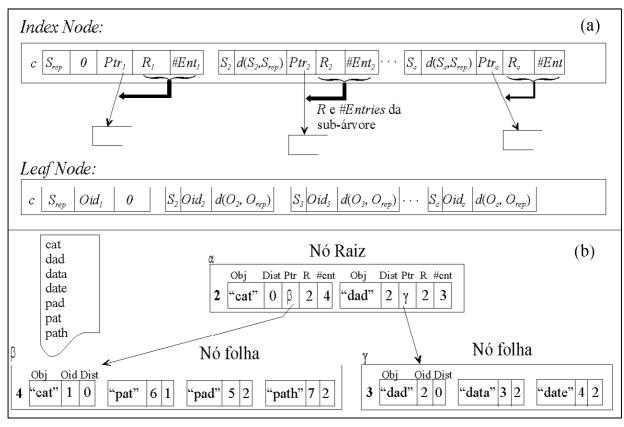
A Slim-tree foi proposta por Traina et. al. em [TrainaJr\_2000]. A Slim-tree é também uma estrutura balanceada e dinâmica, permitindo inserções posteriores à criação da árvore. Quando comparada com a M-tree nas mesmas condições, sempre a sobrepujou, tanto em termos de número de acessos a disco quanto em termos de número de distâncias calculadas para responder consultas por abrangência (*range queries*) e, portanto, também em tempo total de execução. A Slim-tree possui nós internos, que são chamados de *indexnodes*, e nós folha, que são os *leafnodes*. O tamanho da página que armazena cada nó é de tamanho fixo e cada nó pode armazenar um número máximo de objetos *C*. A estrutura dos nós folha que armazenam todos os objetos é:

leafnode [vetor de 
$$\langle Oid_i, d(S_i, Rep(S_i)), S_i \rangle$$
]

onde,  $Oid_i$  é o identificador do objeto  $S_i$  e  $d(S_i, Rep(S_i))$  é a distância entre o objeto  $S_i$  e o objeto central (representative) deste nó folha  $Rep(S_i)$ . A estrutura de um nó interno é:

indexnode [vetor de  $\langle S_i, R_i, d(S_i, Rep(S_i)), Ptr(TS_i), NEntries(Ptr(TS_i)) \rangle$ ]

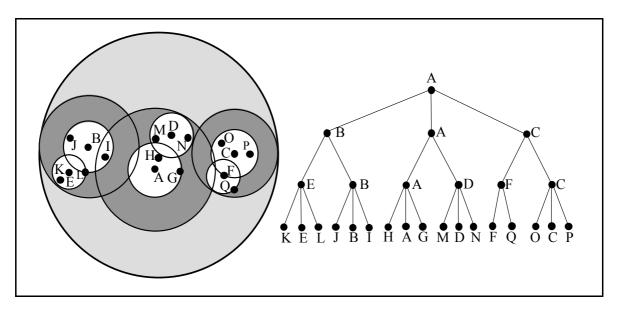
onde,  $S_i$  armazena o objeto que é o centro da subárvore apontada por  $Ptr(TS_i)$ , e  $R_i$  é o raio de cobertura da região. A distância entre  $S_i$  e o centro deste nó  $Rep(S_i)$  é armazenada em  $d(S_i)$ ,  $Rep(S_i)$ . O ponteiro  $Ptr(TS_i)$  indica o nó raiz da subárvore cuja raiz é  $S_i$ . O número de entradas presentes nos nós apontados por  $Ptr(TS_i)$  é armazenado em  $NEntries(Ptr(TS_i))$ . A Figura 4 apresenta graficamente a estrutura da Slim-tree em (a) e um exemplo de sua construção para um conjunto de 7 palavras. A Figura 5 apresenta uma visão geral da organização de 17 objetos, rotulados de A até Q, que são armazenados numa Slim-tree de 3 níveis, na qual a raiz encontra-se no nível zero e os objetos no nível das folhas (nível 2) [TrainaJr\_2000] [Traina\_2001].



**Figura 4 -** (a) Representação gráfica da Slim-tree apresentando os nós internos (*indexnodes*) e os nós folhas (*leafnodes*); (b) um exemplo de uma Slim-tree armazenando 7 palavras utilizando a distância  $L_{edit}^{I}$ .

.

<sup>&</sup>lt;sup>1</sup> Distância Levenshtein ou Edit (L<sub>edit</sub>).



**Figura 5 -** Slim-tree armazenando 17 objetos representada graficamente por raios de cobertura e por uma estrutura de árvore convencional.

#### 3.4.1. Construindo a Slim-tree

O mecanismo de inserção de objetos na Slim-tree é o seguinte: a partir do nó raiz, o algoritmo tenta localizar um nó que possa receber o novo objeto. Se nenhum nó se qualifica, seleciona-se o nó cujo centro está mais perto do novo objeto. Se por outro lado, mais de um nó se qualifica, o algoritmo *ChooseSubtree* é executado para selecionar o nó onde será inserido o novo objeto. Este processo é aplicado recursivamente para todos os níveis da árvore [TrainaJr\_2000] [Traina\_2001].

A Slim-tree possui três opções para o algoritmo *ChooseSubtree*:

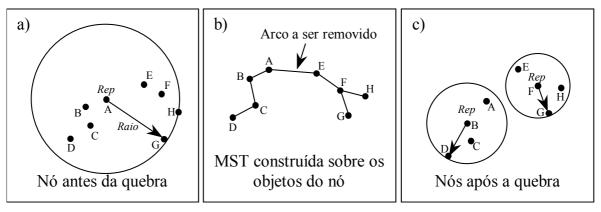
- *random* seleciona aleatoriamente o nó para inserir o novo objeto entre os que se qualificaram.
- mindist seleciona o nó cuja distância de seu representativo (centro) para o novo objeto seja a menor.
- *minoccup* seleciona o nó que esteja com o menor número de objetos armazenados, dentre os que se qualificaram.

O campo *NEntries* presente em todo nó intermediário (*indexnode*) da Slim-tree é utilizado pelo algoritmo *minoccup*, onde o nó com menor valor de *NEntries* é selecionado. É

interessante notar que, utilizando a opção *minoccup* do algoritmo *ChooseSubtree*, obteve-se árvores mais compactas (maior taxa de ocupação dos nós), o que redundou em um número menor de acessos a disco para responder consultas por similaridade [TrainaJr\_2000] [Traina\_2001].

Durante o processo de inserção de objetos pode acontecer do nó escolhido já estar completo (já atingiu sua taxa de ocupação máxima). Nesse caso deve-se alocar um novo nó no mesmo nível do anterior, e os objetos que estavam nesse nó, mais o novo nó a ser inserido devem ser então re-distribuídos entre os dois nós. A Slim-tree possui os seguintes algoritmos para efetuar a quebra de nós (*splitting*) [TrainaJr\_2000] [Traina\_2001]:

- random seleciona aleatoriamente os dois objetos representativos para os novos nós,
   e os demais objetos são distribuídos entre eles pela menor distância entre o objeto e o
   representativo. Deve-se respeitar a taxa de ocupação mínima dos nós.
- *minMax* consideram-se como candidatos todos os possíveis pares de objetos como representativos. Para cada par possível, associa-se os demais objetos a um dos representativos. Serão escolhidos como representativos o par de objetos que minimizar o raio de cobertura da subárvore. A complexidade do algoritmo é  $O(C^3)$  onde C é a capacidade dos nós. Apesar deste algoritmo ser extremamente custoso, ele é designado na literatura como o que consegue obter árvores que possibilitam consultas mais eficientes [Ciaccia\_1998].
- *MST* constrói-se a árvore de caminho mínimo (*minimal spanning tree MST*) [Kruskal\_1956], e um dos arcos mais longos da MST é removido. Dessa forma obtém-se dois agrupamentos, que serão os objetos associados a cada nó. Nos testes realizados, verificou-se que esse algoritmo possibilitou a construção de Slim-trees praticamente equivalentes às construídas utilizando o algoritmo de quebra de nós *minMax*, porém em muito menos tempo. Isso pode ser também notado pela complexidade do algoritmo que é  $O(C^2 \log C)$  com  $O(C^2)$  cálculos de distância. A Figura 6 ilustra o mecanismo de quebra de nós utilizando este algoritmo.



**Figura 6** – Mecanismo de quebra de nós utilizando o algoritmo MST. (a) Nó antes da quebra. (b) Construção da árvore de caminho mínimo (MST). (c) Dois nós gerados pela quebra do nó em (a).

Note-se que a Slim-tree cresce de um nível quando a raiz da árvore está completa e um novo elemento deve ser inserido nela. Nesse caso, a raiz divide-se e uma nova raiz deve ser criada com dois representativos, e dessa forma a árvore cresce um nível [TrainaJr\_2000] [Traina\_2001].

#### 3.4.2. Problema de Sobreposição

Idealmente, os nós de uma árvore para indexação de dados não deveriam se sobrepor, como acontece com as árvores binárias e árvores-B, permitindo podar todos os nós que não possuam objetos candidatos a responder consultas por similaridade. Porém, isso não ocorre nas estruturas métricas, como também não ocorre na maioria dos métodos de acesso espacial, como por exemplo a família R-tree. Da mesma forma que nos demais MAMs, na Slim-tree os nós das regiões podem também ser sobrepostos (ver Figura 5). O aumento de sobreposição entre os nós de uma estrutura de índices é o grande responsável por diminuir sua eficiência para responder consultas. Isso ocorre porque mais nós da árvore serão consultados (todos os que estão se sobrepondo à região de consulta), e a poda de subárvores fica, portanto, prejudicada. Visando suplantar essa deficiência, a Slim-tree foi desenvolvida com o objetivo de diminuir a sobreposição entre os nós da árvore, bem como oferecer mecanismos para verificação da porcentagem de sobreposição existente na árvore [TrainaJr 2000] [Traina 2001].

Note-se que, até a apresentação da Slim-tree, não havia nenhuma forma de se medir a sobreposição entre nós em estruturas métricas, o que era sempre colocado na literatura como

impossível de ser realizado [Ciaccia\_1997A]. Não é sempre possível efetuar medidas de área e volume em espaços métricos por não haver a noção de dimensão para muitos dos conjuntos de dados. A técnica fundamental utilizada nos métodos de acesso espacial para efetuar medidas de interseção entre nós corresponde ao cálculo do "hiper-volume" da interseção entre nós sobrepostos, como por exemplo é feito na R\*-tree [Beckmann\_1990]. Em espaços métricos não é possível determinar como se calcular volumes ou áreas. Então, a proposta que foi efetuada para a Slim-tree em [TrainaJr\_2000] foi de, em vez de calcular o "volume" da intersecção entre nós sobrepostos, utilizar um mecanismo de estimação de sobreposição por contagem dos objetos que estão nesse "volume" hipotético. Assim, a técnica utilizada pela Slim-tree é a de computar o *número de objetos que encontram-se cobertos por mais de uma região (nó)*. Pode-se sistematizar essa idéia a partir das definições a seguir [TrainaJr\_2000] [Traina\_2001].

**Definição 3.4.2.1** - Sejam  $I_1$  e  $I_2$  dois nós internos de uma árvore métrica. A sobreposição entre  $I_1$  e  $I_2$  é definida como o número de objetos cobertos por ambas as regiões (subárvores) dividido pelo número total de objetos presentes nas duas subárvores [Faloutsos 1994].

Utilizando essa definição, pode-se utilizar as técnicas de otimização de desempenho para métodos de acesso espacial já consagradas na literatura da área para dados espaciais. Além disso, pode-se ter uma estimativa de quão apropriada ou 'boa' é uma árvore para um conjunto de dados [Faloutsos 1994].

Em uma árvore ideal, sem sobreposições, a busca a um objeto já indexado deveria permitir o acesso a apenas um nó a cada nível da árvore. Ou seja, para uma consulta puntual (uma consulta por abrangência com raio zero) em uma árvore ideal com, por exemplo, três níveis, somente três nós deveriam ser acessados. Nesse caso, o fator de sobreposição deveria ser zero. Já o pior caso seria quando todos os nós tivessem que ser acessados para responder a uma consulta puntual. Nessa situação o fator de sobreposição deveria ser igual a um. Através dessa intuição, pode-se definir o fator de sobreposição em uma árvore métrica em termos do *absolute fat-factor* definido a seguir [TrainaJr\_2002]:

**Definição 3.4.2.2** - Seja T uma árvore métrica com altura H e com M nós,  $M \ge 1$ . Seja N o número de objetos indexados. Então, o fator de sobreposição *absolute fat-factor*, ou resumidamente *fat-factor* de uma árvore métrica T é:

$$fat(T) = \frac{I_C - H * N}{N} \cdot \frac{1}{(M - H)}$$

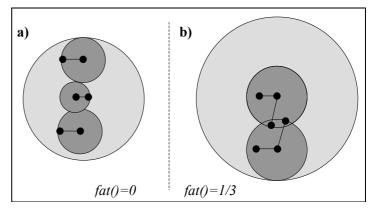
onde  $I_C$  é o número total de nós acessados para responder uma consulta puntual a todos os N objetos armazenados na árvore métrica. As definições 4.1 e 4.2 propiciam a apresentação do próximo Lema [TrainaJr 2002].

**Lema 1 -** Seja T uma árvore métrica. Então fat(T) é sempre um valor do domínio [0,1]. A pior árvore possível retorna o valor um e uma árvore ideal retorna zero [TrainaJr 2002].

**Prova:** Considere uma consulta puntual sobre um objeto armazenado na árvore T. Essa consulta tem que recuperar pelo menos um nó em cada nível da árvore. Particularmente, pelo menos os nós que constituem o caminho de inserção do objeto qualificam e devem ser lidos. Dessa forma, um limite inferior para  $I_C$ , que é o número de nós acessados para responder a uma consulta puntual sobre todos os N objetos, é H\*N. Nesse caso tem-se um *absolute fat-factor* igual a zero. Um limite superior para  $I_C$  ocorre quando é necessário acessar todos os M nós, ou seja M\*N. Isso

resulta em um *absolute fat-factor* igual a um. Como *absolute fat-factor* é uma função linear em  $I_C$ , e  $H^*N \le I_C \le M^*N$ , tem-se que o valor do *absolute fat-factor* está sempre na faixa entre zero e um [TrainaJr 2002].

Observe o exemplo colocado na Figura 7, que apresenta duas árvores e seus *fat-factors*. Nessa



**Figura** 7 – Duas árvores armazenando o mesmo conjunto de dados, porém com número de nós e absolute fat-factors diferentes. Os nós raízes são desenhados com linhas tracejadas e cor mais clara.

figura, o elemento representativo de um nó, que está no centro do mesmo, está sendo indicado conectado ao elemento mais distante dele no nó, o que também delineia o raio deste nó. O cálculo do *fat-factor* dessas árvores é simples. Por exemplo para a árvore da Figura 7(a) tem-se que  $I_C = 12$ , H = 2, N = 6 e M = 4, o que retorna um *absolute fat*=0. Para a árvore da Figura 7(b) tem-se que  $I_C = 14$ , H = 2, N = 6 and M = 3, retornando um *absolute fat*=1/3. Ambas as árvores possuem apenas dois níveis [TrainaJr 2002].

#### 3.4.2.1. Reorganização dos Nós da Árvore – Algoritmo Slim-down

O objetivo dessa subseção é mostrar como melhorar uma árvore métrica já construída para um conjunto de dados. Os fatores de sobreposição *absolute fat-factor: fat*() e *relative fat-factor: rfat*(), além de indicar a quantidade de sobreposição entre nós da árvore bem como a comparação entre árvores, permitem também avaliar se tal árvore pode ser melhorada, em termos de diminuição de número de acessos a disco para responder às consultas por similaridade. Pela própria definição de *fat*(), percebe-se que o desejável é que as árvores construídas apresentem o menor valor possível para esse fator. Ou seja, que o número de objetos nas regiões de sobreposição seja o menor possível. Assim, o que um algoritmo de reorganização da árvore deveria fazer é em primeiro lugar: diminuir o número de objetos nas interseções de nós de mesmo nível; em segundo lugar: diminuir o número de nós da árvore.

Em [TrainaJr\_2000] foi proposto o algoritmo de reorganização para árvores métricas *Slim-down*, que atua sobre uma árvore já construída. Esse algoritmo é apresentado na Tabela 3 e a Figura 8 mostra graficamente o seu funcionamento.

**Tabela 3 -** Algoritmo de Reorganização de árvores métrica: Slim-down [TrainaJr 2000].

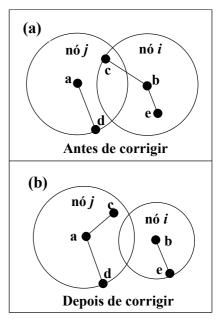
**Entrada**: uma árvore métrica T, e o nível h da árvore; **Saída**: uma árvore T' aprimorada

#### Início

- 1. Para cada nó *i* do nível *h* da árvore, encontre *c*, o objeto mais distante do representativo *b* do referido nó.
- 2. Encontre *j*, um nó irmão do nó *i*, tal que *c* também seja coberto por *j*. Se *j* existe e ainda não está completo, remova *c* de *i* e re-insira-o em *j*.
- 3. Se o nó i não ficou vazio, corrija seu raio. Caso contrário, remova o nó i.
- 4. Os passos 1 a 3 devem ser aplicados sequencialmente sobre todos os nós da árvore. Se após um ciclo completo desses 3 passos, ao menos um objeto moveuse de um nó para outro, o ciclo deve ser aplicado novamente.

Fim

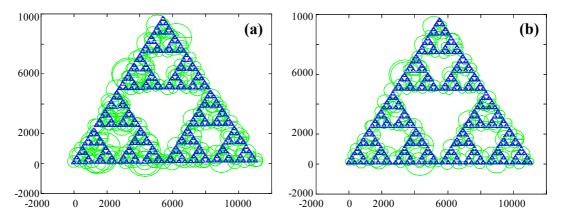
Observe que na Figura 8(a) o objeto c que encontra-se na intersecção entre os nós i e j, é transferido do nó i para o nó j no passo 2 do algoritmo, e como este era o objeto mais distante do representativo do nó i (objeto b), o raio do nó i deve ser diminuído de acordo. Como a figura sugere, o próximo objeto mais distante do representativo, que é o objeto e, passa a definir o novo raio do nó i. Essa diminuição do raio é indicada na Figura 8(b), após a correção da subárvore. Note-se que o objeto c, agora armazenado no nó j, encontra-se fora da intersecção entre os nós, ocorrendo uma diminuição da contagem  $I_C$ . Durante a execução desse algoritmo relaxa-se a restrição de ocupação mínima dos nós da



**Figura 8** – Funcionamento do algoritmo Slim-down.

árvore, e aqueles que se tornam vazios durante o processo podem ser removidos da árvore (passo 3 do algoritmo). Isso leva a uma diminuição do número *M* de nós da árvore [TrainaJr\_2000].

A Figura 9 mostra o primeiro nível da Slim-tree construída sobre o conjunto de dados *Sierpinsky*. Na parte (a) da figura, os nós são apresentados antes da execução do algoritmo Slim-down. O *absolute fat-factor* dessa árvore é igual a 0.03, o que indica uma árvore com pouca sobreposição. Porém, após aplicar o algoritmo de correção da árvore Slim-down, pode-se ver que claramente os nós são menores e que também há menos nós para indexar os dados, após a aplicação do algoritmo Slim-down [TrainaJr 2000].



**Figura 9** - Indexando o conjunto de *Sierpinsky* com 9850 pontos, e utilizando o algoritmo de quebra de nós aleatório. Apresenta os nós acima das folhas. (a) Antes de aplicar o algoritmo *Slim-down*, com *fat-factor*=0,03. (b) Depois de aplicar o algoritmo *Slim-down*, com *fat-factor*=0,01.

A geração da visualização sobre a informação armazenada na Slim-tree é bastante rápida. Por exemplo, utilizando uma máquina com sistema operacional Windows-NT e processador Intel de 750 Mhz e 256 Mb de RAM, a imagem apresentada na Figura 9(a) levou 1.58 segundos para ser gerada. A correção da estrutura pelo algoritmo Slim-down, como apresentado na parte (b) da mesma figura levou 2.72 segundos [TrainaJr 2000].

## 3.4.3. Comparando Árvores Diferentes Construídas sobre o Mesmo Conjunto de Dados

O *absolute fat-factor* permite medir a quantidade de sobreposição presente em uma dada árvore métrica *T*. Porém, este fator não permite a comparação entre árvores diferentes construídas sobre o mesmo conjunto de dados. Isto é, métodos de quebra de nós ou opções do algoritmo de seleção de nós (*ChooseSubtree*) levam, muitas vezes, a árvores com número de nós *M* e alturas de árvore *H* diferentes [TrainaJr 2000] [Traina 2001].

Para possibilitar a comparação entre árvores distintas construídas sobre o mesmo conjunto de dados, torna-se necessária uma abordagem diferente. Para isso utilizou-se a comparação da árvore construída com uma árvore canônica que possua o número mínimo de nós, ou seja, que tenha todos os nós completos, com a possível exceção de um nó em cada nível. Essa árvore canônica além de ter o menor número de nós possível  $M_{\min}$ , também é aquela que possui altura mínima  $H_{\min}$ . Essa nova medida é chamada de *relative fat-factor*, e considera então não mais o número de nós e a altura da árvore, mas o número de nós acessados para responder uma consulta puntual sobre todos os objetos da árvore real, sobre a altura e número de nós de uma árvore canônica [TrainaJr 2000] [Traina 2001]. Tal discussão leva à seguinte definição:

**Definição 3.4.3.1** - O *relative fat-factor* de uma árvore métrica T com mais do que um nó  $(M_{min} > 1)$  é

$$rfat(T) = \frac{I_C - H_{\min} * N}{N} \cdot \frac{1}{(M_{\min} - H_{\min})}$$

O valor retornado por *rfat*() varia entre zero e um número positivo que pode ser maior do que um. Embora não limitado a um, ele permite a comparação direta de duas árvores com

*relative fat-factor* distintos, sendo que a árvore com valor de *relative fat* menor propiciará menos acessos a disco para responder às consultas por similaridade [TrainaJr\_2000] [Traina\_2001].

A altura mínima de uma árvore indexando N objetos é  $H_{\min} = \left[\log_C N\right]$ , e o número mínimo de nós para indexar um conjunto de dados pode ser calculado como  $M_{\min} = \sum_{i=1}^{H_{\min}} \left\lceil N/C^i \right\rceil$  onde C é a capacidade dos nós [TrainaJr\_2000] [Traina\_2001].

Deve-se notar que tanto o *absolute fat-factor* quanto o *relative fat-factor* estão diretamente relacionados com a taxa de sobreposição entre regiões do mesmo nível de uma árvore. O *absolute fat-factor* indica quão boa uma dada árvore é com respeito a sua quantidade de sobreposição, não se preocupando se os nós estão bem ocupados ou não, o que leva à otimização de espaço em disco se os nós estiverem com alta taxa de ocupação. Já o *relative fat-factor* permite a comparação entre duas árvores para o mesmo conjunto de dados, considerando tanto a quantidade de sobreposição quanto à ocupação eficiente de espaço em disco para armazenar os dados [TrainaJr 2000] [Traina 2001].

A possibilidade de avaliar a proposição entre nós através do *absolute fat*- e do o *relative fat-factor* permitiu o desenvolvimento de algoritmos de aprimoramento da árvore Slim-tree tanto durante o processo de construção quanto após a construção de uma árvore. Segundo os testes realizados, em que foram comparados os desempenhos da Slim-tree e da M-tree, verificou-se que a Slim-tree chegou a ter melhorias em termos de diminuição de acesso a disco para responder consultas por abrangência de mais de 200% [TrainaJr 2002].

#### 3.4.4. Visualização dos Dados na Slim-tree

Uma ferramenta muito interessante que está anexada à Slim-tree é o módulo visualizador que permite 'ver' o conjunto de dados indexado, juntamente com a estrutura de nós formada pela hierarquia da árvore. Esta capacidade de visualização permite o tratamento de conjuntos de dados métricos adimensionais, como por exemplo um conjunto de palavras, de uma forma mais intuitiva e prática. Ou seja, pode-se perceber visualmente o relacionamento entre as palavras desse conjunto, além da indicação de agrupamentos e membros isolados. Além disso, a visualização tanto do conjunto de dados, como a estrutura de organização dos mesmos na

indexação auxilia tanto na inspeção visual para verificação de como a árvore está, como em ferramentas para mineração de dados (*data mining*) visuais e interativas [Faloutsos\_1995].

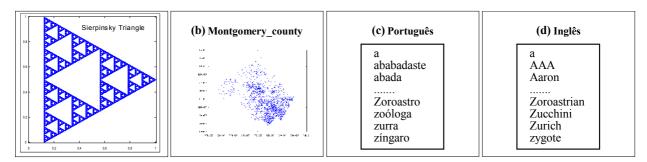
O algoritmo utilizado para construir essa visualização foi baseado no algoritmo *FastMap*, proposto por Faloutsos e Lin em [Faloutsos\_1995]. Tal algoritmo mapeia e permite então desenhar N pontos em k dimensões, onde k é definido pelo usuário. Em uma ferramenta de visualização, k é geralmente igual a 2 ou 3, tornando a visualização mais natural e intuitiva de ser compreendida. Na disposição espacial gerada pelo algoritmo, as distâncias entre os pontos são preservadas o máximo possível, para manter a percepção visual do relacionamento entre os objetos do conjunto original de dados. A distribuição dos nós da árvore é também apresentada graficamente, conforme a solicitação do usuário.

Para ilustrar o funcionamento da visualização sobre os dados indexados na Slim-tree nesse texto, utilizou-se os quatro seguintes conjuntos de dados, que são graficamente representados na Figura 10:

- Sierpinsky um conjunto com 9.941 pontos no espaço bidimensional que ilustra um conhecido fractal matemático. A Figura 10 (a) apresenta esse conjunto.
- Montgomery um conjunto de dados geográficos sobre as interseções de ruas do condado de Montgomery, Maryland - EUA. Este conjunto, possui 27.282 pontos e foi obtido no Censo norte-americano de 1989<sup>2</sup>. A Figura 10 (b) ilustra visualmente esse conjunto.
- *Português* um sub-conjunto das palavras da língua portuguesa, contendo 21.473 palavras retirado de um dicionário<sup>3</sup>), como indica a Figura 10(c).
- Inglês um sub-conjunto das palavras da língua inglesa, como indica a Figura 10(d).

<sup>2</sup> Bureau of the Census - "Tiger/Line Precensus Files: 1990 Technical Documentation". Bureau of the Census. Washington, D.C. 1989.

<sup>&</sup>lt;sup>3</sup> Dicionário utilizado pelo sistema ReGra - R. T. Martins, R. Hasegawa, M. G. V. Nunes, G. Montilha e O. N. Oliveira: "Linguistica Issues in the Development of ReGra: a Grammar Checker for Brazilian Portuguese", *Natural Language Engineering*, Vol. 4, No. 4, dezembro 1997, pp. 287-307.

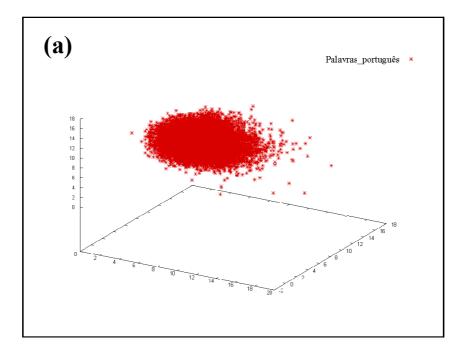


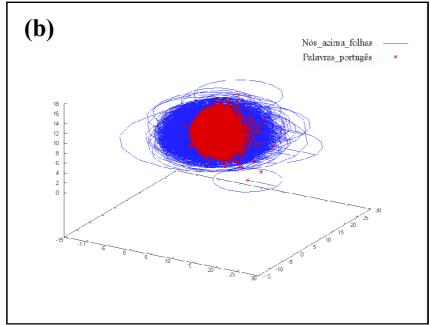
**Figura 10 -** Conjuntos de dados utilizados: (a) triângulo de *Sierpinsky*. (b) *Montgomery*. (c) conjunto de palavras em *Português*. (d) conjunto de palavras em *Inglês*.

A Figura 11(a) apresenta uma visão tridimensional do conjunto *Português*. Para a indexação das palavras, utiliza-se a distância Levenstein<sup>4</sup>, que por retornar somente valores inteiros gera um efeito visual de quantização entre os objetos. Isso ocorre porque a faixa de valores retornado pela distância varia de 0 até a quantidade de letras da maior palavra presente no conjunto de dados, que usualmente é bastante pequeno (esse valor é igual a 26 no conjunto Português). Essa árvore foi gerada utilizando o método de quebra de nós baseado na árvore de caminho mínimo e escolha pela ocupação mínima. Cada nó da árvore pode ter no máximo 30 palavras. A árvore possui então três níveis e 1300 nós [TrainaJr 2000] [Traina 2001].

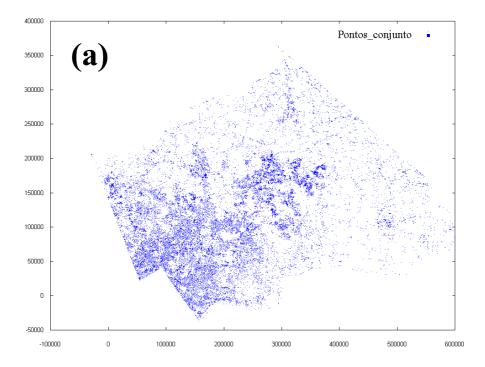
A Figura 11(b) apresenta a estrutura hierárquica dos nós da árvore sobrepostos ao mesmo conjunto de dados visualizado na parte (a). Como não há uma forma característica para a distância Levenstein, pois é uma métrica adimensional, as regiões dos nós estão apresentadas na visualização como projeções de esferas. A árvore apresentada foi construída utilizando o algoritmo de quebra de nós minMax, e somente os nós folhas estão apresentados. Nessa visualização pode-se perceber o alto grau de sobreposição entre os nós, o que obriga as consultas solicitadas sobre os dados a acessarem muito mais nós do que o necessário para respondê-las. Tal fato é demonstrado numericamente pelo alto valor de sobreposição fat() = 0.46 obtido dessa árvore [TrainaJr\_2000] [Traina\_2001].

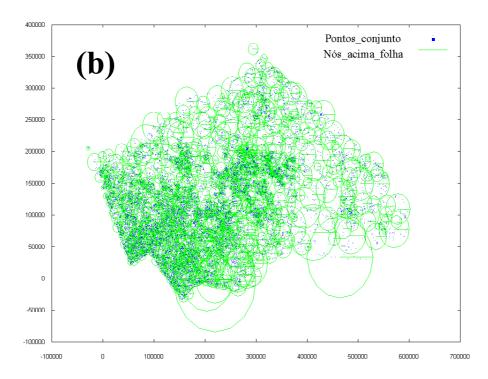
<sup>&</sup>lt;sup>4</sup> Distância Levenshtein ou Edit ( $L_{Edit}$ ) retorna um valor inteiro que é o número de caracteres inseridos, removidos ou substituídos para transformar uma cadeia de caracteres em outra. Por exemplo:  $L_{Edit}$ ("casa", "massa")=2, substitui 'c' por 'm' e insere 's'.





**Figura 11** — Slim-tree indexando um sub-conjunto de 21.473 palavras do dicionário da língua portuguesa. (a) Visualização tridimensional do conjunto de palavras indexadas. (b) Visualização tridimensional das palavras e dos nós da árvore do nível imediatamente superior a elas.

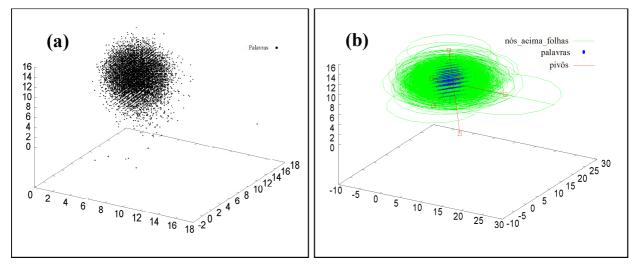




**Figura 12** – Visualização do conjunto Montgomery\_county indexado na Slim-tree. (a) Apenas os objetos do conjunto. (b) Os objetos e os nós imediatamente acima deles.

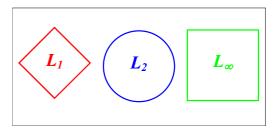
A Figura 12(a) apresenta a visualização dos objetos do conjunto de dados *Montgomery*, que é um conjunto de dados espaciais. Em (b) são sobrepostos os círculos representando os nós imediatamente acima dos nós folha (objetos do conjunto). Como esse conjunto de dados é bidimensional, o algoritmo *FastMap* apenas remapeia os objetos nos pontos do espaço de visualização. Porém, isso confirma que o mapeamento do algoritmo é consistente [Faloutsos\_1995] [TrainaJr\_2000] [Traina\_2001].

A Figura 13 apresenta uma visualização tridimensional do conjunto de palavras da língua inglesa *Inglês*. Na parte (a) da figura apenas as palavras são apresentadas. Pode-se ver que há um pequeno número de palavras que fogem à distribuição usual delas. Essas palavras são as exceções (*outliners*). Na parte (b) da figura estão sobrepostos os nós acima do nível das folhas, e os objetos pivôs e os eixos de projeção. Com isso podemos entender melhor o funcionamento do algoritmo *FastMap* [Faloutsos\_1995] [Traina\_2001] [TrainaJr\_2002].



**Figura 13** – Uma visão tridimensional do conjunto de palavras *English*. (a) As palavras e sua disposição no espaço. (b) O mapeamento do conjunto de palavras, os pivôs utilizados pelo algoritmo *FastMap*, e os círculos representando os nós acima das folhas na Slim-tree. Árvore construída com o algoritmo de quebra de nós minMax, pela seleção do raio mínimo e com 60 palavras por nó.

É interessante notar que a métrica  $L_p^5$  utilizada define o formato para os nós no domínio espacial. A Figura 14 apresenta tais formatos. As métricas não  $L_p$  são apresentadas como projeções de esfera. O sistema da Slim-tree suporta, além das métricas  $L_p$ , funções de distância colocadas por matrizes de distância e a distância de palavras  $L_{Edit}$  [Traina 2001] [TrainaJr 2002].



**Figura 14** – Formatos de nós ao utilizar as distâncias  $L_1$  (*Manhatan*),  $L_2$  (*Euclidian*)  $L_{\infty}$  (*Infinity*).

#### 3.5. Aplicações

A estrutura Slim-tree possui várias inovações sobre as estruturas métricas tradicionais, tais como: o *fat-factor*, que possibilita indicar se a estrutura de índices construída sobre um determinado conjunto de dados é eficiente ou não; o algoritmo Slim-down que efetua a reorganização dos dados organizados por ela, de forma a minimizar a taxa de acesso a disco quando efetuando consultas por similaridade. Além disso, a possibilidade de visualizar a organização da informação armazenada faz com que o usuário possa perceber onde estão os aglomerados e elementos de exceção pertencentes ao conjunto de dados, além do interrelacionamento entre os elementos de dados. Esse recurso presente na Slim-tree possibilita também o acompanhamento visual das consultas efetuadas sobre a base de imagens. Note, que dessa forma abre-se uma nova frente de mecanismos para manusear a informação já armazenada [Traina 2001] [TrainaJr 2002].

<sup>&</sup>lt;sup>5</sup> As distâncias  $L_p$  são definidas da seguinte forma:  $L_p((a_1,...,a_k),(b_1,...,b_k)) = \sqrt[p]{\sum_{i=1}^k |a_i - b_i|^p}$  e  $L_{\infty}((a_1,...,a_k),(b_1,...,b_k)) = \max_{i=1}^k |a_i - b_i|$ 

#### 3.6. Conclusões

Este capítulo apresentou os métodos de acesso métricos que foram desenvolvidos para suportar de forma natural as consultas por similaridade (*range query* e *k-NN*).

As estruturas métricas têm um papel primordial na recuperação de imagens baseada em conteúdo, uma vez que são elas que processam e guardam as informações sobre o grau de similaridade entre imagens.

Neste projeto está sendo utilizada a Slim-tree como estrutura de indexação, pois é a única estrutura métrica dinâmica que permite determinar *a priori* o seu comportamento com relação a recuperação de informações. Isto é feito a partir da medida *fat-factor*. Seu algoritmo foi adaptado para possibilitar consultas a imagens baseadas em conteúdo, como veremos no capítulo 5.

# Capítulo 4

### **Imagens e suas Características Inerentes**

#### 4.1. Introdução

As consultas por similaridade são as ferramentas principais utilizadas no desenvolvimento de sistemas de recuperação de imagens baseada em conteúdo (SiRIC). Pode-se então perceber a importância de tais consultas em sistemas que manipulam imagens. É muito difícil acontecer de uma aplicação precisar comparar imagens iguais, o que seria feito pela comparação pixel a pixel entre elas. O que se deseja, na maior parte dos casos, é a pesquisa por imagens que sejam parecidas ou similares.

O suporte a esse tipo de consultas é dado pelo emprego de uma estrutura de indexação para dados multidimensionais ou adimensionais, que é o caso de dados em espaços métricos, como foi discutido no capítulo 3 desta tese. Embora as estruturas métricas, ou os MAMs, utilizem muita menos informação dos dados para realizar a indexação, pois usam apenas a distância entre pares de objetos, já estão atingindo um desempenho similar e até mesmo sobrepujando as estruturas para dados espaciais no caso de conjuntos de alta dimensão. As estruturas de indexação espaciais utilizam informações sobre posicionamento dos dados, como intersecção e direção dos mesmos. A estrutura utilizada neste trabalho é a Slim-tree, que tem se mostrado a estrutura métrica mais eficiente para suportar a indexação de dados para busca por conteúdo em consultas por similaridade. Isso a torna, no atual estado da arte, a melhor opção para a implementação de um SiRIC como o aqui proposto.

#### 4.2. Extração de Características de Imagens

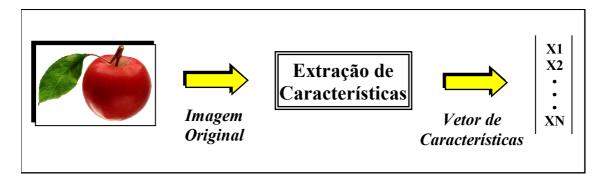


Figura 15 - Processo de Extração de Características de uma imagem.

Uma característica é uma função de uma ou mais medidas, calculadas de forma que quantifique alguma propriedade de um objeto. A Figura 15 apresenta o processo de extração de características. Considerando que o domínio de objetos tratados envolve imagens, temos que este processo produz um conjunto de *n* características que, juntas, formam o *vetor de características* de uma imagem. Pode-se então pensar em um espaço n-dimensional no qual todos os n elementos deste vetor possam ser localizados. Desta forma, qualquer objeto corresponde a um ponto neste espaço, denominado espaço de características. Após uma imagem ser segmentada em regiões, geralmente convém representar e descrever o conjunto resultante de pixels segmentados em uma forma adequada para processamento. Há dois modos de representar uma região, baseando-se nas características externas (isto é, suas fronteiras) ou nas internas (os pixels contidos na região) [Gonzalez 1993]. Geralmente, opta-se por uma representação externa quando o foco principal são as características morfológicas ou formas que estão presentes na imagem. Por outro lado, a representação interna é mais utilizada quando há interesse em propriedades refletivas, tais como cor e textura. Em ambos os casos, é importante que as características selecionadas como descritoras sejam tão insensíveis quanto possível a variações de tamanho, translação e rotação.

Brown em [Brown\_1992] apresenta uma taxonomia dos atributos ou características mais utilizados no processo de extração de características, e agrupa-os em cinco espaços de características. A Tabela 4 sumariza tais espaços e os atributos associados. É interessante notar

que os atributos mais efetivos em sistemas de recuperação de imagens baseada em conteúdo são aqueles que usam características de bordas e características de alto nível, como grafos [Gudivada\_1995] [Petrakis\_1997], e distribuição espacial de padrões [Petrakis\_2001].

**Tabela 4** – Taxonomia sobre espaços de características extraídas das imagens [Brown 1992].

Espaços de Característica	Atributos
Intensidade Bruta(Raw Intensity)	Baseia-se em intensidades de pixels
Bordas	Estrutura intrínseca, menos sensível a ruídos. Incluem
	contornos e superfícies.
	Estrutura intrínseca, posicionamento preciso.
Características Salientes	Envolvem interseção de linhas, cantos, pontos de
	curvatura alta.
Características Estatísticas	Usam toda informação, bons resultados para
	transformações rígidas, suposições.
	Incluem momentos invariantes, eixos
	principais/centróides.
	Utilizam relações e informação de alto nível, bons
	resultados para matching local e impreciso.
Características de Alto Nível	Envolvem características estruturais (grafos de
	configurações de sub-padrões) e sintáticas (gramáticas
	compostas a partir de padrões) e redes semânticas
	(regiões de cena e suas relações).
Matching Versus Modelos	Estrutura intrínseca e precisa, ruídos somente em uma
	imagem.
	Incluem atlas anatômico, mapa geográfico e modelo
	de objeto.

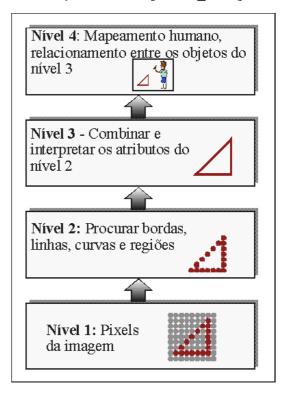
As características estatísticas (histograma de intensidades, média, desvio-padrão, entre outras), por representarem um comportamento mais global da imagem, são mais adequadas nos primeiros passos de seleção ou eliminação de candidatos. Já as características baseadas nas intensidades dos pixels em si são utilizadas quando se buscam imagens exatamente iguais, o que não é o caso geral de consultas por similaridade.

## 4.2.1. Atributos Visuais para Indexação e Recuperação de Imagens Baseada em seu Conteúdo

O processo de análise de imagem baseada no seu conteúdo pode ser modelado como uma hierarquia de abstrações [Aslandogan\_1999]. No primeiro nível estão os pixels da imagem, com a informação sobre cores ou brilho associada ao elemento. O segundo nível da abstração trabalha sobre atributos tais como bordas, cantos, linhas, curvas e regiões de cores. O terceiro nível da abstração procura combinar e interpretar os atributos do nível anterior, colocando-as sobre objetos que possuam tais características. O quarto e último nível da abstração aproxima-se do mapeamento humano, a partir do qual busca-se compreender o relacionamento entre os objetos presentes na imagem. A Figura 16 sintetiza os níveis de abstração descritos [Traina 2001].

Embora existam métodos de detecção e reconhecimento automático para certas classes de objetos e atributos (tipicamente geométricos), a sua eficácia depende muito da complexidade da imagem. Muitos objetos, valores de atributos e conceitos de alto nível, como o relacionamento entre os objetos, não podem ser obtidos por métodos automáticos. Nesses casos, utilizam-se métodos semi-automáticos, nos quais o usuário interage diretamente com a imagem ou o faz através de dicionários ou anotações [Traina\_2001].

A seguir são discutidos os principais atributos visuais e as técnicas para manipular tais atributos.



**Figura 16 -** Níveis de abstração para o processo de reconhecimento de objetos em imagens.

#### 4.2.1.1. Atributo Cor

As cores presentes em uma imagem possuem um papel bastante significativo na indexação e recuperação da mesma. Existem diferentes representações de cores que incluem desde o tradicional RGB (*red, green, blue*), o mais simples modelo que mapeia diretamente as características físicas do dispositivo de exibição, até o HSI (*hue, saturation, intensity*) que reflete mais precisamente o modelo de cores para a percepção humana. Na realidade todas as cores exibidas são criadas por combinações de quantidades apropriadas de vermelho, verde e azul. Um pixel de 24 bits em padrão RGB representa  $2^{24}$  ou aproximadamente 16.7 milhões de cores diferentes. Muitas vezes, para aumentar a eficiência no processamento, as cores da imagem são re-quantizadas de forma a diminuir o número de cores possível e facilitar o tratamento das mesmas através de seu histograma [Traina 2001].

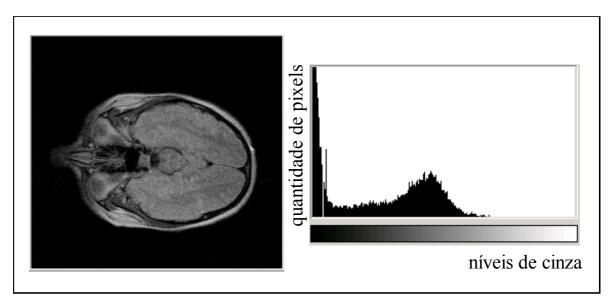


Figura 17 - Imagem quantizada em 256 níveis de cinza e seu histograma associado.

O histograma de cores calcula e apresenta o número de pixels de uma imagem para cada cor. A Figura 17 apresenta uma imagem de tomografia axial de crânio humano quantizado em 256 níveis de cinza e o histograma a ela associado. Dois histogramas de cores podem ser comparados pelo somatório de diferenças absolutas ou quadráticas sobre o número de pixels de cada cor. Tal esquema é bastante simples e tolerante a pequenas alterações na imagem. Dessa forma é natural que os histogramas de cores venham sendo estudados e implementados em sistemas de recuperação de imagens baseado em conteúdo, tanto acadêmicos [Ko 2000]

[Pass\_1996] [Pentland\_1996] [Hafner\_1995] quanto comerciais, como o QBIC [Flickner\_1995] (<a href="http://www.virage.com">http://www.virage.com</a>), e o Excalibur (<a href="http://www.excalib.com">http://www.excalib.com</a>) entre outros.

A popularidade da utilização de histogramas de cores em sistemas de recuperação de imagens baseada em conteúdo deve-se, principalmente, a três fatores [Pass 1996]:

- É computacionalmente simples e barato calcular histogramas de cores.
- Pequenas alterações de movimentação na imagem pouco afetam os histogramas.
- Objetos distintos freqüentemente possuem histogramas diferentes.

Porém, não é possível separar ou reconhecer imagens utilizando apenas o histograma de cores das mesmas, pois duas ou mais imagens bastante diferentes podem ter histogramas semelhantes. Ou seja, não há uma correspondência biunívoca entre a imagem e seu histograma de cores, levando ao surgimento do problema de ambiguidade. Tal fato é exemplificado na Figura 18. As quatro imagens em (a), (b), (c) e (d) possuem o mesmo histograma associado, o qual é apresentado em (e) [Traina\_2001].

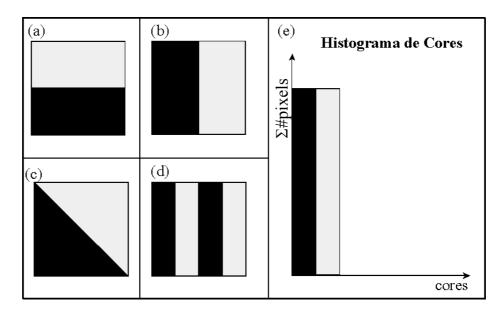


Figura 18 - Mesmo histograma de cores (dois níveis de cinza) associado a quatro imagens distintas.

Devido ao caráter ambíguo do histograma de cores de uma imagem, outros métodos devem ser utilizados conjuntamente. Outro problema dos histogramas é que, como o número de cores é grande (geralmente mais de 100 níveis), indexar vetores dessa dimensão é problemático.

Isso porque um histograma para 100 cores distintas pode ser visto como um ponto 100-dimensional, e para valores dessa ordem a maior parte das estruturas de índices espaciais colapsa, isto é, ocorre a tão falada "maldição da alta dimensionalidade" [Hinneburg\_1999] [Pagel\_2000], e o melhor método de acesso acaba sendo a busca seqüencial.

#### 4.1.1.2. Atributo Textura

Uma textura é um padrão visual no qual há um grande número de elementos visíveis arranjados de forma equânime com densidades variadas. Um elemento de textura é uma região de intensidade uniforme de formas simples que repete-se dentro de um intervalo (veja Figura 19). Assim, uma textura pode ser analisada ao nível de um intervalo (janela) denominando-se análise estatística. Se o procedimento for realizado ao nível do elemento da textura, é então denominado análise estrutural. Geralmente, utiliza-se a análise estrutural sempre que os elementos da textura podem ser claramente identificados. Por outro lado, aplica-se a análise estatística para texturas pequenas e não muito regulares [Tomita 1990].

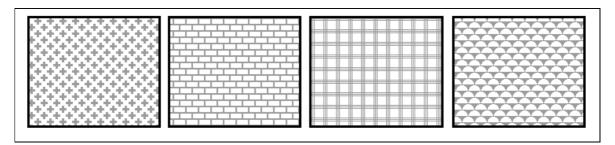


Figura 19 - Exemplificação de quatro elementos de textura.

Medidas estatísticas buscam caracterizar a variação de intensidade em uma janela de textura. Exemplos de tais medidas são: *contraste* (alto contraste, tipo textura de pele de zebra, versus baixo contraste, como a textura da pele de um elefante); *granularidade* (tamanho dos elementos do padrão) e *direcionalidade* (estampa de padrão xadrez em um tecido versus um padrão liso). Uma ferramenta para a manipulação de padrões estatísticos é o espectro de Fourier. Através da transformada de Fourier realizada sobre uma janela de textura gera-se uma assinatura. Janelas que possuam assinaturas próximas ou bastante similares podem, então, ser agrupadas.

A análise estrutural de texturas obtém os elementos de textura presentes na imagem, determinando seus formatos e estimando as regras de posicionamento. As regras de posicionamento descrevem como os elementos de textura são colocados com relação aos demais, além de estabelecer o relacionamento de vizinhança (conectividade), o número de elementos por unidade espacial (densidade), e sua regularidade (homogeneidade).

O tratamento de textura difere do realizado sobre cores devido ao fato de que as texturas são definidas sobre janelas ou regiões da imagem e não sobre pixels como as cores. A segmentação de uma imagem utilizando textura determina quais regiões da imagem possuem textura uniforme. Depois que as regiões são determinadas, os retângulos que as envolvem (*bounding boxes*) podem ser utilizados para construir uma estrutura de indexação tipo R-tree. Porém, da mesma forma que o histograma de cores, há o mesmo problema de ambigüidade, além do da dimensionalidade, para a indexação de dados de textura (informações em espaços de alta dimensionalidade), propiciando a "maldição da alta dimensionalidade".

#### 4.2.1.3. Atributo Forma

A recuperação de imagens baseada em forma é um dos problemas mais difíceis de serem tratados pelos sistemas de recuperação de imagens baseada em conteúdo [Aslandogan\_1999]. Isto se deve, principalmente, à difículdade de segmentar os objetos de interesse presentes na imagem, levando-se a recuperação por formas ser tipicamente limitada aos poucos objetos mais bem discriminados que estão presentes na imagem [Faloutsos\_1994].

A imagem a ser indexada deve ser pré-processada para possibilitar a busca e a determinação da borda de objetos que estão nela presentes. Os filtros ou algoritmos de pré-processamento dependem do domínio da aplicação das imagens em questão. Objetos tais como tumores cerebrais e lesões de pele demandam um conjunto específico de algoritmos que são diferentes dos utilizados para localizar objetos como aviões, carros, etc. Isso porque o primeiro domínio de imagens não pode valer-se de formas pré-definidas que possam auxiliar no processo de reconhecimento de informações morfológicas presentes na imagem. Por outro lado no segundo domínio de objetos, eles possuem formas baseadas em geometria, podendo-se valer da utilização de modelos descritos antecipadamente. O tratamento de imagens mais complexas

demanda, muitas vezes, também o tratamento e remoção de ruídos numa etapa de préprocessamento.

Após o objeto ser encontrado, sua borda precisa ser detectada utilizando algoritmos de perseguição de contornos, como os apresentados em [Duda\_2001] [Russ\_1995] [Gonzalez\_1993] [Mascarenhas\_1989]. O processo de detecção de bordas e formas fica mais difícil e comprometido em cenas complexas, nas quais há, além dos ruídos, oclusão parcial de objetos e sombras sobre regiões das imagens.

Os atributos de forma dos objetos presentes na imagem são também representados através de vetores de valores reais, embora aqui cada vetor possa ter uma dimensão. Nesse caso, um conjunto de vetores não tem uma dimensão característica, embora possa ser definida uma função de dissimilaridade. Dessa maneira, o conjunto de formas extraídas das imagens pode ser visto como elementos de um espaço métrico, e serem indexados dessa forma. Outra técnica é a de aproximar as formas encontradas por outras mais simples e fáceis de manusear. Por exemplo, a triangulação ou a aproximação por retângulos de contorno (*bounding boxes*) pode ser utilizada para representar formas irregulares. Além disso, tem-se a vantagem de que os requisitos de armazenagem são menores, e a comparação fica mais simples. Somente num último passo de comparação faz-se necessário para a manipulação do objeto irregular em si.

#### 4.3. Registros de Imagens

As técnicas de extração de características podem ser usadas como base para a operação de associação (*matching*) entre imagens. Se for especificado um limite de tolerância entre o *matching* entre imagens, pode-se dizer que elas são similares dentro desta tolerância.

Os tipos de variações presentes nas imagens determinarão quais técnicas de extração de características e registro de imagens serão utilizadas, como colocado na introdução deste capítulo. Pois estes tipos de variações dependem da modalidade (Raios-X, ToRM etc), da área considerada (coração, crânio, tórax etc) e talvez até mesmo da patologia considerada (tumores, fraturas, lesões etc) havendo a necessidade da definição de estratégias específicas para o processo de registro entre as imagens médicas.

Em Duncan e Ayache [Duncan\_2000] é apresentada uma visão sobre a evolução das técnicas ao longo das últimas duas décadas [Duncan\_2000]. Por aqui, pode-se verificar que o registro de imagem continua sendo uma importante e desafiante área de pesquisa, focalizando primeiramente o *matching* rígido entre dois modelos 3D reconstruídos a partir de dois conjuntos de imagens 2D, embora haja pesquisadores interessados em mapeamentos não rígidos. As abordagens são classificadas em:

- Aquelas baseadas em superficie/característica: onde alguma forma de informação espacial esparsa tem que ser identificada ou segmentada principalmente para o processo de registro.
- Aquelas baseadas em intensidade de *voxel*: nas quais os registros podem ser realizados diretamente a partir de dados em escala de tons de cinza.

Devido aos avanços tecnológicos em hardware e software, durante as últimas décadas foi possível expandir as técnicas bidimensionais de registro de imagens para tridimensionais, as quais até então eram muito caras, pois envolviam a manipulação de uma imensa quantidade de dados.

Alguns dos primeiros algoritmos de registro não rígido entre modelos 3D se preocupavam em minimizar a distância média Euclidiana entre duas superfícies segmentadas [Duncan\_2000]. Esta abordagem foi adaptada posteriormente para utilizar *matching* de distância "chamfer" [Duncan\_2000] e pode ser vista em softwares disponíveis comercialmente, tal como o ANALYZER [Duncan\_2000].

Estratégias alternativas e poderosas baseadas em características vêm sendo desenvolvidas. Para conjuntos de pontos delineados, surgiu a abordagem atraente do *iterative closest point* (ICP) [Duncan 2000].

Pode-se verificar também que há uma grande quantidade de soluções eficientes baseadas em histogramas [Brunelli 1998].

#### 4.4. Conclusões

Neste capítulo foram, resumidamente, apresentadas as principais técnicas de extração de características de imagens. Tais técnicas auxiliam na indexação e na recuperação de imagens baseadas em seu conteúdo, portanto são importantes para o presente trabalho.

Observe que, em vista da inexistência de um método de extração de características ótimo para cálculo de similaridade em imagens médicas, é nosso objetivo nesse trabalho utilizar vários algoritmos de extração de características de forma integrada, como num *pipeline*. Assim, o problema de identificação de uma imagem é atacado segundo diferentes abordagens. Atuando dessa maneira, pode-se eliminar paulatinamente um grande número de imagens do conjunto candidato para a resposta à consulta por similaridade. Assim, os métodos de extração que utilizam metodologias distintas, ao atuarem de forma integrada, devem propiciar uma melhor discriminação entre as imagens. Deve-se utilizar extratores de característica globais, que utilizem medidas estatísticas da imagem como, por exemplo, histograma. O histograma global não representa univocamente uma imagem, e sim uma classe de imagens, porém pode atuar como um primeiro filtro no processo de seleção, já que não é computacionalmente caro, e vai diminuir bastante o conjunto resposta para o próximo passo de busca.

# Capítulo 5

### O Sistema Mini-PACS proposto

#### 5.1. Introdução

Neste capítulo será apresentada a versão de um sistema PACS que está sendo desenvolvido em pesquisa conjunta entre o Grupo de Bases de Dados e Imagens do ICMC-USP (<a href="http://www.gbdi.icmc.sc.usp.br">http://www.gbdi.icmc.sc.usp.br</a>) e o Centro de Ciências de Imagens e Física Médica (<a href="http://www.cci.fmrp.usp.br">http://www.cci.fmrp.usp.br</a>) da FMRP-USP. Este sistema que está sendo denominado mini-PACS, incorpora facilidades para a recuperação de imagens baseada em seu conteúdo, respondendo dessa forma consultas por similaridade que são efetuadas sobre as imagens. Este sistema está sendo desenvolvido de forma modular, o que permite a atuação simultânea de diversas pesquisas bem como propicia um desenvolvimento mais rápido.

O acesso às imagens, efetuado por meio do mini-PACS, depende de um total controle e integração de vários subsistemas de computação principais, a saber: um Sistema de Processamento de Imagens - SPI, configurado para efetuar eficientemente as manipulações solicitadas sobre as imagens; um Sistema de Gerenciamento de Banco de Dados - SGBD, que permite o armazenamento e a recuperação das imagens utilizando, como dados de busca, informações pictóricas (idealmente de maneira distribuída); além de um Servidor de WWW (World Wide Web) - SW, o qual é utilizado como meio de comunicação e transferência de informações, em um ambiente descentralizado. A Figura 20 ilustra a infraestrutura do mini-PACS.

Os subsistemas atuam de maneira integrada, através de um conjunto de padrões centrado na manipulação de imagens a partir de suas características descritivas. A informação semântica das imagens solicitadas pelos usuários é manipulada internamente de maneira homogênea nos três subsistemas por meio de estruturas sintáticas apoiadas em estruturas de índices. O SPI provê

um conjunto de funções e operadores sobre imagens que extraiam as características das imagens, as quais são armazenadas na base pelo SGBD, além das próprias imagens, e são utilizadas para criar estruturas de índices que permitem agilizar a recuperação das imagens por similaridade Assim, o sistema baseia-se no conceito de busca de imagens através do conteúdo gráfico das imagens, não em textos associados a elas.

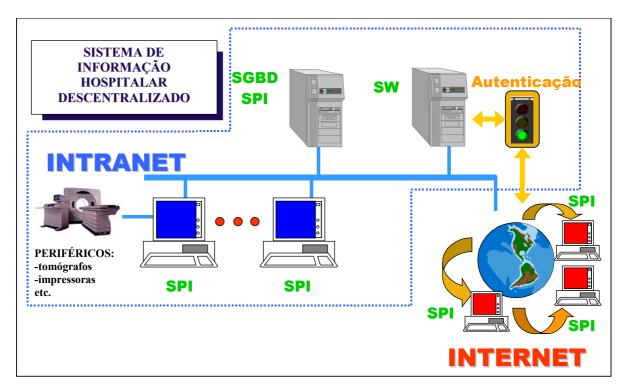


Figura 20 – Infraestrutura do Mini-PACS

O SW está integrado aos SPI e SGBD para disponibilizar seus recursos de maneira distribuída aos diversos usuários do mini-PACS seja via rede local utilizando aplicações cliente/servidor, ou via Intranet/Internet utilizando applets.

Parte do SPI é passada para as estações clientes para processamento de imagens localmente, evitando a sobrecarga no servidor de aplicações.

#### 5.2. Módulos Operacionais

Integrados a estes subsistemas há 3 módulos operacionais ou aplicações que permitem o armazenamento de dados e imagens, o gerenciamento de estruturas métricas e consultas a informações do banco de dados, convencionais ou por similaridade entre imagens:

<u>Sistema de Armazenamento de Dados e Imagens (SADI)</u>: através deste sistema os radiologistas fornecem dados de pacientes e exames que são armazenados no banco de dados.

<u>Sistema de Gerenciamento de Estruturas Métricas (SGEM)</u>: é responsável pela criação de estruturas de índices, baseando-se nas características extraídas de imagens, que são utilizadas para identificar imagens similares. Este sistema somente pode ser utilizado por especialistas em informática.

<u>Sistema de Recuperação de Imagens por Similaridade (SRIS)</u>: este sistema apresenta dois módulos integrados: cliente (*applet*) e servidor (*servlet*). O *applet* é executado remotamente de uma estação conectada a Intranet/Internet a fim de solicitar ao *servlet* informações convencionais tais como prontuários e exames de pacientes, mas também consultas por similaridade dada uma imagem de referência que pode estar armazenada ou não no banco de dados. As imagens consultadas podem ser manipuladas localmente através do SPI que é instalado no cliente ao se executar o *applet*. O acesso a este sistema depende de permissões de acesso e é controlado pelo SW.

Através desta arquitetura, apresentada na Figura 21, usuários podem recuperar imagens do SGBD utilizando um *applet* que deve ser instalado, via download, na estação cliente. O *applet* enviará as solicitações do usuário ao *servlet* no servidor SW, que disparará os mecanismos necessários para executar tarefas e fornecer os dados que o usuário deseja.

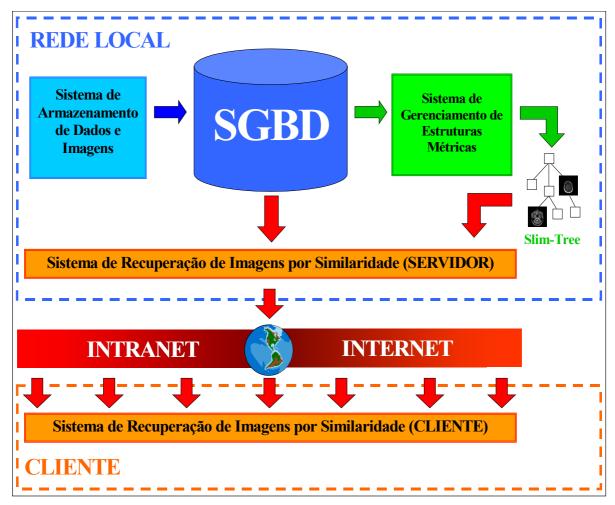


Figura 21 – Módulos Operacionais do Mini-PACS.

#### 5.2.1. Sistema de Armazenamento de Dados e Imagens (SADI)

O SADI é um sistema utilizado exclusivamente por especialistas para a alimentação do banco de dados com informações incluindo exames de pacientes e diagnósticos (laudos). As informações alfanuméricas são armazenadas diretamente no banco de dados, enquanto que as imagens referentes a exames médicos, ao serem armazenadas no banco de dados, passam pelo SPI a fim de identificar descritores básicos de características. Assim, uma imagem e seus descritores são armazenados no banco de dados associados às informações alfanuméricas correlacionadas. Esta operação pode ser executada sobre um único exame ou sobre um conjunto de exames.

As imagens armazenadas no banco de dados podem ser associadas a informações de um Sistema de Informação Hospitalar (HIS) através de um identificador único de paciente ou até mesmo do laudo a ser preenchido pelo médico após a aquisição das imagens.

Na Figura 22 é mostrada a operação do SADI.

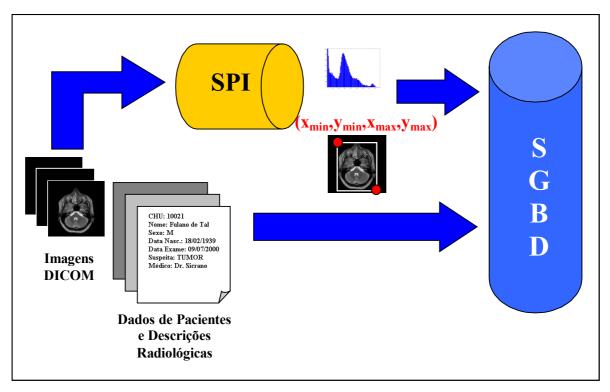


Figura 22 – Sistema de Armazenamento de Dados e Imagens.

#### 5.2.2. Sistema de Gerenciamento de Estruturas Métricas (SGEM)

O SGEM também é um sistema utilizado exclusivamente por especialistas para se fazer a criação e manutenção da estrutura de índices que possibilitará a busca rápida por imagens similares armazenadas no banco de dados. Para cada tipo de extrator de características de imagens é criada uma Slim-tree. A estrutura de índices utiliza os valores gerados pelo banco de dados, chaves primárias, para identificar cada imagem armazenada. Além da chave primária, os nós da Slim-tree contêm descritores de imagens, obtidos do banco de dados e/ou através do SPI, referentes ao tipo do extrator correspondente àquela estrutura. As características não extraídas previamente pelo SADI e processadas pelo SPI nesta etapa podem ser armazenadas no banco de

dados dependendo do especialista. Se o processo de extração for muito demorado é aconselhável que as características extraídas sejam armazenadas no banco de dados para que, em um possível processo de recriação de uma Slim-Tree, o tempo de processamento diminua.

Este sistema pode ser executado de duas formas: logo que uma imagem é inserida no banco pelo SADI o SGEM a inclui nas estruturas índices; ou as imagens são processadas em *batch* periodicamente para que as estruturas se mantenham atualizadas.

Caso um novo extrator seja incorporado ao SPI, as imagens já existentes no banco terão que ser processadas para que se crie a Slim-Tree relacionada ao novo extrator.

Na Figura 23 encontra-se um esquema operacional do SGEM.

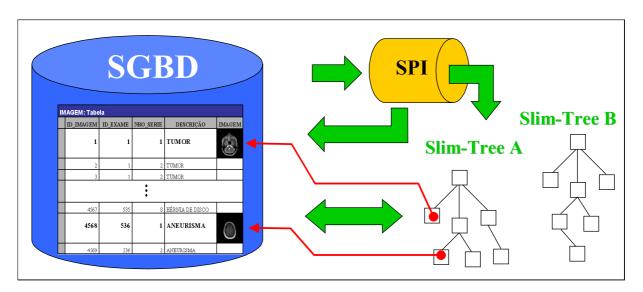


Figura 23 – Sistema de Gerenciamento de Estruturas Métricas.

#### 5.2.3. Sistema de Recuperação de Imagens por Similaridade (SRIS)

O SRIS apresenta dois módulos: o servidor (*servlet*) e o cliente (*applet*). O SRISservlet é responsável pela interpretação das consultas solicitadas pelo usuário através do SRISapplet. O usuário pode fazer qualquer tipo de consulta convencional ao banco de dados utilizando o SRISapplet e também as consultas por similaridade. Neste último caso, o SRISapplet envia ao SRISservlet uma solicitação a qualquer imagem que seja semelhante a uma dada imagem de referência, seja dentro de um intervalo (raio de cobertura), que indica os limites aceitáveis do

grau de similaridade entre as imagens do banco e a de referência (*range-query*), ou seja fornecendo um número inteiro k, que indica as k imagens mais parecidas com a de referência (k-NN). Ao receber a solicitação, o SRISservlet consulta a estrutura de índices previamente criada para determinar quais são os identificadores das imagens mais semelhantes à imagem de referência segundo os parâmetros passados. A partir destes identificadores de imagens o SRISservlet recupera as imagens correspondentes no banco de dados a fim de enviá-las, juntamente com suas informações correspondentes, ao SRISapplet

É importante notar que uma parte do SPI é passada para o cliente junto com o *applet* para que o usuário possa manipular as imagens consultadas e também para que se extraia características de imagens a serem utilizadas como parâmetros em consultas por similaridade.

A Figura 24 ilustra o processo de consulta por *range query* efetuado pelo usuário no cliente. No passo (1) o SRISapplet traduz a consulta visual especificada pelo usuário extraindo as características da imagem de referência, caso esta imagem não faça parte do banco de dados, e envia ao SRISservlet um comando SQL contendo o vetor de características extraído e os demais parâmetros de consulta, tais como tipo de extrator, o tipo de consulta (*range query* ou *k-NN*) etc. No passo (2) o SRISservlet processa o comando SQL identificando qual a estrutura de índice a ser pesquisada e qual o tipo de algoritmo de busca que se deseja executar. A seguir, o SRISservlet executa o algoritmo de busca correspondente e, a partir das chaves retornadas conforme ilustrado no passo (3), identifica quais as imagens armazenadas no banco de dados serão retornadas ao cliente, como ilustrado nos passos (4) e (5). No passo (6) o SRISservlet envia as imagens recuperadas e os parâmetros de resposta (no exemplo, grau de similaridade, pois o tipo de consulta efetuada foi *range query*) ao SRISapplet que se encarregará de exibir os resultados.

Nas Figuras 25 e 26 são ilustradas as telas do protótipo para consulta a imagens por similaridade. Na Figura 25 é apresentada a tela utilizada para que o usuário identifique os parâmetros de consulta. Na Figura 26 é apresentada a tela resultante da consulta por similaridade.

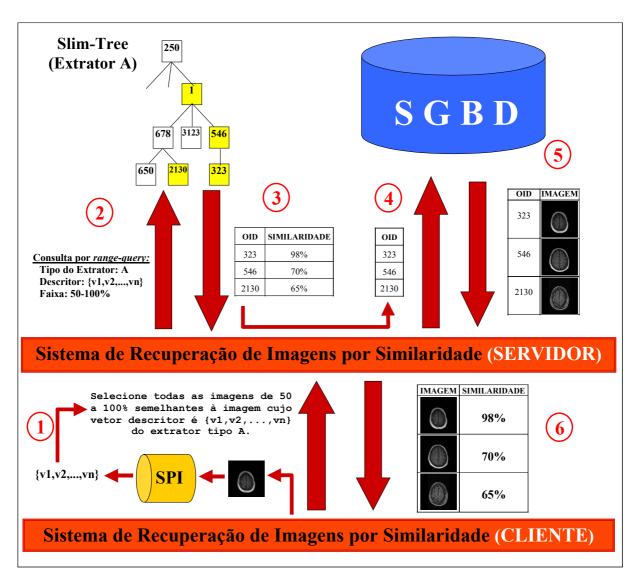
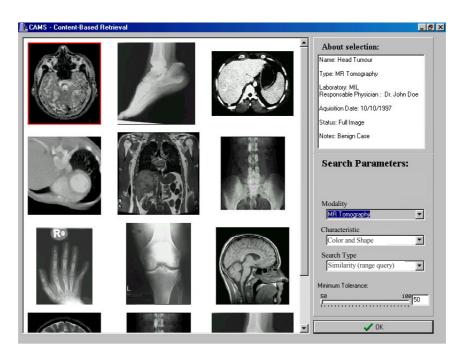
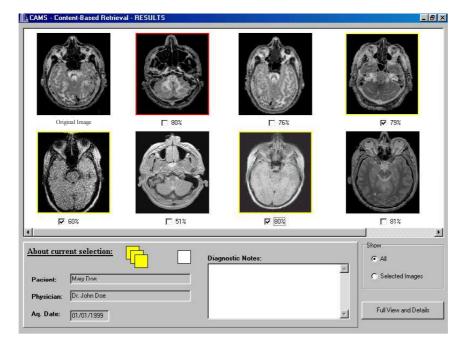


Figura 24 – Sistema de Recuperação de Imagens por Similaridade.



**Figura 25 -** Tela para a definição de consulta a imagens médicas por similaridade. Ao confirmar os parâmetros da consulta, os resultados aparecem como mostrado na Figura 26.



**Figura 26 -** Tela de resultados de consulta baseada em conteúdo de imagens. O botão *Full View and Details* leva a um visualizador genérico contendo funções do SPI.

#### 5.2.4 Categorias de Usuários

O sistema mini-PACS que está sendo descrito neste trabalho pressupõe a existência de três categorias de usuários: os desenvolvedores, o grupo de produção e os clientes. Os desenvolvedores são pessoas responsáveis pela análise e implementação do software de aplicação dentro de um dado domínio. Eles também desenvolvem algoritmos de processamento de imagens, definindo os métodos de processamento de imagens e seus parâmetros associados. Por exemplo, esses usuários estariam encarregados de desenvolver um aplicativo para ensino da técnica de laudo aos residentes em radiologia no hospital.

O grupo de produção consiste nos usuários que configurarão a aplicação e o banco de dados de imagens, tratando-as para uso dos clientes. Eles definirão os métodos de processamento de imagens significativos para uso como um sumarizador para cada atributo de objeto de um tipo de dado imagem definido no escopo da aplicação. Este grupo é dividido em:

- Gerente de Banco de Dados e Imagens: que trabalha em colaboração com os desenvolvedores e especialistas médicos para definir o escopo da aplicação
- Gerentes de aplicação: que trabalham em colaboração com o gerente de banco de dados e imagens para planejar a recuperação e a manipulação das imagens através de macros predefinidas e para conhecer as necessidades de consultas pré-definidas.

Os clientes são os médicos e usuários finais, as pessoas que usam a aplicação e os recursos para recuperação e manipulação de imagens. Permitindo que as consultas sejam executadas através de macros predefinidas, o gerente de aplicação pode definir as estruturas de índice que devem ser mantidas pelo gerente de banco de dados e imagens que , por outro lado, determina quais sumarizadores (métodos de processamento de imagens) os desenvolvedores devem fornecer.

Quando um usuário solicita consultas ao banco de dados através da aplicação, as macros pré-definidas selecionam um conjunto de imagens prospectivas, restringindo o conjunto de imagens que devem efetivamente ser processadas, assim reduzindo drasticamente o tempo de processamento necessário para a consulta. Esta técnica induz a construção de sistemas expansíveis, onde novos algoritmos de processamento de imagens podem ser dinamicamente incorporados em uma aplicação através do gerente de banco de dados e imagens, sem modificações no código. Isto pode ser feito conectando o novo algoritmo ao tipo de domínio. Os

resultados obtidos com a execução do algoritmo criariam uma nova estrutura de índices para cada um de seus parâmetros, cuja navegação pode ser incorporada às macros geradas pelo gerente de aplicação.

Quando o sistema é inicializado, o usuário deve se identificar através de modos usuais, fornecendo seu nome e senha. O sistema recupera a categoria do usuário e usa esta informação para configurar o menu principal. A interface esconde as opções que definem os objetos do escopo dos clientes. As seções seguintes apresentam as opções que os desenvolvedores e usuários do grupo de produção podem usar para definir o escopo da aplicação, cujos formulários permitem a criação, modificação e exclusão de elementos do escopo.

#### 5.3. Extração de Características: Histogramas Métricos

As imagens são representadas como um conjunto de elementos (pixels) colocados em uma grade regular. Os valores associados a cada pixel são aqueles obtidos do processo de quantização e correspondem à luminosidade associada à imagem. Assim, formalmente uma imagem pode ser representada pela seguinte notação:

**Definição 5.3.1.** Uma **imagem** A é uma função definida sobre uma faixa bidimensional  $G=[0,x_{\theta}]$  x  $[0,y_{\theta}]$  tomando valores do conjunto de possíveis de luminosidade  $V=[0,v_{\theta}]$ . Isto é,  $A=\{(x,y,v(x,y))/(x,y)\in G \text{ e } v\in V\}$ .

Um histograma de imagem é composto por um número de *bins* que depende da resolução de quantização da imagem. Geralmente este valor é dado em exponência de 2, isto é: 64, 128, 256 etc. Em imagens médicas, dentro de um mesmo domínio, geralmente este valor é fixo. Formalmente um histograma pode ser explicitado com a seguinte definição:

**Definição 5.3.2.** O **histograma**  $H_A(z)$  de uma imagem A fornece a frequência de cada valor de luminosidade z na imagem. O histograma de uma imagem com t níveis de luminosidade é representado por um vetor com t elementos, chamados bins.

**Definição 5.3.3.** O **histograma normalizado**  $NH_A(z)$  de uma imagem A fornece a frequência em porcentagem de cada valor de luminosidade z na imagem. O histograma normalizado de uma imagem com t níveis de luminosidade é também representado por um vetor com t elementos.

Obter o histograma normalizado de imagens não é uma operação custosa em termos computacionais, pois, para uma imagem com n pixels, o seu custo continua sendo O(n) com uma constante igual a 2. Deve-se observar que o histograma normalizado é invariante em relação às transformações geométricas (escala, rotação e translação). Seria interessante conseguir um histograma que fosse também invariante com relação às transformações lineares de brilho. Tal fato será obtido com a proposta deste trabalho apresentada na seção 5.3.2 a qual denominaram-se histogramas métricos.

Os histogramas normalizados permitem comparações de imagens de qualquer tamanho, assim transformações geométricas realizadas sobre as imagens fontes fornecerão o mesmo histograma. A Figura 27 apresenta uma imagem obtida por ressonância magnética e seu histograma normalizado associado. Esta imagem tem uma resolução espacial de 512x512 pixels exibidos em 256 níveis de luminosidade, assim seu histograma contém 256 *bins*. A indexação de histogramas como este requer a indexação de vetores com 256 elementos ou, em termos de estruturas de indexação, dimensões.

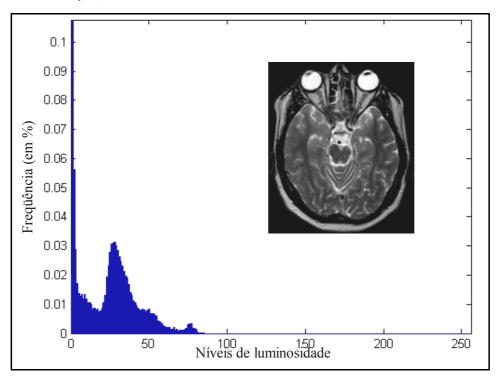


Figura 27 – Uma imagem e seu histograma normalizado.

#### 5.3.1. Área de Interesse (Cropping)

O primeiro passo necessário para permitir a extração de características das imagens foi eliminar o fundo e trabalhar com a imagem em si. Isso é importante, pois permite o tratamento de operação de translação e escala sobre a imagem de forma natural. Note-se que a maior parte das imagens deste projeto, que foram obtidas por tomografia, possui uma característica interessante, que é a imagem estar rodeada pelo fundo.

Na Figura 28 é mostrada uma tela do aplicativo utilizado para capturar automaticamente o menor retângulo que armazena a imagem. O valor de tolerância varia de imagem para imagem dependendo dos parâmetros de aquisição utilizados. O algoritmo utilizado é ilustrado na Figura 29.

Eliminando-se o fundo da imagem podemos também visualizar mais claramente os histogramas, pois com o fundo a densidade de pixels para o tom preto é muito discrepante em relação às densidades para os demais tons.

É comum que os níveis de luminosidade de um histograma sejam semelhantes a seus níveis mais próximos, tal que a forma do

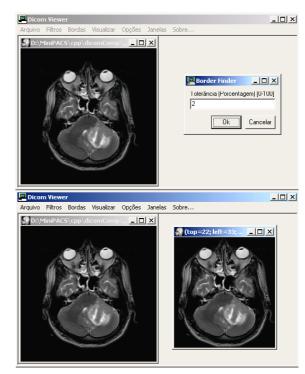
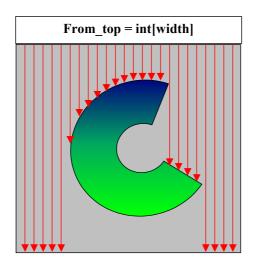
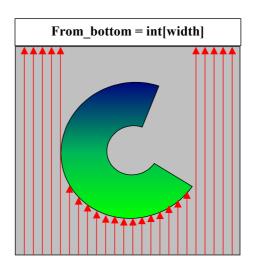


Figura 28 - Resultado do cropping automático.

histograma pode ser mantida utilizando uma curva de aproximação. Um modo de representar esta aproximação é por um conjunto de segmentos de retas. Histogramas de diferentes imagens podem ser aproximados utilizando quantidades diferentes de segmentos de reta, tal que a aproximação seja otimizada para cada histograma. Assim, estas aproximações não possuem um número pré-definido de retas, pois depende de cada imagem. Dessa forma, o domínio de representação para as aproximações não tem um número de dimensões definido. Isso significa que não podem ser vistos como pontos em algum espaço dimensional. Na realidade, as aproximações dos histogramas das imagens são, dessa forma, adimensionais.

Para obtenção da curva (linear por partes) de aproximação de um dado histograma, foi proposto um algoritmo para identificação de pontos de controle baseado em gradiente. Em linhas gerais, na primeira parte do algoritmo definem-se alguns pontos de controle identificando os pontos de máximo e mínimo da função que define o histograma. Na segunda parte do algoritmo, seleciona-se alguns pontos intermediários, como pontos de controle, baseando-se na distância entre pontos adjacentes de máximos e mínimos. O histograma obtido a partir deste processo foi denominado *histograma métrico*.





```
Find top
    Min(from_top) = from_top[top]

Find bottom
    Min(from_bottom) = from_bottom[bottom]

Find left
    left = 0
    While (from_top[left] > from_bottom[left]) and (left < width) do left = left + 1 end while

Find right
    Right = width - 1
    While (from_top[right] > from_bottom[right]) and (right >= 0) do right = right - 1 end while
```

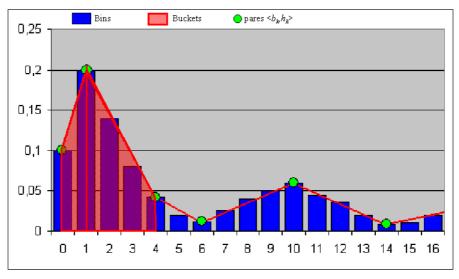
Figura 29 - Algoritmo para eliminação do fundo da imagem (cropping).

#### 5.3.2. Histogramas Métricos

Formalmente um histograma métrico é definido por:

**Definição 5.4.4.** Um **histograma métrico**  $MH_A(z)$  de uma imagem A é definido como  $M_H(A) = \{N_A, \langle b_k, h_k \rangle | 0 \le k \le N_A\}$ , que é um conjunto de  $N_A$  recipientes (*buckets*) formados por pares  $\langle b_k, h_k \rangle$  consecutivos, onde  $b_k$  indica largura e  $h_k$  altura.

Um histograma normalizado é composto por um número de *bins*. Este número depende da resolução de luminosidade da imagem, assim é um número fixo. Em um histograma métrico, o equivalente ao *bin* do histograma é chamado um *bucket*. Cada *bucket* corresponde a um segmento de linha aproximado e, consequentemente, representa um subconjunto de *bins* do histograma original. Os *buckets* não precisam ser regularmente espaçados. O número  $N_A$  de *buckets* em um histograma métrico depende do erro de aceitação no processo de aproximação da curva linear por partes sobre o histograma. Cada *bucket* i é definido por dois pares consecutivos  $\langle b_{i-1}, h_{i-1} \rangle$  e  $\langle b_i, h_i \rangle$ , para  $1 \le i \le N_A$ , onde  $\langle b_{i-1}, h_{i-1} \rangle$  é o *bin* mais à esquerda do histograma original representado no *bucket* i. A Figura 30 representa graficamente os *bins* e *buckets* de um histograma.



**Figura 30** – Histograma normalizado com pontos de controle  $\langle b_k, h_k \rangle$  que definem os *buckets* correspondentes ao seu histograma métrico.

Para obter um histograma métrico que melhor aproxime o histograma original é preciso determinar o menor conjunto de segmentos de retas (*buckets*) da função que o representa. Como cada *bucket* é composto por pares  $\langle b_k, h_k \rangle$  consecutivos, que correspondem a *bins* do histograma original, primeiramente, é necessário encontrar o subconjunto mais representativo de *bins* candidatos a pares  $\langle b_k, h_k \rangle$ , os quais denominamos pontos de controle.

Verificou-se que a grande maioria dos histogramas originais apresentava uma concentração de pixels (freqüências de luminosidade maiores que 0) para um subconjunto de *bins* consecutivos, conforme mostra a Figura 31. Desta forma, o primeiro passo para definir os pontos de controle é identificar a janela (subconjunto de *bins* consecutivos) com a maior concentração de pixels do histograma original. Uma vez identificada a janela de interesse, procura-se pelos pontos de máximo e mínimo, os quais se tornarão pontos de controle. Como visto na Figura 31, os pontos de controle P<sub>1</sub> e P<sub>2</sub> definem esta janela.

Para definir o ponto  $P_1$ , efetuase uma varredura pelos *bins* do histograma original, em ordem <u>crescente</u> de níveis de cinza. Se o primeiro *bin*, tiver uma freqüência de luminosidade maior que 0, o ponto  $P_1$  corresponderá ao par  $\langle b_0, h_0 \rangle$ , onde  $b_0=0$  e  $h_0 \rangle 0$ , como ocorre na Figura 31. Caso contrário,  $P_1$  é definido pelo último *bin* da varredura, cuja freqüência de luminosidade seja igual a 0. Neste caso,  $P_1$  corresponderá ao par  $\langle b_0, h_0 \rangle$ , onde  $b_0 \rangle 0$  e  $h_0=0$ .

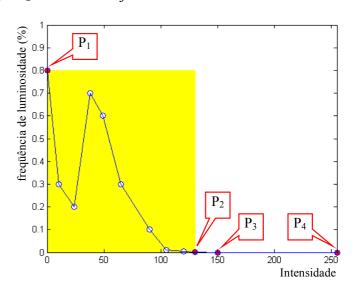


Figura 31 – Identificação da janela de interesse.

O ponto P<sub>4</sub> é definido pelo último *bin* do histograma, o qual, na Figura 31, é representado pelo *bin* cuja intensidade é 255.

Para identificar os pontos  $P_2$  e  $P_3$  é feita uma varredura, em ordem <u>decrescente</u> de intensidade, a partir de  $P_4$ , até que se encontre um *bin* cuja freqüência de luminosidade seja maior que zero. Desta forma,  $P_3$  é definido pelo *bin* anterior ao encontrado. A varredura continua a partir de  $P_3$ , a fim de encontrar o próximo *bin* cuja freqüência de luminosidade seja maior ou

igual a *Dy*, um valor de limiar definido arbitrariamente e relacionado à frequência de luminosidade (Ex: 0,0005). Desta forma, P<sub>2</sub> é definido pelo *bin* anterior ao encontrado.

Quando o valor da frequência de luminosidade do <u>penúltimo</u> *bin* do histograma for maior que 0 e menor que Dy, os pontos  $P_3$  e  $P_4$  coincidirão, assim estes pontos serão representados, unicamente, pelo último par  $\langle b_{NA}, h_{NA} \rangle$  do histograma métrico. O mesmo acontece quando a frequência de luminosidade do <u>penúltimo</u> *bin* do histograma for maior ou igual a Dy, pois  $P_2$  também coincidirá com  $P_3$  e  $P_4$ .

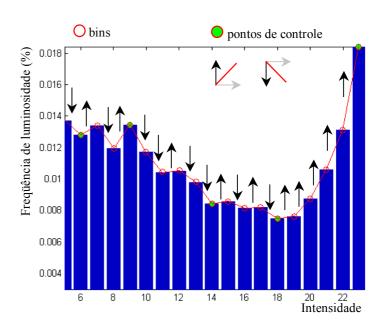
Os pontos  $P_1$  e  $P_2$  definem os limites da janela de interesse que será processada para identificar os pontos de máximos e mínimos representados por pares <u>distintos</u>  $< b_k, h_k >$ .

Definiu-se que o número mínimo de *bins*, compreendidos entre  $P_1$  e  $P_2$ , para as próximas etapas do algoritmo, seja 28. Caso o número de *bins* for menor que 28, estes mais os 4 previamente selecionados ( $P_1$  a  $P_4$ ) serão considerados os pares  $\langle b_k, h_k \rangle$  que constituirão os *buckets* do histograma métrico correspondente ao histograma original, não havendo necessidade de se executar as próximas etapas do algoritmo.

Na segunda etapa do algoritmo, ao percorrer os *bins* da janela de interesse, se houver um decremento entre a frequência de luminosidade de um *bin* em relação ao *bin* anterior, então a

reta, definida por ambos, tem inclinação negativa (*m*=–1). Se houve um incremento entre a frequência de luminosidade de um *bin* em relação ao *bin* anterior, então a reta, definida por ambos, tem inclinação positiva (*m*=+1). A Figura 32 ilustra este passo.

Como a oscilação entre as freqüências de luminosidade dos *bins* é considerável, ou seja é comum encontrar diferentes inclinações de reta para pares consecutivos de *bins*, então, para amenizar esta



**Figura 32** –Sentido entre pares de *bins* e os pontos de controle encontrados pelo algoritmo.

discrepância, calcula-se a média aritmética das inclinações dadas por *Nb* pares de *bins* consecutivos. Se a média for positiva (*média*=+1), define-se o *bin* com menor freqüência de luminosidade, dentre os *Nb* pares de *bins*, como ponto de mínimo. Caso contrário, se a média for negativa (*média*=-1), define-se o *bin* com maior freqüência de luminosidade, dentre os *Nb* pares de *bins*, como ponto de máximo. Encontrado o novo ponto de controle, continua-se o processo para os próximos *Nb* pares de *bins*.

O algoritmo tenta encontrar pontos intercalados de máximo e mínimo. Entretanto, o algoritmo pode identificar dois pontos de máximo ou de mínimo consecutivos. Assim, se o ponto de controle  $\langle b_{i-1}, h_{i-1} \rangle$  for ponto de [máximo/mínimo] e o algoritmo encontrou um novo ponto de [mínimo/máximo]  $\langle b_i, h_i \rangle$ , candidato a ponto de controle, mas, entre estes dois pontos, também foi encontrado um ponto de [máximo/mínimo]  $\langle b_j, h_j \rangle$ , onde  $b_{i-1} \langle b_j \langle b_i, o$  algoritmo determina que:

```
se (b_j-b_{i-l}) < Dx, onde Dx é um valor de limiar definido arbitrariamente e relacionado à intensidade, e |h_j-h_{i-l}| > Dy, então < b_j, h_j >  substitui < b_{i-l}, h_{i-l} >  como ponto de controle; senão  \mathbf{se} \ (b_j-b_{i-l}) > Dx \ \mathbf{e} \ |h_j-h_{i-l}| > Dy, \ \mathbf{então}  < b_j, h_j >  torna-se um novo ponto de controle; senão < b_j, h_j >  é desconsiderado como ponto de controle.
```

Após esta verificação, independente do resultado,  $\langle b_i, h_i \rangle$  torna-se um novo ponto de controle e a busca, pelos próximos pontos de controle, continua até o *bin* anterior a P<sub>2</sub>. Ao final do processo, os pontos de controle P<sub>2</sub>, P<sub>3</sub> e P<sub>4</sub> são acrescentados ao conjunto de pares  $\langle b_k, h_k \rangle$ .

Na terceira etapa do algoritmo, caso a quantidade de *buckets*,  $N_A$ , encontrados na etapa anterior for inferior a 31, então seleciona-se mais (31- $N_A$ ) *bins* como pontos de controle, distribuídos uniformemente entre os primeiros pares  $\langle b_k, h_k \rangle$ , para completar os 31 *buckets*. Isto é o que ocorre com a maioria dos histogramas métricos gerados nos testes.

Em alguns casos, a quantidade *buckets* extrapola 31, pois o resultado do algoritmo depende da distribuição do histograma. Quanto mais homogêneo for o histograma, maior a quantidade de *buckets* são definidos. Assim,  $1 \le N_A < N$ úmero-de-Intensidades(A).

Na Figura 33 é ilustrado um histograma e os pontos de controle que definem os *buckets* de seu histograma métrico correspondente. Os parâmetros utilizados para a geração do histograma métrico foram: Dx=2, Dy=0,0005 e Nb=3.

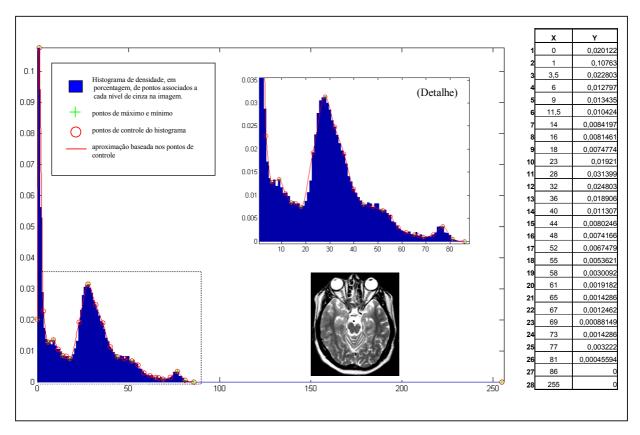


Figura 33 - Histograma da imagem apresentada na figura anterior, com seus pontos de controle.

#### 5.4. Extenção da Slim-tree para Armazenar Imagens

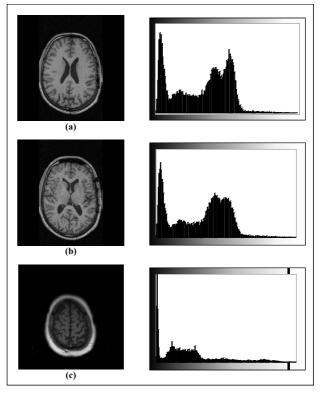
Já existiam aplicações que utilizavam a Slim-tree para indexação e busca de palavras, além de dados espaciais e geográficos. Para suportar imagens e possibilitar a indexação dos vetores de características baseados em histogramas métricos, foi necessário desenvolver uma nova função de distância métrica. Isto foi necessário porque a distância Euclidiana ou qualquer métrica Lp não podiam mais ser utilizadas pois cada vetor de características (histograma métrico) possui um número de elementos particular.

#### 5.4.1. Função de Distância Proposta: por Diferença de Área

Histogramas de imagens semelhantes possuem distribuições parecidas como pode ser

visto na Figura 34. Normalmente, o cálculo de dissimilaridade entre histogramas é dado pela somatória da diferença entre os *bins* de dois histogramas. Com relação aos histogramas métricos a questão é como compará-los, uma vez que o número de *buckets* e a distribuição dos *buckets* de diferentes histogramas são variáveis.

Os histogramas métricos, que possuem dimensionalidade variável, não permitem o cálculo de distância utilizando técnicas usuais como a euclidiana ou qualquer distância Lp [Brunelli\_1998] [Wilson\_1997], pois não é possível calcular a subtração dos pares de elementos dos vetores dos histogramas métricos para todos os pares de elementos. Por exemplo, como calcular a distância euclidiana entre um



vetores dos histogramas métricos para todos os pares de elementos. Por exemplo, como histogramas apresentam a densidade de pixels, em porcentagem, para 256 níveis de cinza das imagens.

histograma com 30 *bins* de outro com 20? Isso porque, considerando os histogramas tradicionais como um conjunto de pares cartesianos, os valores em x serão sempre os mesmos para todos os histogramas, o que não ocorre com os histogramas métricos.

Portanto, para fazer o cálculo da distância entre histogramas métricos foi desenvolvido um novo algoritmo baseado no cálculo da diferença entre histogramas considerando que cada um deles ocupa uma área caracterizada pela distribuição de pixels e que a diferença entre estas áreas indica quanto dissimilares são os histogramas.

Utilizando esta concepção pode-se concluir que quando dois histogramas similares são comparados a diferença entre suas áreas de distribuição é pequena. Formalmente a distância por diferença de área é dada por:

**Definição 5.4.1.** A distância **DM()** entre dois histogramas métricos  $M_H(A)$  e  $M_H(B)$  é dada pela área não sobreposta entre as duas curvas que representam os histogramas métricos, isto é:

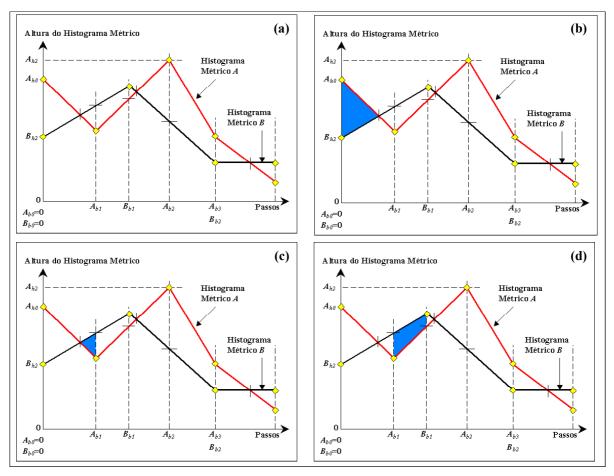
$$DM(M_H(A), M_H(B)) = \int_{0}^{bm} |M_H(A, x) - M_H(B, x)| dx$$

onde bm=máx( $b_{NA-1}$ ,  $b_{NB-1}$ ) e  $M_H$ (Imagem, x) é a função contínua que representa o histograma métrico.

A Tabela 5 descreve o algoritmo desenvolvido para calcular esta função de distância, e a Figura 35 fornece um exemplo de como calcular a distância entre dois histogramas métricos. Para simplificar a notação, definimos a indicação de largura  $b_i$  do histograma métrico da imagem A como  $A_{bi}$ , e a indicação de altura  $h_i$  do histograma métrico da imagem A como  $A_{hi}$ .

Tabela 5 - Algoritmo para cálculo de distância entre histogramas métricos.

```
A função distância DM: calcular a distância entre dois histogramas métricos.
Input: os dois histogramas métricos M_H(A) e M_H(B)
Output: a distância entre M_H(A) e M_H(B).
    dist=0, s=0, a=0, h_a=A_{b0}, h_b=B_{b0}, i=1 e j=1
    enquanto há mais passos para comparar
          se o bucket A_{bi} < B_{bj}, então
                     calcular o valor de B na posição A_{bi} enquanto y=(A_{bi}, y_2)
                                bm=A_{bi}, base=bm-s, e y_1=A_{bi}
                     incrementar i
          senão
                     calcular o valor de A na posição B_{bj} enquanto y=(B_{bj}, y_1)
                                bm=B_{bj}, base=bm-s, e y_2=B_{hj}
                     incrementar j
          se a linha ((a, h_a), (a, h_b)) intersecta a linha ((b_m, y_l), (b_m, y_2)), então
                     calcular a intesecção w=(w_b, w_h)
                     calcular \dot{a}rea_1 = \left| (w_b - a) * \frac{h_a - h_b}{2} \right| e \dot{a}rea_2 = \left| (b_m - w_b) * \frac{y_1 - y_2}{2} \right|
                     adicionar area_1 + area_2 a dist
          senão
                     calcular \acute{a}rea_1 = base * \frac{h_a + y_1}{2} e \acute{a}rea_2 = base * \frac{h_b + y_2}{2}
                     adicionar | área<sub>1</sub> - área<sub>2</sub>| a dist
          h_a=y_1 e h_b=y_2
          a=base+a
    retorna dist
```



**Figura 35** – Distância entre dois histogramas métricos calculando a área entre eles. (a) Dois histogramas métricos A e B, e os pontos usados para especificar os passos do algoritmo. (b) Primeiro passo do algoritmo, exemplificando quando os dois HM se intesectam. (c) Segundo passo do algoritmo. (d) Terceiro passo do algoritmo.

Na Figura 35(a), os dois histogramas são sobrepostos, e são mostrados os pontos de intersecção e aqueles que limitam os *buckets*. Em (b) até (d) é mostrado como tais pontos são utilizados para calcular a área dentro de cada região (nos passos do algoritmo da Tabela 5). Note que o número de passos é maior que ou igual ao número de *buckets* do histograma com mais *buckets*. Isto é devido ao fato de que como o comprimento dos *buckets* é variável, em algumas ocasiões eles devem ser divididos a fim de obter a área entre os dois histogramas considerados. Isto é exemplificado em (b). Em (c) e (d) são mostrados os próximos dois passos realizados para obter DM(), ajudando a compreender como o algoritmo funciona. Quando um dos histogramas métricos termina antes do outro, isto é, quando  $N_A < N_B$ , o cálculo da distância também pára.

A distância DM() atende aos três requisitos de ser uma métrica. A distância DM() é calculada como uma somatória de todos os passos (um *bucket* ou uma divisão dele, se houver intersecção entre os histogramas métricos). Assim, por definição, a distância DM() é claramente simétrica e não negativa. Sem perda de generalidade pode-se observar em cada passo que a desigualdade triangular é atendida. Uma vez que a distância DM() é a somatória da diferença de áreas não sobrepostas entre pares de paralelogramas, a desigualdade triangular também encontrase em cada passo do cálculo de DM(). Assim, por indução dos passos, a desigualdade triangular também é comtemplada em todo o cálculo de distância. Provando formalmente:

1. <u>Simetria</u>: Sejam dois histogramas métricos  $M_H(A)$  e  $M_H(B)$ . A distância:

$$\begin{split} DM\big(M_H\big(A\big), M_H\big(B\big)\big) &= \int\limits_x \big|M_H\big(Ax\big) - M_H\big(Bx\big)\big| dx \\ &= \int\limits_x \big|M_H\big(Bx\big) - M_H\big(Ax\big)\big| dx \qquad (pois\ a\ differença\ \'e\ em\ m\'odulo) \\ &= DM\big(M_H\big(B\big), M_H\big(A\big)\big) \qquad c.q.d. \end{split}$$

2. <u>Não Negatividade</u>: A distância *DM* por definição é um somatório de valores absolutos, logo não pode resultar em valor negativo.

$$MD(M_H(A), M_H(A)) = \int_x |M_H(Ax) - M_H(Ax)| dx$$
  
= 0 (será zero quando for efetuada sobre o mesmo histograma) c.q.d.

3. Desigualdade Triangular: Seja  $M_H(C)$  um terceiro histograma métrico

$$MD(M_{H}(A), M_{H}(C)) = \int_{x} |M_{H}(Ax) - M_{H}(Cx)| dx$$

$$= \int_{x} |M_{H}(Ax) - M_{H}(Cx)| dx + \int_{x} |M_{H}(Bx) - M_{H}(Bx)| dx$$

$$= \int_{x} |M_{H}(Ax) - M_{H}(Bx)| + M_{H}(Bx) - M_{H}(Cx)| dx$$

$$= \int_{x} |M_{H}(Ax) - M_{H}(Bx)| dx + \int_{x} |M_{H}(Bx) - M_{H}(Cx)| dx$$

$$= \int_{x} |M_{H}(Ax) - M_{H}(Bx)| dx + \int_{x} |M_{H}(Bx) - M_{H}(Cx)| dx$$

$$= DM(M_{H}(A), M_{H}(B)) + DM(M_{H}(B), M_{H}(C)) \quad (bastava ser menor ou igual) \quad c.q.d.$$

Os experimentos, como serão vistos no próximo capítulo, mostram que o número de passos no histograma métrico é muito menor que o número de *bins* dos histogramas normalizados, reduzindo de 256 *bins* para 20-32 passos nos histogramas métricos.

É importante recordar que os histogramas métricos são obtidos de histogramas normalizados. Assim, mantêm-se as seguintes propriedades:

**Propriedade 5.4.1.** Uma imagem original e a mesma escalada, transladada ou rotacionada terão o mesmo histograma métrico.

**Propriedade 5.4.2.** Os histogramas métricos são curvas no espaço, assim os histogramas métricos podem ser ajustados no seu começo e fim. Portanto, os histogramas métricos também são invariantes ao brilho da imagem.

Estas duas propriedades realçam que algumas restrições no uso de histogramas para recuperação de imagens são superadas pelo uso de histogramas métricos. Isto é, eles são invariantes a transformações geométricas, incluindo escala, e a brilho.

#### 5.5. Conclusões

Este capítulo apresentou a estrutura geral do sistema mini-PACS que está sendo desenvolvido de forma colaborativa entre o GBDI do ICMC-USP e o CCIFM da FMRP-USP. Para permitir a utilização de histogramas como uma primeira técnica de extração de características de imagens, mas que permitisse a comparação de imagens independente de escala, e de variações lineares de brilho, foi proposta a utilização de histogramas métricos. Tais histogramas suplantam a deficiência de histogramas tradicionais que não permitem a comparação de imagens de tamanhos diferentes e que foram obtidas em níveis de quantização distintos. Além, disso para permitir que histogramas métricos fossem indexados na Slim-tree foi desenvolvida uma função de distância métrica baseada na diferença entre as áreas das curvas dos dois histogramas.

# Capítulo 6

### Resultados

#### 6.1. Introdução

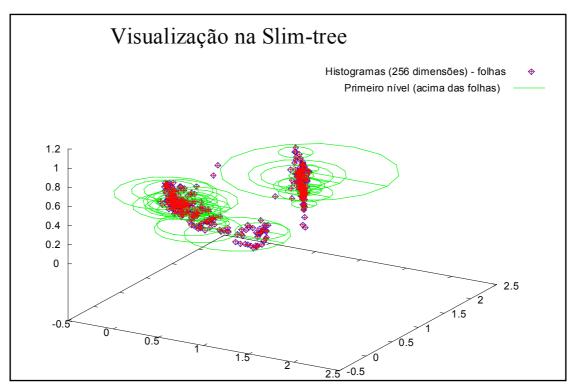
Os testes foram realizados sobre uma plataforma constituída de um servidor Windows NT, um PC Intel Pentium II 450 sob Windows 98, PC AMD K7 1,2 GHz e um notebook Intel Pentium II 750 ambos sob Windows 2000. As imagens DICOM, obtidas de tomógrafos de ressonância magnética, foram tratadas e armazenadas em banco de dados Oracle 8i como blocos binários (BLOBs) juntamente com suas informações textuais correspondentes, as quais foram extraídas do cabeçalho da própria imagem (ver Apêndice B para maiores detalhes sobre o formato DICOM). Os algoritmos foram desenvolvidos em linguagem C (C++Builder 5.0), com exceção do algoritmo para geração de histogramas métricos o qual foi desenvolvido utilizandose o Matlab 6.0, e os aplicativos foram implementados em linguagem C (C++Builder 5.0) e Java (JDK 2.0). A visualização dos resultados foi gerada pelo WGnuPlot, um software de visualização *freeware* (www.gnuplot.info ou www.ucc.ie/gnuplot/).

#### 6.2. Experimentos

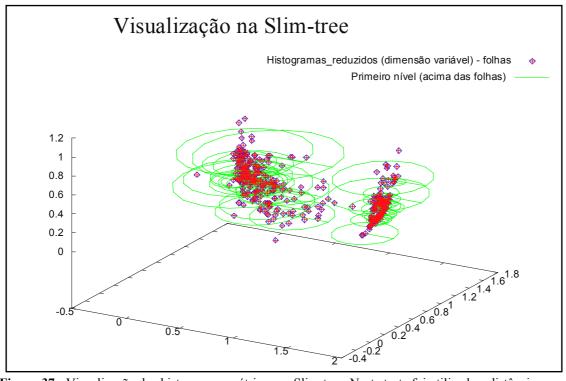
O histograma métrico e a distância *DM* propostos foram implementados utilizando a Slim-tree como método de acesso métrico a consultas por similiaridade. A MAM Slim-tree foi utilizada porque permite a minimização e uma forma de medir a sobreposição entre nós da árvore, além de indicar se a árvore construída propiciará um ganho efetivo de tempo com relação às respostas das consultas por similaridade. Como somente a Slim-tree tem estas funcionalidades, que mostraram-se úteis ao lidar com conjuntos de dados multidimensionais e

até adimensionais como é o caso dos histogramas tradicionais (multidimensionais) e dos histogramas métricos (adimensionais), optou-se por sua utilização.

Foram utilizados dois conjuntos de imagens nos experimentos. O primeiro conjunto, nomeado "RMCrânio500", tem 500 imagens de crânios humanos obtidos de tomografia por ressonância magnética (ToRM), e o segundo, chamado "RMVariados4247", possui 4247 imagens de várias partes do corpo humano obtidas também por ToRM. Cada imagem possui 256 níveis de cinza e diferentes resoluções espaciais. Extraiu-se o histograma normalizado e gerou-se o histograma métrico para cada imagem, criando dois conjuntos de características para cada conjunto de dados. Cada conjunto de características foi indexado utilizando a estrutura Slim-Tree (quatro Slim-trees no total). Os histogramas normalizados foram indexados utilizando a função distância Manhattan sobre os vetores com 256 elementos, e os histogramas métricos foram indexados utilizando a função distância DM com o número de buckets variando de 11 a 32. Na Figura 36 é apresentada a visualização dos histogramas originais (com 256 níveis de intensidade) indexados na Slim-tree para o conjunto RMCrânio500 e na Figura 37 é apresentada a visualização dos histogramas métricos (número de níveis variados) indexados na Slim-tree para o conjunto *RMCrânio500*. Os objetos (histogramas) são indicados pelos caracteres  $\bigoplus$  e elipses representam a projeção planar do nó do primeiro nível da árvore, imediatamente acima das folhas (objetos). Note que as distribuições espaciais dos objetos, histogramas, em ambos os casos, são semelhantes, e a presença de aglomerações se mantém.



**Figura 36 -** Visualização dos histogramas de 256 dimensões na Slim-tree. Neste teste foi utilizada a distância Manhathan (L1).



**Figura 37 -** Visualização dos histogramas métricos na Slim-tree. Neste teste foi utilizada a distância por diferença de área entre histogramas (DM).

Um dos objetivos deste projeto foi definir um modo mais rápido de pré-selecionar imagens em um banco de dados, que é realizado baseado nos histogramas de imagens, a fim de reduzir a necessidade de comparações entre imagens ao responder consultas por similaridade. Para avaliar o método aqui proposto, os experimentos compararam o conjunto de imagens resultante de consultas aos vizinhos mais próximos utilizando histogramas normalizados e histogramas métricos. Para isso, foi considerado que os resultados obtidos a partir dos histogramas normalizados são os corretos. O objetivo foi comparar os conjuntos de imagens recuperadas utilizando seus histogramas normalizados com os conjuntos de imagens recuperadas utilizando os histogramas métricos. Para cada conjunto de imagens obtido foram calculadas medidas de precisão e revocação sobre os resultados (o Anexo I detalha estas medidas). Precisão e Revocação são parâmetros normalmente utilizados para avaliar sistemas de recuperação de informações [Baeza-Yates 1999] e também sistemas de recuperação [Yamamoto 1999]. Revocação indica a proporção de imagens relevantes no banco de dados que foram recuperadas ao responder uma consulta. Precisão, por outro lado, é a proporção das imagens recuperadas que são relevantes para a consulta. Formalmente, seja X o conjunto de imagens relevantes para uma dada consulta, seja Y o conjunto de imagens recuperadas e sejam x, y e z o número de imagens nos conjuntos  $X \cap Y, \ \overline{X} \cap Y$  e  $X \cap \overline{Y}$  , respectivamente. Assim, precisão e revocação podem ser expressas pelas seguintes probabilidades condicionais [Yamamoto 1999]:

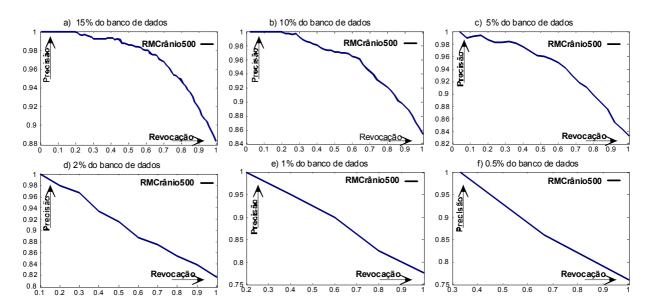
$$precisão = P(X \mid Y) = \frac{P(X \cap Y)}{P(Y)} = \frac{x}{x+y}$$

$$revocação = P(Y \mid X) = \frac{P(Y \cap X)}{P(X)} = \frac{x}{x+z}$$

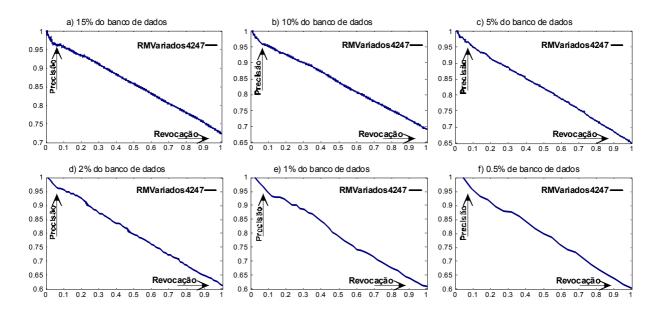
#### 6.3. Medidas de Precisão e Revocação

A Figura 38 e a Figura 39 apresentam as curvas de precisão e revocação obtidas ao solicitar consultas por vizinhos mais próximos para seis diferentes números de vizinhos sobre os bancos de dados *RMCrânio500* e *RMVariados4247* respectivamente. Os números de vizinhos se referem às porções do banco de dados, variando de 0,5% a 15% de cada banco de dados. Isto é,

solicitar consultas *k*-NN requerendo 1% do banco de dados significa solicitar consultas NN com *k*=5 para o banco de dados *RMCrânio500*, e com *k*=43 para o *RMVariados4247*. As curvas apresentam o valor médio de solicitar 50 consultas para cada número de vizinhos, com a imagem central aleatóriamente escolhida entre as imagens do banco de dados. Nestes experimentos, cada consulta é submetida às duas Slim-trees, uma indexando os histogramas normalizados e a outra indexando os histogramas métricos. Cada conjunto de respostas é classificado pela similaridade de cada histograma com o da imagem central da consulta. Os gráficos são obtidos verificando que os números crescentes de histogramas no conjunto de resposta da árvore indexada através do histograma métrico estão presentes no conjunto de resposta da árvore indexada pelo histograma normalizado. Como pode ser visto, a precisão de consulta é alta, sempre em torno de 75% para o conjunto mais homogêneo *RMCrânio500* e em torno de 60% para o conjunto de dados heterogêneo *RMVariados4247*, mesmo para consultas recuperando 0,5% do banco de dados com 100% de revocação.



**Figura 38** – Banco de dados *RMCrânio500*: plotagens de precisão e revocação ao responder consultas por vizinhos mais próximos (k-NN) onde o número k é dado como uma porcentagem do tamanho do banco de dados. (a) 15%, (b) 10%, (c) 5%, (d) 2%, (e) 1%, (f) 0.5%.



**Figura 39** – Banco de dados *RMVariados4247*: plotagens de precisão e revocação ao responder consultas por vizinhos mais próximos (k-NN), em que k é dado como uma porcentagem do tamanho do banco de dados: (a) 15%, (b) 10%, (c) 5%, (d) 2%, (e) 1%.

#### 6.4. Comparação de Tempo

Duas questões discutidas nesta seção são: (a) Qual é a diferença de tempo para indexar as imagens dos bancos de dados pelos histogramas convencionais e métricos? (b) A diferença de tempo para responder consultas sobre histogramas normalizados e histogramas métricos é relevante?

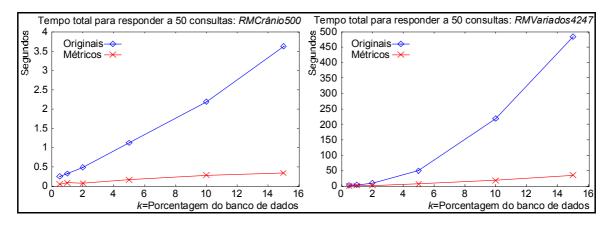
A Tabela 6 apresenta os tempos para a construção da Slim-tree a fim de indexar os histogramas normalizados métricos. Pode-se ver que é muito mais rápido construí-la sobre os histogramas métricos (58% mais rápido para conjunto de dados menor, 690% mais rápido para

**Tabela 6** – Tempo para construir a Slim-tree utilizando histogramas métricos e normalizados a partir dos conjuntos de dados *RMCrânio500* e *RMVariados4247*.

Banco de dados	Histograma	Tempo (s)
RMCrânio500	Métrico	4,62
	Convencional	7,31
RMVariados4247	Métrico	33,76
	Convencional	266,79

conjunto de dados maior). O tempo necessário para criar os histogramas métricos é equivalente ao tempo para construir a árvore de índices utilizando o histograma métrico, mas isto é feito somente uma vez e é menor ao comparar o tempo para responder às consultas.

Foi medido também o número de cálculos de distância realizados por segundo, e encontraram-se 1.289.400 distâncias *DM* por segundo e 269.430 distâncias Manhattan por segundo. Observe que os números são proporcionais ao tamanho do banco de dados e os resultados podem ser comparados a diferentes tamanhos de banco de dados, mas indica que o cálculo da distância *DM* é muito mais rápido do que a distância Manhattan que é considerada a mais simples entre as normas Lp. A Figura 40(a) mostra os tempos de resposta a consultas ao conjunto *RMCrânio500*, e a Figura 40(b) mostra os tempos de resposta a consultas ao conjunto *RMVariados4247*. Como pode ser visto, o ganho varia de 4 vezes mais rápido para pequenos valores de k (0,5% do banco de dados) para mais que 10 vezes mais rápido para grandes porções do banco de dados (15% do banco de dados). Todas as medidas foram obtidas utilizando um computador Intel Pentium II 450 MHz rodando sobre Windows NT. O software foi implementado em Borland C++ Builder.



**Figura 40** – Tempo total para responder 50 consultas por vizinhos mais próximos para cada porção do conjunto de dados: 0,5; 1; 2; 5; 10 e 15%, utilizando os hitogramas normalizados e métricos. (a) conjunto de dados *RMCrânio500*; (b) conjunto de dados *RMVariados4247*.

#### 6.5. Conclusões

Comparar duas imagens é um processo computacionalmente caro. Quando consultas são solicitadas sobre um banco de imagens para recuperar informações baseadas no conteúdo das mesmas, um processo de filtragem é utilizado para tentar reduzir o número de imagens a serem comparadas. Esta operação de filtragem utiliza características extraídas das imagens por algoritmos de processamento de imagens. Uma das características mais freqüentemente utilizadas nos estágios iniciais de filtragem é baseada em histogramas de intensidades ou brilho de imagens.

Um dos principais objetivos desta tese é definir um modo rápido para executar o processo de filtragem sobre um banco de imagens baseado nos histogramas das imagens. Para alcançar esta meta foi proposta uma nova técnica, chamada de **histograma métrico**. Foi mostrado que esta técnica de filtragem chega a ser até 10 vezes mais rápida, ao selecionar imagens por similaridade, do que a dos histogramas convencionais. Além disso, a criação de estruturas de índices utilizando histogramas métricos é ao menos 58% e acima de 700% vezes mais rápida quando comparada à criação de estruturas utilizando os histogramas convencionais.

Os histogramas métricos como o próprio nome indica são definidos em um domínio métrico, assim os parâmetros que melhor descrevem um histograma podem ser utilizados sem compromisso com o conjunto todo de imagens. Para permitir o uso de estruturas de acesso métrico para responder consultas por similaridade, foi definida uma função de distância *DM*, que satisfaz as propriedades de uma função de distância métrica. Esta função é também um modo rápido para comparar histogramas, como pode ser executado em média 4,7 vezes mais rápido que utilizando a função de distância Manhattan sobre vetores de 256 elementos.

Os histogramas métricos têm alguns efeitos colaterais desejados: eles permitem a recuperação de imagens de um modo que a escala, translação e rotação dos objetos é invariante como também o brilho das imagens. Estes efeitos possibilitam que as imagens semelhantes à imagem de referência, mas com diferente brilho, escala, rotação ou translação sejam obtidas sem esforço computacional adicional.

# Capítulo 7

### Conclusões Gerais

#### 7.1. Considerações Gerais e Contribuições

Os sistemas PACS (*Picture Archiving and Communication Systems*) trazem uma enorme contribuição para a organização e recuperação de informações médicas com inclusão de imagens. Há ainda poucos sistemas implantados efetivamente em hospitais pois o processo de transição para tais sistemas é complexo e muito caro. Siegel em [Siegel\_1999] faz uma estimativa sobre a transição da radiologia convencional para a radiologia sem filme nos Estados Unidos, que abrangerá ao menos 90% das operações de radiologia até o ano de 2024, conforme mostra a Figura 41.

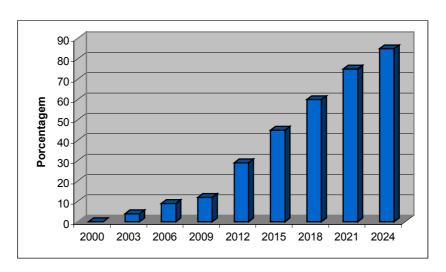


Figura 41 - Taxa de transição para a radiologia sem filme nos Estados Unidos [Siegel\_1999].

A implantação dos sistemas PACS traz, ainda, uma revolução para o trabalho cooperativo, uma vez que os médicos podem ter acesso rápido a documentos e exames e podem compartilhar opiniões e diagnósticos com outros médicos em outras áreas do hospital, ou até em outras unidades médicas sem precisar sair da própria sala [Lundberg 1999].

Os sistemas PACS convencionais baseiam-se em recuperação de imagens por informações textuais [Furuie\_1999] [Cao\_2000] [Zhang\_2001], mas com a contribuição de recentes pesquisas na área de recuperação baseada em conteúdo, eles ficarão mais poderosos auxiliando o diagnóstico médico pela pré-seleção de imagens de interesse baseada em seu conteúdo semântico. Smeulders et al. em [Smeulders\_2000] afirma que os projetos que geraram artigos publicados antes de 1990 sobre recuperação de imagens baseada em conteúdo são raros, freqüentemente obsoletos e de pouco impacto nos dias de hoje, e projetos que geraram artigos publicados depois de 1997 ainda estão começando a evoluir [Yoshitaka\_1999] [Korn\_1998] [El-Kwae\_1999] [Petrakis\_2001].

Embora o reconhecimento de padrões em imagens não seja uma área nova, com o advento de sistemas para recuperação de imagens baseada em conteúdo, e está recebendo um novo enfoque. Pesquisas recentes sobre estruturas métricas dinâmicas têm causado enorme impacto em técnicas de indexação por similaridade, possibilitando recuperar os dados por similaridade de forma rápida, apesar desses dados serem, muitas vezes, multidimensionais ou mesmo adimensionais.

Este trabalho contribuiu para a evolução do estado da arte de sistemas PACS com as seguintes inovações:

- Proposta de uma nova técnica de extração de características utilizando histogramas métricos. Os histogramas de imagens são muito utilizados como um primeiro método de separação entre conjuntos de imagens devido ao fato de ser computacionalmente barato obtê-lo. Porém, os histogramas tradicionais como definidos na literatura não permitem a comparação de imagens de tamanhos e níveis de brilho variados. Os histogramas métricos permitem tal comparação, além de efetuá-la em uma fração do tempo gasto com os histogramas convencionais.
- A definição de uma nova função de distância métrica, a *DM* ( ), que permite a comparação de histogramas métricos. Como o número e a largura dos *buckets* dos

histogramas métricos são variáveis, as medidas usuais de distância, tal como Euclideana, Manhattan e Chebychev não podem ser utilizadas. A distância *DM* foi proposta como o cálculo da área entre as funções de aproximação dos histogramas métricos envolvidos. Apesar do cálculo de área entre curvas ser intuitivamente mais caro do que o cálculo de uma distância Manhattan ou Chebychev, o algoritmo desenvolvido para a distância *DM* é bastante otimizado, resultando em um cálculo em torno de 4 vezes mais rápido do que a distância Manhattan (a distância mais simples).

- A fim de permitir indexar os histogramas métricos a Slim-tree teve que ser estendida para suportar esse novo tipo de dados, bem como para incorporar a distância *DM*.
- Mostrar que é possível ter um sistema PACS que permita a recuperação de imagens baseadas em seu conteúdo utilizando um sistema de gerenciamento de bases de dados comercial, nesse caso o Oracle 8i.

Além das contribuições colocadas acima é importante ressaltar que este é o primeiro trabalho desenvolvido nessa área no GBDI. Dessa forma, foi necessário desenvolver todo um arcabouço de tratamento e visualização de imagens, reconhecimento de formatos DICOM, modelagem e implementação do banco de imagens, entre outras atividades, que apesar de não serem contribuições inovadoras, sem tal desenvolvimento ficaria impossível obter os resultados alcançados.

#### 7.2. Trabalhos Futuros

Este projeto visou mostrar que técnicas de recuperação de imagens baseada no conteúdo das mesmas podem ser integradas a um sistema PACS. As técnicas propostas nesta tese estão sendo incorporadas ao sistema mini-PACS atualmente em desenvolvimento. Como este é um projeto inicial, existem diversas atividades que podem dar continuidade a ele e que serão colocadas a seguir.

#### 7.2.1 – Ampliação do Conjunto de Extratores de Características de Imagens

Da mesma forma que o histograma métrico e a distância *DM* foi incorporada à Slim-tree, outras técnicas que mapeiem a distribuição de textura e formas presentes na imagem poderiam ser incorporadas ao sistema. Ou seja, a estrutura métrica Slim-tree precisaria ser estendida para suportar essas outras técnicas. Cada característica extraída das imagens gera uma árvore de indexação (Slim-tree).

Para obter-se uma simulação do processo realizado pelo cérebro humano no processo de reconhecer as imagens que baseia-se na distribuição de cores, formas e texturas, seria muito interessante poder contar com pelo menos um extrator de características que atuasse sobre cada abordagem. Dessa forma, o sistema estaria mais completo e com maior capacidade de atingir às expectativas dos usuários.

# 7.2.2 – Avaliação por Radiologistas dos Resultados de Recuperação de Imagens por Histogramas Métricos

A avaliação efetuada nesta tese foi embasada na comparação do conjunto de imagens recuperado pelos histogramas métricos com relação ao conjunto recuperado por histogramas tradicionais. A avaliação por especialistas na área médica permitirá a validação dos resultados obtidos na fase final do projeto. Observe que a modelagem do sistema mini-PACS foi validada pela equipe do CCIFM.

Como os histogramas métricos podem ser invariantes ao brilho das imagens, assim como são invariantes às transformações geométricas espera-se que essas duas restrições quanto aos histogramas convencionais que são suplantadas pelos correspondentes métricos, contribua para a recuperação de imagens que sejam realmente relevantes para a consulta solicitada.

#### 7.2.3 – Geração de Assinaturas de Imagens

Outra forma de se indexar e recuperar imagens é por assinaturas. Utilizando os histogramas métricos pode-se gerar assinaturas mais efetivas sobre as imagens do que com os histogramas convencionais. Dessa forma, a recuperação de imagens por conteúdo pode ser feita ainda mais rapidamente. Como o tempo de resposta às consultas é um dos fatores que

contribuem para a aceitação do sistema pelos médicos, diminuir este tempo de resposta mantendo a sua qualidade é um objetivo que deve estar presente na mente dos desenvolvedores de métodos de extração de características.

#### 7.2.4 – Execução de Macros para a Realização de Consultas.

O mini-PACS é contituído por três sistemas, o sistema de processamento de imagens (SPI), o sistema gerenciador de bases de dados e imagens (SGBDI) e o servidor de Web (SW). Tanto o SPI quanto o SGBDI são sistemas abrangentes e flexíveis, que provêm ampla gama de seus respectivos recursos aos usuários, permitindo grande diversidade de opções. No entanto, essa diversidade pode, muitas vezes, confundir e intimidar os usuários típicos, médicos que estão efetuando o acompanhamento clínico dos pacientes, e não estão acostumados a recuperar as informações pelas imagens (médicos radiologistas). Porém, as necessidades desses primeiros usuários tendem a ser previsíveis e repetitivas. Assim, é altamente desejável, para aumentar a usabilidade do sistema, dispor operações pré-configuradas que o clínico possa ter preparadas e personalizadas para si por administradores com conhecimento mais técnico do sistema, sob a supervisão do especialista em radiologia. Isso pode ser feito por um sistema de macros operando no SW, que traz a vantagem adicional de permitir o ajuste (configuração fina) do sistema otimizado para atender ao conjunto de macros disponíveis numa dada aplicação. A definição dos operadores e comandos da linguagem origina-se das necessidades dos radiologistas e técnicos. As solicitações por eles efetuadas para a disponibilidade desses comandos auxiliam a definir os recursos básicos que o sistema deve prover de uma maneira natural para os usuários finais.

#### 7.2.5. – Desenvolvimento do Servidor de Web (SW)

Um servidor ágil e confiável para a manutenção das imagens e informações de pacientes deve suportar o desenvolvimento do mini-PACS. Como tal desenvolvimento transcendeu o escopo deste trabalho, a implementação do SW completo foi deixada para uma fase posterior deste projeto.

#### 7.2.6 - Cruzamento de Informações e Mineração nos Dados do Mini-PACS

O sistema mini-PACS mantém todas as informações referentes aos pacientes do hospital, incluindo suas imagens. Isso gera um volume de informações muito grande cuja tendência é de crescer exponencialmente. Além das consultas tradicionais previamente preparadas para serem respondidas pelo sistema, que são as consultas textuais, numéricas e as que utilizam o conteúdo das imagens; os gerentes do sistema poderiam obter também informações adicionais do tipo: "Existe uma predominância em termos de regiões da cidade/país onde ocorre determinada anomalia?", ou "pacientes que possuem tumor do tipo A também desenvolvem a moléstia B com xx% de probabilidade". Observe que tais consultas são efetuadas sobre os dados armazenados visando obter informações adicionais e valiosas dos mesmos.

## Referências Bibliográficas

- [Alcocer\_1996] P.R.C. Alcocer, C.P. Melo, S.S. Furuie, N. Bertozzo Jr., L.C. Parzianello, M. Rebelo. O Projeto Pacs: um Sistema para Visualização Dinâmica e Armazenamento de Imagens de Angiografía Digital no Incor. *in Anais do 3 Fórum Nacional de Ciência e Tecnologia em Saúde*, Campos do Jordão SP, 2: 695-696, 1996.
- [Aslandogan\_1999] Y.A. Aslandogan e C.T. Yu. Techniques and Systems for Image and Video Retrieval. *IEEE Trans. on Knowledge and Data Engineering*. 11(1):56-63, Janeiro, 1999.
- [Baeza-Yates\_1994] R.A. Baeza-Yates, W. Cunto, U. Manber, S. Wu. Proximity Matching Using Fixed-Queries Trees. *In 5th Annual Symp. on Combinatorial Pattern Matching (CPM)*, Asilomar, California, 1994.
- [Baker\_1997] S. Baker, V. Cahill e P. Nixon. Bridging Boundaries: Corba in Perspective. *IEEE Internet Computing*, 1(5):52-57, Setembro/Outubro, 1997.
- [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 ACM Int'l Conference on Data Management (SIGMOD)*. Proceedings, pp 322-331, Maio, 1990.
- [Bozkaya\_1997] T. Bozkaya, Z.M. Özsoyoglu. Distance-Based Indexing for High-Dimensional Metric Spaces," in ACM Int'l Conference on Data Management (SIGMOD). Proceedings. Tucson, AZ, 1997.
- [Bozkaya\_1999] T. Bozkaya, Z.M. Özsoyoglu. Indexing Large Metric Spaces for Similarity Search Queries. *In ACM Transactions on Database Systems (TODS)*, vol. 24, pp. 361-404, 1999.
- [Brin\_1995] S. Brin. Near neighbor search in large metric spaces. *In Intl. Conf. on Very Large Databases (VLDB)*. Proceedings. Zurich, Switzerland, 1995.
- [Brown\_1992] L.G. Brown. A Survey of Image Registration Techniques. *ACM Computing Surveys*. 24(4):325-376, 1992.
- [Brunelli\_1998] R. Brunelli, O. Mich. On the Use of Histograms for Image Retrieval. *In: International Conference on Multimedia Computing and Systems*, II. Proceedings, vol. 2, Austin, Texas, 1998.
- [Burkhard\_1973] W.A. Burkhard, R.M. Keller. Some Approaches to Best-Match File Searching. *In CACM*. Proceedings. vol. 16, pp. 230-236, 1973.
- [Cao\_2000] X. Cao, H.K. Huang. Current Status and Future Advances of Digital Radiography and PACS. IEEE Engineering in Medicine and Biology, 19(5):80-88, setembro/outubro, 2000.

- [Chieuh\_1994] T-C, Chiueh. Content-Based Image Indexing. *In Intl. Conf. on Very Large Databases (VLDB)*, Santiago de Chile, Chile, 1994.
- [Ciaccia\_1997A] P. Ciaccia, M. Patella, F. Rabitti, P. Zezula. Indexing Metric Spaces with M-tree. apresentado em *Atti del Quinto Convegno Nazionale SEBD*, Verona, Italy, June 1997.
- [Ciaccia\_1997B] P. Ciaccia, M. Patella, P. Zezula M-tree: An efficient access method for similarity search in metric spaces. Proc. VLDB International Conference, pp. 426-435, Athens, Greece, September 1997.
- [Ciaccia\_1998] P. Ciaccia, M. Patella. Bulk Loading the M-tree. Proc. of the 9th Australasian Conference (ADC'98), pp. 15-26, Perth, Australia, February 1998.
- [Ciaccia\_1999] P. Ciaccia, A. Nanni, M. Patella A Query-sensitive Cost Model for Similarity Queries with M-tree. Proc. 10th Australasian Database Conference (ADC'99), pp. 65-76, Auckland, New Zealand, January 1999.
- [Crudele\_1997] M. Crudele, G.J. Clapworthy, M.A. Krokos, G. Salcito, N. Vasilonokolidakis. A Distributed Database on the INTERNET of 3D Models of Human Pathological Organs. *In: Anais of the 10'IEEE Symposium on CBMS*, pp. 256-260, Maribor-Slovenia, Junho 1997.
- [Dev\_1999] P. Dev. Imaging and Visualization in Medical Education. *IEEE Computer Graphics and Applications*, 19(3):21-31, Maio/Junho, 1999.
- [Ducan\_2000] J.S. Ducan e N. Ayache. Medical Image Analysis: Progress over Two Decades and the Challenges Ahead. *IEEE Trans. on Pattern Analysis and Machine Intelligence*. 22(1):85-106, Janeiro, 2000.
- [Duda\_2001] R.O. Duda, P.E. Hart, D.G. Stork. Pattern Classification. John Wiley&Sons, Inc, 2001 2nd Edition, 2001.
- [El-Kwae\_1999] E.A. El-KWae e M.R. Kabuka. A Robust Framework for Content-Based Retrieval by Spatial Similarity in Image Databases. *ACM Trans. On Information Systems*. 17(2):174-198, Abril, 1999.
- [Evans\_1997] E. Evans e D. Rogers. Using Java applets and CORBA for multi-user distributed applications. *IEEE Internet Computing*, 1(3):43-55, Maio/Junho, 1997.
- [Evans\_1996] T. Evans. Construindo uma Intranet. Makron Books, ISBN 85.346.087-2, 1998.
- [Faloutsos\_1994] C. Faloutsos, R. Barber, M. Flickner, J. Hafner, W. Niblack, D. Petkovic, W. Equitz. Efficient and Effective Querying by content. *Journal of Intelligent Information Systems*, vol. 3, pp. 231-262.
- [Faloutsos\_1995] C. Faloutsos and K. Lin. FastMap: A Fast Algorithm for Indexing, Data-Mining and Visualization of Traditional and Multimedia Datasets. apresentado em ACM Int'l Conference on Data Management (SIGMOD), Zurich, Switzerland, 1995.

- [Faloutsos\_1996] C. Faloutsos. Searching Multimedia Database by Content. Kluwer Academic Publishers, 1996.
- [Flickner\_1995] M. Flickner et al. Query by Image and Video Content: The QBIC System. *IEEE Computer*, 28(9): 23-32, setembro de 1995.
- [Furuie\_1999] S.S. Furuie, M.A. Gutierrez, N.B. Bertozzo, J.C.B. Figueiredo, M. Yamagutti. Archiving and Retrieving Long-Term Cineangiographic Images in a PACS. Computers in Cardiology, 26:435-438, 1999.
- [Gaede\_1998] V. Gaede, O. Gunther, O. Multidimensional Access Methods. *ACM Computing Surveys*. vol. 30, pp. 170-231, 1998.
- [Gennip\_1992] E.M.S.J. Van Gennip, A.R. Bakker, M. Greberman. Do the Benefits outweigh the costs of PACS? The results of an International Workshop pn Technology Assessment of PACS. *Computer Methods and Programs in Biomedicine*. 37:265-271, Elsevier Science Publishers, 1992.
- [Gonzales\_1993] R.C. Gonzales, P. Wintz. Digital Image Processing. Addison Wesley, second edition, 1993.
- [Gross\_1998] M. H. Gross. Computer Graphics in Medicine: From Visualization to Surgery Simulation. *ACM SIGGRAPH Computer Graphics*, 32(1):53-56, Fevereiro, 1998.
- [Gudivada\_1995] V.N. Gudivada, V.V. Raghavan. Content-Based Image Retrieval Systems. *IEEE Computer*, 28(9): 18-22, setembro de 1995.
- [Gupta\_1998] A. Gupta et al. Implementing Java computing: Sun on architecture and applications deployment. *IEEE Internet Computing*, 2(2):60-64, Março/Abril, 1998.
- [Guttman\_1984] A. Guttmann. R-trees: A Dynamic Index Structure for Spatial Searching. *In: Anais do 1984 ACM SIGMOD International Conference on Management of Data*, pp. 47-57, junho de 1984.
- [Hafner\_1995] J. Hafner, H. Sawhney, W. Equitz, M. Flickner, W. Niblack. Efficiente Color Histogram Indexing for Quadratic Form Distance Function. *IEEE Transactions on Patterns Analysis and Machine Intelligence*, vol. 17, pp. 729-736, Julho, 1995.
- [Hamilton\_1996] M.A. Hamilton. Java and the shift to net-centric computing. *IEEE Computer*, 29(8):31-39, Agosto, 1996.
- [Hellerstein\_1995] J. M. Hellerstein, J. F. Naughton, A. Pfeffer. Generalized Search Trees for Database Systems. apresentado em *Intl Conf on Very Large Databases (VLDB)*, Zurich, Switzerland, 1995.
- [Hinneburg\_1999] A. Hinneburg and D. A. Keim. Optimal Grid-Clustering: Towards Breaking the Curse of Dimensionality in High-Dimensional Clustering. Apresentado em *Intl Conf on Very Large Databases (VLDB)*, 1999.

- [Horiil\_1994] S.C. Horiil. DICOM: An introduction to the standard. Disponível na URL: <a href="http://www.xray.hmc.psu.edu/dicom/dicom\_intro/DICOMIntro.html">http://www.xray.hmc.psu.edu/dicom/dicom\_intro/DICOMIntro.html</a>.
- [Hua\_1999] K.A. Hua, K. Vu e J-H. Oh. SamMatch: A Flexible and Efficient Sampling-Based Image Retrieval Technique for Large Image Databases. *In: International Conference on Multimedia*, VII. Proceedings. p.225-234, Orlando, FL, 1999.
- [Jagannathan\_1998] V. Jagannathan et al. Objects in healthcare: focus on standards. Versão simplificada do artigo de mesmo título que aparece em ACM Standards View, Summer'98. Disponível na URL: <a href="http://www.acl.lanl.gov/OMG/CORBAmed/Careflow/June16ACMPaper.htm">http://www.acl.lanl.gov/OMG/CORBAmed/Careflow/June16ACMPaper.htm</a>.
- [Kimura\_1998] M. Kimura, K. Ohe, H. Yoshihara, Y. Ando, F. Kawamata, T. Hishiki, et.al.: MERIT-9; a patient information exchange guideline using MML, HL7, and DICOM. *International Journal of Medical Informatics*, 51(1): 59-68, 1998.
- [Ko\_2000] B. Ko, H.-S. Lee, H. Byun. Image Retrieval Using Flexible Image Subblocks. Apresentado em *ACM Symposium on Applied Computing*, 2000.
- [Korn\_1998] P. Korn, N. Sidiropoulos, C. Faloutsos, E. Siegel, Z. Protopapas. Fast and Effective Retrieval of Medical Tumor Shapes. *IEEE Trans. Knowledge and Data Engineering*, 10(6):889-904, Novembro/Dezembro, 1998.
- [Kruskal\_1956] J. B. Kruskal. On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem. *Proc American Math Soc*, 7: 48-50, 1956.
- [Kuzmak\_1998] P. M. Kuzmak, R. E. Dayhoff. Integration of Imaging Functionality into the Healthcare Enterprise Using DICOM. *Journal of Digital Imaging*, 11(3), Supplement 1, Agosto, 1998
- [Lundberg\_1999] N. Lundberg. Impacts of PACS on Radiological Work. *Proceedings of the International ACM SIGGROUP Conference on Supporting Group Work.* pp. 169-178, 1999.
- [Marsh\_1997] A. Marsh. EUROMED The Creation of a Telemedical Information Society. *In: Anais do 10'IEEE Symposium on CBMS*, pp. 86-91, Maribor-Slovenia, junho de 97.
- [Mascarenhas\_1989] N.D.A. Mascarenhas, F.R.D. Velasco. Processamento Digital de Imagens. *IV Escola Brasileiro-Argentina de Informática*, Universidade Católica de Santiago del Estero, Termas do Rio Hondo Argentina, 1989.
- [Mehuys\_1997] A. Mehuys. The project 'Patiënt en Dossier'. *In: Anais of the 10'IEEE Symposium on CBMS*, pp. 54-57, Maribor-Slovenia, junho, 1997.
- [Meire\_1996] H.B. Meire; A. Darzi, N. Lee. Digital Imaging. URL: <a href="http://www.tecc.co.uk/bmj/abcmc/abcmc19.html">http://www.tecc.co.uk/bmj/abcmc/abcmc19.html</a>.

- [Pagel\_2000] B.-U. Pagel, F. Korn, C. Faloutsos. Deflating the Dimensionality Curse Using Multiple Fractal Dimensions. Apresentado em *16th Intl Conf on Data Engineering* (ICDE2000), San Diego, CA, 28 de Fevereiro a 3 de Março, 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. apresentado em *ACM Int'l Conference on Data Management (SIGMOD)*, San Jose, CA, 1995.
- [Pass\_1996] G. Pass, R. Zabih, JustinMiller. Comparing Images Using Color Coherence Vector. Apresentado em *ACM Multimedia*, Boston, MA, 1996.
- [Pentland\_1996] A. Pentland, R. Picard, S. Sclaroff. Content-based Manipulation of Image Databases. *Intl Journal of Computer Vision*, vol. 18, pp. 233-254, Junho, 1996.
- [Perry\_1996] P.J. Perry. Creating cool Web applets with Java. IDG Books Worldwide, ISBN 1-56884-881-1, 1996.
- [Petrakis\_1997] E. G. M. Petrakis, C. Faloutsos. Similarity Searching in Medical Image Databases. *IEEE Transactions on Knowledge and Data Engineering*, 9(3):435-447, Maio/Junho, 1997.
- [Petrakis\_2001] E. G. M. Petrakis, C. Faloutsos, K.-I. D. Lin. ImageMap: An Image Indexing Method Based on Spatial Similarity. *IEEE Trans on Knowledge and Data Engineering*, volume a ser impresso.
- [Raghavan\_1989A] V.V.Raghavan, P.Bollmann, G.S.Jung. Retrieval system evaluation using recall and precision: Problems and answers. Em Proc. of the 12th ACM SIGIR Conference, pp. 59-68, Cambrige, USA, junho, 1989.
- [Raghavan\_1989B] V.V.Raghavan, G.S.Jung, P.Bollmann. A critical investigation of recall and precision as measures of retrieval system performance. ACM Transactions on Office and Information Systems, 7(3):205-229, 1989.
- [Ratib\_1997] O. Ratib. From PACS to the World Wide Web. URL: <a href="http://www.hon.ch/Library/papers/ratib\_t.html">http://www.hon.ch/Library/papers/ratib\_t.html</a>.
- [Rhodes\_1996] M.L. Rhodes, D.D. Robertson. Applications in Surgery and Therapy. *IEEE Computer Graphics & Applications*. 16(1): 28-29. novembro de 1993.
- [Rodley\_1997] J. Rodley. Developing Databases for the Web & Intranets. Coriolis Group Books, ISBN 1-57610-051-0, 1997.
- [Russ\_1995] J. C. Russ. The Image Processing Handbook. 2nd ed. Boca Raton: CRC Press, 1995.
- [Santos\_1996] R.R. Santos, A.J.M. Traina e C. Traina Jr. Uma Linguagem de Definição e Recuperação de Imagens Baseada em Conteúdo em uma Base de Dados Orientada a

- Objetos. *Anais do XI Simpósio Brasileiro de Banco de Dados*. São Carlos: 14 a 16 de outubro de 1996. pp. 127-142.
- [Santos\_2001] R.F. Santos Filho, A.J.M. Traina, C. Traina Jr., C. Faloutsos. Similarity Search without Tears: The OMNI-family of All-purpose Access Methods. *Proceedings of the 17th IEEE Intl. Conference on Data Engineering (ICDE)*, pp. 623-630, Heidelberg, Germany, de 2 a 6 de Abril de 2001,.
- [Shasha\_1990] D. Shasha, T.L. Wang. New techniques for best-match retrieval. *In ACM Transactions on Information Systems*. vol. 8, pp. 140-158, 1990.
- [Sclabassi\_1996] R.J. Sclabassi et al. NeuroNet: Collaborative Intraoperative Guidance and Control. *IEEE Computer Graphics and Applications*. 16(1):39-45, 1996.
- [Sellis\_1987] T. Sellis, N. Roussopoulos, C. Faloutsos. The R+-tree: A Dynamic Index for Multi-dimensional Objects. apresentado em *Intl Conf on Very Large Databases* (VLDB), Brighton, England, 1987.
- [Senzako\_1996] E. Y. Senzako. SisMatch Matching para Imagens de Tomografia por Ressonância Magnética. *Dissertação de mestrado apreentada ao ICMSC-USP*, Outubro de 1996.
- [Siegel\_1999] E.L.Siegel, R.M. Kolodner. Filmeless Radiology. Health Informatics Series, Springer-Verlag, New York Inc., 1999.
- [Smeulders\_2000] A. Smeulders, M. Worring, S. Santini, A. Gupta e R. Jain. Content-Based Image Retrieval at the End of the Early Years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1349-1380, Dezembro, 2000.
- [Stasiu\_2001] R.K. Stasiu, G.L. Bichinho. Components proposal for medical images and HIS. *Proceedings of the 14th IEEE Symposium on Computer-Based Medical Systems. CBMS* 2001, pp. 73 -78, Bethesda, Maryland, 26 e 27 Julho de 2001.
- [Tague-Sutcliffe\_1992] J.Tague-Sutcliffe. Measuring the informativeness of a retrieval process. Em Proc. of the 15th Annual Int. ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 23-36, Copenhagen, Denmark, 1992.
- [Tchounikine\_1997] A. Tchounikine, Y. Amghar, A. Flory. Semantic Interrogation for a Radiological Documentary Record. *In: Anais of the 10'IEEE Symposium on CBMS*, pp. 98-102, Maribor-Slovenia, Junho 1997.
- [Tomita 1990] F. Tomita and T. Saburo. Computer Analysis of Visual Textures. Kluwer, 1990.
- [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.
- [TrainaJr\_1997A] C. Traina Jr., A.J.M. Traina, R.A. Ribeiro, E.Y. Senzako. Content-based Medical Images Retrieval in Object Oriented Database. *Proceedings of 10th IEEE*

- Symposium on Computer-Based Medical System, vol. II., Maribor- Slovenia no período de 11 a 13 de junho de 1997, pp. 67-72.
- [TrainaJr\_1997B] C. Traina Jr., A.J.M. Traina, R.R. Santos, E.Y. Senzako. A Support System for Content-Based Medical Image Retrieval in Object Oriented Databases. *Journal of Medical Systems* (Invited Paper) Plenum Press, 21(6):339-352, dezembro de 1997.
- [TrainaJr\_1998A] C. Traina Jr., A.J.M. Traina, R.R. Santos, E.Y. Senzako. A Tool for Content-based Image Retrieval in Object-oriented Databases. *Proceedings do The Third Biennial World Conference on Integrated Designs and Processs Technology* (IDPT) Issues and Applications of Database Technology (IADT'98) Society for Design and Process Science, Berlin-Alemanha, no período de 6-9 de julho de 1998, pp.344-351.
- [TrainaJr\_1998B] C. Traina Jr., A.J.M. Traina, R.R. Santos, E.Y. Senzako. Support to Content-Based Image Query in Object-Oriented Databases. *in Anais do SAC '98 ACM Symposium on Applied Computing*, realizado no período de 27 de fevereiro a 1 de março de 1998 em Atlanta, Georgia, E.U.A, pp. 241-247.
- [TrainaJr\_2000] C. Traina Jr., A.J.M. Traina, B. Seeger, C. Faloutsos. Slim-Trees: High Performance Metric Trees Minimizing Overlap Between Nodes. In Intl. Conf. on Extending Database Technology. Konstanz, Germany, 2000.
- [TrainaJr\_2002] C. Traina, A. J. M. Traina, C. Faloutsos, B. Seeger. Fast Indexing and Visualization of Metric Datasets Using Slim-trees. *IEEE Transactions on Knowledge and Data Engineering*, volume a ser impresso.
- [Uhlmann\_1991] J.K. Uhlmann. Satisfying General Proximity/Similarity Queries with Metric Trees. *Information Processing Letter*, vol. 40, pp. 175-179, 1991.
- [Vinoski\_1996] S. Vinoski. CORBA: Integrating Diverse Applications within Distributed Heterogeneous Environments. *IEEE Communications Magazine*, Fevereiro, 1997, disponível na URL: http://www.cs.wustl.edu/~schmidt/vinoski.ps.gz, 1997.
- [Wilson\_1997] D.R.Wilson, T.R.Martinez. Improved Heterogeneous Distance Functions. Journal of Artificial Inteligence Research, no 6, pp.1-34, 1997.
- [Yamamoto\_1999] H. Yamamoto, H. Iwasa, N. Yokoya, H. Takemura. Content-Based Similarity Retrieval of Images Based on Spatial Color Distributions. em Proc. of the 10th International Conference on Image Analysis and Processing, pp.951-956, setembro, 1999.
- [Yang\_1997] Z. Yang e K. Duddy. CORBA: A Platform for Distributed Object Computing. Disponível na URL: http://www.infosys.tuwien.ac.at/Research/Corba/archive/intro/OSR.ps.gz, 1997.
- [Yianilos\_1993] P.N. Yianilos. Data Structures and Algorithms for Nearest Neighbor Search in General Metric Spaces. *In Fourth Annual ACM/SIGACT-SIAM Symposium on Discrete Algorithms SODA*. Austin, Texas, USA, 1993.

[Yoshitaka\_1999] A. Yoshitaka e T. Ichikawa. A Survey on content-based retrieval for multimedia databases. *IEEE Trans. Knowledge and Data Engineering*, 11(1):81-93, Janeiro/Fevereiro, 1999.

[Yourdon\_1996] E. Yourdon. Java, the Web and software development. *IEEE Computer*, 29(8):25-30, Agosto, 1996.

[Zhang\_2001] Z.M. Zhang, A. Krol, P. Guangbiao. IBMAS: an Internet Based Medical Archive System. *Proceedings of the 14th IEEE Symposium on Computer-Based Medical Systems CBMS 2001*, pp. 541-546, Bethesda, Maryland 26 e27 de Julho 2001.

### **WEBLIOGRAFIA - Sites utilizados para consulta:**

Banco de Dados - ORACLE: http://technet.oracle.com

Java - SUN Microsystems: http://developer.javasoft.com

http://java.sun.com

http://we.got.net/~rwilkman/java

Jbuilder - INPRISE: <a href="http://www.borland.com/jbuilder">http://www.borland.com/jbuilder</a>

Outros: <a href="http://we.got.net/~rwilkman/java">http://we.got.net/~rwilkman/java</a>

Grupo de discussões: borland.public.jbuilder.applet-issues

inprise.public.as400.jbuilder

borland.public.jbuilder.java.language

DICOM: <a href="http://www.xray.hmc.psu.edu/dicom/dicom\_home.html">http://www.xray.hmc.psu.edu/dicom/dicom\_home.html</a>.

http://www.nema.org/nema/medical/dicom

http://www.siemens.de/med/e/dicom/dicom.html

CORBA: <a href="http://www.omg.org">http://www.omg.org</a>

http://www.infosys.tuwien.ac.at/Research/Corba/intro.html

The Visible Human Project: <a href="http://www.nlm.nih.gov/research/visible/visible-human.html">http://www.nlm.nih.gov/research/visible/visible-human.html</a>

### Applets para visualização de imagens médicas:

The NPAC Visible Human Viewer:

http://www.npac.syr.edu/projects/vishuman/VisibleHuman.html

Visible Human Surface Viewer: <a href="http://lsppc40.epfl.ch/SurfaceServer/surface.html">http://lsppc40.epfl.ch/SurfaceServer/surface.html</a>

## Anexo I

# Avaliação de Performance na Recuperação de Informações<sup>6</sup>

Antes da implementação final de um sistema de recuperação de informações, ~geralmente é necessário executar uma avaliação do sistema. O tipo de avaliação a ser considerada depende dos objetivos do sistema de informação. Obviamente, todo software deve fornecer a funcionalidade para a qual foi concebido. Assim, o primeiro tipo de avaliação a ser considerada é uma análise funcional na qual as funcionalidades do sistema especificado são testadas uma a uma. Tal análise deve, também, incluir uma fase de análise de erro na qual, em vez de procurar funcionalidades, tenta-se fazer o sistema falhar. É um procedimento simples que pode ser muito útil para identificar erros de programa. Dado que o sistema passou na fase de análise funcional, deve-se realizar a avaliação da performance do sistema.

As medidas mais comuns de performance de sistemas são tempo e espaço. Quanto menor o tempo de resposta, menor o espaço utilizado, melhor o sistema será considerado. Há um equilíbrio entre complexidade espacial e temporal que, geralmente, permite trocar um pelo outro.

Em um sistema projetado para fornecer recuperação de dados, o tempo de resposta e o espaço requeridos são, geralmente, métricas de grande interesse e as normalmente adotadas para avaliar um sistema. Neste caso, procura-se pela performance de estruturas de indexação (que são utilizadas para acelerar a consulta), pela interação com o sistema operacional, pela demora nos canais de comunicação, e pela sobrecarga introduzida pelas várias camadas do software que geralmente estão presentes. Esta forma de avaliação é denominada avaliação de performance.

<sup>&</sup>lt;sup>6</sup> Extraído do Capítulo 3 de [Baeza-Yates 1999]

Em um sistema projetado para fornecer recuperação de informação, outras métricas, além de tempo e espaço, são também utilizadas. Na realidade, uma vez que a solicitação de consulta feita por um usuário é inerentemente vaga, os documentos recuperados não são exatamente respostas e têm que ser ordenados de acordo com a relevância de cada um à consulta. Tal classificação por relevância introduz um componente que não está presente em sistemas de recuperação de dados e que têm um papel importante em recuperação de informações. Assim, sistemas de recuperação de informações necessitam da avaliação de quão preciso é o conjunto de resposta. Este tipo de avaliação é denominado avaliação de performance na recuperação.

Quando se deseja fazer a avaliação de performance na recuperação, deve-se primeiro considerar a tarefa de recuperação a ser avaliada. Por exemplo, a tarefa de recuperação poderia consistir simplesmente de uma consulta processada em *batch* (isto é, o usuário submete uma consulta e recebe uma resposta em retorno) ou de uma sessão interativa completa (isto é, o usuário especifica sua necessidade por informação em uma série de passos interativos com o sistema). Além disso, a tarefa de recuperação poderia, também, ser composta por uma combinação destas duas estratégias. Consultas em batch ou interativas são processos bastante distintos e, assim, suas avaliações são também distintas. Na realidade, em uma sessão interativa, o esforço do usuário, características de projeto de interface, diretrizes fornecidas pelo sistema, e duração da sessão são aspectos críticos que deveriam ser observados e medidos. Em uma sessão em batch, nenhum destes aspectos é tão importante quanto a qualidade do conjunto de resposta gerado.

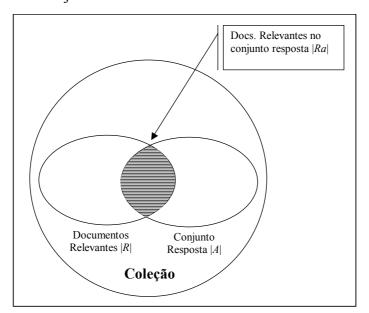
Além da natureza da consulta, deve-se considerar a configuração onde a avaliação será feita e o tipo de interface utilizada. Com relação à configuração, a avaliação de experimentos realizados em um laboratório poderia ser bastante distinta da avaliação de experimentos executados em uma situação real. Com relação ao tipo de interface, enquanto sistemas bibliográficos antigos apresentam ao usuário interfaces que geralmente operam em modo batch, sistemas mais novos geralmente apresentam ao usuário interfaces que operam interativamente.

Avaliação de performance na recuperação em antigos sistemas de recuperação de informação baseados em computador focavam, principalmente, em experimentos de laboratório planejados para interfaces batch. Nos anos 90, as atenções se voltaram para a avaliação de experimentos da vida real. Apesar desta tendência, os experimentos de laboratório ainda são

dominantes. Dois motivos são a repetitividade e a escalabilidade fornecida pela configuração controlada de um laboratório.

### I.1. Precisão e Revocação (Precision and Recall)

Considere como exemplo um pedido de informação I (de uma coleção de referência teste) e seu conjunto R de documentos relevantes. Seja |R| o número de documentos neste conjunto. Assuma que uma dada estratégia de recuperação (que está sendo avaliada) processa o pedido de informação I e gera um conjunto de documentos resposta A. Seja |A| o número de documentos neste conjunto. Além disso, seja |Ra| o número de documentos na intersecção dos conjuntos R e A. A Figura 42 ilustra este conjunto.



**Figura 42** – Precisão e Revocação para um dado exemplo de pedido a informação.

As medidas de precisão e revocação são definidas como:

Revocação é a fração dos documentos relevantes (o conjunto R) que foi recuperado isto é,

$$Re vocação = \frac{|Ra|}{|R|}$$

• **Precisão** é a fração dos documentos recuperados (o conjunto A) que é relevante isto é,

$$\Pr{ecis\tilde{a}o} = \frac{|Ra|}{|A|}$$

Precisão e Revocação, como definidos acima, assumem que todos os documentos do conjunto resposta A foram examinados (ou vistos). Porém, geralmente não são apresentados ao usuário todos os documentos do conjunto resposta A instantaneamente. Ao invés disso, os documentos em A são primeiro ordenados de acordo com o grau de relevância (isto é, é gerada uma classificação). O usuário então examina esta lista classificada começando do documento no topo. Nesta situação, as medidas de precisão e revocação variam enquanto o usuário procede com a análise do conjunto resposta A. Assim, uma avaliação apropriada requer a plotagem de uma curva de precisão versus revocação.

Como antes, considere uma coleção de referência e seu conjunto de pedidos de informação. Vamos focalizar em um dado pedido de informação para o qual uma consulta q é formulada. Assuma que um conjunto Rq contendo os documentos relevantes para q foi definido. Sem perda de generalidade, assuma além disso que o conjunto Rq é composto dos seguintes documentos:

$$Rq = \left\{ d_3, d_5, d_9, d_{25}, d_{39}, d_{44}, d_{56}, d_{71}, d_{89}, d_{123} \right\}$$

Assim, de acordo com um grupo de especialistas, há dez documentos que são relevantes para a consulta q.

Considere agora um novo algoritmo de recuperação que acabou de ser projetado. Assuma que este algoritmo retorna, para a consulta q, uma classificação dos documentos no conjunto resposta como:

1. $d_{123} \bullet$	6. <i>d</i> <sub>9</sub> ●	11. <i>d</i> <sub>38</sub>
2. <i>d</i> <sub>84</sub>	7. $d_{511}$	12. $d_{48}$
3. $d_{56}$ •	8. <i>d</i> <sub>129</sub>	13. $d_{250}$

4.  $d_6$ 9.  $d_{187}$ 14.  $d_{113}$ 5.  $d_8$ 10.  $d_{25}$ 15.  $d_3$ 

Os documentos que são relevantes à consulta *q* são marcados com o ícone ● depois do número do documento. Examinando esta classificação, começando do topo do documento, observa-se os seguintes pontos. Primeiro, o documento *d*<sub>123</sub> que é classificado como número 1 é relevante. Além disso, este documento corresponde a 10% de todos os documentos relevantes no conjunto *Rq*. Assim, pode-se dizer que se tem uma precisão de 100% em 10% de revocação. Segundo, o documento *d*<sub>56</sub> que é classificado como número 3 é o próximo documento relevante. Neste ponto, pode-se dizer que se tem uma precisão aproximada de 66% (dois de três documentos são relevantes) em 20% de revocação (dois de dez documentos relevantes foram vistos). Terceiro, procedendo com a análise da classificação gerada pode-se plotar uma curva de precisão versus revocação como ilustrado na Figura 43. A precisão em níveis de revocação maiores que 50% cai para 0 porque nem todos os documentos relevantes foram recuperados. Esta curva de precisão versus revocação é geralmente baseada em 11 (em vez de 10) níveis de revocação padrões que são 0%, 10%, 20%,..., 100%. Para o nível de revocação 0%, a precisão é obtida por um procedimento de interpolação como detalhada a seguir.

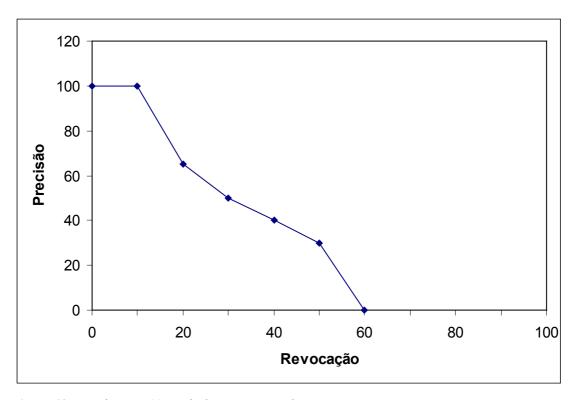


Figura 43 – Precisão em 11 níveis de revocação padrões.

No exemplo, os cálculos de precisão e revocação são dados para uma única consulta. Porém, geralmente, os algoritmos de recuperação são avaliados executando-os para consultas distintas. Neste caso, para cada consulta é gerada uma curva distinta de precisão versus revocação. Para avaliar a performance na recuperação de um algoritmo sobre todas as consultas testes, relacionam-se os cálculos de precisão a cada nível de revocação como:

$$\overline{P}(r) = \sum_{i=1}^{N_q} \frac{P_i(r)}{N_q}$$

onde  $\overline{P}(r)$  é a precisão média no nível de revocação r, Nq é o número de consultas usadas, e  $P_i(r)$  é a precisão no nível de revocação r para a i-ésima consulta.

Uma vez que os níveis de revocação para cada consulta podem ser distintos dos 11 níveis padrões de revocação, frequentemente é necessário um processo de interpolação. Por exemplo,

considere novamente o conjunto de 15 documentos classificados anteriormente. Assuma que o conjunto de documentos relevantes para a consulta *q* mudou e é dado agora por:

$$Rq = \{d_3, d_{56}, d_{129}\}$$

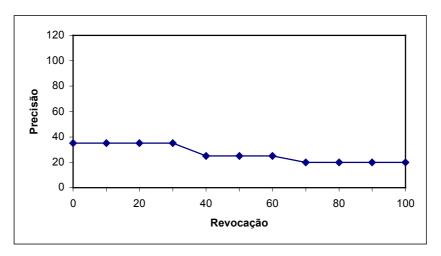
Neste caso, o primeiro documento relevante na classificação para a consulta q é  $d_{56}$  que fornece um nível de revocação de 33,3% (com precisão também igual a 33,3%) porque, neste ponto, um terço de todos os documentos relevantes já foram vistos. O segundo documento relevante é  $d_{129}$  que fornece um nível de revocação de 66,6% (com precisão igual a 25%). O terceiro documento relevante é  $d_3$  que fornece um nível de revocação de 100% (com precisão igual a 20%). Os cálculos de precisão nos 11 níveis padrões de revocação são interpolados como se segue.

Seja  $r_j$ ,  $j \in \{0,1,2,...,10\}$ , uma referência para o j-ésimo nível padrão de revocação (isto é,  $r_5$  é uma referência para o nível de revocação de 50%). Então,

$$P(r_i) = m \acute{a} x_{r_i \le r \le r_i + 1} P(r)$$

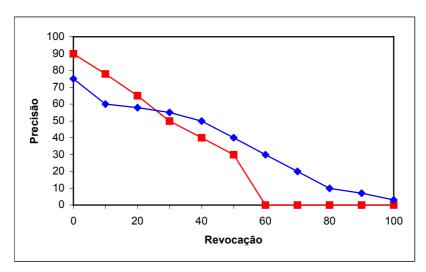
a qual indica que a precisão interpolada no j-ésimo nível padrão de revocação é a máxima precisão conhecida em qualquer nível de revocação entre o j-ésimo nível de revocação e o (j+1)-ésimo nível de revocação.

No último exemplo, esta regra de interpolação abrange os cálculos de precisão e de revocação ilustrados na Figura 44. Nos níveis de revocação de 0%, 10%, 20%, e 30%, a precisão interpolada é igual a 33,3% (que é a precisão conhecida no nível de revocação 33,3%). Nos níveis de revocação de 40%, 50% e 60%, a precisão interpolada é de 25% (que é a precisão no nível de revocação de 66,6%). Nos níveis de revocação de 70%, 80%, 90% e 100% a precisão interpolada é de 20% (que é a precisão no nível de revocação 100%).



**Figura 44** - Precisão interpolada nos 11 níveis padrões de revocação relativa a  $R_q = \{d_3, d_{56}, d_{129}\}$ .

A curva de precisão versus revocação que resulta da média dos resultados para várias consultas é geralmente referida como cálculos de precisão versus revocação. Tais cálculos da média são geralmente utilizados para comparar a performance na recuperação de algoritmos distintos. Por exemplo, pode-se comparar a performance na recuperação de um algoritmo recentemente proposto com a performance na recuperação dos modelos clássicos de espaços vetoriais. A Figura 45 ilustra a cálculos da média de precisão versus revocação para dois algoritmos distintos. Neste caso, um algoritmo tem precisão maior nos níveis mais baixos de revocação enquanto o segundo algoritmo é superior nos níveis mais altos de revocação.



**Figura 45 -** Cálculos da média de precisão versus revocação para dois algoritmos distintos de recuperação.

Uma abordagem adicional é computar a precisão média em valores dados de corte limites de documento. Por exemplo, pode-se calcular a precisão média quando 5, 10, 15, 20, 30, 50 ou 100 documentos relevantes foram vistos. O procedimento é análogo ao cálculo da precisão médio nos 11 níveis padrões de revocação mas fornece informação adicional sobre a performance na recuperação do algoritmo de classificação.

Os cálculos da média de precisão versus revocação são uma estratégia padrão de avaliação para sistemas de recuperação de informação e são usados extensamente na literatura de recuperação de informação. Eles são úteis porque permitem avaliar quantitativamente tanto a qualidade de todo o conjunto de resposta e a abrangência (*breadth*) do algoritmo de recuperação. Além disso, eles são simples, intuitivos e podem ser combinados em uma única curva. Porém, os cálculos de precisão versus revocação também possui desvantagens e seu uso indiscriminado (*widespread*) tem sido criticado na literatura.

### I.1.1 Reduções de Valores Únicos

Cálculos da média de precisão versus revocação são úteis para comparar a performance na recuperação de algoritmos distintos sobre um conjunto de consultas exemplos. Porém, há situações nas quais poderia se desejar comparar a performance na recuperação de algoritmos para consultas individuais. Primeiro, porque o cálculo da precisão sobre muitas consultas poderia camuflar importantes anomalias nos algoritmos de recuperação sob estudo. Segundo, porque quando se comparam dois algoritmos, poderia se estar interessado em investigar se um deles supera o outro em performance para cada consulta de um dado conjunto de consultas exemplos (note que este fato pode ser facilmente escondido por um cálculo da precisão média). Nestas situações, um valor único de precisão (para cada consulta) pode ser utilizado. Este valor único deveria ser interpretado como uma abstração da curva de precisão versus revocação correspondente. Geralmente, esta abstração de valor único é considerada como a precisão em um nível específico de revocação. Por exemplo, poderia se calcular a precisão quando se observa o primeiro documento relevante e considerar esta precisão como a abstração do valor único. Claro que esta não é uma boa aproximação. Estratégias mais interessantes podem ser adotadas:

Precisão média em documentos relevantes analisados: a idéia é gerar uma abstração de valor único da classificação através do cálculo da precisão média obtida depois que cada novo documento relevante é observado (na classificação). Por exemplo, considerando o exemplo na Figura 43, os cálculos de precisão, depois de cada novo documento ser observado, são 1; 0,66;. 0,5; 0,4 e 0,3. Assim, a precisão média dos documentos relevantes observados é dada por (1+0,66+0,5+0,4+0,3)/5 ou 0,57. Os sistemas de medida favorável (*mesure favors sytems*:) recuperam documentos relevantes rapidamente (isto é, no começo da classificação). Pode acontecer de um algoritmo apresentar uma boa precisão média nos documentos relevantes observados mas ter uma performance pobre em termos de revocação global.

<u>Precisão-R</u>: A idéia é gerar uma abstração de valor único da classificação calculando a precisão na R-ésima posição na classificação, onde R é o número total de documentos relevantes para a consulta corrente (isto é, o número de documentos no conjunto  $R_q$ ). Por exemplo, considere os exemplos nas Figura 43 e Figura 44. O valor da precisão-R é 0,4 para o primeiro exemplo (porque R=10 e há quatro documentos relevantes entre os primeiros dez documentos na classificação) e 0,33 para o segundo exemplo (porque R=3 e há um documento relevante entre os primeiros três documentos na classificação). O cálculo da precisão-R é um parâmetro útil para observar o comportamento de um algoritmo para cada consulta individual em um experimento. Além disso, pode-se também computar uma precisão-R sobre todas as consultas. Porém, utilizando um único número para abstrair o comportamento completo de um algoritmo de recuperação sobre muitas consultas poderia ser bastante impreciso.

<u>Histogramas de Precisão</u>: As medidas de precisão-R para muitas consultas podem ser utilizadas para comparar o histórico de recuperação de dois algoritmos. Sejam  $RP_A(i)$  e  $RP_B(i)$  os valores de precisão-R dos algoritmos de recuperação A e B para a i-ésima consulta. Defina, por exemplo, a diferença

$$RP_{A/B}(i) = RP_A(i) - RP_B(i)$$

Um valor de  $RP_{A/B}(i)$  igual a 0 indica que ambos os algoritmos tem performance equivalente (em termos de precisão-R) para a i-ésima consulta. Um valor positivo de  $RP_{A/B}(i)$  indica uma melhor performance na recuperação pelo algoritmo A (para a i-ésima consulta) enquanto um valor negativo indica uma melhor performance na recuperação pelo algoritmo B. A Figura 46 ilustra os valores de  $RP_{A/B}(i)$  (rotulada Precisão-R A/B) para dois algoritmos hipotéticos de recuperação sobre dez consultas exemplos. O algoritmo A é superior em oito consultas enquanto o algoritmo B atua melhor em outras duas consultas. (numeradas por 4 e 5). Este tipo de gráfico de barra é chamado de histograma de precisão e permite comparar rapidamente o histórico de performance na recuperação de dois algoritmos através de inspeção visual.

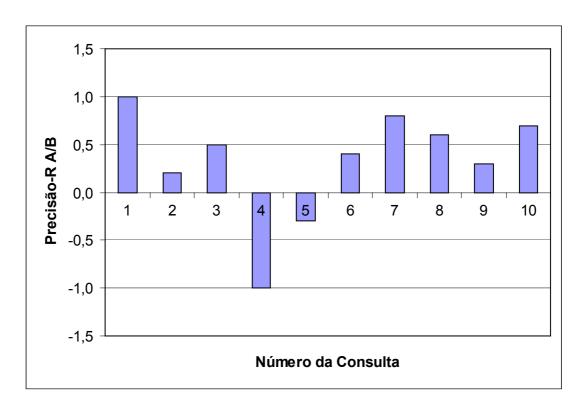


Figura 46 - Um histograma de precisão para dez consultas hipotéticas.

Estatísticas da Tabela de Abstração: cálculos de valor único podem também ser armazenadas em uma tabela para fornecer uma abstração estatística com relação ao conjunto de todas as consultas em uma tarefa de recuperação. Por exemplo, estas

estatísticas da tabela de abstração poderiam incluir: o número de consultas utilizadas na tarefa, o número total de documentos recuperados por todas as consultas, o número total de documentos relevantes que foram eficientemente recuperadas quando todas as consultas são consideradas, o número total de documentos relevantes que poderiam ter sido recuperados por todas as consultas, etc.

### I.1.2. Adequação de Medidas de Precisão e Revocação

Precisão e revocação têm sido utilizadas largamente para calcular a performance na recuperação de algoritmos de recuperação. Porém, uma reflexão cuidadosa adicional revela problemas com estas duas medidas [Raghavan\_1989B] [Raghavan\_1989A] [Tague-Sutcliffe\_1992]. Primeiro, a estimativa adequada de revocação máximo para uma consulta requer conhecimento detalhado de todos os documentos na coleção. Em grandes coleções, tal conhecimento não está disponível implicando que a revocação não pode ser estimada precisamente. Segundo, precisão e revocação são cálculos relacionados capturando diferentes aspectos do conjunto de documentos recuperados. Em muitas situações, o uso de uma única medida combinando precisão e revocação poderia ser mais apropriado. Terceiro, precisão e revocação medem a eficiência sobre um conjunto de consultas processadas em modo batch. Porém, com sistemas modernos, a interatividade (e não o processamento em *batch*) é o aspecto chave do processo de recuperação. Assim, medidas que quantificam o conteúdo informativo do processo de recuperação poderiam ser mais apropriadas. Quarto, precisão e revocação são fáceis de definir quando se força uma ordenação linear dos documentos recuperados. Para sistemas que requerem uma ordenação fraca porém, precisão e revocação podem ser inadequadas.

#### I.2. Medidas Alternativas

Existem, ainda, outras medidas propostas para avaliar performance na recuperação ao longo da literatura, tais como:

 $\underline{\textit{Média Harmônica}}$ : é uma medida única que combina precisão e revocação. Dada uma média harmônica F, esta pode ser representada por

$$F(j) = \frac{2}{\frac{1}{r(j)} + \frac{1}{P(j)}}$$

onde r(j) é a revocação para o j-ésimo documento na classificação, P(j) é a precisão para o j-ésimo documento na classificação, e F(j) é a média harmônica de r(j) e P(j) (assim, relativo ao j-ésimo documento na classificação). A função F assume valores no intervalo [0,1]. Vale 0 quando nenhum documento relevante for recuperado e vale 1 quando todos os documentos classificados forem relevantes. Além disso, a média harmônica F assume um valor alto somente quando ambos precisão e revocação são altos. Portanto, a especificação do valor máximo para F pode ser interpretada como uma tentativa para achar um equilíbrio entre precisão e revocação.

<u>Medida E</u>: combina também precisão e revocação e é chamada de medida de avaliação E. A idéia é permitir que o usuário especifique se está mais interessado em revocação ou em precisão. A medida E é definida como

$$E(j) = 1 - \frac{1 + b^2}{\frac{b^2}{r(j)} + \frac{1}{P(j)}}$$

onde r(j) é a revocação para o j-ésimo documento na classificação, P(j) é a precisão para o j-ésimo documento na classificação, E(j) é a medida de avaliação E relativa a r(j) e P(j), e b é um parâmetro especificado pelo usuário que reflete a importância relativa de revocação e precisão. Para b=1, a medida E(j) funciona como o complemento da média harmônica F(j). Valores de b maiores que 1 indicam que o usuário está mais interessado em precisão que em revocação enquanto que valores de b menores que 1 indicam que o usuário está mais interessado em revocação que em precisão.

<u>Medidas Orientadas a Usuário</u>: Precisão e Revocação são baseadas na proposição de que o conjunto de documentos relevantes para uma consulta é o mesmo, independente do usuário. Porém, usuários diferentes podem ter uma interpretação diferente de qual documento é relevante e qual não é. Para resolver este problema, medidas orientadas ao usuário foram propostas tais como taxa de cobertura, taxa de originalidade, revocação relativo, e esforço de revocação.

Como anteriormente, considere uma coleção de referência, uma solicitação de informação exemplar I, e uma estratégia de recuperação a ser avaliada. Seja R o conjunto de documentos relevantes para I e A o conjunto resposta recuperado. Também, seja U o subconjunto de R que é conhecido pelo usuário. O número de documentos em U é |U|. A intersecção dos conjuntos A e U abrange os documentos recuperados e conhecidos pelo usuário como relevantes. Seja |Rk| o número de documentos neste conjunto. Além disso, seja |Ru| o número de documentos relevantes, recuperados e previamente desconhecidos do usuário. A Figura 47 ilustra a situação. A taxa de cobertura é definida como a fração dos documentos relevantes conhecidos (pelo usuário) que foram recuperados, isto é

$$cobertura = \frac{|Rk|}{|U|}$$

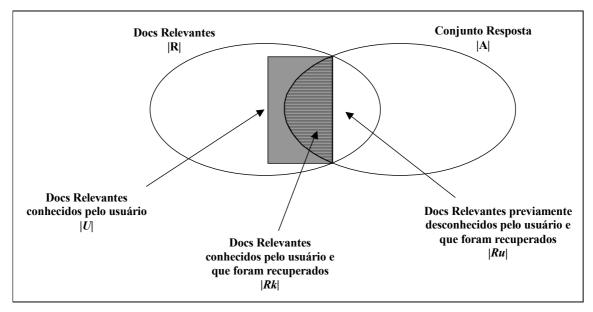


Figura 47 - Taxas de cobertura e de originalidade para um dado pedido a informação.

A taxa de originalidade é definida como a fração dos documentos relevantes recuperados e desconhecidos pelo usuário, isto é

$$originalidade = \frac{|Ru|}{|Ru| + |Rk|}$$

Uma alta taxa de cobertura indica que o sistema está encontrando a maioria dos documentos relevantes que o usuário espera. Uma alta taxa de originalidade indica que o sistema está revelando (ao usuário) muitos documentos relevantes que eram previamente desconhecidos.

A revocação relativa é dada pela taxa entre o número de documentos relevantes encontrados pelo sistema e o número de documentos relevantes que o usuário espera encontrar. Neste caso, quando o usuário encontra tantos documentos relevantes quanto ele espera, ele pára a busca e a revocação relativa é igual a 1. O esforço de revocação é dado pela taxa entre o número de documentos relevantes que o usuário espera encontrar e o número de documentos examinados na tentativa de encontrar os documentos relevantes esperados.

## Anexo II

### Funções de Distância

Funções de distância são utilizadas em várias áreas, incluindo aprendizagem baseada em instância, redes neurais, estatísticas, reconhecimento de padrões e psicologia cognitiva. Muitos sistemas inteligentes dependem da eficiência de uma função para cálculo de distância entre dois vetores. Uma variedade de funções de distância está disponível para tais usos, incluindo as métricas de distância Minkowsky, Mahalanobis, Camberra, Chebychev, Quadrática, Correlação e Chi-quadrado. As funções mais comuns são ilustradas na Figura 48 [Wilson 1997].

Embora tenha sido propostas muitas funções de distância a mais utilizada é a distância Euclideana, que é ilustrada na Figura 48 e onde x e y são dois vetores de entrada (um sendo vetor de referência e o outro um vetor a ser classificado) e m é o número de variáveis de entrada (atributos) na aplicação. A raiz quadrada não é computada freqüentemente, porque a(s) instância(s) mais próximas ainda serão as mais próximas a despeito do cálculo da raiz quadrada [Wilson 1997].

Uma função alternativa, a função de distância Manhattan, requer menos tempo computacional [Wilson\_1997].

As funções de distância Euclideana e Manhattan são equivalentes a função de distância Minkowskian, com *r*=2 e *r*=1 respectivamente [Wilson 1997].

Um ponto fraco da função de distância Euclideana é que se um dos atributos de entrada tem uma faixa relativamente larga, então isto pode sobrepujar os outros atributos. Por exemplo, se uma aplicação tem apenas dois atributos, A e B, e A pode ter valores variando de 1 a 1000, e B tem valores variando apenas de 1 a 10, então a influência de B sobre a função de distância será superada pela influência de A. Por isso, as distâncias geralmente são normalizadas dividindo-se a distância para cada atributo pela faixa (isto é, máximo-mínimo) daquele atributo, tal que a distância para cada atributo esteja na faixa aproximada de 0 a 1. A fim de evitar os elementos de

exceção, é também comum dividir a distância pelo desvio padrão invés da faixa. É também possível mapear qualquer valor fora desta faixa para o valor mínimo ou máximo para evitar valores normalizados fora da faixa de 0 a 1. Conhecimento de domínio pode ser utilizado para decidir qual método é mais apropriado [Wilson\_1997].

Relacionada à idéia de normalização estão os esquemas que utilizam pesos de atributos. Muitos sistemas inteligentes que usam funções de distância incorporam vários esquemas de pesos em seus cálculos de distância [Wilson\_1997].

Nenhuma das funções apresentadas na Figura 48, incluindo a distância Euclideana, tratam apropriadamente atributos de entrada não-contínuos.

Um atributo pode ser lienar ou nominal, e um atributo linear pode ser contínuo ou discreto. Um atributo contínuo (ou continuamente valorado) utiliza valores reais, tal como a massa de um planeta ou a velocidade de um objeto. Um atributo linear discreto (ou inteiro) pode ter somente um conjunto discreto de valores lineares, tal como o número de filhos. Um atributo nominal (ou simbólico) é um atributo discreto cujos valores não estão necessariamente em qualquer ordem linear. Por exemplo, uma variável representando cor poderia ter valores como vermelho, azul, marrom, branco e preto, que poderiam ser representadas por inteiros de 1 a 6, respectivamente. Utilizando medidas de distância lineares como a Euclideana ou a Manhattan sobre tais valores faz pouco sentido neste caso.

Em [Wilson\_1997] são apresentadas algumas funções que tratam tanto atributos contínuos como nominais.

**Minkowsky:**  $D(x,y) = \left(\sum_{i=1}^{m} |x_i - y_i|^r\right)^{\frac{1}{r}}$ 

**Euclidean:**  $D(x,y) = \sqrt{\sum_{i=1}^{m} (x_i - y_i)^2}$ 

**Manhattan**/ *city-block*:  $D(x, y) = \sum_{i=1}^{m} |x_i - y_i|$ 

Camberra:  $D(x,y) = \sum_{i=1}^{m} \frac{|x_i - y_i|}{|x_i + y_i|}$ 

**Chebychev:**  $D(x,y) = \max_{i=1}^{m} |x_i - y_i|$ 

Quadratic:  $D(x,y) = (x-y)^T Q(x-y) = \sum_{i=1}^m \left( \sum_{j=1}^m (x_i - y_j) q_{qi} \right) (x_j - y_j)$ 

Q é uma matriz de tamanho m x m finita, positiva e específica ao problema.

**Mahalanobis:**  $D(x, y) = \left[\det V\right]^{\frac{1}{m}} (x - y)^T V^{-1} (x - y)$  V é matriz de covariância de  $A_1...A_m$ , e  $A_j$  é o vetor de valores para o atributo j que aparece nas instâncias do conjunto de treinamento 1...n.

Correlação:  $D(x, y) = \frac{\sum_{i=1}^{m} (x_i - \bar{x}_i)(y_i - \bar{y}_i)}{\sqrt{\sum_{i=1}^{m} (x_i - \bar{x}_i)^2 \sum_{i=1}^{m} (y_i - \bar{y}_i)^2}}$ 

 $\overline{x}_i = \overline{y}_i$  é o valor médio para o atributo i que aparece no conjunto de treinamento.

Chi-quadrado:  $D(x, y) = \sum_{i=1}^{m} \frac{1}{soma_i} \left( \frac{x_i}{tam_x} - \frac{y_i}{tam_y} \right)^2$ 

soma, é a soma de todos os valores para o atributo i que aparece no conjunto de treinamento, e  $tam_x$  é a soma de todos os valores no vetor x.

Correlação de Posição de Kendall:

 $D(x, y) = 1 - \frac{2}{m(n-1)} \sum_{i=1}^{m} \sum_{j=1}^{i-1} sinal(x_i - x_j) sinal(y_i - y_j)$ sinal(x)=-1, 0 ou 1 se x<0, x=0, ou x>0, respectivemente.

Figura 48 - Equações de funções de distância mais comuns (x e y são vetores de m valores de atributo) [Wilson 1997].

# Apêndice A

### Desenvolvimento de aplicações para a WWW

Com a evolução da Internet e dos navegadores, percebeu-se que ao invés de desenvolver várias versões de um mesmo aplicativo para Unix, Macintosh, OS/2, Windows 95 e NT, seria muito mais fácil encapsular a parte cliente em navegadores e deixá-los se preocuparem com o suporte a múltiplos ambientes. Desta forma surgiram as Intranets [Yourdon 1996].

Uma Intranet corresponde a um ou mais *web-sites* que pertencem a uma organização e podem ser acessados somente por membros de uma organização. Ou seja, uma intranet pode ter um único servidor *Web* ou vários servidores *Web*, cada um em um departamento da empresa. Muitas empresas usam uma intranet, ao invés da internet, para oferecer aos seus empregados facilidades de acesso às informações corporativas, de forma a controlar o acesso externo a elas. Em uma intranet, o acesso a estas informações pode ser realizado através de navegadores [Evans 1996] [Evans 1997].

Os navegadores comunicam-se através de uma rede, inclusive através da Internet, com os servidores *Web* utilizando um protocolo de comunicação chamado HTTP (*HyperText Transfer Protocol*). Este protocolo é utilizado para trocas de pacotes de informações entre computadores conectados à Internet através de protocolos TCP/IP (*Transfer Control Protocol/Internet Protocol*). Em uma transação HTTP, o sistema abre uma conexão entre um navegador cliente e um servidor Web, submete uma solicitação ao servidor *Web*, retorna a resposta do servidor ao cliente e então encerra a conexão. Nenhuma informação sobre a solicitação é mantida após encerrada a conexão.

Tecnicamente, ao se acessar um site, por exemplo http://www.icmc.sc.usp.br, um mecanismo conhecido por DNS (*Domain Name Service*) converte o *site* em um endereço IP, que é um identificador de máquina para a Internet. Ao se conectar ao endereço IP, o navegador então

passa a localização e o nome do documento para o servidor como uma solicitação. O servidor responde enviando o documento solicitado ao navegador, que o interpreta e o exibe na tela [Rodley 1997].

Um fator muito crucial no projeto de sistemas distribuídos e principalmente em intranets que possuem ao menos uma conexão com a Internet é a segurança.

Geralmente os desenvolvedores de sistemas baseados no paradigma cliente/servidor assumem que senhas e mecanismos de proteção a arquivos sejam suficientes para garantir a segurança de suas aplicações. Na Internet, uma atitude casual em relação à segurança pode levar a graves conseqüências. As ferramentas de desenvolvimento devem ajudar na criação de recursos para acesso seguro a funcionalidades e dados e também para transmissão segura de dados confidenciais através da Internet [Yourdon 1996].

Em geral há quatro aspectos de segurança a serem considerados [Hamilton 1996]:

- Policiamento: É crucial a definição de uma boa estratégia de policiamento dos dados que trafegam pela intranet/internet.
- Privacidade: A única solução para manter a privacidade dos dados que trafegam por uma Intranet é encriptar os dados.
- Autenticação: É necessário que haja a verificação da identidade das partes envolvidas na comunicação de dados através da Internet.
- Integridade: É preciso assegurar que as aplicações acessadas através da Internet executarão exatamente o que se espera delas, sem efeitos subversivos.

Em um modelo ideal, privacidade e autenticação são de responsabilidade da camada de rede ao invés da camada da aplicação. Adicionando estas características à camada da aplicação pode-se causar problemas de usabilidade quando são utilizados mecanismos de encriptação e autenticação. A linguagem de programação Java concentra-se em manter a integridade na camada da aplicação [Hamilton 1996].

Em aplicações típicas baseadas em arquitetura cliente/servidor, ao solicitar e validar a senha de um usuário, determina-se quais funcionalidades ou dados o usuário pode ter acesso.

Desta forma a aplicação precisará interagir com o navegador para encriptar/descriptar transmissões entre a estação cliente e o servidor, e deverá informar ao usuário qualquer situação que exponha informação confidencial. É importante também integrar pacotes que fornecem altos níveis de segurança contra ataques deliberados por partes de *hackers* que tentem enviar dados falsos ou tentem substituir o applet original por outros que provoquem um ataque nocivo ao servidor.

Para auxiliar na segurança da rede existem dispositivos, chamados *firewalls*, que são instalados entre a Intranet e a Internet com a finalidade de limitar o acesso aos dados privados. Um *firewall* constitui-se de uma ou mais máquinas que executam software de proteção que ajuda a manter a rede interna segura, não permitindo a entrada de dados gerados exteriormente. Todo o tráfego que entra e sai da Internet passa por esta máquina. A maior parte das técnicas envolve a inspeção da origem, destino e conteúdo de cada pacote de informações e autoriza ou não a saída ou entrada do pacote baseado nestes parâmetros. O servidor Web geralmente é instalado na máquina que contém o *firewall* e por conseguinte não é protegido por ele. Se houver um servidor de banco de dados então é aconselhável que haja um *firewall* entre a rede interna e a Internet [Evans\_1996] [Evans\_1997] [Rodley\_1997]. Na Figura 49 é mostrada uma configuração para uma Intranet.

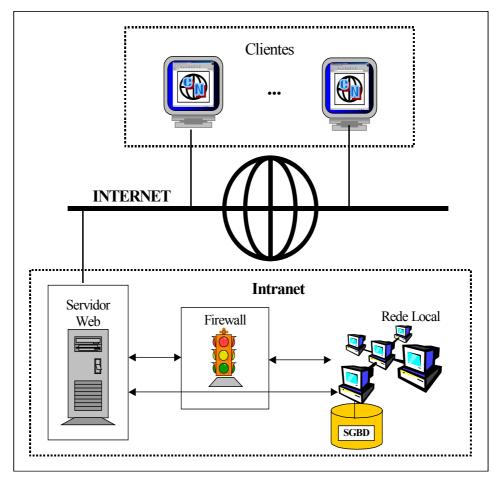
Um modo simples de encarar o desenvolvimento para a Internet é em termos de relações cliente/servidor. A maior parte das interações na Internet/Intranet pode ser feita através de solicitações e respostas.

A arquitetura mais simples para aplicações voltadas para a Internet envolve a simples recuperação de documentos HTML da WWW. A linguagem HTML (*Hypertext Markup Language*) permite que os documentos enviados pelo servidor sejam exibidos em um navegador no cliente [Yourdon\_1996]. A Figura 50 ilustra o acesso a páginas HTML

O formato HTML é um método muito limitado no controle das informações que fluem pela Web, pois exibe informações pré-determinadas e estáticas, ou seja, não recupera nada de banco de dados e nem solicita entrada de dados ao usuário. Para poder exibir dados atualizados, utilizando este formato, seria necessário a edição freqüente do arquivo HTML.

Uma arquitetura mais dinâmica e comum na Internet envolve formulários que solicitam entrada de dados, exibem informações e fornecem uma combinação familiar de componentes de

interface tais como campos de texto, botões de opções, barras de rolagem e listas de escolhas. Neste tipo de arquitetura, após o preenchimento dos dados requisitados estes são submetidos ao servidor através do acionamento de um botão na página HTML. Este tipo de procedimento leva à execução de um comando CGI (*Common Gateway Interface*) que chama um programa que processa os dados fornecidos[Yourdon\_1996]. A Figura 51 ilustra a comunicação através de CGI.



**Figura 49**– Uma rede virtual privada – INTRANET. Nesta configuração, embora o servidor web esteja desprotegido, os dados da intranet podem ser divulgados na internet de modo seguro. Aqui o firewall é configurado para ignorar o tráfigo da base dados, deixando a segurança para o sistema de gerenciamento de banco de dados.

O CGI é um padrão de comunicação com regras bem definidas para a criação de aplicações cooperativas. Ao contrário do formato HTML, um programa ou script CGI pode exibir informações dinâmicas obtidas de recursos disponíveis no servidor Web, por exemplo de um banco de dados.

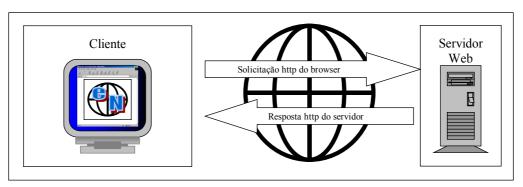
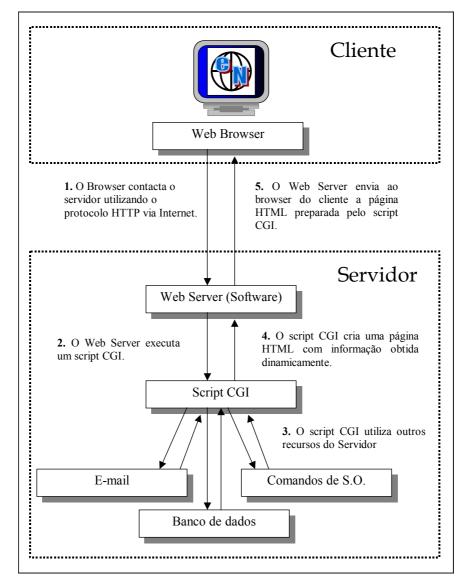


Figura 50 - Acesso a páginas HTML.



**Figura 51** - Como funciona o acesso a programas CGI (Ref. <a href="http://wdvl.internet.com/Authoring/Scripting/Tutorial">http://wdvl.internet.com/Authoring/Scripting/Tutorial</a>)

Os programas CGI podem ser escritos em diversas linguagens de programação: C, C++, Java, Perl, Visual Basic etc; e também podem trabalhar em diferentes tipos de sistemas: Mac, Windows, OS2, Unix etc. A diferença entre Java e outras linguagens de programação é que Java foi desenvolvido para redes de computadores, como a Internet. Programas CGI escritos em Java ou Perl são independentes de plataforma, ou seja, o mesmo programa CGI pode ser levado de uma plataforma para outra não tendo a necessidade de revisão de código.

Os programas CGI são independentes dos documentos HTML e a maior parte do processamento geralmente é feito no servidor. Existem outros tipos de scripts que podem ser integrados a páginas HTML e executar operações dinamicamente sem sobrecarregar o servidor, tais como JavaScript e VBScript. Neste tipo de arquitetura a maior parte do processamento é feito no cliente, liberando o servidor da sobrecarga de processamento. Por outro lado, este tipo de script depende da capacidade do navegador interpretar a linguagem utilizada.

### A.1. A Linguagem JAVA

Java é uma linguagem de programação orientada a objetos com uma sintaxe semelhante ao C e C++, embora mais simples. Por ser uma linguagem interpretada, todo o ciclo de vida para o desenvolvimento de software pode ser realizado dentro de um navegador.

As aplicações em Java são também mais robustas que aplicações correspondentes em C ou C++, porque um sistema runtime Java ou JVM (*Java Virtual Machine*), que geralmente acompanha os navegadores, gerencia toda a memória. As mesmas características que fornecem robustez também fornecem segurança, mesmo quando as aplicações trafegam pela Internet (*download*) pois o JVM possui mecanismos de segurança que protegem contra interferências indevidas. Aplicações que executam tarefas concorrentes são mais rápidas pois o Java possui suporte próprio para tratamento de concorrência [Hamilton\_1996] [Perry\_1996].

A principal vantagem em Java é que as aplicações são completamente portáteis. Uma vez que o código é escrito não há a necessidade de adaptá-lo ou mesmo recompilá-lo. Para um programa qualquer rodar em um computador, ele deve primeiro ser traduzido de uma linguagem como Pascal ou C++ para a língua nativa da máquina, o que é feito pelo compilador. Como geralmente os softwares já vêm compilados, podem ser criadas diferentes versões para

plataformas diferentes. Ao invés de produzir instruções específicas para máquinas distintas, o compilador Java produz *bytecode* independente de plataforma. O ambiente *runtime* Java, ou máquina virtual, traduz o *bytecode* em instruções específicas para a máquina em uso. A máquina virtual Java é instalada na máquina do usuário, ou como parte de um navegador ou como parte do sistema operacional [Hamilton\_1996][Perry\_1996].

A linguagem Java originou-se em 1990 com James Gosling, um desenvolvedor da Sun Microsystems, que fazia parte de um grupo de pesquisa para investigação de técnicas avançadas para desenvolvimento de software voltadas para uma grande variedade de dispositivos de rede e sistemas integrados. A meta deste grupo era simplificar o desenvolvimento de aplicações seguras, altamente robustas e de alta performance em múltiplas plataformas e em redes heterogêneas distribuídas [Hamilton 1996].

Como a maioria dos programas orientados a objetos, todas as variáveis e funcionalidades em um programa Java estão contidos dentro de objetos que por sua vez são definidos através de classes. Como C, Java contém tipos de dados numéricos padrões, porém independentes de hardware ou sistema. Em Java todas as variáveis são fortemente "tipadas" e não há ponteiros. Isto produz uma redução de erros de programação e um aumento de segurança [Hamilton\_1996]. As aplicações em Java são livres de vírus, pois não podem acessar a memória do sistema como os programas em C ou C++ [Perry 1996].

Até o surgimento da linguagem Java era difícil gerenciar as informações na Internet, construir e utilizar aplicações que acessassem estas informações e executar estas aplicações de forma eficiente [Perry\_1996]. Neste caso a integridade do programa é garantido em tempo de execução como mostrado na Figura 52.

A linguagem Java suporta as seguintes principais características [Perry 1996]:

- Multimídia interativa: Java fornece o controle para que os diversos tipos de mídia trabalhem em conjunto e interação em tempo real.
- Independência de Plataforma: O desenvolvedor pode manter somente uma fonte de um programa Java para diferentes plataformas.
- Muitas camadas de segurança: Java fornece verificação de código pós-compilação, restrições de acesso a arquivos e verificação de classes em tempo de execução.

 Distribuição e Instalação de Software: Java permite a distribuição de mensagens e arquivos na Internet através da utilização de uma biblioteca de rotinas para suporte a protocolos TCP/IP.

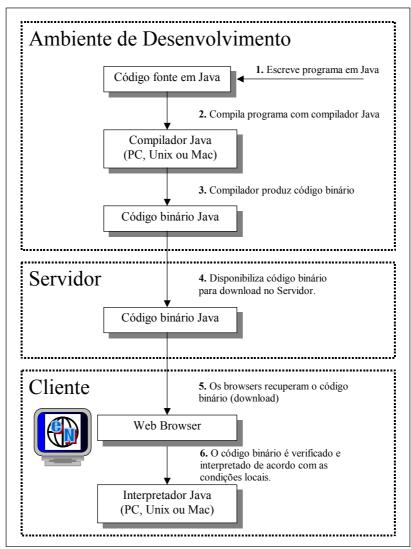


Figura 52 - Como funciona programas Java na WWW.

 $(Ref.\ \underline{http://wdvl.internet.com/Authoring/Scripting/Tutorial})$ 

Há dois tipos de aplicações Java: *stand-alone* e *applets*. As aplicações *stand-alone* são programas que não se integram a páginas Web enquanto que os *applets* são programas que podem ser integrados a elas. Pode-se utilizar ambas em diferentes tipos de plataformas que suportam Java [Perry 1996].

Para criar applets é preciso:

- 1. definir o layout da página *Web*: A maioria dos *applets* mostram informação ao usuário. É preciso definir uma área na página para o *applet*, como se faz com uma figura qualquer.
- 2. Escrever o código fonte para o *applet*. Usar um editor de texto, tal como o Windows NotePad ou o shareware PCGrasp.
- 3. Compilar o *applet* com o compilador Java e testá-lo na página *Web*, incorporando-o a um documento HTML.

#### A.2. Banco de dados na WWW

Todo gerenciador de banco de dados (*database engine*) é acompanhado por um conjunto de bibliotecas (*Dinamic Link Libraries* - DLLs) que permitem a conexão e o uso de aplicações. O acesso direto a estas bibliotecas é denominado *chamadas nativas*.

O uso de chamadas nativas por um programa requer um conjunto de operações específicas ao banco de dados utilizado além das fornecidas pelo compilador de linguagem. Assim, chamadas nativas para um banco de dados Oracle não podem ser utilizadas para um banco de dados Sybase.

Há uma alternativa para as chamadas nativas conhecida como ODBC (*Open Database Connectivity*), que é uma interface de programação de aplicação (API – *Application Programming Interface*) para acesso a banco de dados. A principal vantagem do ODBC sobre as chamadas nativas convencionais é a criação de programas independentes do banco de dados utilizado. O ODBC possibilita que um gerenciador de dispositivos ODBC controle a comunicação entre a aplicação e o gerenciador de banco de dados. Assim cada gerenciador de banco de dados registra o seu próprio dispositivo ODBC através deste gerenciador de dispositivos o qual, após receber uma chamada ODBC de uma aplicação, interpreta a solicitação traduzindo-as em chamadas nativas ao gerenciador de banco de dados utilizado [Rodley 1997].

Com o intuito de utilizar os recursos da linguagem Java, a Sun Microsystems especificou um padrão para acesso a banco de dados por programas implementados em Java chamado JDBC

(*Java Database Connectivity*). Como o ODBC, o JDBC é uma API que determina como os programas Java devem interagir com os bancos de dados. A principal diferença entre estes dois padrões é que o JDBC é inspirado em uma linguagem orientada a objetos, assim em vez de especificar um conjunto de funções, o JDBC administra um conjunto de objetos e interfaces Java. A Figura 53 mostra a comunicação de dados via JDBC.

Para o desenvolvimento de sistemas direcionados a Internet que envolvam um modelo de armazenamento baseado em banco de dados relacional, é muito mais fácil utilizar formulários HTML como um *front-end* do que portar a aplicação para múltiplas plataformas. Porém, uma abordagem convencional, utilizando CGI com chamadas nativas ao banco de dados, pode apresentar alguns problemas. O HTML não incorpora nenhum recurso para validação dos dados entrados. A validação de formulários pode ser feita somente no servidor. Isto significa que erros na entrada de dados não podem ser corrigidos até que o formulário seja processado pelo servidor Web, levando a interfaces não-interativas [Yourdon\_1996].

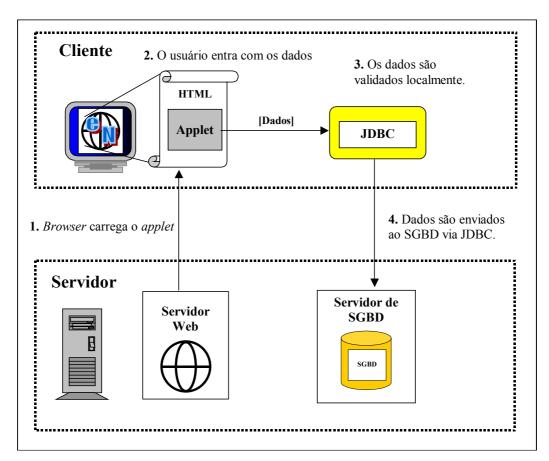


Figura 53 - Comunicação de dados via JDBC.

Outro problema neste tipo de abordagem é o consumo significativo, dependendo da complexidade do formulário e quantidade de usuários simultâneos, dos recursos do servidor para se executar a validação do formulário. Porém o problema mais sério é a propensão para gargalos no servidor [Yourdon 1996].

Uma API JDBC permite que uma aplicação se comunique diretamente com um sistema gerenciador de banco de dados (SGBD), eliminando a maioria dos problemas de um sistema convencional para a *Web*. Neste caso, o navegador carrega um *applet*, o usuário entra com os dados e o *applet* valida os campos preenchidos localmente antes que os dados sejam enviados para o SGBD via JDBC [Yourdon\_1996].

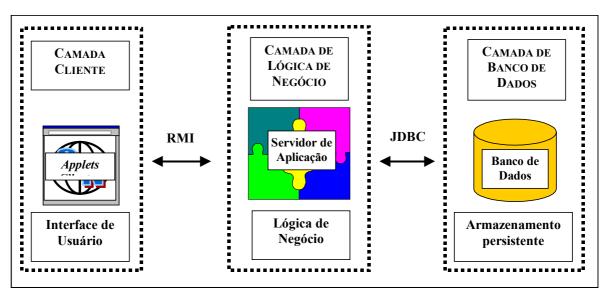
### A.3. RMI – Remote Method Invocation

Por ser uma linguagem de propósitos gerais, Java pode ser utilizada tanto em implementações para o cliente (applets) quanto para o servidor (servlets). Para tal abordagem, a Sun Microsystems definiu uma interface para a comunicação entre servidor e cliente baseados em Java chamada Remote Method Invocation (RMI) [Hamilton 1996].

RMI é um conjunto de classes Java, ferramentas de desenvolvimento e outras características que permitem a clientes Java executar métodos em objetos Java que estejam em diferentes executáveis Java [Evans\_1997].

Um esquema envolvendo o RMI é muito parecido com o do JDBC, como mostrado na Figura 53, exceto que o servidor de SGBD é substituído por um servidor Java e o JDBC pelo RMI. Este tipo de interface não substitui a necessidade de JDBC para comunicação com o SGBD, como é mostrado na Figura 54[Hamilton\_1996][Gupta\_1998].

É importante observar que o RMI não suporta objetos distribuídos independentes de linguagem. O software utilizado tanto no cliente quanto no servidor deve ser implementado em Java, executando dentro de JVMs.



**Figura 54 -** Arquitetura para desenvolvimento de aplicação Java em 3 camadas.

### A.4. Arquitetura do Gerenciador de Requisições de Objetos Comuns

A OMG (*Object Management Group*) é um consórcio industrial internacional que promove a teoria e a prática de desenvolvimento de software orientado a objeto. Sua meta é fornecer uma arquitetura comum, considerando plataformas de hardware e sistemas operacionais heterogênios, para a comunicação entre objetos de aplicação.

A OMG desenvolveu um modelo conceitual, conhecido como o modelo de objeto básico,

e uma arquitetura de referência, chamada OMA (*Object Management Architecture*) sobre a qual as aplicações podem ser construídas. A OMA/OMG tenta definir, em um alto nível de abstração, as várias facilidades necessárias para a computação distribuída e orientada a objetos. A OMG/OMA constitui-se por 5 componentes, conforme pode ser visto na Figura 55:

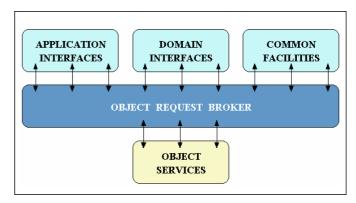


Figura 55 – Arquitetura de Gerenciamento de Objetos.

- Object Request Broker (ORB): é uma via de comunicação comum através do qual objetos distribuídos de software e seus clientes podem interagir. Usando um ORB, um objeto e seus clientes podem residir em um mesmo processo, ou em processos diferentes, o qual pode ser executado em diferentes máquinas conectadas em rede. O ORB adota uma tecnologia conhecida como Common Object Request Broker Architecture (CORBA) responsável pela especificação de uma estrutura para comunicação transparente entre objetos da aplicação [Evans\_1997] [Yang\_1997] [Vinoski 1997].
- Object Services: correspondem às interfaces independentes de domínio que são utilizadas por muitos programas de objetos distribuídos. Por exemplo, um serviço fornecendo a descoberta de outros seviços disponíveis é quase sempre necessário a despeito do domínio da aplicação. Dois exemplos de Object Services que satisfazem esta regra são: o Naming Service, que permite aos clientes encontrarem objetos baseando-se em nomes, e o Trading Service, que permite aos clientes encontrarem objetos baseando-se em suas propriedades. Há também especificações do Object Service para gerenciamento, segurança, transações, notificação de eventos entre outras [Yang\_1997].
- Common Facilities: Como as interfaces do Object Service, estas interfaces são também horizontalmente orientadas, mas diferente do Object Services elas são voltadas para as aplicações do usuário final. Um exemplo de uma facilidade é a DDCF (Distributed Document Component Facility), uma facilidade de documento componente baseada no OpenDoc. A DDCF permite a apresentação e troca de objetos baseando-se em um modelo de documento facilitando, por exemplo, a anexação de um objeto planilha a um documento de relatório [Yang\_1997].
- Domain Interfaces: Estas interfaces seguem regras semelhantes aos Object Services e
  às Common Facilities mas são voltadas para domínios específicos de aplicação. Por
  exemplo, uma das primeiras propostas apresentadas pelo OMG referentes às Domain
  Interfaces foram para os PDM (Product Data Management) para o domínio do

manufaturamento. Pretende-se também apresentar propostas para os domínios médicos, financeiros e das telecomunicações [Yang\_1997].

 Application Interfaces: Estas interfaces são desenvolvidas especialmente para uma dada aplicação. Por elas serem específicas à aplicação, e pelo OMG não desenvolver aplicações, mas somente especificações, estas interfaces não são padronizadas. Porém, se com o tempo surgirem certos serviços úteis de um domínio específico de aplicação, eles podem vir a se tornarem candidatos para futuras padronizações da OMG.

A especificação CORBA fornece uma visão muito abstrata de objetos. Ela somente é visível através de operações definidas por uma interface e executadas utilizando-se uma referência a objeto. A referência a objeto por si só é uma abstração, e está disponível aos clientes somente como um valor opaco, o qual pode ser passado como um parâmetro em uma operação, ou restringida a uma dada representação externa que pode ser armazenada e consequentemente transformada em uma referência a objeto.

A interface determina as operações que um cliente pode realizar utilizando a referência a objeto. O único modo de um cliente poder observar ou afetar o objeto é através de operações definidas para aquela interface. Os outros detalhes do objeto ficam escondidos [Yang 1997].

CORBA fornece uma linguagem própria e declarativa, chamada IDL (*Interface Definition Language*) que pode especificar as operações que um cliente pode solicitar de um objeto. O ORB fornece o software necessário para transmitir as solicitações a objetos feitas pelo cliente e as respostas fornecidas pelos objetos ao cliente. Na maioria dos casos, este software é automaticamente gerado como código fonte a partir da descrição de interface IDL através da ferramenta de compilação IDL de um produto ORB. Partes deste código fonte são compilados e "linkados" ao objeto e outras partes aos clientes. A IDL tem sido mapeada para várias linguagens tais como C, C++, Java e outras [Evans\_1997] [Baker\_1997].

A especifição CORBA define uma arquitetura de interfaces consistindo de componentes mostrados na Figura 56 e descritos a seguir:

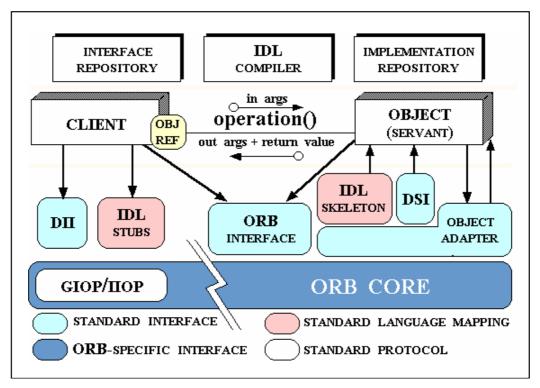


Figura 56 - Common Object Request Broker Architecture (CORBA).

- Object Implementation: define operações que implementam uma interface CORBA/IDL. As implementações podem ser escritas em várias linguagens incluindo C, C++, Java, Smalltalk e Ada.
- Client: Este é uma entidade de programa que executa uma operação sobre uma implementação de objeto. O acesso aos serviços de um objeto remoto deveriam ser transparentes ao solicitador. Idealmente, deveria ser tão simples quanto uma chamada a métodos em um objeto. Os demais componentes na Figura 56 ajudam a suportar este nível de transparência.
- Object Request Broker (ORB): O ORB fornece um mecanismo para comunicação transparente entre cliente e implementações de objeto. O ORB simplifica a programação distribuída escondendo do cliente os detalhes sobre a execução dos métodos. Desta forma, as solicitações do cliente parecem ser chamadas locais a procedimentos. Quando um cliente executa uma operação, o ORB é responsável por achar a implementação do objeto, ativá-lo

transparentemente se necessário, entregar o pedido ao objeto, e retornar qualquer resposta ao solicitador.

- ORB Interface: Um ORB é uma entidade lógica que pode ser implementada de vários modos. Para separar as aplicações dos detalhes de implementação, a especificação CORBA define uma interface abstrata para um ORB. Esta interface fornece várias funções auxiliares de convertem referências a objetos em cadeias de caracteres e vice-versa, cria listas de argumento para pedidos feitos através de uma interface dinâmica de execução (Dynamic Invocation Interface DII) descrita abaixo
- CORBA/IDL stubs e skeletons: os stubs e skeletons servem como uma cola entre as aplicações do servidor e do cliente, respectivamente, e o ORB. A transformação das definições CORBA IDL na linguagem destino de programação é automatizada por um compilador CORBA/IDL. O uso de um compilador reduz o potencial para inconsistências entre stubs clientes e skeletons servidores e aumenta as oportunidades para otimizações automática de compilador.
- Dynamic Invocation Interface (DII): esta interface permite que um cliente acesse diretamente os mecanismos inerentes de requisição fornecidos por um ORB. As aplicações usam o DII para fazer pedidos aos objetos sem a necessidade de stubs IDL específicos a interface para se conectarem. Ao contrário dos stubs IDL, a DII também permite aos clientes fazerem chamadas síncronas deferidas, que separam operações de enviar e receber, e chamadas de uma direção, ou seja só para envio.
- Dynamic Skeleton Interface (DSI): esta é a interface do servidor análoga a DII do cliente. A DSI permite que um ORB entregue requisições a uma implementação de objeto que não tenha conhecimento em tempo de compilação do tipo do objeto que está implementando. O cliente que faz a requisição não tem idéia se a implementação está usando os skeletons IDL de tipo específico ou está usando os skeletons dinâmicos.
- Object Adapter: auxilia o ORB com a entrega de requisições ao objeto e com a ativação dele.
   O mais importante é que um Object Adapter associa implementações do objeto com o ORB.
   Os Object Adapters podem ser especializados para fornecer suporte a certos estilos de

implementação de objeto, tal como *Object Adapters* relacionados a banco de dados orientado a objeto para persistência e *Object Adapters* relacionados a bibliotecas para objetos não-remotos).

Uma vez que o software CORBA implementa os detalhes de transmissão das solicitações e respostas a operações, os objetos e seus clientes podem interagir independente de sua localização, sistema operacional ou mecanismos de comunicação, principalmente através da WWW. Assim, as aplicações distribuídas podem incorporar componentes escritos em diferentes linguagens fontes e ser executadas em diferentes plataformas[Vinoski\_1997][Yang\_1997].

#### A.5. CORBA na WWW

CORBA também padroniza um conjunto de protocolos para interoperabilidade entre ORBs conhecido com GIOPs (General Inter-ORB Protocols). O protocolo principal deste conjunto é o IIOP (Internet inter-ORB Protocol).

Uma vez que a WWW é basicamente um conjunto de protocolos, parece natural estendêla. A inclusão de CORBA/IIOP neste conjunto forneceria um modo alternativo para integrar a WWW e CORBA.

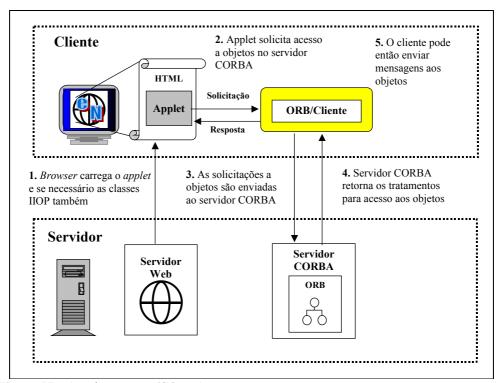


Figura 57 – Arquitetura Java/CORBA para a WWW.

Baseada em princípios de hipertexto, a WWW foi desenvolvida como uma tecnologia padrão para distribuição e apresentação de dados, permitindo que documentos sejam localizados em diferentes nós da rede.

As primeiras versões do CORBA não especificavam um protocolo de comunicação ORB interno. Isto é, cada vendedor ORB determinava como seu ORB se comunicaria internamente. Por esta razão, era impossível enviar mensagens de um ORB para um objeto conectado a outro processo servidor ORB de outro vendedor. Conseqüentemente, em CORBA 2.0 definiu-se o IIOP. O IIOP, como um protocolo padronizado, permite a comunicação entre diferentes produtos ORB.

Graças ao Java e aos ORBs/Java, o conceito inter-ORB foi levado à Internet. Atualmente, os navegadores podem agir como clientes que acessam objetos em um servidor via IIOP. O protocolo tradicional da WWW, o HTTP, deixa de ser um gargalo entre a comunicação cliente/servidor, como mostra a Figura 57.

As classes Java solicitadas ao servidor HTTP na realidade não são *applets* comuns mas ORBlets, que possibilitam ao navegador se comunicar com o servidor CORBA. No caso do navegador não possuir o arquivo Java IIOP, o servidor passa também as classes IIOP básicas durante o *download*. Estabelecida a conexão com o servidor CORBA, o cliente pode então trocar mensagens com objetos residentes nele, evitando o gargalo do HTTP.

As vantagens desta abordagem são:

- A comunicação cliente/servidor fica menos sobrecarregada;
- As funções do servidor podem possuir um valor de revocação;
- Os tipos de dados reais, não somente os textos, podem ser trocados.
- Os objetos do servidor podem ser implementados em qualquer linguagem, desde que haja um ORB compatível com CORBA2.0.

O IIOP permite ao navegador agir apropriadamente como um ambiente de aplicação. Algumas empresas, tais como a Netscape e a Oracle, reconheceram este potencial e incluiram os componentes ORB necessários em seus produtos baseados em arquitetura cliente/servidor.

A Microsoft trilhou um caminho parecido, mas com o uso de sua tecnologia proprietária ActiveX/DCOM (*Distributed Component Object Model*) ao invés de CORBA.

### A.6. CORBA e JAVA

A Sun Microsystems utiliza o JAVA para permitir o uso de sistemas CORBA por qualquer usuário que tenha um navegador, sem a necessidade da instalação de um módulo CORBA no cliente como é mostrado na Figura 57. Em vez de se ter um componente ORB/Cliente a Sun propõe o uso de um produto chamado *Java Object Environment* (JOE) [Hamilton\_1996]. Desta forma, um sistema distribuído pode ser implementado utilizando um back-end (servidor) baseado em CORBA e um *front-end* (cliente) baseado em JAVA.

Em [Evan\_1997] pode-se encontrar uma tabela comparativa entre uma abordagem CGI e outra JAVA/CORBA, mostrando claramente as vantagens desta última em relação a primeira.

O RMI possui algumas vantagens em relação a uma abordagem CORBA/Java:

- Não há necessidade de uma linguagem especial IDL. O RMI simplesmente utiliza o tipo de interface Java para descrever os métodos de um objeto remoto, e este por sua vez implementa esta interface de um modo usualmente Java.
- Servidores de objetos remotos não precisam gerenciar a exclusão de instâncias de objetos uma vez que eles podem confiar em coleções de pacotes distribuídos pertencentes ao RMI.
- Se os objetos passados como argumentos a métodos ainda não estão presentes então eles podem automaticamente ser carregados (download).
- As ferramentas de desenvolvimento RMI/Java são partes da versão padrão do JDK (Java Developer Kit), e as extensões utilizadas em tempo de execução podem estar diponíveis em qualquer executável que suporte Java, assim não há necessidade de se obter licensas de uso adicionais para propósitos de desenvolvimento e utilização.

Porém, quando comparado ao CORBA, o RMI tem ao menos duas desvantagens. A falta de independência de linguagem pode ser um problema para aplicações que precisam incorporar software de legado desenvolvidos em outras linguagens. As aplicações CORBA também podem se utilizar dos *CORBAservices*, um conjunto de interfaces CORBA padrões que podem ser muito úteis para certos tipos de aplicações. Estes serviços incluem o serviço de denominação (*Naming Service*), o serviço de transação (*Transaction Service*), e o serviço de eventos (*Event* 

*Service*). A Sun propôs uma arquitetura de componentes para o desenvolvimento e utilização de aplicações orientadas a objeto e distribuídas chama Enterprise JavaBeans (EJB) e que promete ser compatível a CORBA.

#### A.7. Conclusões

Levando-se em consideração que é pouco provável que universalmente se utilize uma única plataforma ou sistema operacional, a abordagem de computação distribuída orientada a objetos é muito promissora. A participação ativa de organizações internacionais, como a OMG, que promova a integração entre as indústrias de tecnologia a fim de possibilitar a interoperabilidade entre produtos concorrentes, é uma necessidade.

Utilizando a WWW que fornece um ambiente comum, a linguagem Java que é independente de plataforma e a especificação CORBA para integração de sistemas desenvolvidos em linguagens diferentes e em plataformas diferentes é possível se obter soluções mais fáceis e consistentes para a integração de sistemas heterogêneos.

O fato destas tecnologias estarem sendo bem aceitas entre as grandes corporações está possibilitando a sua evolução contínua, como mostra a Figura 58.

Os ambientes hospitalares, que são inerentemente distribuídos e heterogêneos, claramente se beneficiarão destas tecnologias.

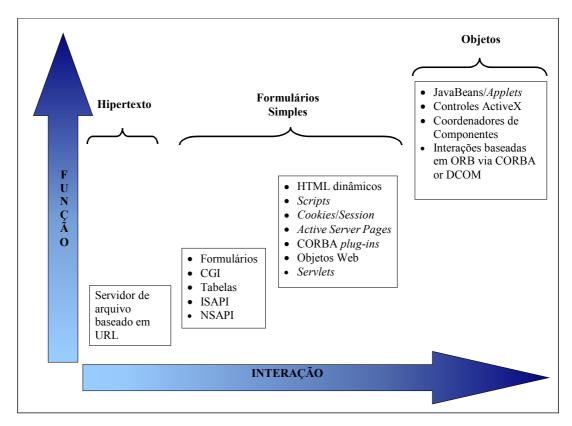


Figura 58 - Evolução das tecnologias para a WWW.

# Apêndice B

## Padrões de Comunicação de Dados na Medicina

#### B.1. CORBAmed

Uma das áreas que a OMG (*Object Management Group*) tem se dedicado bastante é a área médica. A identificação de pacientes pode ser surpreendentemente difícil, devido ao fato de muitas pessoas possuirem o mesmo nome, uso ilegal de registros de identidade etc. Assim a OMG criou o CORBAmed.

O CORBAmed é responsável pela especificação de interfaces interoperáveis para serviços de objeto distribuídos, tais como serviço de identificação de paciente, serviço de consulta léxica e serviço de acesso a observações clínicas, os quais poderão estar disponíveis em vários produtos comerciais da área médica [Jagannathan\_1998].

A OMG tem uma abordagem interessante e provada para padronizar tecnologia que é intrinsicamente diferente do processo de padronização convencional. Isto leva os fabricantes e usuários de tecnologia de componentes a participarem de fóruns abertos com o intuito de identificar áreas que requerem padronização prática e viável, procurando consenso entre competidores de mercado e adotando os resultados encontrados. Neste contexto o CORBAmed vem desenvolvendo especificações de serviço em numerosas áreas da saúde levando em consideração padrões existentes como o HL7 (*High Level Seven*, protocolo para comunicação de informações relacionadas a saúde), DICOM (ACR-NEMA Digital Imaging and Communications in Medicine [Horiil 1994]), NCPDP, X12 e outros [Jagannathan 1998].

O foco do CORBAmed está sobre serviços interoperáveis, levando-se em consideração os recursos do CORBA (para maiores detalhes ver item A.4 do Apêndice A). A missão do CORBAmed é:

- Melhorar a qualidade de tratamentos e reduzir custos através do uso de tecnologias
   CORBA para interoperabilidade por toda a comunidade global de saúde.
- Definir interfaces padronizadas orientadas a objeto entre serviços e funções relacionados a saúde.
- Promover a interoperabilidade entre uma variedade de plataformas, sistemas operacionais, linguagens e aplicações.
- Utilizar o processo de padronização da OMG.

Um *toolkit* de soluções CORBAmed foi publicado incluindo:

- Especificações padrões
- Produtos para avaliação e demonstração
- White papers e apresentações
- Produtos disponíveis
- Companhias que contribuem para a força-tarefa

O sucesso do CORBAmed encontra-se no fato das soluções serem gratuitas. A tecnologia no entanto não envolve ainda sistemas comerciais. Uma arquitetura de integração deve ser adotada para possibilitar a migração gerenciada e contínua de tecnologia, infraestrutura e serviços comerciais.

A principal vantagem em se utilizar componentes padronizados em arquiteturas de sistemas de informação para a saúde é que qualquer componente padronizado pode ser substituído, de forma consistente e barata, a qualquer hora por um componente mais recente e melhorado enquanto que o resto dos processos e tecnologias envolvidas possam permanecer intactas e inalteradas.

#### **B.2. Padrão DICOM**

O Digital Imaging and COmmunications in Medicine (DICOM) é um padrão para comunicação e armazenamento de imagens médicas e informações associadas a elas. Atualmente

o DICOM é utilizado por diversas modalidades de equipamentos de geração de imagens médicas. Este padrão possui mecanismos para a troca de informações entre diversos tipos de imagens, assim como para a comunicação das mesmas.

O padrão foi desenvolvido por um comitê de trabalho formado por membros do American College of Radiology (ACR) e do National Electrical Manufactures Association (NEMA) que iniciou os trabalhos em 1983, este comitê foi formado com o intuito de desenvolver um padrão para comunicação digital e informações de imagens, o comitê publicou a primeira versão em 1985, que foi chamada de ACR-NEMA 300-1985 ou (ACR-NEMA Version 1.0) e a segunda versão em 1988, chamada de ACR-NEMA 300-1988 ou (ACR-NEMA Version 2.0). A terceira versão do padrão, nomeada de DICOM 3.0 foi apresentada em 1993, onde foi substancialmente enfatizado, o conteúdo alterado, discutido, alguns problemas da primeira e da segunda versão e criado um novo processo, principalmente o protocolo de comunicação para rede [Steven 1999].

Os objetivos iniciais do padrão eram:

- promover a comunicação de informações de imagens digitais, sem levar em consideração os fabricantes dos aparelhos;
- facilitar o desenvolvimento e a expansão dos sistemas PACS que também podem se comunicar com outros sistemas de informação hospitalar;
- permitir a criação de uma base de dados de informações de diagnósticos que possam ser examinados por uma grande variedade de aparelhos distribuídos geograficamente [ACR-NEMA-1999].

O padrão hoje está essencialmente completo, apesar das mudanças que ainda possam acontecer devido à evolução da área, pois ele é um padrão multi-partes, o qual significa que as informações podem ser acrescidas quando há necessidade. Como um padrão estável e perfeitamente desenvolvido, ele está sendo implementado por diversas companhias tecnológicas e produtoras de imagens médicas. Estas implementações podem apresentar algumas falhas no padrão, que necessitarão ser corrigidas e podem induzir ao futuro desenvolvimento de um padrão melhor. Porém, o padrão é correntemente adequado para o desenvolvimento e implementação de sistemas de radiologia *filmless* [Steven 1999].

O padrão DICOM facilita a capacidade de atividade conjunta de equipamentos de imagens médicas porque especifica:

- um conjunto de protocolos a serem obedecidos pelos equipamentos exigindo a adaptação deles ao padrão;
- a sintaxe e semântica de comandos e informações associadas, as quais devem ser trocadas usando estes protocolos;
- informações que devem ser fornecidas com uma implementação para que a adaptação para o padrão seja cumprida.

#### O padrão DICOM não especifica:

- a implementação detalhada de algumas características do padrão sobre um equipamento que exige adaptação;
- o conjunto completo de características e funções esperadas de um sistema implementado integrando um grupo de equipamentos, cada um exigindo adaptação DICOM;
- um processo de teste e validação para avaliar uma adaptação aplicada ao padrão.

#### **B.2.1. Objetivos do Padrão DICOM 3.0**

Entre os principais objetivos do padrão DICOM podem ser considerados:

- Endereçar a semântica de comandos e dados associados. Para que equipamentos possam atuar uns sobre os outros, deve haver padrões de modo que equipamentos estejam esperando para reagir a comandos e dados associados;
- É explícito em determinar a adaptação necessária de implementações do padrão. Em particular, uma instrução de adaptação deve especificar informações suficientes para determinar as funções para que a necessidade de trabalho conjunta possa ser esperada com outro equipamento que também se ajuste ao padrão;
- Facilitar operações em ambiente de rede, sem a necessidade de um mecanismo de interface de rede;

- É estruturado para acomodar a introdução de novos serviços, facilitando assim suporte para futuras aplicações em imagens médicas;
- Faz uso de padrões internacionais existentes sempre que aplicável, e adequa-se à documentação estabelecida para padrões internacionais.

O padrão tem sido desenvolvido com uma ênfase em imagens médicas para diagnóstico como as geradas pela radiologia e técnicas a fins; de qualquer forma, ele é aplicável para uma grande linha de imagens relativas à troca de informações em ambientes médicos.

#### B.2.2. Definições Utilizadas pelo DICOM

- Attribute (Atributo) uma propriedade de um Objeto de Informação. Um atributo tem um nome e um valor que são independentes de qualquer método de codificação;
- *Command* (Comando) um meio genérico para conduzir uma solicitação para operar sobre Objetos de Informação através de uma interface de rede;
- Command Element (Elemento de Comando) uma codificação de um parâmetro de um comando que conduz este valor de parâmetro;
- *Command Stream* (Fluxo de Comandos) o resultado de um conjunto de comandos de elemento DICOM usando o DICOM;
- Conformance Statement (Adaptação Relatada) uma expressão formal associada com uma implementação específica do padrão DICOM. Ela especifica as Classes de Serviço, os Objetos de Informação, e protocolos de comunicação suportados pela implementação;
- Data Dictionary (Dicionário de Dados) um registro dos elementos de dados que determinam uma tag única, um nome, valores característicos, e a semântica para cada elemento de dado;
- Data Element (Elemento de Dado) uma unidade de informação definida por uma única entrada no dicionário de dados;

- Data Set (Conjunto de Dados) Informações trocadas devem estar formadas por um conjunto estruturado de valores atribuídos diretamente ou indiretamente por Objetos de Informação relatado. O valor de cada atributo no Conjunto de Dados é expresso como um Elemento de Dado;
- Data Stream (Dado Corrente) O resultado da codificação de um conjunto de dados usando o método de codificação DICOM (número de elemento de dados e representações como especificada pelo Dicionário de Dados);
- Information Object (Objeto de Informação) Uma abstração de uma entidade de informação real (ex: CT Image, Study, etc) o qual é influenciada por um ou mais comandos DICOM;
- Information Object Class (Classe Objeto de Informação) Uma descrição formal de um objeto de informação que inclui uma descrição desses propósitos e os atributos que possui. Ele não inclui valores para esses atributos.
- Information Object Instance (Instância de Objeto de Informação) Uma representação de uma ocorrência de uma entidade do mundo real, que inclui valores para os atributos da Classe Objetos de Informação para qual entidade deverá pertencer.
- Message (Mensagem) Um dado único da Message Exchange Protocol trocado entre duas cooperações DICOM Aplication Entities. Uma Messag é composta de um Command Stream acompanhado por um Data Stream opcional.
- Service Class (Classe de Serviço) uma descrição estruturada de um serviço que é suportado pela cooperação DICOM Aplication Entities usando especificação DICOM Commands acionando uma classe específica de Objetos de Informação.

O objeto de informação e a classe de serviço são dois dos componentes fundamentais do DICOM. Um entendimento destes componentes faz possível entender, pelo menos ao nível funcional, o que o DICOM faz e porque então ele é útil. Objetos de Informação definem o conteúdo de imagens médicas, e Classes de Serviços definem como aqueles conteúdos se interrelacionam.

As Classes de Serviços e Objetos de Informação são combinadas para formar as unidades funcionais do DICOM. Esta combinação é chamada *service-object pair*, ou SOP. A classe SOP é a unidade elementar do DICOM, tudo que DICOM faz quando implementado é baseado no uso de classes SOP. Além disso, sempre que os atributos no objeto de informação e as variáveis da classe de serviço são preenchidos por valores representando um paciente real, a classe SOP torna-se uma instância SOP e recebe um identificador único (UID) [RSNA-1997].

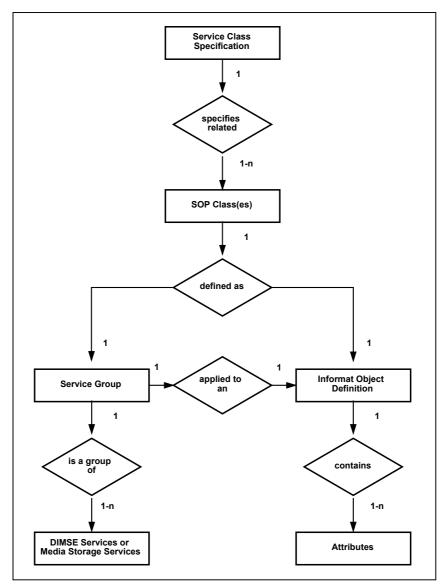


Figura 59 - Estruturas principais do modelo de informações do DICOM.

Para exemplificar a estrutura acima faremos uma analogia entre criar uma sentença e os conceitos definidos pelo padrão. Os elementos à esquerda representam parte de uma sentença e a direita estão os conceitos correspondentes do DICOM.

Tabela / - Analogia entre a	construção de uma	sentença e uma Classo	e DICOM SOP

Verbo: "Store"	Serviço (DIMSE)
Substantivo: "Image CT"	Definição do Objeto de Informação
	(IOD)
Sentença genérica: "Store in a CT	Classe SOP
image"	
Sentença específica: "Store this CT	Instância da classe SOP
image"	

A Tabela 7 mostra uma analogia entre construir uma sentença e construir uma Classe DICOM SOP. Podemos notar a distinção entre uma Classe SOP e uma instância SOP. No segundo caso, uma imagem específica foi e não é mais requerida combinando um serviço e um objeto de informação que é direto. Por exemplo, DICOM define uma série de armazenamento de classes SOP (ex: classe SOP para armazenamento de imagem de CT), classe para armazenamento de Ressonância Magnética Nuclear. A definição de objeto de informação CT e a classe de serviço de armazenamento são combinadas para formar a classe SOP armazenamento de imagem CT outras classes SOP de armazenamento são formadas de modo similar. Porque as classes SOP são requeridas para descrever o caminho e a funcionalidade de DICOM.

Em alguns equipamentos, para uma classe SOP em particular poder exercer dois papéis distintos como provedor da classe de serviço (SCP – Service Class Provider), ele disponibiliza e executa os serviços da classe SOP. E como usuário da classe de serviço (SCU – Service Class User), ele utiliza os serviços. Para cada combinação de classe SOP e papel exercido pelo equipamento, SCP ou SCU, o padrão define um conjunto básico de comportamentos padrão que irão governar a comunicação.

Segundo [Honeymann\_1999], DICOM é um padrão orientado a objeto, definindo objetos de informação, serviços e classes de serviços para executar estes serviços. Cada dispositivo tem

um conjunto de objetos definidos e é preparado para reconhecer o arquivo e permitir o acesso a ele, e também os serviços providos, e negociar entre dois dispositivos qual tem a necessidade de transferir a imagem.

O modelo fundamental contido no DICOM é o *Dicom Information Model*. Ele define a estrutura e a organização da informação relatada para a comunicação de imagens médicas. Ele especifica o conteúdo de objetos de informação que se juntam com os serviços DICOM requeridos, do *Service Object Pairs* (SOPs).

Cada serviço pode ser um serviço de rede ou um serviço de armazenamento de mídia, assim como a *Store* ou *Query*. Cada objeto é especificado por um *Information Object Definition* que define um conjunto de atributos relatados de alguns objetos do mundo real. Uma classe SOP é um conjunto de serviços e um conjunto de permissões *Information Object Definitions* (IODs), para que qualquer serviço possa ser aplicado. Um exemplo é a *Storage Service Class*, que especifica a aplicação do serviço de armazenagem para um objeto *Imagem CT* [Steven 1999].

#### B.2.3. Serviços do padrão DICOM

O padrão original DICOM publicado em 1993, define oito classes de serviços:

- 1. Verificação
- 2. Armazenamento
- 3. Query/Retrieve
- 4. Study Content Notification
- 5. Gerenciamento de Paciente
- 6. Gerenciamento de Estudo
- 7. Gerenciamento de Resultados
- 8. Gerenciamento de Impressão

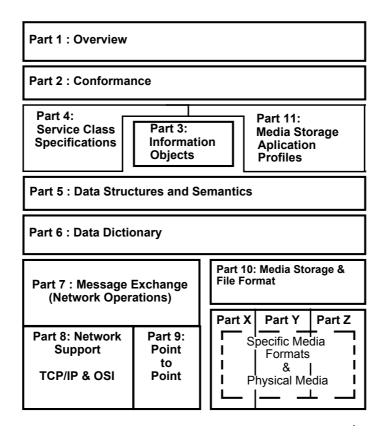
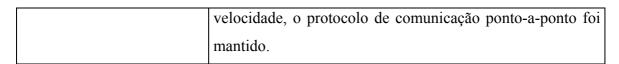


Figura 60 – Partes correntes do padrão DICOM e partes propostas para extensão do padrão.

Tabela 8 - Tabela do Resumo das partes do padrão DICOM 3.0.

Parte 1 : Overview	Apresentação do padrão, com uma descrição dos princípios	
	de desenvolvimento utilizados, definição da terminologia e	
	descrição das demais partes do padrão.	
Parte 2 : Conformance	Define os termos de conformidade com o padrão, indicando	
	como os fabricantes devem descrever sem ambigüidade	
	como seus produtos estão em conformidade com o padrão.	
Parte 3: Information Objects	Descreve como os IOs são definidos e especifica as diversas	
	classes de IO usadas no padrão. Muitos IODs possuíam	
	grupos de atributos comuns ou similares, de forma que estes	
	foram reunidos para criar módulos comuns que podem ser	
	usados por mais de um IOD. Assim foram criados IODs	
	compostos e IODs normalizados.	

Contém as especificações das classes de serviço, que 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	
Define como um conjunto de informações provenientes de	
objetos de informação e de classes de serviços devem ser	
codificadas para formar parte de uma mensagem.	
Fornece uma lista de todos os elementos de dados, ou	
atributos, que compõem todos os IOs. Para cada elemento	
de dado é fornecido o seu código numérico, o seu nome, a	
sua representação (texto, número em ponto-flutuante, etc), a	
multiplicidade e o domínio de valores permitidos.	
Esta parte descreve a dinâmica de comunicação, indicando	
o que é necessário para uma aplicação interagir no padrão	
de comunicação do DICOM. Também define como são	
construídas as seqüências de comandos, da mesma forma	
que a parte 5 define como são construídas as sequências de	
dados.	
Define o suporte de rede necessário para a troca de	
mensagens do DICOM. Atualmente o protocolo TCP/IP e	
qualquer outro que satisfaça o modelo de camadas ISO-OSI	
são suportados, mas a estrutura do padrão permite que	
novos protocolos sejam incorporados no futuro.	
Para manter compatibilidade com as versões anteriores do	
padrão, que usavam interfaces paralelas de dados de alta	



Além dessas partes acima listadas que são as principais, ainda existem as seguintes partes:

- Parte 10: Mídia de Armazenamento e Formato de Arquivo;
- Parte 11: Perfil da Aplicação e Mídia de Armazenamento,
- Parte 12: Mídia de Formatos e Mídia Física e;
- Parte 13 Gerenciamento ponto a ponto.

#### **B.2.4.** Suporte para rede DICOM

O padrão original ACR-NEMA definia uma simples interface paralela de "50 pinos", como meio de troca de mensagem. Isto limitava o padrão para operações ponto a ponto, com uma rede conectando pontos externos.

A última versão DICOM 3.0 está voltada para comunicação entre equipamentos, seja através de redes ou por ligações ponto a ponto. O propósito é que cada equipamento utilize seus próprios padrões e formatos para armazenar e gerenciar seus dados, mas quando for necessária a comunicação com outros equipamentos de diferentes fabricantes, é fundamental a existência de uma linguagem comum para que equipamentos de diversos fabricantes sejam capazes de se entender.

Uma outra implementação do DICOM permite que as informações possam ser armazenadas em meios físicos removíveis para que ela possa ser transportada, essa necessidade surgiu devido a alguns especialistas terem a necessidade de trabalhar com essas imagens externamente, por exemplo, imagens cardíacas que podem ser expostas em forma de cinema [Freire 1997].

Um passo para atender esta necessidade foi a criação da Parte 10, que faz descrições genéricas de estruturas de arquivos e diretórios para meios removíveis, além de serviços básicos para gerenciamento dos arquivos. Para especificar completamente o armazenamento em meios removíveis, foram elaboradas as Partes 11 e 12. A Parte 11 exerce uma função semelhante à Parte 4 e à Parte 12, ela especifica os detalhes de armazenamento em cada tipo de meio físico escolhido para fazer parte do padrão, como por exemplo: disquetes, CD-R, fitas, etc.

#### **B.2.5. Vantagens do padrão DICOM**

O padrão DICOM diferencia-se dos outros formatos de imagens tais como (JPEG, TIFF, GIF e outros) por permitir que as informações dos pacientes sejam armazenadas, de forma estruturada, juntamente com a imagem, isto é, elas são armazenadas contendo ponteiros, conhecidos como *tags* que identificam e limitam as informações. A imagem propriamente dita no padrão DICOM é baseada no formato JPEG com ou sem compressão, dependendo do equipamento que a gerou, pois cada companhia de tecnologia em imagem, pode implementar de uma forma, deste que obedeça a adaptação do padrão.

A grande vantagem dessa estrutura é permitir fazer a leitura do arquivo e extrairmos as informações necessárias para uma comunicação direta, ou seja, gerenciar as imagens e informações dos pacientes de forma coerente, mantendo a integridade; outra vantagem é que ele possibilitou melhorar a performance e auxilia no desenvolvimento de sistemas PACS [Kimura\_1998].

Outra vantagem descrita por [Kuzmak\_1998], é que o uso do padrão DICOM reduz custos, por permitir soluções para abrir sistemas consistindo de programas *in-house* e comerciais. Com o DICOM e consequência de anos de trabalho, o *US Department Veterans Affairs*, onde a idéia do formato surgiu, possui uma variedade de diferentes opções para sistemas de imagens radiológicas.