UNIVERSIDADE DE SÃO PAULO
INSTITUTO DE FÍSICA DE SÃO CARLOS
DEPARTAMENTO DE FÍSICA E INFORMÁTICA

*Incorporação do Tipo de Dado
Imagem em um Banco de Dados
Orientado a Objetos*

RILDO RIBEIRO DOS SANTOS

Orientador: Prof. Dr. Caetano Traina Júnior
Co-Orientadora: Profa. Dra. Agma Juci Machado Traina OK

Tese apresentada ao Instituto de Física de São Carlos, Universidade de São Paulo, para obtenção do título de Doutor em Ciências "Física Aplicada - opção Física Computacional".

SÃO CARLOS

SÃO PAULO

1997

IFSC-USP SERVIÇO DE BIBLIOTECA
INFORMAÇÃO

1785

Santos, Rildo Ribeiro dos

Incorporação do tipo de dado imagem em um Banco de Dados Orientado a Objetos/Rildo Ribeiro dos Santos.-São Carlos, 1997.

300 p.

Tese (Doutorado) - Instituto de Física de São Carlos, 1997.

Orientador: Prof. Dr. Caetano Traina Jr.

Co-Orientador(a): Profa. Dra. Agma J.M. Traina

1. Banco de Dados com Imagens. 2. Sistemas de Informação com Suporte a Imagens.

I. Título.

**UNIVERSIDADE
DE SÃO PAULO**

Instituto de Física de São Carlos

Av. Dr. Carlos Botelho, 1465
CEP 13560-250 - São Carlos - SP
Brasil

Fone (016) 274-3444

Fax (016) 272-2218

**MEMBROS DA COMISSÃO JULGADORA DA TESE DE DOUTORADO DE RILDO
RIBEIRO DOS SANTOS APRESENTADA AO INSTITUTO DE FÍSICA DE SÃO CARLOS,
DA UNIVERSIDADE DE SÃO PAULO, EM 26 DE NOVEMBRO DE 1997.**

COMISSÃO JULGADORA:

Prof. Dr. Caetano Traina Júnior/ICMSC-USP

Prof. Dr. Horácio Carlos Panepucci/IFSC-USP

Prof. Dr. Jan Frans Willem Slaets/IFSC-USP

Prof. Dr. Clézio Luis Tozzi/UNICAMP

Prof. Dr. Eduardo Massad/FM-USP

*À minha esposa, Deysinha, por seu grande
amor e compreensão nos momentos de
ausência a quem submeti.*

Agradecimentos

Ao Prof. Dr. Caetano Traina Jr. por sua orientação ao longo de todo este período, deste a iniciação científica, que em muito contribuiu em minha formação profissional, e em particular pela sua amizade e confiança.

À Profa. Dra. Agma Juci Machado Traina, por sua valiosa contribuição ao trabalho, com o auxílio no desenvolvimento dos procedimentos de processamento de imagens, parte básica fundamental neste projeto.

Ao componentes do Instituto de Pesquisa e Desenvolvimento em Banco de Dados e Sistemas de Informação, USP - São Carlos, que proporcionaram um ambiente valioso para a discussão e validação de muitas das propostas contidas neste trabalho.

À Wladerez A.G. Caiado pelo trabalho sério e pelo suporte operacional junto à Secretaria de Pós-Graduação do IFSC - USP.

Aos Diretores da Visual Systems, por proporcionar o ambiente computacional para os testes de validação do protótipo desenvolvido neste projeto.

Aos Colegas do Departamento de Computação e Estatística do IBILCE - UNESP, São José do Rio Preto, pelo apoio às atividades de pesquisa contida neste trabalho.

Ao Instituto de Física de São Carlos, IFSC-USP e ao Instituto de Ciências Matemáticas de São Carlos, ICMSC-USP, pelo apoio institucional.

Ao CNPq, FAPESP e PICD-CAPES pelo apoio financeiro ao desenvolvimento deste projeto.

SUMÁRIO

ABSTRACT	XVIII
----------	-------

RESUMO	XIX
--------	-----

CAPÍTULO 1

INTRODUÇÃO	1
------------	---

1.1. CONSIDERAÇÕES INICIAIS.	1
1.2. OBJETIVOS.	3
1.3. ORGANIZAÇÃO DA TESE.	5

CAPÍTULO 2

CONCEITOS PARA A INCORPORAÇÃO DE IMAGENS EM UM SISTEMA DE INFORMAÇÃO	7
---	---

2.1. CONSIDERAÇÕES INICIAIS.	7
2.2. SISTEMAS DE REPRESENTAÇÃO DE INFORMAÇÃO COM IMAGENS.	8
2.2.1. DEFINIÇÃO E ESPECIFICAÇÃO.....	8
2.2.2. COMPONENTES DE UM SISI.	13

2.2.3. CARACTERÍSTICAS DO TIPO DE DADO IMAGEM.	16
2.3. ELEMENTOS DE UM SISTEMA DE PROCESSAMENTO DE IMAGEM DIGITAL.	18
2.4. PROCESSAMENTO DE IMAGENS PARA DESCRIÇÃO DE SEU CONTEÚDO.	19
2.4.1. DESCRIÇÃO TEXTUAL DO CONTEÚDO DA IMAGEM.	20
2.4.2. EXTRAÇÃO DE CARACTERÍSTICAS GRÁFICAS DA IMAGEM.	20
2.5. ESTRUTURAS DE ORGANIZAÇÃO E ACESSO A IMAGENS.	22
2.5.1. QUAD-TREES.	25
2.5.2. STRINGS 2D.	30
2.5.3. ÁRVORES-R.	35
2.5.4. ÁRVORES-R ⁺	41
2.5.5. ÁRVORES-R*.	43
2.5.6. ÁRVORE DE CÉLULAS.	49
2.5.7. CONSIDERAÇÕES SOBRE AS ESTRUTURAS DE ACESSO.	52
2.6. DEFINIÇÃO DE OPERAÇÕES DE CONSULTA AO BANCO DE IMAGENS.	55
2.6.1. OPERADORES ESPACIAIS.	56
2.6.1.1. JOIN ESPACIAL.	59
2.6.1.2. DETERMINAÇÃO DE VIZINHANÇA.	63
2.6.2. OPERADORES DE CONJUNTO.	67
2.6.3. OPERAÇÃO DE COMPARAÇÃO ENTRE CARACTERÍSTICAS DAS IMAGENS.	67
2.6.3.1. SIMILARIDADE DE CARACTERÍSTICAS GRÁFICAS.	68
2.7. LINGUAGENS DE ACESSO E INTERFACE PARA IMAGENS.	70
2.8. CONSIDERAÇÕES FINAIS.	72

CAPÍTULO 3

IMPLEMENTAÇÕES DE SISTEMAS DE INFORMAÇÃO COM SUPORTE A**IMAGENS** **75**

3.1. CONSIDERAÇÕES INICIAIS.	75
3.2. LINGUAGENS DE CONSULTAS ESPACIAIS.	77
3.2.1. LINGUAGEM PSQL.	78
3.2.2. LINGUAGEM SQL ESPACIAL.	80
3.3. LINGUAGEM DE CONSULTA PICQUERY+.	82
3.4. SISTEMA DE CONSULTA PARA IMAGENS 3D.	87
3.5. SISTEMA QBIC.	90
3.6. MODELO DE IMAGEM ESPACIAL BASEADO EM CONHECIMENTO - KSIM.	96
3.7. SISTEMA PROBE.	100
3.8. SISTEMA PICTION.	102
3.9. SISTEMA DE RECUPERAÇÃO DE IMAGENS POR FORMAS SIMILARES (FIBSSR).	103
3.10. SISTEMA CHABOT.	106
3.11. RECUPERAÇÃO DA INFORMAÇÃO EM SISTEMA DE MULTIMÍDIA.	108
3.12. SISTEMA DE INFORMAÇÃO GEOGRÁFICA.	110
3.13. CONSIDERAÇÕES FINAIS.	111

CAPÍTULO 4**BANCO DE DADOS ORIENTADO A OBJETOS** **113**

4.1. CONSIDERAÇÕES INICIAIS.	113
4.2. CONCEITOS GERAIS DE ORIENTAÇÃO A OBJETOS.	114

4.3. LINGUAGENS DE CONSULTA ORIENTADA A OBJETOS.	120
4.3.1. MODELO ORIENTADO A OBJETOS DE REFERÊNCIA.....	120
4.3.2. PREDICADOS DE IGUALDADE.....	124
4.3.3. PREDICADOS SOBRE AS PROPRIEDADES DOS OBJETOS.....	126
4.3.4. QUANTIFICADORES RESTRITOS.	127
4.3.5. NAVEGAÇÃO.....	127
4.3.6. MÉTODOS.....	128
4.3.7. HIERARQUIA DE CLASSES.....	130
4.3.8. OPERAÇÕES DE <i>JOIN</i> E PROJEÇÕES GENÉRICAS.	133
4.3.9. CONSULTAS RECURSIVAS.....	134
4.3.10. ANÁLISE DOS RESULTADOS DAS CONSULTAS.....	135
4.3.11. COMPARAÇÃO COM A LINGUAGEM RELACIONAL.....	136
4.5. CONSIDERAÇÕES FINAIS.	137

CAPÍTULO 5

<u>SIRIUS - MODELO DE BANCO DE DADOS ORIENTADO A OBJETOS</u>	139
5.1. CONSIDERAÇÕES INICIAIS.	139
5.2. CONCEITUAÇÃO GERAL DO MODELO SIRIUS.	140
5.2.1. OBJETOS E TIPOS DE OBJETOS.....	140
5.2.2. ATRIBUTOS, TIPOS DE ATRIBUTOS E CARACTERÍSTICAS DE TIPOS DE ATRIBUTOS.	142
5.2.3. ABSTRAÇÃO DE AGREGAÇÃO.....	146

	ix
5.2.4. ABSTRAÇÃO DE CLASSIFICAÇÃO.....	150
5.2.5. ABSTRAÇÃO DE GENERALIZAÇÃO.....	146
5.2.6. ABSTRAÇÃO DE COMPOSIÇÃO.....	153
5.3. SIRIUS COMO MODELO CONCEITUAL PARA REPRESENTAÇÃO DE IMAGENS.	155
5.4. CONSIDERAÇÕES FINAIS.	156
<u>CAPÍTULO 6</u>	
<u>IMAGEM: CONCEITOS E MODELAGEM EM UM BANCO DE DADOS</u>	
<u>ORIENTADO A OBJETOS</u>	<u>157</u>
6.1. CONSIDERAÇÕES INICIAIS.	157
6.2. REPRESENTAÇÃO DE IMAGENS NO SIRIUS.	158
6.2.1. CONSTANTES IMAGENS.	158
6.2.2. SUMARIZADORES.....	162
6.2.3. INSTÂNCIAS DAS VARIÁVEIS DESCRITORAS.	164
6.2.4. LIMITE DE ACEITAÇÃO.	166
6.2.5. MODIFICADORES.....	167
6.3. MODELAGEM DO CONCEITO DE IMAGEM NO SIRIUS.	170
6.3.1. DEFINIÇÃO DO ESQUEMA DE IMAGENS PARA UMA APLICAÇÃO.	177
6.3.2 EXEMPLO DA MODELAGEM DE UMA APLICAÇÃO COM IMAGENS MÉDICAS NO SIRIUS.	180
6.4. ESTRUTURAS DE ACESSO A IMAGENS.	189
6.5. LINGUAGEM DE ACESSO A IMAGENS NO SIRIUS.	191

7.4. ARQUITETURA DO SGI.	254
7.5. MODELO DE CONSTRUÇÃO DE UMA APLICAÇÃO.	260
7.6. CONSIDERAÇÕES FINAIS.	261

CAPÍTULO 8

CONCLUSÕES E PROPOSTAS DE FUTURAS PESQUISAS **263**

8.1. CONSIDERAÇÕES INICIAIS.	263
8.2. CONTRIBUIÇÕES INOVADORAS DO TRABALHO.	263
8.3. PROPOSTAS DE FUTURAS PESQUISAS.	267
8.4. CONSIDERAÇÕES FINAIS.	261

REFERÊNCIAS BIBLIOGRÁFICAS **275**

APÊNDICE A

SUMÁRIO DA LINGUAGEM DE ACESSO A IMAGENS **1**

1. LINGUAGEM DE DEFINIÇÃO DE IMAGENS NO SIRIUS.	1
1.1. CRIAÇÃO DOS OBJETOS DE DESCRIÇÃO DO DOMÍNIO DA APLICAÇÃO.	1
1.2. ASSOCIAÇÃO ENTRE OS OBJETOS DO ESQUEMA DA APLICAÇÃO.	1
1.3. ALTERAÇÃO DA DESCRIÇÃO DOS OBJETOS NO ESQUEMA.	1

FIGURAS

CAPÍTULO 02

<i>Figura 1: Estágios de Construção de um Sistema de Informação com Suporte a Imagens.</i>	9
<i>Figura 2: Componentes de um Sistema de Informação com Imagens</i>	10
<i>Figura 3 - Arquitetura dos três Esquemas.</i>	13
<i>Tabela 1 - Níveis Conceituais na Arquitetura de um SISI.</i>	14
<i>Figura 4 - Elementos Básicos de um Sistemap de Processamento de Imagens.</i>	19
<i>Figura 6: Divisão do polígono para a Quad-tree.</i>	27
<i>Figura 7: Aspectos lógicos da quadtree totalmente invertida.</i>	29
<i>Figura 8: Aspectos lógicos da quadtree ID.</i>	29
<i>Figura 9:A imagem e sua representação simbólica.</i>	31
<i>Figura 10: Padrões ambíguos.</i>	34
<i>Figura 11: Retângulos para a árvore-R</i>	37
<i>Figura 12: Árvore-R correspondente a Figura 11.</i>	37
<i>Figura 13: Retângulos organizados em uma árvore-R+.</i>	42
<i>Figura 14:Árvore-R+ correspondente a Figura 13.</i>	43
<i>Figura 15: Representação para uma Árvore de células</i>	51
<i>Figura 16: Relações Topológicas</i>	58
<i>Figura 17:Conjunto de retângulos para ordenação.</i>	63
<i>Figura 18: Exemplo de um teste de intersecção.</i>	64
<i>Figura 19: MINDIST não é sempre a melhor ordem.</i>	66
<i>Figura 20 - Operador de União em imagens simbólicas.</i>	69

CAPÍTULO 03

<i>Figura 1 - Consulta do Exemplo a).</i>	86
<i>Figura 2 - Consulta do Exemplo b).</i>	87
<i>Figura 3: Arquitetura de Instanciação (acima) e do Banco de Dados (abaixo).</i>	91
<i>Figura 4: Representação de Hierarquia de Tipos Abstratos do KSIM.</i>	97
<i>Figura 5 - Diagrama de Fluxo do Processamento de Consultas Baseado em Conhecimento</i>	97
<i>Figura 6: Arquitetura do Sistema PROBE</i>	101

CAPÍTULO 04

<i>Tabela 1 - Principais Características do Modelos de Dados Orientado a Objeto.</i>	118
<i>Figura 1 - Esquema da Aplicação Exemplo para o OOPC.</i>	122
<i>Figura 2 - Exemplo de uma Hierarquia de Classes.</i>	131

CAPÍTULO 05

<i>Figura 1 - Representação de Objetos.</i>	141
<i>Figura 2 - Exemplo de Representação de Atributos.</i>	142
<i>Figura 3 - Notação utilizada na agregação de atributos.</i>	146
<i>Figura 4 - Notação utilizada na agregação de objetos.</i>	148
<i>Figura 5 - Exemplo de relacionamento Triplo.</i>	149
<i>Figura 6 - Abstração de classificação e seus atributos de Classificação.</i>	151
<i>Figura 7 - Exemplos de ocorrências de abstração de Generalização.</i>	153
<i>Figura 8 - Representação de Composição (colônia de objetos).</i>	154

CAPÍTULO 06

<i>Figura 1 - Meta Esquema para o Atributo com Característica de Imagem.</i>	172
--	-----

<i>Figura 2 - Esquema de Exames médicos realizados em Pacientes.</i>	182
<i>Figura 3 - Descrição do Conteúdo do Atributo Exame de Tomografia.</i>	183
<i>Figura 4 - Descrição do Conteúdo do Atributo Tomografia do Crânio.</i>	184
<i>Figura 5 - Definição dos Operadores.</i>	184
<i>Figura 6 - Descrição das Constantes Imagens: Medula e Tálamo.</i>	186
<i>Figura 7 - Descrição das Constantes Imagens - Coração e Pulmão</i>	186
<i>Figura 8 - Descrição da Constante Imagem: Hipotálamo</i>	188
<i>Figura 9 - Descrição da Distribuição de Cores da Constante Coração.</i>	188
<i>Figura 10 - Representação de Instâncias para o Atributo com Característica de Imagem.</i>	190

CAPÍTULO 07

<i>Figura 1 - Esquema do Atributo com Característica de Imagens no SIRIUS.</i>	251
<i>Figura 2 -DE-R do Atributo com Característica de Imagem.</i>	252
<i>Figura 3 - Arquitetura do Sistema de Gerenciamento de Imagens.</i>	254

CAPÍTULO 08

<i>Figura 1 - Transformação na Estrutura de Indexação.</i>	271
--	-----

APÊNDICE B

<i>Figura 1 - Interface do Sistema para os Grupos de Desenvolvimento e Produção.</i>	2
<i>Figura 2 - Definindo um Atributo com Característica de Imagem.</i>	3
<i>Figura 3 - Definindo uma Constante Imagem.</i>	4
<i>Figura 4 - Definindo os Sumarizadores do Esquema da Aplicação.</i>	5
<i>Figura 5 - Definindo as Variáveis Descritoras do Esquema.</i>	6
<i>Figura 6 - Descreve o Conteúdo do Atributo com Característica de Imagem.</i>	7
<i>Figura 7 - Descrevendo as Constantes Imagens do Esquema.</i>	8

<i>Figura 8 - Instâncias do Tipo de Objeto Paciente.</i>	10
<i>Figura 9 - Inserindo Imagens no Banco de Dados.</i>	11
<i>Figura 10 - Definindo o Contexto das Variáveis Descritoras.</i>	12
<i>Figura 11 - Definindo Contexto dos Modificadores.</i>	13
<i>Figura 12 - Formulando uma Consulta ao Banco de Imagens.</i>	14
<i>Figura 13 - Resultados das Consultas ao Banco de Imagens.</i>	15

ABSTRACT

Picture Archiving and Communication System (PACS) applied in medical image storage provides graphical information of many aspects of the health, diseases and treatment of each patient for the physician and technicians of a health care center.

Nowadays, medical exams can generate a large amount of data due to the computerized medical instruments used to collect the graphical information about the patients. This work describes the concepts used in an Object Oriented Data Base System to deal with medical images, so that it can be retrieved through queries based on the graphic contents of the stored images. The usual approach uses icons and textual attributes stored with the images to specify the queries. This work uses a novel modeling technique to define the "image data type", through which it is possible to decide, beforehand the query itself, the valuable data of each image when it is stored in the database, so the search can be accelerated when queries are issued.

RESUMO

Os Sistemas de Armazenamento e Comunicação de Imagens Médicas fornece, para os físicos, médicos e técnicos de um centro de saúde, informações gráficas sobre vários aspectos envolvidos no diagnóstico de cada paciente. Atualmente, os exames médicos produzem uma grande quantidade de informação em função dos equipamentos médicos computadorizados, utilizados principalmente para se obter dados gráficos (imagens) sobre os pacientes. Este trabalho descreve os conceitos utilizados em Sistema de Banco de Dados Orientado a Objetos para manipular imagens médicas, de tal forma que possam ser recuperadas através de consultas, baseadas na descrição de seu conteúdo gráfico. A abordagem tradicional se utiliza de ícones e atributos textuais, armazenados juntamente com as imagens, para especificar as consultas. Neste trabalho é utilizado uma nova técnica de modelagem para definir o “tipo de dado imagem”, através do qual é possível decidir, previamente à execução da consulta, os dados que sejam significativos para cada imagem, no instante que esta seja armazenada no Banco de Dados. Desta forma, a busca por uma determinada informação pode ser acelerada, durante a avaliação de uma consulta.

INTRODUÇÃO

1.1. CONSIDERAÇÕES INICIAIS.

A área de estudos em Banco de Dados (BD) preocupa-se com o desenvolvimento de sistemas e tecnologias que permitam a armazenagem, busca e recuperação de informações em meios digitais. Para isso, é necessário a identificação de elementos conceituais, formalizados como construtores semânticos, que atuam de forma a propiciar a representação e a manipulação da informação de uma maneira padronizada e com um grau de consistência controlável. Os Sistemas Gerenciadores de Banco de Dados (SGBD) têm sido utilizados com sucesso em diversas aplicações em áreas distintas do conhecimento. No entanto, novas aplicações têm apresentado como requisito a necessidade de manipular informações não textuais, ou seja, que possam, por exemplo, estar contidas no interior de imagens digitais, gráficos e sons.

As áreas de aplicação são diversas incluindo, Sistemas de Informação Geográfica (GIS), Automação de Escritório (AE), Sistema de Comunicação e Armazenamento de Imagens Médicas (PACS), Projeto e Manufatura Assistido por Computador (CAD/CAM), aplicações científicas de BD e em Robótica, entre outras. Todas estas aplicações caracterizam-se por produzir um grande volume de informações contidas em imagens que devem ser manipuladas e armazenadas para uma posterior recuperação através de consultas. No entanto, cada domínio de

aplicação possui imagens com características distintas e com requisitos diversos referentes às consultas a serem formuladas. Em um Sistema de Informação Geográfica, as imagens caracterizam-se por apresentar informações definidas em um espaço de duas dimensões (2D), no qual as consultas podem ser formuladas sobre os objetos presentes no interior de uma imagem, considerando suas posições relativas no espaço ou mesmo medidas de distâncias entre eles. Nos Sistemas de Automação de Escritório (AE) podem surgir consultas em que a comparação de padrões de assinaturas em diversos documentos seja o objetivo principal. Nos Sistemas de Informação que manipulam Imagens Médicas, as consultas podem envolver a aplicação de operações mais sofisticadas em seu plano de execução, em consequência da diversidade e complexidade da informação presente. Portanto, o Banco de Dados consiste em uma parte fundamental do Sistema de Informação com Suporte a Imagem (SISI).

A criação de um Sistema Gerenciador de Banco de Dados (SGBD) para manipular um grande conjunto de imagens, uniformemente com os seus demais elementos, foi inicialmente difícil em decorrência da sobrecarga computacional imposta pela maioria dos algoritmos de processamento e armazenamento de imagens. Outro aspecto a ser mencionado consiste na ausência de estruturas conceituais nos modelos de dados que permitissem a representação de imagens e das informações a estas associadas. Os sistemas iniciais propunham a incorporação da imagem ao BD, considerando-a como um elemento externo ao Banco, associado às demais informações através de um atributo de ligação ao arquivo indicando o seu armazenamento físico no disco. Com esta organização, cada registro físico que represente algum item de informação possui um campo com a indicação do arquivo com a imagem.

1.2. OBJETIVOS.

A Medicina vem cada vez mais utilizando-se de equipamentos que geram grande quantidade de dados em formato digital, muitos dos quais devem ser armazenados durante longo tempo [Dutton_90]. Isso pressupõe sua armazenagem em Bancos de Dados, porém a natureza altamente variável das informações não torna adequado o uso dos SGBDs tradicionais. Por exemplo, a operação de busca de imagens baseada em conteúdo não tem sido contemplada. Um dos equipamentos que têm despertado grande interesse é o aparelho de Tomografia por Ressonância Magnética (ToRM) [Panepucci_87] [Tannús_87] para a obtenção de imagens da estrutura interna do corpo humano.

Um Sistema de Banco de Dados adequado para suportar a construção dos aplicativos que permitam a manipulação eficiente desse tipo de dados deve ter a capacidade de suportar a armazenagem de imagens em geral e, particularmente, imagens médicas, e permitir ao mesmo tempo a sua recuperação das imagens desejadas, juntamente com os dados que não são imagens, tais como critérios utilizados e o resultado de diagnósticos. Este trabalho define as atividades de incorporação do tratamento de imagens como um tipo de dado em um Gerenciador de Banco de Dados. Encontra-se integrado com o projeto de pesquisa em desenvolvimento junto ao Departamento de Física e Informática, do Instituto de Física de São Carlos (IFSC) e do Departamento de Computação e Estatística (DCCE) do Instituto de Ciências Matemáticas de São Carlos (ICMSC), ambos pertencentes à Universidade de São Paulo (USP). Esse projeto consiste no desenvolvimento de um sistema completo de tomografia por RM que inclui desde a construção do tomógrafo propriamente dito até o sistema de suporte a diagnósticos, utilizando tecnologia

totalmente dominada e desenvolvida no Brasil. O desenvolvimento descrito nesta tese propõe a incorporação de imagens como um tipo de dados de atributos em um Banco de Dados Orientado a Objetos, através de um conjunto de conceitos que permite aos usuários definir aplicações, nas quais a imagem seja manipulada uniformemente, como os demais tipos de dados disponíveis. Para tal, são propostos mecanismos que possibilitam a representação das informações que possam ser extraídas das imagens. Estas consistem na descrição de seu conteúdo, a qual será realizada considerando os aspectos gráficos da imagem, ou seja, distribuição de cores, padrões, texturas e outros. O objetivo destes conceitos é permitir a formulação de consultas sobre o domínio da imagem, no qual sejam especificadas informações do seu conteúdo gráfico, utilizando-se de elementos significativos para este fim. Neste projeto não é desconsiderado o uso de informações descritivas do conteúdo da imagem, que estejam na forma de texto. Propõe-se também um conjunto de comandos especificados em uma Linguagem de Manipulação e Definição de Imagens de alto nível, a qual possui os seus elementos formalizados através dos operadores da álgebra de imagens.

1.3. ORGANIZAÇÃO DA TESE.

A apresentação dos elementos conceituais envolvidos no desenvolvimento deste projeto, assim como dos resultados obtidos das atividades de pesquisa, encontra-se organizada nos seguintes capítulos:

Capítulo 2: Apresenta-se os conceitos envolvidos na incorporação de um tipo de dado Imagem em um Banco de Dados. Estes conceitos são tratados considerando-se

as etapas necessárias a este processo, ou seja, aquisição, processamento, armazenamento, organização e recuperação das imagens.

Capítulo 3: Realiza-se a revisão dos principais trabalhos disponíveis na literatura envolvendo a incorporação e manipulação de imagens em um Sistema de Informação, realizando-se uma análise comparativa das características relevantes de cada projeto.

Capítulo 4: Realiza-se a revisão geral dos conceitos de Banco de Dados Orientado a Objetos, discutindo principalmente os aspectos referentes às linguagens de manipulação de objetos.

Capítulo 5: Descreve-se o Modelo SIRIUS, utilizado como suporte para o desenvolvimento deste trabalho. É realizada uma breve descrição dos seus componentes principais e dos construtores semânticos que permitem a sua classificação como um Modelo de Dados Orientado a Objetos. Uma atenção maior é dedicada às características de atributos e dos mecanismos disponíveis para o suporte ao conceito de imagem pelo Modelo.

Capítulo 6: Apresenta-se a formalização teórica dos aspectos envolvidos na manipulação de imagens. São descritas a modelagem do atributo com característica de imagem, a metodologia de desenvolvimento de uma aplicação com o domínio no conjunto de imagens e a linguagem de acesso proposta.

Capítulo 7: Descreve-se os principais aspectos da implementação do Sistema de Gerenciamento de Imagens, através da apresentação detalhada dos módulos que compõem a sua arquitetura.

Capítulo 8: Descreve-se as principais contribuições deste trabalho, assim como de algumas linhas de pesquisas a serem desenvolvidas como consequência imediata dos resultados desta Tese.

CONCEITOS PARA A INCORPORAÇÃO DE IMAGENS EM UM SISTEMA DE INFORMAÇÃO

2.1. CONSIDERAÇÕES INICIAIS.

A construção de um Sistema de Informação com Suporte a Imagens (SISI) envolve a utilização de conhecimentos provenientes de várias áreas de pesquisa. Em Processamento de Imagens tem-se disponível um conjunto de técnicas para se extrair características de descrição de uma imagem. O campo de Redes Neurais fornece conhecimento alternativo na implementação das operações de processamento de imagens, assim como métodos para a classificação de informações e/ou formulação de estruturas na sua organização. A disciplina de Banco de Dados apresenta os conceitos básicos necessários ao suporte deste novo tipo de dado, com o objetivo de fornecer construtores semânticos para a representação do conjunto de informações que possam ser extraídas de cada imagem. Além disso, pesquisas na área de Interação Usuário-Computador e de Sistemas de Gerenciamento de Interface

colaboram para a formulação de um meio de definição e manipulação dos aplicativos, que seja natural e transparente ao usuário final.

Neste capítulo serão expostos os principais aspectos envolvidos em um Sistema de Representação de Informação com Imagens. A organização do capítulo está em concordância com a sequência de incorporação do tipo de dado imagem em um SISI. Na seção 2 é apresentada uma conceituação geral sobre um SISI e de seus componentes principais. Nas seções 3 e 4 realiza-se uma revisão sobre as principais técnicas de aquisição e processamento de imagens necessárias ao processo de extração das informações contidas no interior de uma imagem. Uma vez obtido o conjunto de características de descrição de uma imagem, este pode ser organizado em uma série de estruturas que compõem os seus métodos de acesso. Na seção 5 é apresentada uma descrição de alguns métodos de acesso a estas informações, a saber “quadrees”, “string-2D”, árvores-R e suas variações. Definindo-se as estruturas de acesso às imagens, na seção 6 são estudados os principais tipos de operadores presentes nas expressões de consultas, quando estas estão armazenadas em um Banco de Dados. E, finalizando o capítulo, são enumerados os principais requisitos e características do projeto do sistema de interface e das linguagens de consulta a imagens baseadas na descrição de seu conteúdo.

2.2. SISTEMAS DE REPRESENTAÇÃO DE INFORMAÇÃO COM IMAGENS.

2.2.1. Definição e Especificação.

As imagens consistem de um item de dado integrante da realidade de vários domínios de aplicação, e os Sistemas de Informação definidos para o seu

gerenciamento devem suportar também este tipo de dado. Um Sistema de Informação com Suporte a Imagens (SISI) permite a definição e manipulação uniforme da informação contida em elementos textuais e gráficos, na forma de imagens, vídeos e som. Para uma caracterização precisa dos requisitos de funcionalidade do SISI, deve-se considerar as etapas básicas envolvidas em sua construção, apresentada de forma esquemática na Figura 1, e descrita a seguir.

Figura 1: Estágios de Construção de um Sistema de Informação com Suporte a Imagens.

A. *Análise da Imagem e Reconhecimento de Padrões.*

A imagem original, obtida a partir de um meio de aquisição qualquer, é submetida a uma série de algoritmos com o objetivo de se extrair informações sobre o seu conteúdo. Estas informações podem consistir na identificação dos objetos que estejam presentes no interior da imagem, no levantamento de características gráficas que permitam sintetizá-la sob um determinado aspecto, como por exemplo, sua distribuição média de cores, ou identificação da textura dominante. O procedimento de análise das imagens pode ser realizado de forma automática, ou seja, todas as informações são extraídas através da aplicação dos algoritmos apropriados sem a intervenção do usuário na formulação do resultado final, ou com um acompanhamento deste, inserindo informações que sejam necessárias à completa descrição do conteúdo apresentado pela imagem. Esta fase apresenta como principal

requisito a escolha apropriada do conjunto de algoritmos, que devem extrair valores possíveis de serem estruturados e utilizados como elementos que caracterizam a informação presente na imagem. Os tipos de algoritmos a serem aplicados são determinados a partir das características do domínio da aplicação no qual se insere a imagem, indicando o tipo de informação relevante a ser extraída para a descrição de seu conteúdo. Nas aplicações que envolvam a manipulação de imagens geográficas (GIS - Sistemas de Informações Geográficas), passa a ser relevante a identificação da posição relativa ou mesmo a distância entre os objetos. Esta informação pode não ser significativa em uma imagem resultante de um exame de tomografia por ressonância magnética, que apresentará o seu próprio conjunto de requisitos.

Os objetos e as informações resultantes desta fase devem ser organizados em estruturas de dados que permitam a posterior manipulação e especificação das operações de acesso. As estruturas escolhidas são determinadas a partir dos requisitos próprios de cada aplicação e das formas de consultas realizadas, e devem ser compatíveis com a técnica de acesso a ser formulada para o Banco no qual estas imagens ficarão armazenadas.

B. Estruturação e Organização da Informação na Imagem.

Para recuperar uma imagem ou uma coleção destas, serão fornecidas informações que sejam significativas em sua identificação. Estas devem ser as mesmas resultantes da atividade de análise realizada anteriormente, pois são as únicas que o sistema irá conseguir manipular e reconhecer até o momento como item de identificação de uma imagem. Estas informações consistem no elemento de acesso

às imagens, e devem ser organizadas com o objetivo de tornar eficiente as operações de consulta sobre o Banco de Dados. Isto implica na criação de índices de acesso para as imagens, que sejam dependentes das informações extraídas e possam definir a sua organização [Chiueh_94].

A especificação da estrutura de indexação é realizada sob três aspectos [Chang_92b]: o mecanismo de extração, sua representação e a organização da estrutura de acesso. O *processo de extração* consiste na fase descrita anteriormente, a qual é altamente dependente da evolução das técnicas de processamento de imagens. A *representação do índice* define o elemento a ser utilizado como chave de acesso, ou seja, o tipo de dado da informação extraída da imagem e principalmente a sua estruturação. Neste caso podem ser citados o uso de palavras chaves, descritores de formas, distribuição de cores e outros. A estruturação desta informação apresenta uma importância fundamental na *organização dos índices* de acesso às imagens. Esta define como as informações estarão classificadas para facilitar a localização de uma determinada imagem no BD.

Além dos índices para acesso, formula-se mecanismos através dos quais possa-se representar o significado semântico dos objetos e das informações extraídas em cada imagem. Estes mecanismos, definidos como construtores semânticos, são dependentes do domínio de conhecimento da aplicação e da informação manipulada. Neste ponto torna-se relevante o aspecto de modelagem do tipo de dado imagem e do seu conteúdo gráfico a ser realizado no modelo de dados escolhido para a construção do Sistema. Portanto, o resultado desta fase consiste na estrutura contendo os índices

de acesso às imagens e na representação semântica dos objetos e informações presentes em seu conteúdo.

C. Recuperação das Informações nas Imagens.

Uma vez representado o conjunto de informações que descreve o conteúdo gráfico da imagem e se encontre definida a estrutura de indexação, estes são utilizados na especificação das operações de consulta ao BD. A adequação das estruturas às características do domínio de conhecimento da aplicação faz com que os requisitos de recuperação das informações sejam validados de forma natural. Como exemplo, em aplicações que envolvem imagens de tomografia computadorizada, o usuário pode escolher recuperar as que apresentam um determinado tumor com a forma semelhante a algum previamente selecionado. Para outras aplicações, a estrutura de acesso deve permitir a realização de operações que envolvam a manipulação de informações ou coordenadas espaciais definindo posições relativas dos objetos.

Estas três etapas podem ser consideradas como os estágios necessários à transformação de uma imagem contendo apenas informação digital, em um elemento de informação com um significado semântico associado a cada item de dado que seja identificado em seu interior. Assim, definindo o conhecimento presente em uma imagem através da tupla (x_i, x_m) , onde x_i consiste na imagem digital e x_m , uma lista na forma $(x_{1m}, x_{2m}, \dots, x_{nm})$ representando os significados semânticos que possam ser associados, então no início deste processo pode-se considerar a imagem como (x_i, null) , ou seja, sem conhecimento da informação que contém. Ao final do processo de

identificação e representação da semântica de cada elemento presente na imagem, esta torna-se uma estrutura da forma (x_i, x_m) . Isto significa que o tipo de dado imagem se encontra preparado para a sua inserção em um Sistema de Informação, de forma a possibilitar uma manipulação uniforme.

2.2.2. Componentes de um SISI.

Figura 2: Componentes de um Sistema de Informação com Imagens

Em função da descrição anterior com as etapas de construção de um SISI, pode-se formular a sua arquitetura composta pelos seguintes módulos: Sistema de Aquisição de Imagens,

Sistema de Processamento e Análise de Imagens, Sistema de Banco de Dados e Imagens, Sistema de Comunicação de Imagens e Sistema de Apresentação de Imagens (Figura 2) [Chang_92b]. Os elementos e estruturas envolvidas na

Figura 3 - Arquitetura dos três Esquemas.

especificação e construção de cada módulo serão apresentados em detalhes nas próximas seções deste capítulo.

Um dos elementos de fundamental importância na arquitetura descrita anteriormente, ao lado do Sistema de Processamento e Análise de Imagens, consiste na existência de um Banco de Imagens. Para que este Banco apresente o suporte à manipulação de imagens e dos elementos de descrição do seu conteúdo, foi proposta uma arquitetura [Chang_92b] complementar à estruturação tradicional da abordagem de BD (Figura 3), composta por três esquemas ou níveis [Elmasri_94]. Nesta arquitetura o nível conceitual é refinado com o objetivo de fornecer um tratamento adequado às várias visões das informações nas imagens. Assim, a arquitetura fica com os níveis indicados na Tabela 1 a seguir.

Nível	Tarefas	Exemplo
Usuário	Raciocínio Espacial	Encontre todos os veículos com rodas.
Caracterização Semântica	Estruturação do Conhecimento contido na Imagem	Encontre imagens com (objeto_imagem, roda);
Caracterização da Imagem	Descrição da Imagem	Encontre imagens com (objeto_imagem, círculo);
Representação das Características	Estrutura de Representação das Características da Imagem	Encontre imagens, com (objeto_imagem, contorno_do_círculo);
Organização das Características	Armazenamento/Recuperação do Dado Imagem	Localiza/Insere imagem (objeto_imagem, estrutura_dados_contorno);

Tabela 1 - Níveis Conceituais na Arquitetura de um SISI.

O *nível do Usuário* de um sistema de informação envolvendo imagens fornece informações que descrevem o domínio da aplicação, e os tipos de operação, basicamente as consultas, a serem realizados sobre as imagens. Neste nível as operações realizadas consistem na execução do raciocínio espacial, comparação de padrões selecionados, intensidade de cores, entre outros. O *nível de Caracterização*

Semântica descreve o domínio da aplicação sob a perspectiva do usuário.

Representa-se os tipos de imagens e o que se espera identificar em seu conteúdo, ou seja, imagens do domínio de aplicação médica contendo órgãos do corpo humano, por exemplo, coração, pulmão, rim, ou aplicação de gerenciamento de projetos de engenharia mecânica, no qual nas imagens são identificados pistões, válvulas ou modelos de peças internas do motor. A estes vários elementos correspondem características que, uma vez extraídas da imagem, possam ser utilizadas para representá-las. Por exemplo, um pistão no projeto do motor corresponde à identificação de um círculo; um coração pode ser identificado a partir da distribuição de níveis de cinza que seja representativo para este órgão, e assim com cada elemento indicado como significativo na imagem. A organização destas informações corresponde ao *nível de Caracterização da Imagem*.

O *nível de Representação das Características* suporta estruturas que permitem representar os elementos identificados em uma imagem. Um círculo localizado pode ser representado através dos pontos que definem o seu contorno ou por um ponto com as coordenadas do centro e a medida de seu raio. Uma distribuição de cores pode ser indicada por um vetor que contenha todo o seu histograma, ou, simplifiadamente, através dos valores calculados para sua média, desvio padrão e variância. Por fim, o nível de *Organização das Características* corresponde ao nível físico da arquitetura de BD, indicando as estruturas de dados para o armazenamento físico dos elementos descritivos das características da imagem e de suas estruturas de acesso.

Resumindo, um Banco de Imagens manipula informações que podem ser classificadas nas seguintes categorias [Grosky_89]:

- Imagem: informação em formato digital representando algum aspecto relevante da aplicação, armazenada no BD;
- Dados relacionados com a Imagem: inclui informação de registro da imagem, tais como meio de aquisição, data, formato, resolução, tamanho, descrição textual de seu conteúdo e outros;
- Informação extraída da Imagem: objetos e características de descrição do conteúdo da imagem, resultantes de seu processamento e análise;
- Relacionamentos com os demais elementos presentes no domínio da aplicação: entidades do mundo real que a imagem descreve ou com as quais possui algum relacionamento;
- Entidades e relacionamentos identificados na aplicação: consistem no restante das informações manipuladas por um BD.

Um Sistema de Gerenciamento de Banco de Dados e Imagens deve, portanto, fornecer mecanismos uniformes para o armazenamento e manipulação dos vários aspectos das informações envolvidas com imagens. Todas as vantagens presentes nos Sistemas de Gerenciamento de Banco de Dados (SGBD), tais como independência, integridade, compartilhamento, controle de redundância e segurança de acesso aos dados, são importantes para o Sistema de Informação com Suporte a Imagens.

2.2.3. Características do tipo de dado Imagem.

A arquitetura descrita no item anterior permite identificar os elementos a serem especificados no processo de incorporação de uma imagem no BD. No

entanto, para que este processo atinja o seu objetivo final de forma satisfatória, devem ser consideradas algumas características particulares das informações contidas em uma imagem, tais como:

1. *Impossibilidade de se descrever precisamente o conteúdo de uma imagem:*

a forma exata de descrição da informação presente na imagem consiste da própria imagem. Qualquer metodologia utilizada em sua representação irá provocar perdas de informação. Assim, a descrição do conteúdo de uma imagem consiste em um processo dependente da análise e processamento da imagem. Interpretações sobre os seus elementos podem sofrer alterações à medida que estas técnicas sejam aprimoradas. Além disso, nem sempre o conjunto de interpretações obtido é único e preciso;

2. *Dificuldade na associação de significado semântico à imagem e seus*

componentes: as entidades espaciais ou objetos e seus relacionamentos não carregam um significado semântico em si. Em uma aplicação convencional, quando existe um relacionamento do tipo “Empregado Gerencia Departamento”, claramente está indicado o significado da associação existente entre as duas entidades, através dos seus nomes de indentificação. No entanto, a definição da semântica através de um nome descritivo não é suficiente para representar a informação contida na imagem como consequência de alguns problemas. Primeiro, uma mesma imagem pode ser interpretada de diferentes maneiras. Segundo, esta imagem pode ser utilizada para diferentes necessidades em períodos distintos. Por fim, uma vez que a interpretação indicada consiste de uma aproximação da informação realmente presente, esta pode

sofrer alterações à medida que as técnicas de processamento sejam aprimoradas e/ou a aplicação seja melhor compreendida.

3. *Consultas ao BD de imagens formuladas através da indicação de outras imagens*: uma imagem (ou a informação baseada em imagens) pode ser consultada através de outras imagens. A forma de consulta a ser definida também depende dos elementos de modelagem presente, ou seja, das informações sobre as imagens que se encontram disponíveis e possam ser utilizadas em sua formulação. Neste aspecto o domínio de conhecimento sobre as imagens pode desempenhar um papel de auxílio na solução das consultas, fornecendo um elemento de apoio ao usuário, para que este possa refinar ou dirigir a sua consulta em direção ao resultado desejado.

2.3. ELEMENTOS DE UM SISTEMA DE PROCESSAMENTO DE IMAGEM DIGITAL.

Os componentes gerais de um sistema de processamento de imagens digitais pode ser visualizado na Figura 4 [Gonzalez_87]. O **processador de imagens** consiste nos elementos de hardware para desempenhar as seguintes funções: aquisição de imagens, armazenamento, processamento de baixo nível (ou processamento rápido) e exibição da imagem final.

O **digitalizador** é responsável pela incorporação da imagem no sistema, através da transformação de informações presentes em sinais analógicos em dados digitais que possam ser interpretados na forma de imagens. Estes elementos podem ser generalizados para a terminologia de dispositivos de captura de imagens. Nesta categoria se encaixa, por exemplo, equipamentos como o tomógrafo por ressonância magnética, que fornece imagens de porções internas do corpo humano. Os

dispositivos de armazenamento fornecem o meio físico em que as imagens permanecem após serem apresentadas no elemento de exibição.

Figura 4 - Elementos Básicos de um Sistemap de Processamento de Imagens.

2.4. PROCESSAMENTO DE IMAGENS PARA DESCRIÇÃO DE SEU CONTEÚDO.

Uma vez que o sistema possui um conjunto de imagens a serem armazenadas, o processo seguinte consiste na extração de características que possam ser utilizadas como elemento descritivo do seu conteúdo, representando-o da forma mais precisa possível. Estes elementos serão importantes também para o processo de especificação das estruturas de organização e acesso às imagens.

As técnicas e algoritmos desenvolvidos na área de Processamento de Imagens são empregadas para que características significativas presentes na imagem possam ser identificadas e utilizadas em sua descrição. Neste sentido podem ser utilizadas técnicas que procuram melhorar a qualidade de uma imagem, aplicando por exemplo os algoritmos de realce, filtros de suavização e eliminação de ruídos, que produzem

uma imagem com maior precisão de detalhes de acordo com um determinado critério, escolhido a partir dos requisitos da aplicação.

Além disso, as principais técnicas de processamento de imagens para a formação do Banco de Imagens são as consideradas descritivas, ou seja, determinação de histograma com a distribuição de cores, texturas, formas de objetos e comparação (“matching”) de padrões. Estas técnicas extraem informações das imagens que podem ser utilizadas como elementos para a identificação e formulação de consultas que descrevam as imagens através de características que sejam estritamente gráficas.

No entanto, além das características gráficas, torna-se necessário a presença de elementos descritos textuais representando informações que não possam ser extraídas através de um procedimento automático. As descrições gerais sobre o tema principal da imagem, objetos presentes, identificação da aplicação e outras são também importantes para a sua completa caracterização. Nesta seção serão discutidos os métodos e abordagens utilizadas para a descrição da informação presente em uma imagem.

2.4.1. Descrição Textual do Conteúdo da Imagem.

Uma das abordagens usuais para a descrição do conteúdo de uma imagem consiste em indicá-lo através de uma informação textual. Esta informação pode ser armazenada na forma de atributos associados à imagem. O procedimento de extração destas informações é manual e se utiliza de um especialista no domínio da aplicação que seja capaz de identificar e reconhecer todos os objetos e detalhes apresentados na imagem. No entanto, por maior que seja o conhecimento do especialista, este não irá

conseguir descrever com detalhes todo o conteúdo da imagem. Por outro lado, esta descrição representa a “visão” particular de um especialista sobre a imagem, e cada usuário pode apresentar uma percepção diferente e a indicar de forma distinta.

Outro problema com esta abordagem consiste na representação do conjunto de atributos que possam ser apropriados na descrição do conteúdo. Estes atributos não podem ser puramente textuais, mas podem representar de forma apropriada os detalhes relevantes ao domínio da aplicação sobre a imagem, uma vez que devem possibilitar a sua recuperação a partir de consultas submetidas ao BD.

2.4.2. Extração de Características Gráficas da Imagem.

A disciplina de Processamento de Imagens fornece uma série de métodos que possam ser aplicados sobre uma determinada imagem e resultar em um conjunto de parâmetros tais que a caracterizem [Gonzalez_87] [Schalkoff_89]. Dentro da abordagem da construção de um Banco de Imagens, os aspectos que envolvem o comportamento de uma determinada técnica devem ser tratados pela área de pesquisa equivalente. No entanto, o resultado da aplicação de cada método é importante, uma vez que terá consequência direta sobre as estruturas de organização e representação a serem formuladas para as imagens.

Na análise de imagens, a entrada do processamento é uma imagem, enquanto a saída é uma descrição não gráfica da mesma. Este processo pode ser entendido como uma redução ou compactação de dados, em que se diminui o volume de dados mas mantém-se o conteúdo da informação relevante para uma determinada aplicação [Mascarenhas_89]. Na manipulação de imagens, em geral, tanto as entradas quanto

as saídas são imagens. As transformações que realizam manipulação nas imagens são divididas em:

- a) Radiométricas, em que os valores de níveis de cinza dos pontos da imagem são alterados, sem modificação de sua geometria; e
- b) Geométricas, em que a geometria da imagem é alterada, mantendo-se o máximo possível os valores dos níveis de cinza.

Os dois tipos de transformações podem ser realizadas com a finalidade de eliminar distorções na imagem, introduzidas geralmente pelo próprio sistema de aquisição de imagens, denominando-o de restauração, ou enfatizar certas características das imagens. Para a definição de um Banco de Imagens, é relevante o conjunto de técnicas que definem a sua Análise, uma vez que resulta em parâmetros mais apropriados à formulação de uma estrutura de acesso e consultas. As técnicas de manipulação de imagens são úteis no processo de apresentação da imagem adquirida, na interface do sistema, ou como elemento de auxílio na extração semi-automática de informações da imagem.

No entanto, o método de análise a ser aplicado é dependente do domínio da aplicação. As aplicações de GIS, sistemas de gerenciamento de projetos CAD/CAM, apresentam como requisito a identificação de elementos geométricos, ou seja, pontos, polígonos, regiões, planos de visões de sólidos, e outros. Em aplicações médicas os algoritmos podem procurar por elementos que não apresentam um formato geométrico claramente definido, operações de busca de padrão baseadas na definição de regiões [Senzako_96], descrição da imagem por distribuição de cores e outros.

2.5. ESTRUTURAS DE ORGANIZAÇÃO E ACESSO A IMAGENS.

O mecanismo de representação de um índice para imagens define como os elementos de descrição do seu conteúdo serão estruturados para constituir em sua chave de acesso, e, conseqüentemente, às demais informações que se encontram associadas à imagem. Assim as estruturas de índices são adequadas para cada esquema de representação que seja proposto.

Com relação aos elementos de acesso às imagens, ou seja, as técnicas de indexação, estes podem ser estudados sobre três aspectos: extração, representação e organização dos índices. A **extração** é altamente dependente das técnicas de processamento de imagens que estejam disponíveis e de sua precisão no reconhecimento de características das imagens, as quais foram discutidas na seção anterior. Em aplicações convencionais de sistemas de Banco de Dados, que não envolvam informações associadas às imagens, técnicas de indexação baseadas em palavras-chaves, tais como, nome do empregado, código do projeto, são suficientes para suportar as necessidades do usuário, expressadas através das consultas. Uma aplicação na qual a presença de imagens seja relevante em sua modelagem apresenta requisitos de recuperação de informações que não podem ser adequadamente indicadas somente através de palavras-chaves. Além deste aspecto, torna-se necessário o acesso a imagens com a indicação de outros parâmetros de descrição, tais como, forma, textura, distribuição de cores, relações espaciais, entre outras. Isto significa que as características das imagens definem elementos que podem ser utilizados como índices de imagens. Em consequência das particularidades

apresentadas por estas informações, a **representação** da estrutura de índice deve considerar alguns aspectos peculiares, tais como [Chang_92b]:

- Os índices, por indicar características extraídas das imagens, consistem em uma representação aproximada para a chave de acesso;
- Os índices de imagens não apresentam uma ordem intrínseca, no sentido em que sendo definido a , b , e c como valores para um determinado índice, e estabelecendo $a < b < c$, não necessariamente implica que a imagem com chave de acesso b seja mais similar à imagem a do que à imagem c . Além disso, naturalmente para algumas características de descrição de imagens, os valores que as descrevem não possuem representação direta de seu domínio no espaço euclidiano, o que pode tornar complexa a atividade de definir uma operação de comparação;
- As representações dos índices podem apresentar atributos interrelacionados, ou seja, se $a1$ e $a2$ são atributos de um índice, então o resultado da aplicação de uma função envolvendo a tupla $(a1, a2)$ poderá ser diferente da interseção do resultado da aplicação da mesma função separadamente sobre $a1$ e $a2$.

Uma vez definido o método de indexação para acesso às imagens deve-se estabelecer as estruturas de dados necessárias para a **organização** destas informações na forma de chaves. Cada parâmetro selecionado na representação das chaves de acesso pode ser descrito por outros elementos, tais como, a forma de um objeto, que pode ser definida a partir de sua borda de contorno, maior eixo de direção, retângulo ou raio do círculo de delimitação e outros. Assim a especificação da estrutura de

dados deve suportar o tratamento de chaves de acesso multidimensionais, ou seja, vários atributos descrevendo o mesmo aspecto da informação ou imagem.

Figura 5: Dimensões da Indexação de Imagens

Para se classificar as abordagens das técnicas de indexação pode-se utilizar três dimensões (Figura 5): a) Organização: define como estruturar as informações das imagens, através da utilização de árvores-B, Quad-tree, árvores-R e outras; b) Representação: seleciona os índices para as imagens, ou seja, corresponde à indicação de palavras-chaves, descritores de padrões, string-2D, assinaturas, e outros e, c) Extração: define as técnicas para obter os valores dos índices de imagens. Nos itens seguintes serão apresentadas algumas estruturas que permitem realizar tanto a representação quanto a organização dos índices de acesso às imagens.

2.5.1. Quad-Trees.

A estrutura de dados quad-tree [Samet_84] é definida com base no princípio da decomposição recursiva em sub-regiões do espaço representado em uma imagem, e assim sucessivamente até que um determinado critério seja completamente obedecido pela sub-região. As quad-trees podem ser diferenciadas através do tipo de

dado representado, o princípio utilizado no processo de decomposição, e sua resolução (número de decomposições aplicadas). No geral estes fatores são pré-definidos ou se adaptam às características dos dados de entrada. Com respeito ao tipo de dado têm-se definido quad-trees que suportem a representação de pontos, linhas, regiões e volumes (octrees).

A estrutura da quad-tree para a representação de informações pontuais, por exemplo, a coordenada de localização de uma cidade em uma aplicação de Geoprocessamento, pode ser definida como a generalização da árvore de busca binária para um espaço multidimensional. Em duas dimensões cada ponto consiste em um nó da árvore com quatro filhos, cujas subárvores definem as principais direções a partir do ponto inicial, indicadas na sequência Noroeste, Nordeste, Sudoeste e Sudeste. O processo de inserção de um ponto nesta quad-tree é semelhante ao utilizado na árvore de busca binária. O nó apropriado para a inserção é definido através de comparações com os valores das coordenadas x e y , realizadas em todas as direções. De forma semelhante, o padrão da árvore resultante depende da ordem de inserção dos registros. Esta estrutura é particularmente válida para aplicações que envolvem a realização de busca contendo parâmetros de localização ou a determinação da vizinhança de um determinado ponto. Realizando comparações com os valores das coordenadas, pode-se delimitar o espaço de busca e consequentemente as subárvores a serem percorridas.

Para dimensões elevadas, cada nó da quad-tree passa a apresentar um número elevado de filhos (2^k , para k dimensões). Neste caso, um incremento em sua representação é realizado através da árvore $k-d$ (" $k-d$ tree") [Robinson_81], também

semelhante à árvore binária; no entanto, em cada nível é realizado o teste sobre uma coordenada diferente. Assim, para o caso de duas dimensões (árvore 2- d), para o nó raiz e todos os ímpares testa-se o valor da coordenada- x , e os demais dos níveis pares, testa-se o valor da coordenada- y . Esta organização é superior à das quad-trees para pontos, sendo possível processar operações de comparação em paralelo, percorrendo partes distintas da árvore.

A abordagem mais utilizada de quad-tree consiste na representação de *regiões*, a qual é baseada em sucessivas divisões do espaço da imagem em quatro

Figura 6: Divisão do polígono para a Quad-tree.

quadrantes com áreas iguais, até que toda a informação presente em uma subregião seja uniforme. Isto significa que esta organização é adequada quando as imagens apresentam partes homogêneas. Este critério de

homogeneidade depende do tipo de dado que pertence à região a ser representada. Considerando uma imagem binária contendo apenas os valores branco (0) e preto(1), uma região homogênea é definida como inteiramente composta por 0's ou 1's. No entanto, pode-se definir intervalos para a uniformidade, constituindo em, por exemplo, níveis de cinza na imagem, ou estabelecer critérios que indiquem a presença de algum elemento, como por exemplo, a presença da borda de um desenho. Neste caso a divisão ocorrerá para os quadrantes que apresentarem intersecção com a

borda (Figura 6), repetido até que todos os contornos da figura se encontrem representados [Hunter_79].

O tratamento de informações volumétricas, ou seja, objetos em três-dimensões (3D), é realizado através das *octrees*, nas quais o processo de divisão é semelhante ao utilizado para a representação de regiões, no entanto produzindo “octantes”, representando cubos no espaço 3D contendo informações homogêneas. Além do tratamento de informações no espaço 3D, esta estrutura pode ser utilizada no auxílio à soluções de problemas em robótica, como, por exemplo, a determinação de caminhos através do conhecimento prévio dos obstáculos.

A forma tradicional de se representar uma quad-tree é a estrutura de árvore. No entanto, algumas extensões têm sido propostas no sentido de elaborar alternativas a este processo. A *quad-tree linear* é uma organização na qual uma quad-tree é representada como uma lista linear [Unnikrishnan_88] [Wang_91]. Cada nó desta lista, denominado de *código da quad-tree*, é definido para uma imagem de $2^n \times 2^n$ da seguinte forma:

$$Q = d_1 d_2 \dots d_n,$$

onde, $d_i \in 0,1,2,3,X$ para $i=1,2,\dots,n$, e X é um dígito maior do que 3, e sendo $d_i = X$, então para cada $i \leq j \leq n, d_j = X$; n define a resolução. Estes elementos decrevem o quadrante, em função de sua localização e do número de divisões necessárias à sua identificação.

A quad-tree linear consiste em uma lista ordenada de códigos de quad-tree, representando as informações espaciais em um formato semelhante às linhas de rastreamento (“raster”). Através desta representação pode-se definir uma série de

operações que sejam fundamentais para a manipulação da informação espacial, tais como, área, continência, localização, distância e determinação de vizinhança, realizando cálculos sobre os valores dos códigos dos quadrantes [Wang_91]. Esta representação pode ser diretamente utilizada em conjunto com um modelo de dados, por exemplo, o Relacional, para modelar as informações presentes em uma imagem com elementos espaciais. A sua descrição é tratada como um conjunto de valores, os

Figura 7: Aspectos lógicos da quadtree totalmente invertida.

códigos de quadrantes, que podem representar regiões, linhas ou pontos na imagem. Uma região é caracterizada como um conjunto de códigos de quad-tree $R = \{Q_1, Q_2, \dots, Q_n\}$, onde cada $Q_i, (1 \leq i \leq n)$

indica um quadrante em R . Um ponto pode ser representado através de um código simples, e as linhas são definidas como uma

sequência de códigos indicando seus vértices.

Algumas extensões das quad-trees foram propostas com o objetivo de torná-la uma estrutura para a indexação de imagens, realizando suporte de gerenciamento do meio de armazenamento externo. A quadtree totalmente invertida [Vassilakopoulos_95] define uma estrutura de indexação para um banco de imagens com a utilização de espaços contínuos no disco para

Figura 8: Aspectos lógicos da quadtree ID.

o armazenamento dos nós da árvore. Cada nó é formado por um número de bits igual ao número máximo de imagens que possam ser armazenadas no Banco. Cada bit é utilizado como um “flag” para sinalizar a presença da imagem neste nó da árvore, significando que a imagem possui uma informação no quadrante representado por este nó. O termo *invertida* é utilizado em consequência de que a árvore armazena identificadores da imagem, e não os dados que a compõem (Figura 7). Outra variação utilizada como organização da estrutura de índices consiste na *quadtree invertida dinâmica* (quadtree ID) [Vassilakopoulos_95] definida pelos componentes: i) Posterior, consistindo de um arquivo de listas com identificadores distintos das imagens, e ii) Anterior, consistindo em uma quad-tree, na qual cada nó contém uma indicação para as listas. Um exemplo da quadtree ID aparece na Figura 8 onde cada nó da lista pode suportar no máximo duas referências às imagens. Para inserir uma nova imagem nesta estrutura, inicialmente é realizada a representação na quadtree, produzindo um conjunto de nós, onde o critério de homogeniedade seja válido. Para cada um destes nós equivalentes presente na estrutura anterior da quadtree ID, é inserida uma entrada de identificação da nova imagem no arquivo de listas. Esta organização é considerada superior à da quadtree totalmente invertida, uma vez que realiza uma melhor utilização do espaço em disco, e consiste de uma estrutura dinâmica, não sendo obrigatória uma reorganização da árvore após uma série de inserções de novos elementos.

2.5.2. Strings 2D.

A utilização de strings 2D [Chang_87] [Costagliola_95] como estrutura de acesso a imagens, sugere um processo de indexação através de ícones. Esta

abordagem baseia-se no princípio de que as informações espaciais presentes em uma imagem real podem ser adequadamente representadas por uma imagem simbólica, ou seja, uma matriz de símbolos, em que cada um corresponde a algum objeto da imagem. A representação de uma imagem simbólica através de uma string 2D é derivada a partir de projeções ortogonais nas linhas e colunas dos símbolos presentes. Portanto através das strings 2D obtém-se uma representação compacta da informação existente em uma imagem real, sendo adequada à especificação de consultas do tipo: “encontre todas as imagens contendo uma árvore à esquerda de uma casa”.

A abordagem de descrição do conteúdo de uma imagem através de elementos simbólicos, tais como ícones, sugere a definição de uma imagem simbólica, formalmente descrita como [Costagliola_95]: “Dado um conjunto V de símbolos,

uma *imagem simbólica* f

sobre V é um mapeamento

$M \times N \rightarrow W$, onde $M =$

$\{1, 2, \dots, m\}$, $N = \{1, 2, \dots,$

$n\}$ e W é um produto

cartesiano em V .”

Figura 9: A imagem e sua representação simbólica.

A Figura 9 mostra o

exemplo de uma imagem e sua representação simbólica. O conjunto de símbolos é dado por $V = \{a, b, c, d\}$, os quais correspondem aos objetos encontrados na imagem real, ou seja, a - conqueiros, b - ave, c - barco, d - sol, e $m=n=3$, sendo que m e n especificam o tamanho da imagem em função da resolução espacial da grade. Além

disso, a posição dos símbolos na grade corresponde à localização espacial dos objetos na imagem. No entanto, dependendo da aplicação, os elementos significativos de uma imagem podem ser pontos, linhas, regiões, objetos ou outros.

A representação do conteúdo de uma imagem por um conjunto de símbolos sugere definir uma estrutura de indexação baseada em ícones. Os ícones correspondem diretamente aos símbolos utilizados neste processo. Diferente dos índices convencionais que referenciam diretamente a classe de objetos indexada, a indexação baseada em ícones indica como é representado o objeto a ser acessado.

Uma estrutura de indexação por ícones apresenta as seguintes propriedades:

- a) *Construção*: dado um objeto, deve ser possível extrair um ícone que o represente;
- b) *Reconstrução*: a partir de um índice de ícones, será reconstruído de forma eficiente o objeto representativo da classe correspondente. Esta propriedade permite a formulação de consultas, em que é apresentado o esboço da imagem através da utilização dos ícones;
- c) *Não Ambigüidade*: duas classes de objetos não são representadas pelo mesmo ícone, ou seja, fornecido um ícone deve-se restaurar somente um objeto;
- d) *Compactação*: um ícone irá capturar apenas as informações que sejam necessárias para distinguir entre classes de objetos, eliminando as que não sejam necessárias, reduzindo o tamanho do BD.

A partir da definição de uma imagem simbólica, seja V um conjunto de símbolos ou vocabulário onde cada símbolo representa um objeto, tais como uma “casa”, uma “árvore”, e outros. Considerando $A = \{ "=", "<", ":" \}$ como o conjunto dos relacionamentos espaciais entre os objetos, uma string 2D é definida como

[Costagliola_95] [Chang_87]: “Uma string 2D, escrita como (u,v) , sobre o conjunto V é definida através do par de sequências:

$$(x_1y_1x_2y_2\ldots y_{t-1}x_t, x_{p(1)}z_1x_{p(2)}z_2\ldots z_{t-1}x_{p(t)}),$$

onde $x_i \in V$ e y_i, z_i são símbolos extraídos de A , e $p: \{1, \ldots, t\} \rightarrow \{1, \ldots, t\}$ consiste de uma função de permutação, que permite com que ambas as sequências sejam consistentes na representação da imagem simbólica. O símbolos do conjunto A apresentam como significado:

“=” : define a relação espacial “na mesma posição espacial que”;

“<” : na string u indica a relação espacial “esquerda-direita”, e na string v “embaixo-acima”;

“.” : representa a relação “no mesmo conjunto que”.

A string-2D referente à Figura 9 consiste em $(u,v) = (a=d<a=b<c, a=a<b=c<d)$, onde $p=13452$ e :

$x_1x_2x_3x_4x_5$ consiste em $a d a b c$;

$y_1y_2y_3y_4$ consiste em $= < = <$;

$z_1z_2z_3z_4$ consiste em $= < = <$.

Uma representação reduzida para a string 2D indicada anteriormente é obtida, eliminando-se os símbolos “.” e “=” e suas combinações com “<”, resultando em $(u,v) = (ad<ab<c,aa<bc<d)$. A string 2D denominada de absoluta para o exemplo anterior é a seguinte: $(a = = d < a = b = < = c =, a = a = < = b = c < d =)$, na qual encontram indicadas todas as relações espaciais dos símbolos presentes na Figura 9, não sendo, entretanto, uma representação eficiente, por produzir uma sequência longa a ser processada e apresentar relações espaciais redundantes.

A partir da existência de uma string 2D pode-se realizar o processo inverso, ou seja, obter a imagem simbólica que se encontra representada. Este procedimento fica facilitado através da presença da função de transição como parte integrante da string 2D. Para o exemplo anterior, a string 2D fica $(ad<ab<c, aa<bc<d, 13452)$, sendo que a segunda sequência é obtida a partir da inicial, pode-se indicar apenas os índices necessários à sua formação a partir da função de transição, ou seja, $(ad<ab<c, 13<45<2)$. A string 2D da forma (u,v,p) é denominada de string aumentada.

A utilização da string-2D como chave de acesso para imagens, representadas através de símbolos ou ícones, pode ser atingida de forma natural. A consulta por uma imagem é realizada indicando-se a sua representação simbólica, a qual convertida para uma string 2D, será comparada com as demais existentes, na procura da imagem mais semelhante à apresentada como parâmetro de busca. Desta forma, implementa-se uma estrutura de índices baseada em ícones representados por sua string 2D, na qual a operação de recuperação das imagens é indicada através da comparação de duas strings 2D [Chang_88], que consiste basicamente em localizar uma subsequência de símbolos no interior de alguma string 2D representando as imagens presentes no Banco de Dados.

a	a
a	a

Figura 10: Padrões ambíguos.

A representação de uma imagem simbólica através de uma string 2D pode não ser única, e também a imagem obtida a partir do processo de reconstrução pode ser diferente da original. Extensões como a string 2D não redundante [Costagliola_95] têm sido

especificadas com o objetivo de solucionar este problema. Considere as imagens simbólicas f e g presentes na Figura 10. Estas são ambíguas, pois apresentam a mesma string 2D de representação, sendo imagens distintas:

e		e	d
	c		c
b		b	d
a	a	a	d

($a < a$, $a < a$).

Em string 2D não redundante os símbolos repetidos são representados uma única vez. Considerando a imagem simbólica representada ao lado, a string 2D é ($abe < ac < abe < ddcd$, $aaad < bbd < cc < eed$), e a string 2D não

redundante consiste em ($abe < ac < abe < dc$, $ad < bd < c < ed$). Com esta formulação obtém-se uma simplificação no índice gerado para as imagens, e a mesma pode ser reconstruída, considerando que a imagem F representada seja não ambígua, então um símbolo a terá múltiplas ocorrências em u (ou v) se e somente se possuir um única ocorrência em v (ou u). A partir desta propriedade as coordenadas de cada objeto, indicado pela tripla (a , $x\text{-coord}$, $y\text{-coord}$), onde a é o símbolo, são determinadas por:

- Definindo $X_a = \emptyset, Y_a = \emptyset$ para cada símbolo a de V e seja $R =$ o conjunto das coordenadas;
- Para cada ocorrência do símbolo a em u (ou v), insira o valor inteiro ($1 +$ o número de $<$ precedendo a) no conjunto X_a (ou Y_a);
- Para cada símbolo a em u , adicione ($\{a\}, X_a, Y_a$) em R .

Para a string 2D não redundante apresentada anteriormente, estes passos produzem os seguintes conjuntos:

$$\begin{array}{llll} X_a = \{1,2,3\} & Y_a = \{1\} & X_b = \{1,3\} & Y_b = \{2\} \\ X_c = \{2,4\} & Y_c = \{3\} & X_d = \{4\} & Y_d = \{1,2,4\} \end{array}$$

$$X_e = \{1,3\} \quad Y_e = \{4\}$$

e o conjunto $R = \{(a \ 1 \ 1), (a \ 2 \ 1), (a \ 3 \ 1), (b \ 1 \ 2), (b \ 3 \ 2), (c \ 2 \ 3), (c \ 4 \ 3), (d \ 4 \ 1), (d \ 4 \ 2), (d \ 4 \ 4), (e \ 1 \ 4), (e \ 3 \ 4)\}$. Desta forma obtém-se o conjunto com a descrição precisa dos objetos presentes na imagem e suas respectivas localizações espaciais.

As imagens simbólicas bidimensionais são reduzidas para representações em uma dimensão utilizando-se de projeções dos símbolos nos eixos de coordenadas. Um processo análogo ocorre para a representação de imagens simbólicas em um espaço 3D. Neste caso, as projeções são realizadas em direção aos planos cartesianos (xy, yz, zx) , resultando em uma representação em 2D [Costagliola_92].

2.5.3. Árvores-R.

As árvores-R [Guttman_84] foram propostas como uma extensão das árvores-B para a manipulação de informações com mais de uma dimensão, combinando suas vantagens com as apresentadas pelas “quadrees”. Como as árvores-B, esta é uma estrutura dinâmica, ou seja, permanece balanceada durante as operações de inserção e remoção, e de forma semelhante às “quadrees” suporta o tratamento dos espaços vazios (“dead spaces”) presentes nas imagens, não provocando a criação de novos nós que os representem na árvore. Além disso, ao nível dos nós folhas, encontra-se referenciada toda a abrangência espacial de um objeto, e por sua semelhança com as árvores-B, possuem uma implementação eficiente com relação à paginação do espaço em disco.

As árvores-R foram propostas como uma técnica de indexação que permite

Figura 11: Retângulos para a árvore-R

pesquisa espacial com acesso direto ao objeto indicado. No entanto, ela pode ser utilizada como uma metodologia de representação da informação espacial presente na imagem, consistindo de uma estrutura adequada para chaves de

acesso multidimensionais.

Um nó folha da árvore-R contém entradas da forma $(oid, rectg)$, onde oid refere-se a um registro no BD contendo o objeto espacial, e $rectg$ define um

Figura 12: Árvore-R correspondente a Figura 11.

retângulo com n-dimensões que consiste no limite de contorno do objeto espacial indexado:

$rectg = (I_0, I_1, \dots, I_{n-1})$, onde n é o número de dimensões e I_i consiste em um intervalo fechado $[a, b]$ descrevendo a extensão do objeto ao longo da dimensão i . O espaço delimitado por $rectg$ é definido como o retângulo mínimo de delimitação (MBR) de um objeto.

Por outro lado, um nó não-folha da árvore-R contém entradas que são da forma $(cp, rectg)$, onde cp é um indicador para o nó filho da árvore a partir do atual, e $rectg$ é o retângulo de delimitação de todos os retângulos presentes nos nós filhos. Por exemplo, para uma aplicação onde a chave de acesso contenha dimensão igual a

2 (dois), o retângulo apresenta coordenadas $(x_{inf}, y_{inf}, x_{sup}, y_{sup})$. As figuras 11 e 12 apresentam a formação de uma árvore-R para uma imagem simbólica.

Definindo M como o número máximo de entradas para cada nó da árvore e seja m o número mínimo de chaves por nó, obedecendo a $2 \leq m \leq M / 2$, uma árvore-R satisfaz as seguintes propriedades:

- Cada nó folha apresenta entre m e M valores chaves, do contrário consiste do nó raiz;
- Cada nó não-folha apresenta entre m e M nós filhos, do contrário consiste do nó raiz;
- O nó raiz apresenta no mínimo dois filhos, do contrário consiste em um nó folha;
- Todos os nós folhas da árvore-R aparecem em um mesmo nível.

Para efeito de análise da eficiência da operação de busca em uma árvore-R, devem ser considerados os conceitos de *sobreposição* e *abrangência*. A abrangência é definida como a área total de todos os retângulos presentes nos nós-folha da árvore-R, e a sobreposição como o total de área em comum entre dois ou mais retângulos. Assim, os algoritmos de busca e inserção de um novo objeto e seu retângulo de delimitação deve provocar um valor mínimo para estes dois parâmetros. Do contrário, não há como garantir a existência de um caminho único de busca envolvendo uma determinada consulta por algum objeto ou imagem.

O algoritmo de escolha de uma subárvore para a inserção de uma chave procura determinar o melhor retângulo considerando a menor área de expansão necessária para acomodar o novo objeto. Para os casos nos quais ocorram igualdade neste critério, é selecionado o retângulo com menor área, em função de se procurar

diminuir o fator de abrangência da árvore. Quando a inserção de um novo objeto provocar uma violação no limite máximo de chaves, ocorre divisão deste em dois outros. Este algoritmo procura minimizar a área necessária à formação dos dois novos retângulos, selecionando para a presença em cada um as entradas que se encontram mais distantes entre si, não aumentando desta forma a região de sobreposição.

Além de considerar os conceitos de sobreposição e de abrangência para a definição da eficiência dos algoritmos da árvore-R, outros critérios de otimização podem ser estabelecidos [Beckmann_90]:

- A área do retângulo de diretório a ser minimizada: significa que a área coberta pelo retângulo de delimitação de um nó que não se encontra contida nos retângulos presentes em seu interior (“espaços vazios”) deve ser minimizada. A melhoria no desempenho é obtida, uma vez que a decisão sobre os caminhos a serem percorridos pode ser realizada em níveis mais elevados da árvore;
- A área de sobreposição dos retângulos de diretório deve ser minimizada, o que pode também provocar uma redução no número de caminhos a serem percorridos;
- As margens do conjunto de retângulos devem ser minimizadas: definindo-se margem como a soma dos comprimentos das bordas de delimitação, minimizar o seu valor implica em obter um retângulo que se aproxime da forma quadrática;
- Otimização do espaço de armazenamento: a melhor utilização do espaço disponível para o armazenamento provoca um menor custo no processamento de uma consulta, uma vez que a altura da árvore diminui.

Com o objetivo de atingir estes critérios foram propostas variações na definição original da árvore-R, que basicamente inserem alterações na forma como os algoritmos se comportam para a determinação de uma subárvore e no procedimento de divisão de um nó. Outros incrementos alteram a definição do retângulo de delimitação, permitindo que seus eixos de formação não sejam necessariamente paralelos aos eixos do sistema de coordenadas do espaço. Estas modificações serão discutidas com maiores detalhes nos itens a seguir.

No entanto, propostas foram realizadas no sentido de incrementar o desempenho da árvore-R sem inserir alterações em sua conceituação original. O objetivo principal consiste em explorar o paralelismo presente nas operações envolvendo consultas espaciais, minimizando o tempo necessário para processar a estrutura de acesso [Kamel_92]. A técnica sugere uma alteração na organização do hardware, através da utilização de várias unidades de disco para armazenar partes da árvore-R, controladas por um único processador, com o objetivo de proporcionar um paralelismo nas operações de acesso a disco. As operações sobre a árvore-R permanecem as mesmas quando considerada a existência de um único disco, sendo inserido apenas o aspecto de selecionar em qual disco será criado um novo nó. O princípio básico consiste em definir um índice de proximidade, que indica a similaridade do novo nó com os demais da árvore existente em cada unidade de disco. Dois critérios podem ser considerados para definir a eficiência da estratégia de particionamento:

- consultas envolvendo pequenas regiões de busca devem ativar o menor número possível de unidades de disco;

- consultas envolvendo a recuperação de um grande volume de dados devem ativar o número máximo de unidades de disco, explorando o paralelismo nas operações de entrada/saída em disco.

As estratégias de particionamento consistem em:

- a) Árvores-R independentes: considera a existência de várias árvores-R independentes para cada unidade de disco. Os métodos para a distribuição dos nós podem percorrer a árvore, associando cada nó a uma unidade, ou aplicando uma função de busca (*hash*) que proporcione uma distribuição uniforme. Outra alternativa, denominada de particionamento do espaço, consiste em particionar a árvore-R a partir do primeiro nível após o nó raiz, definindo cada nó filho a partir da raiz, como raiz das subárvores independentes, alocadas em cada disco;
- b) Super-Nós: considerando que cada nó da árvore-R utilize d páginas no disco, então estas podem ser uniformemente distribuídas ao longo de d unidades de disco;
- c) Árvores-R Multiplexada: este esquema utiliza uma única árvore-R, cujos nós são integralmente inseridos em cada unidade de disco, existindo elementos de ligação entre estas. A distribuição de nós dentro desta organização será realizada para obter um balanceamento das informações no disco de acordo com os *dados*, contendo um número de nós iguais, a *área* coberta pelos retângulos alocados a cada unidade de disco, e a *proximidade*, considerando a associação em uma mesma unidade, de retângulos que possam ser recuperados em uma única consulta. Esta medida de proximidade é realizada através da proporção de consultas que recuperam ambos conjuntamente, dentre o total submetido ao BD.

Dos métodos citados anteriormente, a árvore-R multiplexada apresenta um resultado mais adequado, considerando os critérios de eficiência que foram descritos. As árvores-R independentes podem provocar a realização de busca em várias unidades de disco, mesmo em consultas por pequenas regiões, quando for adotada a distribuição dos nós. Problema semelhante apresenta a definição de super-nós, uma vez que para recuperar um nó deve-se percorrer as d unidades de disco. A estratégia alternativa de distribuição por particionamento do espaço adotada nas árvores-R independentes também apresenta problemas em consultas envolvendo uma área maior, uma vez que poderá ativar poucas unidades de disco.

2.5.4. Árvores-R⁺.

Para aplicações de BD com informações espaciais, nas quais os objetos apresentam grande quantidade de área de sobreposição, a árvore-R pode não se

Figura 13: Retângulos organizados em uma árvore-R⁺.

revelar adequada, pois provoca um aumento deste fator em função da construção do retângulo de delimitação. Com o objetivo de minimizar tal parâmetro, foi proposta a árvore-R⁺ [Roussopoulos_88], que provoca divisão no espaço da imagem em

subespaços que não necessariamente delimitam todo o objeto. Desta forma, se em um nó folha o retângulo que delimita um objeto possui uma região de sobreposição com outro, estes são decompostos em um conjunto de subretângulos menores sem

sobreposição, e cuja união torne possível a restauração do original. Todos os ponteiros dos subretângulos produzidos indicam o mesmo objeto. Este processo de divisão pode ser expandido para os nós não-folha, com o objetivo de provocar uma diminuição no fator de sobreposição.

Figura 14: Árvore- R^+ correspondente a Figura 13.

Este procedimento, no entanto,

provoca aumento na altura da árvore final produzida. Na Figura 14 encontra-se a representação da árvore- R^+ para a imagem simbólica da Figura 13. Comparando o procedimento de divisão do espaço com o realizado pela árvore- R , nota-se a sensível redução nas áreas de sobreposição dos retângulos produzidos.

Formalmente uma árvore- R^+ possui a mesma estrutura de nós que a apresentada pela árvore- R , mas com algumas características particulares:

- Cada entrada $(p, RECT)$ em um nó intermediário da subárvore indicada por p contém um retângulo R se e somente se R está contido em $RECT$. A única exceção ocorre quando R é um retângulo em um nó folha, e neste caso R deve simplesmente sobrepor $RECT$;
- Para quaisquer duas entradas $(p_1, rect_1)$ e $(p_2, rect_2)$ de algum nó intermediário, a região de sobreposição entre $rect_1$ e $rect_2$ possui área igual a zero.

Definindo M como o número máximo de entradas que possam existir em cada nó da árvore- R , uma árvore- R^+ não satisfaz as condições referentes ao número mínimo e máximo de chaves por nó, ou seja respectivamente $M/2$ e M . Apesar deste

aspecto, alguns resultados experimentais [Roussopoulos_88] têm demonstrado uma degradação do espaço físico utilizado da ordem de 10%, quando comparado às árvores-R* (descritas a seguir). Este valor pode ser ponderado com relação à eficiência alcançada nas operações de busca. Além disso, testes realizados através da utilização de segmentos uniformemente distribuídos no espaço, e formulando-se consultas envolvendo a localização de objetos que contenha um determinado ponto, têm demonstrado uma maior eficiência da estrutura de árvore-R⁺ com relação à árvore-R [Faloutsos_87]. As árvores-R apresentaram problemas com relação ao número de acessos a disco para segmentos de pequeno tamanho e para os segmentos longos, o que não ocorreu para as árvores-R⁺. No entanto, nestas últimas podem ocorrer em um grande número de acessos a disco, caso seja frequente a presença de segmentos longos, provocando várias divisões dos nós com o objetivo de evitar a sobreposição.

2.5.5. Árvores-R*.

As árvores-R* [Beckmann_90] seguem uma estrutura e definição de nós semelhantes à árvore-R, e com relação às operações de inserção e remoção são completamente dinâmicas, ou seja, podem ser ativadas sem a necessidade de uma reorganização global periódica. Ao contrário das árvores-R⁺, a estrutura permite a presença de retângulos com sobreposição, e, portanto, não há como garantir a existência do caminho único de busca para um determinado objeto. As árvores-R* procuram obter ganho de eficiência na estrutura, através da otimização das operações de inserção de uma nova chave. Esta operação apresentará o seu desempenho dependente de duas operações básicas, a saber, a escolha da subárvore mais

apropriada à nova chave, e do algoritmo de divisão de um nó, quando este ultrapassar o número máximo de chaves permitidas. Uma melhoria nestes dois algoritmos pode tornar a operação de busca por um determinado objeto mais eficiente.

Para solucionar o problema de escolha do caminho apropriado para inserção de uma nova chave, os algoritmos propostos para as árvores-R consideravam como parâmetro apenas a área do retângulo. Isto significa que um dos principais elementos a ser minimizado consiste na menor área de crescimento de um retângulo para acomodar o novo objeto. A abordagem da árvore-R* inclui o tratamento das margens e regiões de sobreposição, além da área do retângulo. Desta forma, antes de verificar o custo da incorporação de um novo objeto, será selecionado o que apresentar o menor valor de sobreposição com os demais retângulos, a qual é definida como:

$$Sobrepos(E_k) = \sum_{i=1, i \neq k}^p area(E_k.Rect \cap E_i.Rect), 1 \leq k \leq p.,$$

sendo $E_1, \dots, E_p$ as p entradas presentes do nó corrente, e $Rect$ os respectivos retângulos de delimitação.

O algoritmo que define a função de escolha da subárvore para a inserção é descrito como [Beckmann_90]:

Algoritmo EscolhaSubárvore:

ES1: Posicione N para ser a raiz da árvore.

ES2:

Se (N é folha) retorne o valor de N;

Senão Se (os ponteiros para nós filhos indicam folhas)

[determina o custo de sobreposição mínimo]

Escolha a entrada em N , na qual o retângulo necessita do menor crescimento de sobreposição para incorporar o novo retângulo de informação.

Solucione as igualdades, escolhendo a entrada cujo retângulo apresente o menor crescimento de sua área.

então

Entrada selecionada com o retângulo com a menor área.

Se (os ponteiros para nós filhos não indicam folhas)

[determina o custo de crescimento mínimo da área]

Escolha a entrada em N , cujo retângulo necessita de menor crescimento em sua área para incorporar o novo retângulo de informação.

Solucione as igualdades, escolhendo a entrada cujo retângulo apresente a menor área.

ES3: Posicione N para o nó filho a partir da entrada selecionada e repita o passo ES2.

Com o objetivo de reduzir o número de entradas envolvidas no cálculo da menor área de sobreposição, uma abordagem alternativa não considera os retângulos que se encontram distantes entre si. Portanto, esta porção do algoritmo fica alterada para:

[determine o custo de sobreposição mínimo para os retângulos mais próximos]

Ordene os retângulos em N em valores crescentes do aumento necessário em sua área para incluir o novo retângulo de dados;

Seja A o conjunto com as primeiras p entradas;

A partir das entradas em A, considerando todas as entradas em N, escolha o retângulo que necessita de menor crescimento de sobreposição;

Solucione as igualdades, escolhendo a entrada cujo retângulo apresente o menor crescimento de sua área.

O algoritmo de divisão de um nó na árvore-R* utiliza-se da seguinte abordagem. Inicialmente, todas as entradas deste nó são ordenadas a partir do menor valor e em seguida, a partir do maior valor, para cada eixo de seu retângulo. Os conjuntos obtidos possuem as suas entradas combinadas entre si, produzindo dois novos conjuntos, indicando a distribuição dos possíveis nós a serem produzidos como resultado da divisão. O número total de combinações permitidas é dada por $M-2m+2$, onde a k -ésima distribuição é descrita como: o primeiro grupo contém as primeiras $(m - 1) + k$ entradas, e o segundo grupo as restantes. Para cada distribuição obtida será calculada um conjunto de valores que irão definir qual será a mais adequada. Estes valores consistem em:

- *área*: $\text{área} [\text{bb}(\text{primeiro grupo})] + \text{área}[\text{bb}(\text{segundo grupo})]$;
- *margem*: $\text{margem} [\text{bb}(\text{primeiro grupo})] + \text{margem}[\text{bb}(\text{segundo grupo})]$;
- *sobreposição*: $\text{área} [\text{bb}(\text{primeiro grupo}) \cap \text{bb}(\text{segundo grupo})]$;

onde *bb* indica o retângulo de delimitação que contém todas as entradas de um grupo.

Estes valores são utilizados para calcular o possível eixo de divisão do nó e a melhor distribuição obtida. Possíveis métodos para o processamento destes valores consistem em: a) menor valor sobre um determinado eixo; b) menor valor obtido através da soma de todos os valores para cada eixo, ou c) o menor valor global.

O algoritmo para uma árvore- R^* realizar a divisão das chaves em um determinado nó seleciona inicialmente a dimensão do retângulo a ser utilizada como elemento ao longo do qual esta operação será processada. A seleção do eixo é indicada para os que possuem menor valor para a margem entre os retângulos. Uma vez selecionado o eixo da divisão, as entradas serão distribuídas considerando os menores valores de sobreposição entre dois retângulos, solucionando os possíveis conflitos, escolhendo o que apresentar menor área. Os algoritmos são apresentados a seguir:

Algoritmo de Divisão:

D1: Chama EscolhaEixoDivisão para determinar o eixo, perpendicular ao qual será realizada a divisão;

D2: Chama EscolhaÍndiceDivisão para determinar a melhor distribuição nos dois grupos ao longo do eixo selecionado;

D3: Distribui as entradas nos dois grupos.

Algoritmo EscolhaEixoDivisão:

EED1:

Para cada eixo faça:

Ordene as entradas pelo menor valor e depois pelo maior valor de seus retângulos;

Determine todas as distribuições como indicado anteriormente;

Calcule S , a soma de todos os valores de margens para as diferentes distribuições;

fim.

EED2: Escolha o eixo com o valor mínimo para S como o eixo de divisão.

Algoritmo EscolhaÍndiceDivisão:

EID1: Ao longo do eixo de divisão, escolha a distribuição com a área de sobreposição mínima;

Solucione as igualdades, escolhendo a distribuição com o menor valor para área.

As árvores-R e suas variações são não-determinísticas na alocação de entradas em seus nós, ou seja, diferentes sequências de inserção do mesmo conjunto de chaves irão produzir árvores diferentes. Por esta razão as árvores-R possuem as suas características de desempenho dependentes das chaves anteriores já inseridas. Estas podem ter produzido uma estrutura de retângulos que não seja a mais adequada ao estado atual do sistema. Uma reorganização da localização das chaves é realizada durante o procedimento de divisão dos nós, entretanto, é insuficiente pois não envolve toda a árvore. Este problema pode se tornar crítico em decorrência da eliminação de chaves de um nó, de tal forma que este fique com um número de chaves abaixo do mínimo permitido, fazendo com que este seja agrupado com outro nó vizinho na árvore, mas que pode aumentar significativamente a sua área de abrangência.

Uma abordagem inicial, realizada pelas operações definidas para as árvores-R, consiste em processar uma re-inserção destas chaves na árvore. Este procedimento pode determinar nós mais adequados para as chaves, e como consequência, uma estrutura mais apropriada ao instante atual. As árvores-R* adotam uma estratégia diferenciada, forçando a re-inserção de um valor de chave sempre que esta provocar um “overflow”(ultrapassa o limite máximo de chaves para o nó). As chaves a serem

selecionadas para a re-inserção são as que possuem a maior distância com relação ao centro do retângulo de delimitação deste nó. Caso a reinserção do valor chave venha novamente provocar o problema neste mesmo nó, a opção a ser adotada consiste em realizar a sua divisão. O algoritmo para reinserção das chaves na árvore- R^* fica:

Algoritmo ReInserção:

RI1: Para todas as $M + 1$ entradas do nó N , determine a distância entre os centros de seus retângulos e o centro do retângulo de delimitação de N ;

RI2: Ordene as entradas em ordem decrescente de suas distâncias;

RI3: Remova as primeiras p entradas em N e ajuste o retângulo de delimitação para as chaves restantes;

RI4: No conjunto ordenado, definido em RI2, iniciando com a distância máxima ou a mínima, utilizar o algoritmo da inserção para reinserir as entradas.

Resumidamente, a operação de re-inserção possui algumas vantagens, tais como:

- A re-inserção pode alterar as entradas de nós vizinhos, diminuindo a área de sobreposição de seus retângulos;
- Incremento na utilização do espaço de armazenamento;
- Provoca uma diminuição do procedimento de divisão de um nó;
- A re-inserção será realizada considerando os fatores de otimização para a árvore atual, tornando-a relativamente independente da sequência de inserção dos valores de chaves.

2.5.6. Árvore de Células.

Uma proposta de generalização das árvores-R consiste em realizar modificações para que se possa facilitar a busca por objetos que apresentem formas arbitrárias, procurando minimizar a região de abrangência necessária a sua representação, definindo a estrutura *árvore de células* (“cell tree”) [Günther_91]. Os objetos são descritos como uma união de células convexas, definidas no espaço euclidiano com d -dimensões. Um determinado ponto do espaço se encontra no interior de um objeto, se o mesmo pertencer ao interior de qualquer célula. A Figura 15 apresenta a descrição de objetos com formas irregulares, inclusive contendo buracos e componentes disjuntos. Importante observar que as células consideradas na descrição de um objeto não são necessariamente disjuntas.

Figura 15: Representação para uma Árvore de células

De forma semelhante à árvore-R, uma árvore de células consiste de uma estrutura dinâmica e balanceada, sendo que cada nó corresponde a uma página no disco de armazenamento. Cada nó folha possui uma entrada E representando uma célula z através da tupla (G, D, A) , onde G indica a geometria da célula; D é o identificador do objeto cuja sequência de células convexas contém a célula z ; e A é o

conjunto de todos os atributos de D necessários para sua caracterização, inclusive suas células de descrição. Os nós internos contém entradas da forma (cp, P, C) , onde cp é o indicador do nó filho, P consiste de um poliedro convexo com d -dimensões. Todas as células que sejam subconjunto de P pertencem à subárvore a partir deste nó. O elemento C é denominado de *container*, sendo um subconjunto de P , contendo também cada célula z na subárvore fornecendo um mecanismo mais preciso de localização destas células.

Com esta estrutura pode-se realizar buscas da seguinte forma: encontre todos os objetos com sobreposição a um espaço de busca indicado que possa ser de forma arbitrária. Para solucionar esta consulta, inicialmente é calculado o conjunto de células convexas que descreve o espaço de busca. Para cada componente convexo, será realizada uma busca pela árvore para determinar o objeto que se sobrepõe a esta célula. Encontrando-se em um nó interior da árvore, o espaço de busca é novamente decomposto em células convexas, e pode gerar também componentes não convexas, os quais são descartados por não influenciarem na busca. Estes são comparados com os componentes deste nó, e aquele que possui sobreposição é selecionado como a raiz da subárvore a ser percorrida. Este procedimento se repete até que seja localizado um nó folha. Neste ponto, para a célula que apresentar sobreposição, retorna o objeto que se encontra representado.

Para inserir um novo objeto na árvore de células, inicialmente calcula-se a região convexa que o delimita. Cada componente desta região é inserido separadamente, e os que apresentam vários pontos de sobreposição podem ser colocados em subárvores distintas. Como consequência, a inserção de um nó pode

provocar a criação de várias células novas, dependendo do estado atual da árvore.

A inserção irá se processar de forma similar à realizada nas árvores-R, sendo que os nós em que ocorra “overflow” serão divididos, percorrendo cada hiperplano de delimitação com o objetivo de determinar o mais adequado, considerando: a) dividir o nó de forma a produzir dois outros que possam ser criados na mesma página do disco e, b) selecionar o hiperplano que apresente interseção com o menor número de células. Quando não for possível determinar um hiperplano ótimo de divisão, este nó folha fica armazenado em uma página de “overflow”, ultrapassando o seu limite de armazenamento de objetos. Esta abordagem é aceitável [Günther_91], pois considera-se uma grande ocorrência destes casos apenas em banco de dados que se encontrem densamente populados, com vários objetos ocupando posições próximas no espaço.

2.5.7. Considerações sobre as Estruturas de Acesso.

Nos itens anteriores foram descritas várias estruturas que possibilitam a representação e organização dos elementos utilizados como índices de acesso a um BD. Estas podem ser classificadas em estruturas que se utilizam de informações simbólicas para a caracterização do conteúdo da imagem, como a string 2D, e nas estruturas de organização das informações na forma de árvores, as quad-tree e árvore-R e suas variações.

A quad-tree consiste de uma estrutura utilizada na representação de imagens em que possa ser identificada a presença de regiões definidas através de algum critério, como por exemplo, a distribuição de cores. Neste caso, é produzida uma árvore na qual os nós folhas representam regiões da imagem contendo a mesma cor,

ou valores em um intervalo de tolerância. Se a imagem possui uma grande variedade de cores, a sua representação irá resultar em uma árvore com altura elevada, indicando a utilização de um maior espaço para o armazenamento da estrutura produzida, além de incrementar o tempo de processamento necessário para se percorrer toda a árvore. Dentre as alterações propostas para que a definição da quad-tree possa ser utilizada como uma estrutura de organização dos índices de acesso às imagens, a quad-tree invertida dinâmica é considerada a mais adequada, por permitir uma boa utilização do espaço disponível em disco e não necessitar de reorganizações periódicas após uma série de inserções.

A string 2D define de forma adequada a representação de índices baseados em ícones presentes em imagens simbólicas. Técnicas de organização de índices, tais como *hashing*, têm sido empregadas para implementar o acesso e recuperação de imagens com o seu conteúdo descrito através de uma string 2D [Chang_92a]. O principal problema desta estrutura encontra-se na utilização de símbolos representativos como motivação fundamental, e a possível presença de redundância na reconstrução das imagens. Além disso, as strings 2D representam de forma efetiva a informação presente em uma imagem simbólica através de símbolos decorendo objetos em um espaço 2D. Para a descrição de cenas contendo objetos no espaço 3D, outros aspectos devem ser adicionados à string 2D de forma que estas imagens possam ser descritas com precisão [Bimbo_93]. Dentre as dificuldades encontradas, uma string 2D não permite a representação efetiva da noção de profundidade presente em uma cena 3D, mesmo permitindo a sobreposição de símbolos, o que pode

dificultar a formulação de uma consulta. Além disso, os símbolos a serem utilizados devem ser naturalmente indicados em três dimensões.

A árvore-R é adequada para se representar imagens simbólicas com requisitos de consultas utilizando-se o raciocínio espacial, na indicação das várias dimensões presentes na especificação das características gráficas da imagem, revelando-se também eficientes na manipulação de informação pontual. No entanto, apresenta problemas de utilização de espaço e de crescimento quando manipula informação com dimensões elevadas [Flickner_95] [Faloutsos_87]. As suas extensões, ou seja, árvore-R, árvore-R* e árvore de células, procuram otimizar aspectos conceituais e principalmente operacionais da árvore-R, através da criação de novos algoritmos para as operações de inserção e localização de um objeto no interior da estrutura. O objetivo destas alterações consiste em facilitar o trabalho de localização da subárvore de inserção de um novo objeto, e proporcionar uma melhor representação do espaço ocupado pelos objetos presentes em uma imagem.

A árvore de células introduz uma generalização na conceituação da árvore-R, na qual o retângulo de delimitação de um objeto seja apresentado de forma arbitrária, como uma região convexa. Esta definição tem como objetivo flexibilizar a representação de objetos com a forma arbitrária, e, conseqüentemente, minimizar a área da região de delimitação. No entanto, a representação de regiões convexas é mais complexa do que a de intervalos de valores que define uma região retangular. Como consequência imediata, a árvore de células apresenta a característica de uma maior utilização do espaço de armazenamento em disco para a sua estrutura do que o necessário nas árvores-R e suas variações [Günther_91]. Com relação às operações

de busca, as árvores-R demonstram-se mais eficientes, principalmente nas consultas envolvendo pontos, tanto com relação ao tempo de processamento, quanto ao número de acessos a discos que se fazem necessários. No caso de consultas por regiões, as estruturas, incluindo a árvore de células, apresentam um desempenho semelhante. No entanto, todas as estruturas irão apresentar um incremento em sua eficiência com o aumento do tamanho de cada página do disco. Uma página maior permite uma redução no número de operações de divisões em cada nó da árvore, o que consiste em um custo elevado de processamento.

De uma forma geral, os requisitos para a definição da informação a ser utilizada como índices, bem como sua estrutura de representação e organização, são os seguintes [Chang_92b]:

- a) Um índice ativo no lugar de uma estrutura passiva, permitindo que esta realize a inicialização de operações;
- b) Um índice parcial e não total, no qual apenas um conjunto das imagens ou de objetos são indexados, enquanto que os demais permanecem sem mecanismos de acesso;
- c) Um índice dinâmico, ou seja, a estrutura de organização deve possibilitar uma evolução, crescimento ou divisão caso seja uma aplicação distribuída;
- d) Um índice visível tal que o usuário tenha consciência da sua existência, estrutura e significado, deixando de ser uma estrutura transparente, uma vez que as consultas são especificadas através da utilização destes índices;

- e) Uma estrutura de índice imprecisa que possibilite a resolução de consultas com esta característica, ou seja, a indicação da informação a ser recuperada pode ser realizada de forma aproximada.

2.6. DEFINIÇÃO DE OPERAÇÕES DE CONSULTA AO BANCO DE IMAGENS.

Uma vez extraída, representada e organizada as informações contidas em uma imagem, pode-se especificar as operações de acesso e consulta envolvendo este tipo de dado. A especificação destas operações é importante para se definir o comportamento da linguagem de consulta formulada em um Banco com Imagens, seja esta uma linguagem que se utilize puramente de aspectos visuais ou com a presença de uma estrutura de comandos.

As operações são especificadas seguindo os requisitos do domínio da aplicação, ou seja, as características dos objetos e informações extraídas da imagem. Nas aplicações denominadas de Banco de Dados Espaciais, que envolvem por exemplo, as áreas de GIS, CAD/CAM, robótica, as operações devem manipular dados com n-dimensões apresentando o conhecimento dos objetos que representam sua extensão e posição relativa no espaço [Günther_90]. Neste caso as operações suportam a formulação de consultas, genericamente classificadas como [Faloutsos_87]: a) *consultas de pontos*, na qual dado um determinado ponto no espaço seja localizado todos os objetos que o contenha e, b) *consultas de regiões*, dada uma determinada região, encontre todos os objetos que tenham alguma intersecção.

Em aplicações que se localizem em outros domínios, tais como suporte a imagens médicas, as operações devem suportar consultas que envolvam a

manipulação de elementos descritivos das características visuais do conteúdo das imagens, como por exemplo, distribuição de cores, textura e padrões [Caeli_93] [Kankanhalli_96].

2.6.1. Operadores Espaciais.

As operações espaciais suportadas podem ser divididas em três categorias [Wang_91] [Sistla_94]:

- Busca Espacial*: localizar a posição de um objeto espacial em uma determinada área, obtendo suas coordenadas no espaço;
- Medidas Espaciais*: determinar propriedades espaciais das entidades, tais como, tamanho, área, distâncias, perímetros e outros;
- Comparação Espacial*: realizar comparações entre os objetos espaciais para determinar, por exemplo, continência, sobreposição, localização de vizinhança e posicionamento relativo.

Estes operadores (Figura 16) consideram a definição das posições relativas entre os objetos presentes em uma imagem, indicadas como um conjunto de relacionamentos topológicos, consistindo em [Nabil_96] [Bimbo_93] [Papadias_95]:

- disjunção*: um objeto não apresenta nenhuma sobreposição com outro objeto da

Figura 16: Relações Topológicas

imagem;

- *tocar*: um objeto apresenta um ponto ou aresta em comum de sua superfície com outro objeto;
- *continência ou interior*: um objeto está contido ou se encontra no interior de outro da imagem;
- *sobreposição*: um objeto se sobrepõe a outro, ocultando parcialmente a sua superfície;
- *sobreposto-por*: um objeto se sobrepõe a outro, ocultando totalmente a sua superfície;
- *igualdade*: um objeto se sobrepõe totalmente a outro, sendo que ambos são exatamente iguais.

Para formular estas categorias de operações sobre objetos espaciais torna-se desejável a definição de uma estrutura abstrata de dados que seja independente da representação escolhida. Desta forma a noção de conjunto de pontos de formação de um objeto pode se revelar adequada a este propósito. Alguns operadores da álgebra de conjuntos podem ser diretamente aplicados, como pertinência, união, intersecção, diferença e mesmo noções de proximidade ou sobreposição. O conjunto de pontos permite a especialização em novos subtipos para o tratamento de objetos 3D ou com n-dimensões, e cada um destes, refinados para definir o conjunto de pontos específicos à representação. As operações definidas neste nível são consideradas como uma implementação das operações mais genéricas.

A definição das operações espaciais, através das estruturas utilizadas na representação e organização das informações presentes no conteúdo da imagem,

consiste em abordagens distintas para a implementação das operações existentes em níveis genéricos. Isto significa que as operações espaciais irão apresentar comportamentos distintos, como consequência das estruturas de representação e acesso utilizadas [Hoel_92]. Realizada esta consideração inicial, nos itens seguintes serão apresentadas as operações espaciais considerando a existência das estruturas de representação e organização de índices apresentadas na seção anterior deste capítulo, exemplificando através das operações de “*join*” (junção) espacial e de determinação de vizinhança.

Para suportar consultas por regiões com eficiência, o método de acesso deve permitir o armazenamento, em uma mesma página de disco, dos objetos que se encontram próximos no espaço, provocando uma redução no número de acessos a disco. Algumas estruturas não manipulam diretamente os objetos em toda a sua extensão, uma vez que seu tamanho não é necessariamente limitado à extensão da página em disco. Estas estruturas, tais como as árvores-R e suas extensões, representam os objetos através de uma aproximação definida como retângulo mínimo de delimitação (MBR). Em uma estrutura com suporte a este conceito, as consultas por regiões podem ser processadas em duas etapas [Brinkhoff_93] [Orenstein_90]:

- a) *Filtro*: consiste em localizar todos os objetos, nos quais seu MBR apresente alguma intersecção com o retângulo da consulta;
- b) *Refinamento*: para os objetos determinados anteriormente, verificar os que obedecem à condição da consulta.

2.6.1.1. Join Espacial.

A operação de **join espacial** é utilizada para combinar objetos pertencentes a dois conjuntos através de alguma característica espacial. Como exemplo, considere dois objetos espaciais “*floresta*” e “*cidades*”, cujos atributos representam as bordas de delimitação de ambos. Uma consulta na qual seja necessária a aplicação do operador de *join* espacial consiste em “encontrar todas as florestas que apresentem cidades em seu interior”. Para sua conceituação considere os conjuntos $A = \{a_1, \dots, a_n\}$ e $B = \{b_1, \dots, b_m\}$ de objetos espaciais, podendo ser denominados de relações espaciais, que representam objetos espaciais através de seus polígonos de delimitação. Seja $Id()$ a função que associa um identificador único para cada objeto espacial no BD, e $Mbr()$ a função que calcula o retângulo mínimo de delimitação de um objeto. As operações de *join* espacial a serem formuladas são:

- a) **Join espacial do MBR**: determina todos os pares de identificadores de objetos $(Id(a_i), Id(b_j))$ que apresentem sobreposição em seus retângulos de delimitação $Mbr(a_i) \cap Mbr(b_j) \neq \emptyset$;
- b) **Join espacial de ID**: determina todos os pares de identificadores de objetos $(Id(a_i), Id(b_j))$, com alguma região em comum, $a_i \cap b_j \neq \emptyset$;
- c) **Join espacial de objetos**: determina a região comum entre os objetos $(a_i \cap b_j)$ que apresentem esta sobreposição, $a_i \cap b_j \neq \emptyset$;

Pelas definições acima, o *join* espacial do MBR pode ser utilizado como um filtro na avaliação dos demais tipos apresentados em b) e c). No entanto os métodos disponíveis para o processamento das operações de *join* em relações não espaciais, não pode ser diretamente aplicado para o *join* espacial. Vários métodos para a

avaliação do *join* espacial foram propostos através da utilização da estrutura de acesso às imagens que se encontram disponíveis.

O processamento de *join* espacial com a utilização de árvores-R* [Brinkhoff_93] é realizado tendo como princípio básico a propriedade que cada retângulo de delimitação de um nó consiste no MBR de todos os retângulos presentes na subárvore deste nó. Desta forma, se dois retângulos correspondentes a dois nós E_R e E_S não apresentarem uma região em comum, não irá existir nenhum par $(rect_R, rect_S)$ de retângulos pertencentes às subárvores distintas no resultado do *join*. Do contrário, nestas subárvores podem existir objetos cujos retângulos de delimitação apresentem uma região de sobreposição, pertencendo, portanto, ao resultado do *join*. A determinação da região em comum entre os retângulos é realizada em níveis equivalentes das subárvores, comparando-se cada entrada presente em um nó (E_R) com todas as demais presentes no outro nó (E_S), até atingir os nós folhas onde se encontra a identificação dos objetos. A redução no tempo de processamento necessário nestes passos é obtida restringindo-se o número de comparação entre os retângulos. Considerando dois nós que obedecem à condição do *join*, ou seja $E_R.rect \cap E_S.rect \neq \emptyset$, para as subárvores indicadas por $E_R.ref$ e $E_S.ref$ serão testadas apenas as entradas que apresentem alguma intersecção com o retângulo $E_R.rect \cap E_S.rect$. Estas são as possíveis entradas candidatas a possuírem uma região comum entre si, e, portanto, a pertencerem ao resultado final do *join*.

Outra abordagem para incrementar a eficiência no processamento do *join* espacial consiste em realizar uma ordenação das entradas em cada nó da árvore-R* de acordo com suas localizações espaciais. No caso de retângulos com duas

dimensões, um método que permite a ordenação de forma adequada para determinar a intersecção consiste no seguinte: seja uma sequência $Rseq = \langle r_1, \dots, r_n \rangle$ com n retângulos; um retângulo r_i é delimitado pelas coordenadas de suas extremidades inferiores e superiores respectivamente, ou seja, $(r.xl, r.yl)$ e $(r.xu, r.yu)$. Definindo a projeção das coordenadas de cada retângulo ao longo dos eixos por $\pi_x(r_i)$ e $\pi_y(r_i)$, uma sequência $Rseq = \langle r_1, \dots, r_n \rangle$ está ordenada no eixo X se $r_i.xl \leq r_{i+1}.xl$, para $i \leq n$. Por exemplo, a sequência

Figura 17: Conjunto de retângulos para ordenação.

ordenada para os retângulos presentes na Figura 17 é $\langle r_1, r_4, r_2, r_5, r_3, r_6 \rangle$.

Para se determinar a intersecção entre retângulos presentes em uma sequência ordenada, utiliza-se uma linha de varredura que

percorre perpendicularmente ao longo de um eixo, da esquerda para a direita. Considerando duas sequências ordenadas $Rseq = \langle r_1, \dots, r_n \rangle$ e $Sseq = \langle s_1, \dots, s_n \rangle$ referentes às entradas em nós da árvore- R^* , realiza-se uma varredura até localizar um retângulo t pertencente a $Rseq \cup Sseq$ com o menor valor para xl . Se o retângulo t pertencer a $Rseq$, então percorre-se $Sseq$ a partir de seu primeiro elemento até localizar o retângulo s_h , cujo valor da coordenada inferior (xl) seja maior do que a coordenada superior de t . Neste ponto, pode-se afirmar que o retângulo t apresenta intersecção ao longo do eixo X para todos os retângulos de s_j com $j \leq h$. Se o

Figura 18: Exemplo de um teste de intersecção.

mesmo ocorrer ao longo do eixo Y, então o retângulo t apresenta intersecção com s_j . Se o retângulo t pertence à

sequência $Sseq$, então realiza-se uma busca análoga em $Rseq$. Ao finalizar este passo, o retângulo t é marcado como analisado, e o procedimento se repete para os demais retângulos em $Rseq \cup Sseq$. Quando a última entrada de R ou S for processada, então todas as intersecções foram calculadas. A Figura 18 apresenta a sequência de testes de intersecção realizada para os retângulos indicados.

A definição do *join* espacial para retângulos com duas dimensões pode ser estendida para o tratamento de retângulos com n -dimensões, e desta forma adaptado à realização de consultas envolvendo a recuperação de imagens através de vários atributos distintos associados às suas características, tais como para histograma com as cores, textura, ou comparação de padrão. Nestas características, considerando a representação de cada atributo como uma dimensão do retângulo, se terá montado uma chave de acesso com n -dimensões. Assim pode-se formular consultas tais como: “dada uma imagem retorne todas as que apresentem cores (ou textura, ou padrão) semelhantes à indicada”.

2.6.1.2. Determinação de Vizinhança.

Outras categorias de consultas comuns de serem executadas em Banco de Dados com Imagens Espaciais consistem em: dado um determinado objeto no espaço, determine os cinco demais que se encontram mais próximos, ou aqueles que estejam a uma distância máxima indicada. Esta consulta pode ser expandida para se determinar o conjunto de objetos que sejam mais semelhantes a algum outro fornecido. As consultas com estas características são definidas como **consultas dos k-vizinhos mais próximos** (*k-Nearest Neighbors* - k-NN). A versatilidade deste tipo de busca aumenta quando são consideradas algumas variações como a localização dos k-vizinhos posteriores, ou combinadas com outros operadores espaciais e, desta forma, consultando pelos k-NN à oeste de uma localidade, ou ainda, *join* espacial com os k-NN, tal como localizar os três restaurantes mais próximos de dois cinemas distintos. Para um processamento eficiente desta categoria de consulta, a estrutura de representação dos objetos espaciais deve permitir a conceituação de proximidade e sua generalização para a definição de vizinhos. Assim serão discutidos abordagens de solução deste problema considerando a estrutura de árvores-R e das árvores de k-dimensões.

Para a especificação da consulta dos vizinhos mais próximos em uma árvore-R [Roussopoulos_95], é necessária a indicação de algumas medidas. Dado um ponto de consulta P, um objeto *ob* e seu retângulo de delimitação (MBR), define-se a menor distância entre P e *ob* (MDIST) como o menor valor da distância euclidiana do ponto P à superfície do retângulo mais próxima. Define-se também a menor entre todas as distâncias máximas possíveis do ponto P aos vértices do retângulo MBR

(MMDIST). Estes dois valores permitem delimitar as subárvores de busca e sua ordem de visita, e, conseqüentemente, os retângulos necessários para localizar os vizinhos mais próximos.

Considerando a definição utilizada para o retângulo mínimo de delimitação, pode-se afirmar que uma das superfícies apresenta no mínimo um ponto dos objetos que contém. Generalizando a definição de MDIST, a menor distância de um ponto P no espaço a um objeto ob qualquer é definida como o menor valor da distância entre P e os pontos que compõem o objeto. Assim, MDIST pode ser escolhido como um

parâmetro para direcionar a busca através da árvore, no entanto, deve ser também considerado o tamanho e a forma do retângulo,

genericamente indicado pelo *Figura 19: MINDIST não é sempre a melhor ordem.*

valor de MMDIST. A

Figura 19 apresenta um exemplo no qual a escolha do valor de MDIST para selecionar a busca não produz a melhor sequência, uma vez que inicialmente será pesquisado o MBR1, e, a seguir, consultar em M11 e M12. De outra forma, o parâmetro MMDIST indica a primeira visita ao retângulo MBR2, do qual resulta em um valor estimado para o vizinho mais próximo, através de M21. A partir deste limite superior, torna-se desnecessária a busca em M11 e M12. Assim, pode-se definir uma estratégia para restringir os retângulos de delimitação a serem consultados, ou seja:

- a) Para cada retângulo M cujo valor para $MDIST(P, M)$ seja maior do que $MMDIST(P, M')$ para um outro retângulo M' qualquer, então M pode ser eliminado da busca;
- b) Se a distância real de P a um objeto ob for maior do que $MMDIST(P, M)$ para um retângulo M , então ob pode ser descartado pois M contém um objeto ob' que se encontra mais próximo de P ;
- c) Cada retângulo M com $MDIST(M, P)$ maior do que a distância efetiva do ponto P ao objeto ob pode ser descartado, pois não pode conter um objeto ob' mais próximo a P .

O algoritmo para encontrar os vizinhos mais próximos percorre a árvore ordenando os retângulos de cada nó através, por exemplo, dos valores de $MDIST$, e aplicando as estratégias a) e b) para restringir a busca. Este procedimento se repete até localizar um nó, quando as distâncias efetivas entre o ponto P e o objeto ob é determinada, sendo selecionados os que apresentam valores menores do que o atual que esteja representando o mais próximo. Ao retornar deste processo recursivo, a estratégia c) é empregada na eliminação de alguns retângulos adicionais. Para localizar os k vizinhos mais próximos, retorna-se os k primeiros objetos presentes na lista final após se percorrer todos os nós válidos da árvore.

A busca pelos vizinhos mais próximos em uma árvore de k -dimensões consiste em encontrar os registros na árvore próximos ao registro de consulta. Sendo euclidiano o espaço onde estão presente os objetos da árvore, a proximidade é indicada através da fórmula de distância entre dois pontos, generalizada para n -dimensões [Sproull_91].

Importante ressaltar que estas consistem em operações básicas de acesso a um Banco de Dados contendo informações espaciais, e cada uma apresenta uma especificação distinta dependendo da estrutura utilizada para a representação da informação espacial. As aplicações nas quais as imagens presentes não possuem como requisitos fundamentais a manipulação de informações espaciais, ou seja, posições relativas, distâncias, e outras, podem apresentar as suas operações fundamentalmente definidas sobre os operadores espaciais. Neste domínio de aplicação as imagens são descritas através de suas características gráficas ou visuais, tais como forma, cor e textura. Cada uma destas características apresenta uma série de atributos que a descrevem e, uma vez que representa a informação presente na imagem, consiste de uma chave de acesso com n -dimensões. Portanto, as imagens, de forma semelhante às aplicações espaciais, são descritas por elementos presentes em um espaço com n -dimensões, o que torna possível a utilização das estruturas de quad-tree, árvores com k -dimensões, ou árvores-R para representar estas informações.

As operações são conceitualmente as mesmas, inserindo alterações em suas implementações, principalmente para suportar a definição de distâncias entre objetos em um espaço com n -dimensões representando cores, texturas e formas. Como o espaço de representação de cor e textura consiste de um espaço não euclidiano, torna-se necessário a definição de transformações responsáveis por torná-lo euclidiano. Com isto uma imagem passa a ser representada através de um ponto do espaço euclidiano com n -dimensões, em que são válidas as medidas de distância existentes para os objetos espaciais. Assim a semelhança entre duas imagens com relação à sua

distribuição de cores pode ser definida através de uma operação de *join* espacial e conjunto resultante formado através da procura dos seus vizinhos mais próximos.

2.6.2. Operadores de Conjunto.

A definição de imagem simbólica representa a informação presente no interior de uma imagem através de um conjunto de símbolos ou ícones representativos. Com esta consideração pode-se definir uma semântica para as operações existentes na álgebra de conjuntos, tais como união, intersecção, diferença, pertinência e outros, tal que possam ser aplicadas sobre uma imagem simbólica. Assim, a união entre duas imagens simbólicas pode ser definida como o conjunto de ícones representativos que apareçam em ambas. Define-se a intersecção como os símbolos existentes em comum entre as imagens envolvidas, e assim por diante, acompanhando a definição existente no tratamento de conjuntos [Racine_92] [Schene_92].

Figura 20 - Operador de União em imagens simbólicas.

Além de tratar a imagem como um conjunto de símbolos, estas operações devem preservar a disposição espacial dos objetos representados, conforme pode ser observado no exemplo da Figura 20.

2.6.3. Operação de Comparação entre Características das Imagens.

As operações descritas anteriormente apresentam em sua definição semântica a realização de comparação entre valores, sejam estes a indicação de uma localização espacial, posição relativa, ou os elementos gráficos tais como cor, textura e padrão. Nas aplicações contendo imagens espaciais, a operação de comparação é formulada diretamente sobre a estrutura de organização e representação do conteúdo da imagem, determinando regiões em comum presentes nos retângulos ou através de buscas por sub-cadeias em uma 2D-string. Para a comparação das características gráficas são utilizados métodos que não dependem da estrutura de representação utilizada, mais sim dos elementos extraídos para a sua descrição. Dentro da abordagem a ser adotada para o projeto atual, torna-se importante analisar a definição de semelhança entre características descritivas do conteúdo gráfico de uma imagem.

2.6.3.1. Similaridade de Características Gráficas.

Quando as características a serem comparadas entre as imagens não consistem apenas de posições relativas, esta operação deve ser precisamente definida para produzir um resultado adequado. As características de distribuição de cores, textura, e busca de padrão se tornam convenientes no processo de descrição do conteúdo da imagem em um sistema de banco de dados, se estas forem representadas através de elementos mensuráveis. Para o usuário é trivial realizar a comparação de imagens que apresentem cores semelhantes, ou mesmo localizar imagens na qual um determinado objeto apareça. Estas consultas, quando formalizadas no acesso a um BD, devem ser avaliadas considerando-se as características e informações disponíveis para a produção do resultado final.

Com a representação dos histogramas, textura e comparação de padrão através de elementos mensuráveis, a definição da igualdade entre duas imagens com relação a um destes aspectos é realizada através da especificação de fórmulas com medidas de distâncias, que possam indicar este fator. Para a característica de distribuição de cores, alguns métodos são propostos com o objetivo de definir a similaridade entre duas imagens [Kankanhalli_96]. O projeto QBIC [Niblack_93] [Faloutsos_93] apresenta a definição do histograma de uma imagem, através de uma distribuição normalizada, com as k cores mais representativas de seu conteúdo. Esta representação provoca uma partição do espaço de cores da imagem em um número k de células, em que a definição do histograma é realizada indicando-se o número de pontos que ocorrem em cada célula. O método utilizado para calcular a distância entre dois histogramas x e y é definido pela fórmula:

$$d_{hist}^2(x, y) = (x - y)' A (x - y) = \sum_i^K \sum_j^K a_{ij} (x_i - y_i)(x_j - y_j)$$

onde t indica a matriz transposta, e a matriz A possui entradas que descrevem o fator de similaridade entre as cores i e j . Os valores contidos na matriz A são responsáveis por transformar o conceito de semelhança entre duas intensidades de cores para o espaço euclidiano, facilitando o cálculo da similaridade entre duas distribuições de cores. Com esta medida de distância é possível indicar que imagens alaranjadas são semelhantes a imagens vermelhas, ou ainda especificar consultas que envolvam a especificação parcial do histograma, ou seja, exibir todas as imagens que apresentem de 20% a 30% de cor azul, sem realizar nenhuma consideração sobre as demais cores.

A textura de uma imagem pode ser descrita em vários aspectos, dentre os quais estão a rugosidade, o contraste e a direcionalidade. A característica de rugosidade auxilia na medida da escala da textura (*pebbles X boulders*), podendo ser calculada de forma eficiente com a utilização de janelas de movimentação com tamanhos diferentes. O contraste descreve a imagem em função da variação dos níveis de cinza que se encontra presente. A direcionalidade descreve se uma imagem apresenta uma direção em sua textura, como por exemplo, o vidro, ou é isotrópica, consistindo de um objeto suave. A medida de semelhança para a textura é calculada como uma distância Euclidiana ponderada para o seu espaço de três dimensões. Os pesos ou fatores de normalização utilizados consistem na variância inversa de cada componente, determinada sobre os elementos existentes no BD [Faloutsos_93]. Assim a distância entre dois objetos i e j em consulta envolvendo a textura é calculada como:

$$d_{ij} = \frac{(R_i - R_j)^2}{\sigma_R^2} + \frac{(C_i - C_j)^2}{\sigma_C^2} + \frac{(D_i - D_j)^2}{\sigma_D^2},$$

onde, R , C , e D representam as características de rugosidade, contraste e direcionalidade, e σ consiste no fator de normalização de cada característica.

As consultas envolvendo a recuperação de imagens através da indicação de sua forma consiste de uma operação sofisticada em função do número elevado de aspectos que podem ser considerados, além destes serem altamente dependentes do domínio da aplicação. Algumas características que possam ser utilizadas na descrição da forma são a área, circunflexão, excentricidade, orientação do eixo principal e invariantes de momentos. A medida de semelhança é definida tal como realizada para

a textura, ou seja, através de uma distância Euclidiana ponderada, onde os pesos consistem na variância inversa de cada característica.

2.7. LINGUAGENS DE ACESSO E INTERFACE PARA IMAGENS.

Os aspectos de extração, representação e organização dos índices associados às imagens presentes no sistema, junto com o conjunto de operadores definidos anteriormente, formam o suporte teórico para a formulação de uma linguagem de acesso às imagens em um Banco de Dados. Várias abordagens têm sido adotadas para a construção desta linguagem, procurando seguir a estrutura utilizada na representação das informações extraídas das imagens. Dentre as abordagens mais usuais se enquadram as extensões à linguagem de consulta relacional (SQL) [Roussopoulos_88] [Egenhofer_94], em que são incorporados novos operadores e comandos que possibilitam a manipulação das informações extraídas a partir das imagens, permitindo a sua descrição e formulação de comparações. Outra abordagem consiste na utilização do conceito da elaboração de consultas baseada em exemplos (QBE “Query-by-Example”) para descrever a imagem a ser recuperada do BD através de um modelo no qual sejam listadas as suas características, constituindo na linguagem de consulta baseada em exemplos de imagens (QPE - “Query by Picture Example”) [Cardenas_93].

Outros sistemas se utilizam da mesma idéia da linguagem QBE para descrever o conteúdo da imagem. No entanto, o exemplo representando a imagem a ser recuperada é descrito através de um esboço realizado pelo usuário final: este desenha a aparência final da imagem procurada, utilizando-se dos elementos disponíveis e suportados pelo sistema na representação interna das características de

descrição da imagem. Se a imagem é descrita através de símbolos, então a consulta é formulada selecionando-se os ícones a partir de uma biblioteca disponível no sistema [Bimbo_93] [Orenstein_88]. Caso os objetos presentes no interior da imagem possam ser identificados através de suas bordas de contorno, a consulta é formulada desenhando-se as formas a serem localizadas [Flickner_95].

Para esta abordagem, a interface do sistema deve apresentar como requisito principal a facilidade de uso, através do suporte de dispositivos apontadores e mecanismos que facilitem a realização do esboço da imagem exemplo da consulta. De uma forma geral, a interface para um sistema de informação com recuperação de imagens deve permitir a apresentação e manipulação de dados que possam ser do tipo gráfico ou textual, de uma maneira uniforme. Maiores detalhes sobre a sintaxe das linguagens de definição e acesso a imagens e das características do sistema de interface serão apresentados no capítulo 3, quando estas serão estudadas dentro do contexto dos projetos de Sistemas de Recuperação de Informação com Imagens.

2.8. CONSIDERAÇÕES FINAIS.

Neste capítulo foram apresentados os principais conceitos referentes à introdução do tipo de dado imagem em um Banco de Dados, para a construção de um Sistema de Informação com Suporte à realização de consultas às imagens baseado em seu conteúdo gráfico. A especificação e o projeto de um sistema com estas características devem ser realizados seguindo algumas etapas básicas, a saber:

- a) Extração do conhecimento contido na imagem, ou seja, obter um conjunto de características que possibilite descrever o seu conteúdo da forma mais precisa possível;

- b) Representação e Organização do conjunto de características utilizadas na descrição da imagem, com o objetivo de definir os seus mecanismos de acesso;
- c) Definição da estrutura de Banco de Dados necessária ao armazenamento da imagem de forma semelhante e integrada aos demais tipos de dados manipulados no domínio da aplicação.

Uma vez com as imagens e as demais informações relacionadas disponíveis, estes elementos sofrerão as operações naturais do ambiente de banco de dados, ou seja, consultas e alterações dos seus atributos. Dentro deste contexto, foram apresentadas as principais abordagens para a extração da informação presente em uma imagem, sua forma de organização e as estruturas de acesso disponíveis para definir as operações de acesso e recuperação das imagens, os tipos de operações que podem ser realizadas sobre o tipo de dado imagem, considerando os elementos utilizados na descrição do seu conteúdo, e uma introdução às abordagens utilizadas na especificação de uma linguagem de alto nível para consulta a imagens.

No próximo capítulo será realizado um levantamento de alguns dos principais sistemas de informação que realizaram a incorporação do tipo de dado imagem e das abordagens utilizadas na solução dos problemas presentes em cada etapa da transformação de uma imagem em um tipo de dado com um significado semântico precisamente definido, assim como das características da linguagem e da estrutura de interface disponível para a apresentação dos dados.

IMPLEMENTAÇÕES DE SISTEMAS DE INFORMAÇÃO COM SUPORTE A IMAGENS

3.1. CONSIDERAÇÕES INICIAIS.

Em função do conjunto de operadores especificados para um Banco de Imagens pode-se definir as classes de consultas genéricas a serem submetidas ao Gerenciador. Assim, de acordo com a abordagem utilizada formula-se consultas baseadas em cores, textura, desenho, forma, volume, restrições espaciais, navegação (*browsing*), atributos objetivos e/ou subjetivos, movimento, texto e informações sobre o domínio [Gudivada_95b]. Um Sistema de Informação que apresente suporte a todas estas classes de consultas é considerado genérico o suficiente para ser utilizado em qualquer tipo de aplicação [Hofstede_96]. No entanto, pela própria complexidade inerente ao tipo de dado, este processo de generalização fica comprometido, sendo formuladas abordagens que procuram contemplar as características mais marcantes a um domínio de aplicação específico.

A formulação de consultas por cores e texturas permite ao usuário selecionar imagens contendo os objetos descritos através destes parâmetros. A própria noção dos objetos presentes em uma imagem é específica da aplicação, caracterizando-a dentre as demais, representando uma entidade semântica de interesse no domínio da

aplicação. A recuperação através de desenhos possibilita ao usuário produzir um esboço da imagem, a ser utilizada como elemento de comparação na recuperação das semelhantes que possam existir no Banco. Este tipo de consulta pode ser elaborada como um processo de comparação de padrão das bordas dominantes entre as imagens, ou seja, uma busca por forma, ou referenciada em um espaço 3D como uma busca volumétrica. A categoria de consultas baseadas em informações espaciais realizam comparações entre imagens através da indicação de uma série de parâmetros, tais como, posição espacial, adjacência, sobreposição, continência e outros.

Uma consulta por navegação pode ser considerada quando o usuário não possui a princípio uma noção clara da informação que procura. A recuperação de imagens através de atributos específicos ou por textos associados ocorre com uma comparação exata entre os valores assumidos pelos atributos, sendo semelhante a uma linguagem de consulta estruturada (SQL) disponível nos sistemas tradicionais. Consultas envolvendo atributos abstratos recuperam imagens baseando-se em valores que podem apresentar interpretações distintas para cada tipo de usuário. As consultas envolvendo noções de movimento recuperam imagens contendo sequências de objetos que apresentam variações no espaço geográfico e temporal.

Como consequência desta ampla classe de consultas, as abordagens apresentadas pelos sistemas de informação com suporte a imagens procuram fornecer um tratamento completo ao conjunto de requisitos que caracterizam o seu domínio de aplicação. Desta forma, neste capítulo serão apresentadas uma série de sistemas e abordagens para a consulta de imagens em um Banco de Dados, descrevendo-os em

abordagens para a consulta de imagens em um Banco de Dados, descrevendo-os em função do tipo de aplicação a que se destina, modelo de dados utilizado, formas de consultas suportadas e estruturas de organização e acesso às imagens implementadas.

3.2. LINGUAGENS DE CONSULTAS ESPACIAIS.

As linguagens de consultas espaciais permitem a definição e manipulação de dados em um banco de imagens, no qual o seu conteúdo é descrito através de uma imagem simbólica [Chang_81]. Nesta categoria de aplicação o principal requisito consiste na realização de consultas nas quais as imagens são descritas através da disposição espacial dos objetos que possam ser localizados em seu interior [Herot_80]. Portanto, as operações básicas desta categoria de aplicação consistem na manipulação das posições relativas dos objetos no espaço, ou seja, sobreposição, proximidade, localização e outros.

De forma geral, as linguagens de consultas espaciais apresentam o seguinte conjunto de requisitos [Egenhofer_94]:

- a) Presença do tipo abstrato de dados *espacial* com as operações correspondentes e relacionamentos necessários para que os usuários possam tratar os dados espaciais em um nível independente da sua representação interna;
- b) Exibir o resultado da consulta em uma forma gráfica como a maneira mais natural de analisar as informações espaciais, permitindo a apresentação da geometria dos objetos;
- c) Combinação do resultado de uma consulta com outras realizadas anteriormente, permitindo um dinamismo na interação com o sistema;

- d) Apresentação de informações associadas ao contexto da imagem simbólica, ou seja, dados que não sejam explicitamente solicitados mas que são importantes na interpretação do resultado;
- e) Possibilidade de indicar os símbolos que compõe a imagem final a ser recuperada;
- f) Fornecer mecanismos que permitem restringir áreas de interesse de uma geografia em particular, sobre a qual as consultas serão formuladas.

Neste item serão apresentadas duas propostas de linguagens de consultas espaciais que seguem a abordagem de alterações sobre a linguagem relacional SQL. Os projetos descritos consistem na linguagem PSQL e no SQL Espacial.

3.2.1. Linguagem PSQL.

A linguagem PSQL [Roussopoulos_88] realiza extensões à linguagem SQL proporcionando o tratamento de dados espaciais e outros tipos de domínios definidos pelo usuário. Os domínios básicos suportados pela PSQL são pontos, segmentos de linhas e regiões, e a especificação de um novo domínio é realizada com a indicação dos seguintes aspectos:

- *A Forma de representação* que, por exemplo, para uma região consiste no conjunto de segmentos de reta que a define. A representação interna é uma extensão do conceito de relações, uma vez que as tuplas que descrevem um objeto da imagem são ordenadas e agrupadas em classes. Por exemplo, um segmento de reta pode ser armazenado como uma sequência de coordenadas X e Y que correspondem a seus pontos;
- *Operadores para a comparação* entre os elementos do domínio. Basicamente, a linguagem suporta operadores sobre os tipos básicos, ou seja, posições relativas de

pontos (N, S, NE, SE, e outros), segmentos de retas (interseção ou não) e regiões (sobreposição, cobertura, e outros);

- *Funções* para o cálculo dos atributos de um elemento ou para todo o domínio, tais como, a proximidade entre dois pontos, o tamanho de um segmento de reta, e a área de uma região, além de outras;
- o *domínio* definindo o pano de fundo do ambiente de apresentação necessário para exibir os elementos do domínio.

A definição dos dados na PSQL se complementa com a presença de funções que permitem indicar restrições sobre os valores dos domínios básicos, permitindo a sua validação. Com esta definição é possível restringir o conjunto de pontos que podem formar um segmento de reta, ou janela na qual possa estar contida uma região. A especificação de uma relação na PSQL se realiza definindo-se a apresentação de cada tupla, consistindo em: a) *Localização*: define a posição do elemento na tupla; b) *Granularidade*: define quantas tuplas serão exibidas ao mesmo tempo; c) *Associação visual*: define o conjunto de atributos textuais a serem exibidos juntamente com esta tupla.

A manipulação de dados através da PSQL é obtida com o seguinte comando:

```
SELECT <lista_atributos>
FROM <lista_relações>
[ ON <lista_imagens> ]
WHERE <qualificação>
```

Um exemplo de uma consulta realizada utilizando-se esta linguagem é apresentado a seguir. Nesta consulta são recuperadas informações (atributos textuais) sobre cidades localizadas em uma determinada região específica (definida por uma

janela de delimitação) e com restrições sobre a sua população (seja mais do que 450.000 habitantes).

```
SELECT cidade, estado, populacao, localizacao
FROM Cidades
ON Mapas_Brasil
WHERE localizacao within window(4+/-4, 11+/-9)
and populacao > 450.000
```

A cláusula opcional ON, presente na consulta anterior, indica as imagens sobre as quais a consulta será processada, e no caso, se refere aos Mapas_Brasil. Esta cláusula é importante quando o domínio da relação apresenta várias imagens de tipos distintos, do contrário a indicação desta cláusula pode ser omitida. A área de busca para a consulta é especificada na qualificação, a qual pode ser montada a partir dos operadores espaciais, aplicados sobre variáveis ou coordenadas espaciais constantes. Na consulta anterior, a região de busca da informação espacial, é delimitada pelas coordenadas de definição da janela (*window()*), sendo utilizada para verificar o valor do atributo espacial, localização.

Uma característica importante suportada pela linguagem consiste na síntese das informações presentes nas imagens ou mapas distintos, mas que se referem à mesma posição geográfica no espaço. Esta operação é semelhante à definição da operação de junção (ou *join*), onde as tuplas presentes em duas relações devem se referenciar a uma mesma entidade, e, neste caso, a área geográfica representa o atributo para o *join*. Além disso, a linguagem proporciona a integração da informação textual com as imagens simbólicas.

3.2.2. Linguagem SQL Espacial.

A linguagem de consulta SQL Espacial (“Spatial SQL - SSQL”) [Egenhofer_94] realiza a separação entre linguagem de *recuperação* das imagens e *apresentação* gráfica dos resultados da consulta. Desta forma, se encontra disponível um ambiente de visualização, cuja configuração é mantida pela linguagem de apresentação, denominada de GPL (*Graphical Presentation Language*).

A linguagem SSQL apresenta uma abstração para o tipo de dado espacial, estendendo os domínios existentes no cálculo relacional em quatro outros denominados de: *spatial_0*, *spatial_1*, *spatial_2* e *spatial_3*, para objetos espaciais contendo 0, 1, 2 e 3 dimensões, respectivamente. Sobre as tuplas da relação define-se operações e relacionamentos espaciais. As operações podem ser unárias, quando manipulam uma única tupla, ou binárias quando mais de uma tupla se encontra envolvida. Dentre as operações unárias estão presentes a determinação da dimensão de um objeto, sua superfície e interior. Além destas, existe a classe dos operadores aritméticos que, dependendo da dimensão, pode realizar cálculos como tamanho (para 1-D), área (2-D) e volume (3-D). De outra forma, as operações espaciais binárias determinam valores como a distância entre dois objetos ou sua posição relativa medida por um vetor de direção.

Os relacionamentos espaciais binários são definidos entre dois atributos espaciais e resultam em valores lógicos do tipo verdadeiro ou falso, como, por exemplo, a associação de vizinhança entre duas áreas é estabelecida como a interseção entre suas superfícies e seus interiores. Além disso, semelhantemente à

PSQL, define-se associações espaciais do tipo sobreposição, disjunção e outras, e posições relativas tais como direita/esquerda e norte/sul.

A linguagem de apresentação (GPL) apresenta comandos para a definição do ambiente através de sua especificação (SET), cancelamento (CANCEL) e verificação (SHOW). Os parâmetros definidos são:

- *Modo de Apresentação*: definindo características para a apresentação do gráfico, tais como, “indicar um novo desenho”, “sobrepôr um resultado anterior”, “realçar o resultado” e outros;
- *Apresentação Gráfica*: define aspectos tais como o padrão de cores e os símbolos a serem utilizados para a exibição da imagem;
- *Escala e Janela de Exibição*;
- *Contexto*: define o contexto no qual a imagem apresentada deve ser interpretada, como, por exemplo, um ponto representando uma cidade no mapa pode não ser significativo quando exibido de forma isolada;
- *Conteúdo*: permite examinar o resultado de uma consulta, quando este se apresentar combinado na tela com outras imagens.

Para formular uma consulta em SSQL, inicialmente define-se o ambiente de apresentação do resultado. Com os comandos a seguir, indica-se a cor e o padrão de desenho para um objeto denominado de rodovia:

```
SET LEGEND
    COLOR black
    PATTERN dashed
FOR SELECT superficie (geometria)
FROM Rodovia;
```


Neste ambiente de apresentação, pode-se selecionar a informação desejada, indicando-a como um novo desenho a ser realizado:

```
SET MODE new;  
SELECT Rodovia.geometria  
FROM Rodovia R, Cidade C  
WHERE C.Nome = "SJRP" and R.Nome = "Washington Luis"  
      and R.geometria INSIDE C.geometria;
```

A consulta anterior recupera o traçado da rodovia, que passa pelo interior da cidade de São José do Rio Preto, indicando através do operador espacial *inside*.

3.3. LINGUAGEM DE CONSULTA PICQUERY+.

A Linguagem PICQUERY+ consiste de uma linguagem de consulta de alto nível independente do domínio proposta para o gerenciamento de um banco de dados com informações alfanuméricas e imagens. A motivação para o seu desenvolvimento consiste em fornecer uma interface fácil de ser utilizada aos usuários do sistema de BD Médico baseado em conhecimento (KMED - UCLA) [Cardenas_93]. O projeto teve origem com a definição da linguagem PICQUERY [Joseph_88], que consiste em uma expansão à linguagem de consulta baseada em exemplos (QBE - *Query by Example*), permitindo ao usuário escrever comandos de acesso ao Banco de Imagens, descrevendo características do conjunto que se espera obter.

O modelo utilizado para o armazenamento de imagens propõe a representação dos dados em uma estrutura com grades de células. Define-se uma estrutura de dados dinâmica com uma pilha de imagens, consistindo de variáveis de duas dimensões que as representam. Todos os valores de dados para a mesma célula ou pixel da grade são armazenados em um registro convencional, em suas posições equivalentes, sendo de tamanho variável dependendo do número de imagens existentes no Banco. O acesso

aos dados é obtido utilizando-se de um algoritmo de varredura que percorre uma janela retangular sobre a pilha de imagens até localizar a informação desejada. A linguagem de consulta PICQUERY permite ao usuário do sistema formular as suas consultas através da indicação de exemplos com o resultado esperado, com a especificação de campos em uma tabela.

A linguagem permite que seja realizado uma série de operações pertencentes a um conjunto específico de processamento de imagens, tais como detecção de bordas e contornos, busca e reconhecimento de padrões, operações geométricas e espaciais (rotação, zoom, cálculos de distâncias, etc), e algumas operações funcionais (cálculos matemáticos de agregação, ou seja, mínimo, máximo, média, ou estatísticas) ou funções definidas pelo usuário. Os comandos da linguagem com estes operadores podem se aplicar sobre todo o BD ou sobre um conjunto de objetos de uma imagem.

A extensão realizada a esta linguagem foi denominada de PICQUERY+, que incorpora algumas características adicionais, tais como:

- Definição de tipos abstratos de dados;
- Habilidade de realizar comparações entre objetos de dados baseando-se em descritores imprecisos;
- Avaliação de consultas considerando eventos temporais e o aspecto de evolução da informação;
- Habilidade para especificar operações alfanuméricas e de processamento de imagens nos dados recuperados;

- Habilidade para especificar operações de visualização dos dados resultantes de consultas.

Para a incorporação destas características, o modelo de representação de imagens por pilha sofreu alterações em sua definição, incluindo novos construtores. Dentre as novas funcionalidades incluídas tem-se os construtores tradicionais de objetos (herança, agregação, e outros), hierarquias de conhecimento e regras de domínio, e elementos semânticos que permitam a manipulação de eventos temporais e processos evolucionários. Além disso, as consultas formuladas através da linguagem PICQUERY+ apresentarão como domínio um banco de dados federativo, contendo informações que se referem a cada aspecto de um sistema hospitalar.

A formulação de uma consulta utilizando-se da linguagem PICQUERY+ segue três estágios básicos. O primeiro estágio consiste na *Especificação do Domínio do Banco de Dados*, realizado através da seleção do banco de dados a partir do qual deverá ser realizada a busca pelos resultados da consulta, de acordo com os requisitos da aplicação. A complexidade inerente a partir dos tipos diferentes de BD existentes no ambiente do sistema fica transparente para o usuário da aplicação. O próximo estágio é a *Definição do Padrão para a Consulta*, ou seja, especificar as restrições para a formação do resultado da busca. Estas restrições são indicadas utilizando-se três tipos de construtores de filtros:

- a) Predicados do tipo “WHERE”, estabelecendo condições sobre os atributos alfanuméricos e com os identificadores da imagem;
- b) Predicados Evolucionários, especificando restrições associadas com diferentes fases de desenvolvimento ou versões de um objeto. Inclui operadores do tipo i)

- EVOLVES_INTRO evolução do objeto em relação ao tempo ou algum processo de maturação; ii) FUSES_INTRO, dois ou mais objetos se fundem em um novo objeto; iii) SPLITS_INTRO, um objeto se divide em dois ou mais objetos;
- c) Predicados Temporais, permitindo a especificação de período de tempo para a existência de um objeto ou para a ocorrência de um evento, sequência temporal de eventos e comparação de períodos de tempo através de operadores especiais, tais como DURING, OVERLAPS, MEETS, EQUIVALENT, ADJACENT, FOLLOWS e PRECEDES.

A especificação do padrão com o resultado da consulta é realizada através de uma tabela composta pelas seguintes colunas: a) Objeto: definição do objeto ou atributos de interesse; b) Operadores de relação: definição de como os dados devem se correlacionar com o valor do objeto padrão para que satisfaça a condição; c) Valor do Objeto: valor para o padrão de comparação; d) Operador Lógico: permite a construção de predicados complexos e e) Grupo: definição de sub-cláusulas em predicados complexos.

O último estágio consiste na especificação dos aspectos de *Visualização e Processamento dos Resultados* de uma consulta, permitindo ao usuário indicar os objetos de interesse e como estes serão exibidos. Com o objetivo de extrair um conhecimento maior sobre os dados apresentados como resultado da consulta, pode-se aplicar operações de processamento e análise ou métodos de visualização dos dados.

A seguir são apresentados alguns exemplos de consultas especificadas na linguagem de consulta PICQUERY+:

a) Encontre todas as imagens de adolescentes que apresentem alguma

anormalidade na formação da

mão (Figura 1):

A determinação da anormalidade em uma imagem da mão consiste, na linguagem PICQUERY+, de uma derivação do conhecimento específico do domínio implementado através de uma série de subconsultas geradas pelo sistema de

Figura 1 - Consulta do Exemplo a).

processamento de consultas. No domínio da aplicação médica, estas consistem em;

- i. Execução de uma série de passos associados com a extração de características da imagem, incluindo, detecção de borda, classificação de pixel, realce e *thresholding*;
- ii. Determinação subsequente de medidas dos objetos, tais como tamanho, média de valores e outros;
- iii. Comparação destas medidas com outras características do domínio de conhecimento do que seja estatisticamente definido como uma mão normal.

O sistema de processamento de consultas procura deixar a necessidade de execução de todas estas atividades transparente para o usuário final.

b) Encontrar todos os objetos presentes na imagem P2 que apresentem a forma semelhante ao formato dos objeto “Metacarpal” considerado anormal na imagem P1 (Figura 2).

Figura 2 - Consulta do Exemplo b).

Nesta consulta se encontra associado o operador SIMILAR-TO, que na linguagem PICQUERY+ apresenta alguns parâmetros que permite ao usuário definir o “nível de conferência” para a operação, definindo a aceitação ou não de uma

determinado resultado na consulta.

3.4. SISTEMA DE CONSULTA PARA IMAGENS 3D.

O sistema descrito por Bimbo et. all. [Bimbo_93] descreve o conteúdo de uma imagem, na qual se encontre presente uma cena qualquer, através da associação de símbolos (ícones) a cada objeto identificado, e dos relacionamentos espaciais entre estes. No entanto, os ícones devem representar os objetos da cena, e não os presentes na imagem desta cena, ou seja, um objeto 2D na cena deve ser representado por um ícone 2D, ao passo que um objeto 3D o será por um ícone 3D. Assim, a consulta sobre um banco de imagens com esta representação é indicada através da

seleção dos ícones com os objetos que devem estar presentes na cena, estabelecendo uma correspondência não ambígua com descrição interna.

A representação simbólica de uma cena e dos objetos identificados é realizada de uma forma genérica com a utilização de ícones 3D, sendo constituída basicamente por fórmulas que expressam os relacionamentos espaciais existentes entre estes. Cada objeto é representado através do seu menor retângulo de delimitação (*mbr*) e por um sistema de eixos que contem o centro do *mbr* e seja ortogonal ao seu menor eixo de alongamento.

As fórmulas que definem os relacionamentos podem ser:

- a) *Posicionais*: considera intervalos originados por projeções dos *mbrs* em cada eixo e são expressas como:

$$\phi_i ::= p_i | p_i < O_p > q_i,$$

onde p_i e q_i são projeções sobre os eixos de coordenadas dos objetos genéricos p e q , e O_p é um operador espacial que relaciona os dois intervalos originados pelas projeções sobre um mesmo eixo, em suas posições relativas no espaço, ou seja, antes, depois, estritamente depois, antes com uma adjacência à esquerda, e outros.

- b) *Direcionais*: considera os relacionamentos de colinearidade e paralelismo entre os planos que definem pares de objetos *mbrs*. Estas fórmulas são expressas como:

$$\psi ::= < p | q > | < p || q > | < p \perp q > ,$$

onde, $|$ indica que colinearidade, ou seja, os objetos apresentam eixos em comum, $||$ indica paralelismos entre os eixos e $\perp$ significa que os objetos não são paralelos.

Uma vez que entre dois objetos pode-se definir uma única fórmula posicional e direcional, o seu relacionamento espacial é genericamente definido como:

$$p R q, \quad \text{onde, } R = (O_{px}, O_{py}, O_{pz}, O_d),$$

sendo que O_{pk} representa algum operador posicional avaliado ao longo do eixo k do sistema de coordenadas de referência, e O_d o operador direcional.

As imagens e as suas descrições são armazenadas em um Banco de Dados Orientado a Objetos, juntamente com o conjunto de ícones utilizados. Os ícones consistem em sua forma gráfica e nos atributos textuais que descrevem a entidade representada. Associado a cada imagem, há um conjunto de atributos textuais que possibilitam descrever a cena ou objetos que possam estar presentes.

Uma consulta é submetida ao sistema através de uma interface gráfica, na qual são selecionados o conjunto de ícones para descrição do conteúdo gráfico e operadores lógicos que possam formar a expressão. O posicionamento relativo é indicado através da própria construção da imagem, existindo também a possibilidade da definição de restrições sobre os valores dos atributos textuais. O conjunto contendo o resultado de uma consulta é montado se a descrição fornecida para a cena puder ser localizada no interior de alguma outra descrição, e os valores dos atributos se encontram em um intervalo válido.

A extensão deste projeto possibilita a recuperação de uma sequência de imagens [Bimbo_95] consistindo na presença de objetos em movimentos. Neste caso a expressão de descrição da cena é aprimorada com incorporação de operadores temporais, os quais são responsáveis por definir a posição de um determinado objeto no interior de uma cena em função do instante de tempo no qual esta imagem foi registrada. A linguagem utilizada para esta representação é definida como Lógica Espaço-Temporal (STL).

3.5. SISTEMA QBIC.

O sistema QBIC [Flickner_95] deve permitir a realização de consultas sobre Banco de Dados (BD) com Imagens que possam ser baseadas em:

- imagens exemplos;
- esboços ou desenhos construídos pelo usuário;
- padrões de cores e texturas selecionados;
- movimentação de objetos e câmera;
- outras informações gráficas que sejam relevantes.

Figura 3: Arquitetura de Instanciação (acima) e do Banco de Dados (abaixo).

Estes elementos representam as características básicas do sistema QBIC, para permitir a formulação de consultas baseadas em conteúdo através da utilização de uma linguagem de consulta gráfica. Para que estas características possam ser atingidas, o sistema QBIC descreve dois aspectos como de fundamental importância: (a) instanciação do BD e (b) os modelos de consulta que sejam suportados. Durante a instanciação do BD as imagens e os vídeos são inseridos e processados para a

extração de suas características de descrição de seu conteúdo, as quais são também armazenadas no BD. A Figura 3 apresenta a arquitetura do sistema.

O modelo de dados do QBIC suporta a definição de imagens ou cenas, nas quais podem existir objetos (subconjunto de uma imagem), e trechos de vídeo que contenha uma sequência de quadros ou de objetos em movimento. A partir destes últimos são extraídos quadros (*frames*) representativos denominados de *r-frames*, utilizado como um elemento de descrição do vídeo. Os *r-frames* são tratados como imagens e também possuem suas características extraídas e armazenadas no BD para constituir a descrição de seu conteúdo. O sistema QBIC permite a realização de consultas sobre objetos (“encontre imagens que apresente um objeto vermelho redondo”), cenas (“encontre imagens com uma distribuição de cores aproximada em 30% de vermelho e 15% de azul”), trechos de vídeo (“encontre todos os trechos de vídeo com tomadas realizadas da esquerda para a direita”), ou qualquer combinação destes elementos. O conjunto de imagens que irá compor o resultado da consulta é obtido através da aplicação de funções comparativas de determinação da semelhança (através de medidas de similaridade) entre as características extraídas de cada imagem, com as indicadas pelo usuário.

No processo de instanciação do BD, uma das etapas fundamentais consiste na identificação dos objetos presentes em uma imagem ou cena de vídeo. Esta identificação pode ser realizada de forma manual, no entanto, apresentando como requisito a presença de especialistas nas áreas de conhecimento que constituem nos temas das imagens, ao mesmo tempo que dificulta a realização de consultas baseadas em seu conteúdo. Com o objetivo de aprimorar este processo foram desenvolvidas

ferramentas que possam reconhecer automaticamente os objetos, sendo inicialmente aplicadas técnicas de segmentação para a separação do fundo e consequente delimitação dos objetos. Este princípio se aplica a uma classe de imagens nas quais os objetos sejam facilmente distintos dos elementos que compõe o seu pano de fundo. Além disso, assume-se que estes objetos normalmente se localizam em uma posição central na imagem.

Para esta classe de imagens é fornecido um método alternativo que permite a determinação dos objetos através de colorização. A região de delimitação de um objeto é preenchida por pontos com cores cujos valores se encontram dentro de um determinado intervalo, obtido a partir do valor inicial de um pixel selecionado no interior deste objeto. Para facilitar a determinação das bordas dos objetos desenvolveu-se uma técnica que utiliza das curvas de *spline* para traçar os seus limites a partir de pontos nesta região indicados pelo usuário.

Outro elemento que o BD manipula consiste na sequência de quadros que define um vídeo ou filme. A etapa de instanciação deste elemento apresenta os seguintes componentes: (a) delimitação de trechos no vídeo; (b) criação de quadros representativos para cada sequência e (c) derivação de uma representação em camadas para os objetos em movimento.

A definição de trechos de um vídeo é possível devido a vários fatores, tais como:

- representar uma mesma cena;
- significar uma operação na câmera;
- contém um evento distinto ou uma ação, como a presença de um objeto;

- são escolhidos como uma possível entidade de acesso pelo usuário.

Os primeiros elementos indicadores de fronteiras ou limites em trechos de um vídeo consistem na mudança ou corte de cenas. A localização destes pontos pode ser realizada de forma automática com a utilização de (a) métodos que se baseiam na representação de elementos globais da cena, tais como histogramas de cores e intensidades, sem considerar qualquer informação espacial, ou (b) métodos que procuram modelar movimentações de câmera ou tomadas distintas, determinadas através de diferenças espaciais entre as características registradas. O método desenvolvido para o Sistema QBIC procura utilizar das vantagens apresentadas pelas duas abordagens, ou seja, diferenças de histogramas e movimentação de câmeras. Os limites em trechos de vídeo podem ser definidos apresentando como base outros eventos tais como surgimento ou desaparecimento de um objeto, mudança na movimentação de objetos, ou eventos similares.

Cada sequência de vídeo delimitada apresenta um quadro representativo (*r-frame*). Os *r-frames* são tratados como imagens comuns durante o processo de instanciação do BD, e portanto sofrerá um pré-processamento para a extração de suas características descritivas. Além disso, o *r-frame* consiste no elemento a ser inicialmente retornado como resultado de uma consulta sobre informações em vídeo. Sendo o trecho desejado, o mesmo poderá ser completamente apresentado ao usuário. A escolha de um *r-frame* é realizada através da indicação de um quadro qualquer deste trecho do vídeo. No entanto para trechos nos quais ocorrem tomadas com uma movimentação de câmera de forma suave, a abordagem adotada consiste em se obter um *r-frame* de sintetização. Inicialmente procura-se determinar o fundo que seja

dominante ao longo de toda a cena representada durante essa sequência do vídeo. Este fundo é determinado utilizando transformações de *warping* entre os quadros, que resulta na computação automática do movimento dominante ao longo da cena. A partir desta definição, os objetos que estejam presentes na cena podem ser sobrepostos ao fundo para a caracterização final do *r-frame* dessa sequência do vídeo.

Para facilitar a segmentação de objetos independentes em movimentos e de estruturas significativas, utilizou-se da natureza dependente com o tempo do vídeo, para definir a sua representação em camadas. As diferentes camadas são utilizadas para identificar objetos em uma cena e auxiliar na determinação de suas características e na formulação de consultas.

Com relação à estrutura de acesso, o Sistema QBIC adota duas estratégias principais. A primeira consiste na definição de um filtro que, aplicado sobre um grande conjunto de imagens, produz um conjunto menor tal que sobre este possa ser aplicado um procedimento de cálculo de similaridade das características indicadas [Niblack_93] [Faloutsos_93]. As imagens são, desta forma, ordenadas pelo nível de similaridade apresentado, sendo as primeiras definidas como o resultado esperado para a consulta.

A segunda trata da formulação de uma estrutura de indexação com o objetivo de otimizar o acesso às imagens. Esta estrutura será dependente da dimensão da chave de busca. Para características com dimensão baixa, tais como cor média e textura, foram utilizados métodos de indexação multidimensional como as árvores R^* . Para características de dimensões mais elevadas, estas podem ser reduzidas através da aplicação de métodos, como a transformada do componente principal,

obtendo-se um espaço com dimensão menor, e assim possível de ser utilizado em uma árvore-R*.

3.6. MODELO DE IMAGEM ESPACIAL BASEADO EM CONHECIMENTO - KSIM.

O sistema KSIM [Hsu_96] introduz a abordagem de tratamento de conhecimento na formulação de consultas baseada em conteúdo sobre um banco de imagens. Para esta integração, o sistema define um modelo de representação da imagem e interpretação de seu conteúdo dividido em três camadas:

- a) Camada de Representação: realiza o armazenamento da matriz de pixel da imagem e dos possíveis objetos que possam ser localizados em seu interior, representando-os através de seus contornos;
- b) Camada Semântica: modelagem das características extraídas da representação da imagem, as quais podem ser espaciais, indicando relacionamentos sobre a posição relativa dos objetos ou sobre a forma destes, descritas através de parâmetros tais como o centro e raio da circunferência de sobreposição. As características são classificadas em *derivadas*, obtidas diretamente a partir da imagem, *compostas*, obtidas através da composição de outras, ou *conceituais*, apresentando um conhecimento associado;
- c) Camada de Conhecimento: contem a lógica para a interpretação da semântica da imagem e da similaridade entre estas, definidas a partir do conjunto de características extraídas. Para tal, utiliza-se de hierarquias de tipos abstratos (TAH) para representar os conceitos gerais referentes a imagens (Figura 4).

presença de termos conceituais. Consultas tais como “localizar os pacientes que apresentem um grande tumor próximo à lateral do ventrículo” são formuladas como:

```
SELECT    pacienteComImagem (paciente:  t.paciente,  imagem:
t.imagem)
FROM      tumor t, Ventrículo_Lateral v
WHERE     NEARBY v and t.size IS "large";
```

onde, `pacienteComImagem` consiste em um tipo construído utilizado para exibir o resultado da consulta.

O operador `SIMILAR-TO` é utilizado na busca por objetos que sejam semelhantes a algum específico, baseando-se (`BASED-ON`) em um conjunto de características especificadas na consulta. Como exemplo, considere uma consulta para recuperar os pacientes com tumor no cérebro similar ao apresentado pelo paciente “P0-01”, considerando o seu tamanho e a localização próxima ao ventrículo lateral:

```
SELECT
pacienteComImagem(paciente:il.paciente,imagem:il.imagem)
FROM Imagem il, ip
WHERE il SIMILAR_TO it (it.paciente.id = 'P0-01')
      BASED_ON (it.tumor.tamanho,
                it.[tumor, ventriculo_lateral].( $x_c, y_c, q_c, d_c$ ))
```

(x_c, y_c, q_c, d_c) consistem de parâmetros com os seguintes significados, respectivamente: coordenadas x e y do centro da figura, distância ao centro (raio), e ângulo de cobertura.

O conceito de tipo abstrato de dados possui uma importância fundamental no processamento das consultas no sistema KSIM, cujas hierarquias permitem a classificação do conjunto de imagens no BD. O conceito semântico presente em cada nó do TAH é representado através de intervalos de valores das características, ou

seja, a tupla (x_c, y_c, q_c, d_c) , o que permite a formulação de consultas considerando a similaridade de imagens.

O processamento de uma consulta no sistema KSIM é dividido em três fase (Figura 5):

- Análise da consulta e seleção das características;
- Matching* baseado em conhecimento do conteúdo da imagem;
- Flexibilização da consulta.

Figura 5 - Diagrama de Fluxo do Processamento de Consultas Baseado em Conhecimento

Na fase inicial, baseada na imagem alvo, contexto da consulta e tipo do usuário, o sistema seleciona as características relevantes e os relacionamentos espaciais para o processamento da consulta. Para as consultas com o operador SIMILAR-TO, são selecionadas as características e relacionamentos espaciais especificados na cláusula BASED-ON. Caso esta cláusula não esteja presente, o tipo

do usuário e os objetos presentes na imagem alvo são utilizados para a obtenção destes parâmetros.

A fase seguinte consiste na seleção dos TAH que podem ser utilizados para validar os possíveis resultados da consulta. O conjunto de imagens presentes no TAH que mais se aproximam dos valores presentes no intervalo de validação representa o conjunto de imagens semelhantes ao objetivo da consulta. A fase de flexibilização consiste em percorrer a hierarquia dos TAH com valores distintos para o intervalo de validação, obedecendo aos requisitos estabelecidos pelo usuário.

3.7. SISTEMA PROBE.

PROBE [Orenstein_88] consiste de um sistema de banco de dados orientado a objetos com suporte ao tratamento de informações espaciais, no qual o objeto básico consiste no ponto. A modelagem da informação espacial é realizada no sistema PROBE através de dois tipos básicos de objetos: *entidades* e *funções*, definindo o seu modelo de dados espacial (PDM). As entidades correspondem ao mesmo conceito existente nos modelos de dados, ou seja, representa conceitos ou objetos presentes nas aplicações, e os que apresentam características semelhantes são agrupadas em *tipos de entidades*, as quais podem ser divididas em *subtipos*, formando uma hierarquia de generalização.

As propriedades, relacionamentos entre entidades e as operações possíveis são representadas pelas funções, que podem ser de dois tipos:

- a) Funções Computacionais: produzem valores a partir da execução de um procedimento;

b) Funções Armazenadas: produzem valores a partir de uma busca convencional no banco de dados, realizada pela função armazenada *extent*.

A arquitetura do sistema PROBE está descrita na Figura 6 e, genericamente possui os componentes principais de um banco de dados orientado a objetos, ou seja,

a) *Núcleo* do sistema para o processamento das linguagens de consulta e gerenciamento dos tipos de objetos e b) *Classes de Objetos*, representando objetos de um novo tipo e as operações suportadas.

Figura 6: Arquitetura do Sistema PROBE

O conjunto de características que podem descrever a informação espacial presente em uma determinada imagem é suportada pela classe “IMAGE”. O processo de extração e associação destas características a cada objeto é realizado pelos métodos desta classe, em particular este último processo é realizado pela função

occurrences(). A forma de armazenamento das imagens também pode ser especificada na classe “IMAGE”, como, por exemplo, um conjunto de pixel. Uma vez com as características de cada imagem descritas e representadas, as consultas são formuladas e avaliadas, seguindo uma estratégia denominada filtro ou aproximação geométrica [Orenstein_86]. Este filtro realiza um processamento inicial do conjunto de objetos presentes no banco, selecionando um grupo que possa satisfazer a

condição da consulta. Para formalizar o conjunto com o resultado, novos métodos mais restritivos devem ser aplicados.

O objetivo do projeto PROBE com a sua conceituação e métodos de construção consiste em fornecer suporte ao tratamento de informações espaciais que possam ser representadas como pontos. Assim, operações como pertinência de um objeto a uma região do espaço, sobreposição de regiões, continência e posicionamento relativo são definidas sobre os conjuntos de pontos que delimitam cada região.

3.8. SISTEMA PICTION.

O Sistema Piction [Srihari_95] realiza a descrição do conteúdo das imagens através de uma análise das legendas que se encontram associadas, especificamente de imagens de pessoas ou fatos jornalísticos. Com um processo de extração da informação presente na legenda, é possível realizar operações de recuperação da imagem através de seu conteúdo e também identificar os rostos que se encontram presentes. A abordagem para o desenvolvimento deste projeto realiza uma divisão do problema em duas áreas: a) processamento da linguagem de entrada (legenda) e b) projeto de uma arquitetura que suporte o gerenciamento desta informação e incorpore as técnicas de análise das imagens.

Para análise das legendas associadas a cada imagem, segue-se o princípio de que o texto apresenta informações sobre quem e o que se encontra na imagem e fornece elementos que permitem localizar e identificar os objetos no interior da cena. Através da combinação destas informações com o conhecimento prévio da aparência e tipo de objetos que possam compor a cena, é possível definir a sua semântica

visual, ou seja, compreender o seu conteúdo. A semântica visual define um conjunto de restrições sobre a cena que acompanha, divididas em:

- *Espaciais*: restrições sobre a topologia e geometria dos objetos, definindo o seu posicionamento relativo no interior da cena;
- *Características*: descreve as propriedades de cada objeto, como cor do roupa, cabelo e outras;
- *Contextual*: descreve o contexto geral da cena e os objetos existentes.

A arquitetura do sistema Piction é composta pelos seguintes módulos:

- a) Processamento de Linguagem Natural: este módulo recebe como entrada a legenda original da imagem e produz o conjunto de restrições de definição da semântica visual da cena, através da identificação de elementos importantes no texto que possa definir algum dos aspectos da semântica, tais como nomes próprios, posicionamento relativo, adjetivos, cores, e outros;
- b) Módulo de Controle: gerencia o funcionamento dos demais módulos do sistema;
- c) Interpretação da Imagem: responsável pela localização e segmentação dos objetos, e pela extração das propriedades visuais de faces humanas.
- d) Base de Conhecimento Integrada para Visão/Linguagem: apresenta tipos distintos de base de conhecimento para auxiliar no processo de extração da informação visual a partir do texto. Estas bases consistem em: i) conhecimento léxico, definição das palavras e semântica associada; ii) conhecimento visual, definindo esquemas de descrição dos possíveis objetos; iii) conhecimento do mundo, contendo informações sobre pessoas, locais, eventos e descrição geral do domínio

e iv) conhecimento específico da imagem, disponibilizando informações sobre as imagens e legendas previamente analisadas e processadas.

3.9. SISTEMA DE RECUPERAÇÃO DE IMAGENS POR FORMAS SIMILARES (FIBSSR).

O sistema denominado de Sistema de Recuperação de Imagens por Formas Similares (FIBSSR) [Mehrotra_95] define mecanismos que permitem a descrição e a elaboração de consultas sobre um banco de imagens, utilizando-se da semelhança entre as formas de objetos presentes no interior de cada uma. Os objetos localizados podem se apresentar genericamente como compostos por elementos rígidos e/ou articulados.

As imagens são armazenados no banco de dados na forma de elementos que descrevem a forma dos objetos que esta possa conter. Assim, a imagem é considerada pelo sistema como uma forma a ser descrita e inserida no banco de dados para um acesso posterior em decorrência de uma consulta. Cada forma a ser armazenada no BD é processada para se obter a sua superfície e pontos de interesse que a definam, consistindo em vértices ou pontos máximos de curvatura. A superfície é, desta forma, representada como uma sequência de segmentos definidos pelos pontos de interesse, proporcionando uma aproximação polinomial para a forma da imagem. Cada segmento é definido no sistema como uma característica da superfície, formada por um conjunto de pontos de interesse adjacentes.

A representação de cada característica da superfície é realizada de forma a ser invariante quanto a rotação, translação e mudança de escala. Assim, seja uma característica F com n -pontos de interesse, um par é escolhido como o vetor de base. Este é colocado coincidente ao eixo- x e definido como o vetor unitário para a

determinação das demais características, obtendo a sua representação normalizada. Uma característica F é portanto descrita por:

- conjunto de pontos $((x_1, y_1), \dots, (x_n, y_n))$, onde (x_i, y_i) é a coordenada normalizada do i -ésimo ponto de interesse;
- vetor de parâmetros de transformação $P = (S, T_x, T_y, \theta)$, onde S é a escala (ou tamanho) do vetor base da normalização, T_x e T_y são os parâmetros de translação, e θ consiste no ângulo que o vetor base forma com o eixo x .

Um objeto rígido é representado por uma coleção de características como descrito anteriormente. Um objeto articulado é representado através do conjunto de objetos rígidos e dos pontos de articulação existentes em comum entre cada um.

O vetor de características descrevendo a forma dos objetos é utilizado na definição da chave de acesso multidimensional para o banco de imagens. A similaridade entre duas características é definida pela distância Euclidiana entre os seus vetores representativos. A árvore k-D-B é utilizada como estrutura de indexação, na qual cada nó folha possui associada uma lista contendo informação sobre a localização e a forma desta característica. Para objetos rígidos, a lista contém elementos da forma (parâmetros de transformação, identificador do objeto) e para os objetos articulados, apresenta uma entrada para cada ponto de articulação na forma de parâmetros de transformação, identificador do objeto, localização dos pontos de articulação, e identificador dos componentes irmãos.

O processamento de consulta neste sistema envolve três etapas principais: seleção da característica indicada na consulta, formulação das possíveis formas similares e formalização do resultado final. O processo de seleção da característica da

consulta para a busca, indicada como robusta e relativamente insensível a ruídos e efeitos de distorção, é o que apresentar ângulos retos e longos segmentos de reta. Cada característica similar encontrada na busca, dentro do limite de tolerância definido, apresenta uma lista de formas que a contém. Estas formas, articuladas ou rígidas, consistem no possível resultado final para a consulta. Se o resultado formar um conjunto pequeno, o usuário poderá selecionar manualmente as formas que sejam mais adequadas para o objetivo da consulta. Do contrário, o sistema poderá auxiliar neste processo, adotando a estratégia de determinação das regiões de sobreposição existente entre a forma sugerida na consulta e as que possam compor o resultado final.

O sistema apresenta uma interface gráfica na qual pode-se selecionar a forma a partir de um conjunto pré-definido, ou esta pode ser desenhada livremente, de acordo com o requisito atual para a consulta. Uma vez indicada a forma, será obtida a sua descrição e seleção da melhor característica para que se possa processar a consulta sobre o banco de imagens.

3.10. SISTEMA CHABOT.

O projeto CHABOT [Ogle_95] consiste de um sistema para a consulta de imagens baseada no histograma das cores que se encontram presentes. As imagens são armazenadas no banco de dados relacional com características de orientação a objetos, como a definição de classes e mecanismos para a herança de atributos, denominado de Postgres [Stonebraker_90]. O domínio da aplicação utilizado inicialmente para o projeto, consiste de imagens em que se encontram presentes

elementos da natureza, como florestas, rios, lagos, obtidas pelo Departamento de Recursos de Água do Estado da Califórnia.

As imagens são descritas por uma série de atributos textuais que descrevem o processo de aquisição, local, data e tema a que se refere, além do histograma de cores. A distribuição de cores é determinada no instante da inserção de cada nova imagem no BD. As consultas realizadas sobre o BD com imagens são formuladas indicando-se alguns valores e condições para os atributos de descrição da informação a ser recuperada. O mesmo ocorre para o atributo que consiste no histograma, o qual pode-se indicar critérios que definem a comparação com as imagens do BD. Estes critérios consistem de distribuições de cores padrão como, por exemplo, *SomeOrange*, *MostlyRed*, *WhiteStuff* e outros semelhantes, e a comparação é realizada através de uma função denominada de *MeetsCriteria()*, que define a semelhança entre os histogramas das imagens presentes no BD com a condição da busca.

A linguagem de consulta ao BD consiste de uma extensão à PostQuel, definida para o Postgress, na qual se encontra presente a chamada à função de avaliação do critério de comparação para o histograma das imagens. Uma consulta para recuperar todas as imagens da área da Baía de São Francisco existentes no BD que foram obtidas a partir do dia primeiro de janeiro de 1994 e que apresentam uma coloração alaranjada, é formulada como:

```
retrieve (q.all) from q in photocd_bib where
    q.shoot_date > "1_jan_1994" and
    q.location "San Francisco Bay Area"
    MeetsCriteria ("SomeOrange", q.histogram);
```

Além deste tipo de consulta, o sistema oferece a possibilidade da definição de novos critérios de comparação, utilizando-se de comandos próprios da linguagem. Estes elementos definem as *consultas conceituais*. Assim, para especificar um novo critério de busca denominado de flores púrpuras, então pode-se escrever:

```
Concept Name: purple flowers
Description:  q.description ~ "flower" and
              MeetsCriteria("SomePurple",
q.histogram);
```

3.11. RECUPERAÇÃO DA INFORMAÇÃO EM SISTEMA DE MULTIMÍDIA.

Em aplicações de banco de dados contendo informações presentes em vídeo, vídeo sob demanda, monitoramento industrial, monitoramento de tráfico de veículos, e outras, apresentam como requisito o armazenamento e processamento de uma grande quantidade de imagens contendo objetos em movimento. Neste domínio, para manipular toda esta quantidade de informação, a operação de classificação em categorias semelhantes representa um auxílio importante na solução de consultas baseadas em conteúdo do vídeo.

Além disso, a presença de objetos em movimento consiste em uma característica única deste tipo de imagens. A análise e identificação dos movimentos existentes complementa o processo de classificação das sequências de vídeo, e este pode ser realizado de forma independente da identificação dos objetos presentes em cada cena [Flickner_95]. Portanto, uma consulta envolve não apenas a descrição dos possíveis objetos a serem recuperados, mas também uma indicação sobre os movimentos apresentados por estes. O resultado desta consulta pode consistir de uma

única cena ou uma sequência de imagens descrevendo o vídeo [Weiss_95] [Woelk_86].

As abordagens para a recuperação de imagens presentes em vídeo, descrevem a informação sob dois aspectos: *temporais*, contendo a indicação dos movimentos relativos dos objetos na cena, e *espaciais*, que consiste na elaboração da posição relativa dos objetos e das formas por estes apresentados. A abordagem fornecida por Dimitrova & Golshani [Dimitrova_95] descreve a informação contida no vídeo pela tripla (O, M, V) , onde:

- O : representação do objeto através das características extraídas (texturas, esboço do objeto, outros objetos que o compõe, e outras);
- M : descrição do movimento dos objetos, ou seja, velocidade e curvatura da trajetória;
- V : identificação da sequência de vídeo a que o objeto pertence, indicando os seus *frames* iniciais e finais.

Uma consulta sobre este Banco de Dados Multimídia é formulada genericamente como $\{f(x) \mid P(x)\}$, onde $f(x)$ indica os objetos a serem recuperados, e $P(x)$ consiste no predicado de validação. Esta linguagem suporta operadores que pertencem às categorias de conjunto, igualdade, sincronização temporal, composição espacial, e outras necessárias para o tratamento de informações textuais.

O projeto MACS (*Media Abstraction Creation System*) [Brink_95] define uma abstração para a informação presente em uma sequência de imagens em movimento pela tupla $M = (ST, fe, \lambda, R, F, Var_1, Var_2)$, onde:

- ST : conjunto de objetos da sequência, denominados de estados;

- fe : características dos objetos, definindo informações que descrevam o domínio de interesse, tais como pessoas a serem identificadas, peças de fabricação, carros e outras;
- λ : consiste em um mapeamento que indica a possibilidade da ocorrência de cada objeto com uma determinada característica em uma cena;
- R : define relacionamentos entre os possíveis objetos presentes na cena, podendo indicar posições espaciais, ou relações da forma: “uma pessoa *está usando* um camisa azul clara, calça jeans e tênis”;
- F : define relacionamentos entre as características, os quais podem ser ou não independentes dos estados (objetos);
- Var_1 : conjunto de objetos denominados variáveis de estado, definido com valores nos objetos da aplicação (ST);
- Var_2 : conjunto de objetos denominados de variáveis de características, definido com valores sobre as características suportadas pela aplicação (fe).

Desta forma, um banco de dados multimídia é definido como um conjunto de abstrações de mídia, conforme definido acima, e as consultas são especificadas em comandos de forma semelhante ao cálculo de predicados apresentado pela linguagem PROLOG.

O processo de identificação dos objetos em movimento e sua representação, além do sincronismo necessário para exibição da informação resultante de uma consulta que possa envolver outras mídias (som, por exemplo), torna o sistema de banco de dados multimídia distinto dos demais descritos neste capítulo. O aspecto de semelhança, consiste na abordagem de descrição da informação presente nas imagens

de definição do vídeo, através de características gráficas, e a utilização destas como itens de formulação de consultas e acesso ao conteúdo do BD.

3.12. SISTEMA DE INFORMAÇÃO GEOGRÁFICA.

Os sistemas de informação geográfica, conhecidos como GIS (*Geographic Information Systems*), manipulam um grande volume de informações provenientes de diversas origens, a partir de especialistas no domínio da aplicação, ou através das imagens, obtidas pelos satélites de rastreamento da superfície. As informações extraídas desta categoria de imagens consistem na descrição dos objetos presentes na cena, a qual pode ser realizada por métodos automatizados de processamento de imagens específicos ao tratamento destas quando obtidas por satélites e complementadas com a descrição de um especialista. Outro aspecto importante consiste na especificação da posição relativa dos objetos presentes na imagem, definindo-se relações espaciais entre estes, conforme mencionado em alguns sistemas anteriores (PSQL, PROBE e SQL Espacial).

Considerando a descrição da imagem geográfica através de informações espaciais, vários dos sistemas anteriores que suportam este tipo de dado, podem ser utilizados neste domínio de aplicação. No entanto, alguns sistemas definem operadores específicos para realizar suporte a alguns requisitos da área, tais como a determinação e planejamento de rotas [Holmes_92] [Kasturi_89] em mapas digitalizados [Tanaka_88]. De forma semelhante aos requisitos apresentados por um Sistema de Informação com Suporte a Imagens (SISI), a arquitetura de um GIS deve ser definida de maneira a proporcionar um tratamento uniforme ao conjunto de

informações manipulado e apresentar uma interface adequada para a sua utilização pelo usuário [DeWitt_92] [Oliveira_96].

3.13. CONSIDERAÇÕES FINAIS.

As diversas implementações dos Sistemas de Informação com Suporte a Imagens procuram cada uma à sua maneira solucionar o problema da formulação da consulta baseada no conteúdo gráfico da imagem a ser recuperada. Desta forma, várias abordagens foram utilizadas para atingir este objetivo, trabalhando nas várias etapas existentes, ou seja, extração, representação, organização, formas de acesso, linguagens de consulta e apresentação do resultado final.

A seguir é apresentado um quadro resumo com os principais aspectos que caracterizam a implementação destes sistemas, que consistem em:

- a) Modelo de dados: qual é o modelo de BD utilizado para o suporte às informações manipuladas pelo sistema;
- b) Representação da imagem: estrutura interna existente para o armazenamento da imagem;
- c) Definição das Características: abordagem utilizada para definir as características de descrição do conteúdo da imagem;
- d) Método de indexação: indica quais elementos definidos como chaves de acesso às imagens presentes no banco;
- e) Estrutura de acesso: definição da estrutura de dados utilizada para organizar os índices de acesso às imagens;
- f) Domínio da aplicação: indica o tipo de aplicação que se destina o sistema;

- g) Linguagem de Consulta: abordagem utilizada na especificação da linguagem de consulta;
- h) Interface: característica da forma de entrada dos dados de descrição da imagem e a forma de apresentação ao usuário final.

Os pontos do quadro que não apresentam informações significam que estas não foram mencionadas de forma explícita nas referências pesquisadas para o sistema apresentado.

Sistema	Modelo de Dados	Representação da Imagem	Definição das Características	Método de Indexação	Estrutura de Acesso	Domínio de Aplicação	Linguagem de Consulta	Interface
PICQUERY+ [Cardenas_93]	Orientado a Objetos	Pilha com a matriz de pixel das imagens	Sistema de Processamento de Imagens			Imagens Médicas	Extensão ao QBE	Gráfica e Textual
PSQL [Roussopoulos_88]	Relacional com modificações	Imagem Simbólica		atributos textuais e ícones	Árvores-R e R*	GIS	Extensão ao SQL	Gráfica (ícones) e textual
SSQL [Egenhofer_94]	Relacional com modificações	Imagem Simbólica		atributos textuais e ícones		GIS	Extensão do SQL	Gráfica (desenho) e textual
Sistema de Imagens 3D [Bimbo_93]	Orientado a Objetos	Imagem Simbólica		ícones e extensões a string 2D		Imagens e Vídeo	Desenho Gráfico	Gráfica
QBIC [Flickner_95]	Relacional	Características gráficas	Extração de informação e descrição textual	Características gráficas mensuráveis	Árvores-R	Imagens e Vídeo	Desenho Gráfico	Gráfica
KSIM [Hsu_96]	GemStone	Características gráficas	Extração da forma e relações espaciais	Características gráficas mensuráveis	MDISC	Imagens Médicas	Linguagem SQL OO	Textual
Piction [Srihari_95]	Redes Semânticas	Rede Semântica	Processamento de Linguagem Natural e Proc. de Imagens			Imagens com legendas	Linguagem Natural	Textual
FIBSSR [Mehrotra_95]		Vetor descritivo da forma	Aproximação por figuras geométricas	vetor de características	Árvore k-D-B	Imagens Geográficas	Desenho Gráfico	Gráfica
PROBE [Oresntein_88]	Orientado a Objetos	Conjunto de pontos						
CHABOT [Ogle_95]	Postgres	Histograma	Determinação do Histograma	Textual		Imagens da Natureza	Postquel (extensão do SQL)	Gráfica e textual

BANCO DE DADOS ORIENTADO A OBJETOS

4.1. CONSIDERAÇÕES INICIAIS.

Como consequência imediata da complexidade das aplicações surgidas em domínios distintos, foram propostos diversos modelos que procuram fornecer aspectos conceituais que possibilitam a incorporação de um maior significado semântico às informações representadas em uma modelagem. Os modelos de dados Orientado a Objetos fornecem mecanismos de modelagem e organização da informação mais adequados aos requisitos apresentados por uma série de aplicações denominadas não convencionais [Atkinson_89] [Kim_89] [Nierstrasz_89] [Kim_90a] [McLeod_91] [Bertino_93] [Cattell_94]. Dentre elas pode-se citar o desenvolvimento dos banco de dados com suporte a imagens.

Um banco de dados orientado a objetos consiste de uma coleção de objetos, cujos aspectos comportamentais e seus estados, além dos relacionamentos existentes entre eles, são representados por um modelo de dados orientado a objetos. Neste capítulo serão apresentados os principais conceitos e abordagens que definem um Sistema de Gerenciamento de Banco de Dados Orientado a Objetos (SGBDOO), que demonstram como estas construções podem auxiliar no processo de desenvolvimento de um Sistema de Informação com Suporte a Imagens (SISI).

4.2. CONCEITOS GERAIS DE ORIENTAÇÃO A OBJETOS.

No contexto de modelagem de dados, o paradigma de orientação a objetos resume-se na concentração de toda a informação em elementos denominados “objetos”. Dessa forma, as entidades conceituais de uma aplicação são modeladas através de objetos, sendo estes dispostos de maneira que representem da melhor forma possível a semântica do problema a ser resolvido.

A seguir serão apresentados alguns conceitos propostos por Kim [Kim_90b] como sendo os elementos que um modelo de dados orientado a objetos deve suportar, e que formam um conjunto teórico básico sobre o qual fundamentam-se as diversas variações sobre o tema.

- *Objeto*: consiste de um item de dado representado por alguma memória privada onde se armazena um conjunto de informações. A estrutura da memória privada é determinada por definições de propriedades ou atributos, e as operações são especificadas pela definição de métodos.
- *Tipo*: uma coleção de definições (especificação) para a estrutura de um objeto e seus métodos. Um tipo pode ser construído sobre outros métodos. A linha de distinção entre o conceito de classe e tipo de objetos não é completamente clara, mas pode-se definir como [Bertino_93] a) tipo: define a especificação de um conjunto de objetos e, b) classe: define a estrutura e a implementação de um conjunto de objetos.
- *Classe*: uma coleção de objetos similares, ou seja, que compartilham o mesmo tipo. Uma classe funciona com um gabarito (*template*) [Bertino_91] para a definição de um conjunto de atributos de instância, um conjunto de mensagens para uso externo e um conjunto de métodos que se aplicam aos objetos daquela

classe. O relacionamento semântico existente entre um objeto e a classe a qual pertence é denominado “instância_de” (*instance_of*).

- *Instância*: ocorrência de um objeto dentro de uma classe. Um objeto pode pertencer a uma única classe como instância desta.
- *Metaclasse*[Bertino_91]: uma classe cujas instâncias são classes. Desta forma pode ser possível definir atributos e métodos que caracterize a classe e não os objetos que sejam instâncias desta. Como exemplo, considere um atributo para a classe cujo valor seja uma sumarização dos valores assumidos por seus objetos.
- *Subtipo*: um tipo construído sobre outros tipos (supertipos), com o objetivo de obter uma descrição mais especializada em relação aos seus supertipos.
- *Supertipo*: um tipo que representa informações mais genéricas em relação a outros tipos, os quais são denominados subtipos.
- *Método*: uma descrição de como executar uma operação de um determinado objeto, representando o seu comportamento. Ele é composto por: uma assinatura (identificação do método), na qual é especificado o seu nome, os nomes e classes dos argumentos e a classe do resultado, se o método retornar algum; e pelo corpo, que consiste na implementação através de um conjunto de instruções escritas em uma determinada linguagem de programação.
- *Mensagem*: requisito para a ativação de métodos entre os objetos.
- *Objetos complexos*: são objetos ligados via um relacionamento denominado “é-parte-de” (*is-part-of*). Um objeto participa de um relacionamento “é-parte-de” se ele possuir uma referência a outro objeto participante no relacionamento.

O paradigma de orientação a objetos é baseado nos conceitos acima, e organizados segundo os seguintes princípios [Bertino_91]:

- Encapsulamento de dados e métodos: define que o acesso ao estado de um objeto seja realizado somente através dos seus métodos. A representação do estado de um objeto é geralmente realizada por variáveis que são ocultas aos usuários do objeto, consistindo em seus atributos.
- Métodos e passagem de mensagens: O conjunto de métodos determina o comportamento do objeto, e a passagem de mensagens é o meio pelo qual os objetos se comunicam.
- Classificação e instanciação: objetos são organizados dentro de classes. Uma classe determina as propriedades estruturais e comportamentais de suas instâncias. As propriedades estruturais são especificadas por um conjunto de variáveis, e as propriedades comportamentais por um conjunto de métodos.
- Herança: uma classe é interpretada como um *template* para a criação de instâncias daquela classe, isto é, define como são as instâncias de um tipo e a forma como estas instâncias são criadas como representação de conceitos do mundo real. Com o mecanismo de herança, uma classe, denominada *subclasse*, pode ser definida apresentando como base uma outra classe (*superclasse*). Uma classe pode herdar métodos das suas *superclasses*, além de apresentar os seus próprios atributos e métodos, os quais, por sua vez, podem ser herdados por suas *subclasses*, definindo uma hierarquia de classes. Herança é definida como um mecanismo para compartilhar recursos nas hierarquias de classes.
- Identidade de objetos: o sistema define e mantém um identificador único para os objetos. Isto permite que objetos possam ter os mesmos atributos e valores e sejam identificados sem a necessidade de chaves únicas para suas instâncias.

Vários projetos de SGBDOO estão sendo realizados tanto na indústria quanto em centros de pesquisa. Um resumo comparativo de alguns dos principais projetos desenvolvidos é apresentado por Zand (Tabela 1) [Zand_95], onde são analisados os seguintes aspectos:

- *Gerenciamento de Versões*: define o suporte oferecido para o gerenciamento de versões do Esquema;
- *Gerenciamento de Transações*: define se o gerenciador oferece mecanismos para o controle de transações;
- *Objetos Compostos*: indica se o modelo suporta a representação de objetos definidos através da abstração de composição;
- *Herança Múltipla*: indica se o modelo suporta o mecanismo de herança múltipla no processo de especialização de objetos de um determinado tipo;
- *Distribuição*: define se o Gerenciador permite o controle da distribuição de dados;
- *Evolução Dinâmica do Esquema*: indica se o sistema permite a evolução do esquema de uma aplicação, após a sua definição;
- *Multimídia*: indica se o Gerenciador apresenta suporte aos tipos de dados necessários em aplicações multimídia, ou seja, imagem, som e vídeo;
- *Linguagem*: define as linguagens utilizadas na definição e manipulação dos tipos de dados suportados pelo modelo;
- *Extensão do Relacional / Orientado a Objetos*: define se a abordagem para a elaboração do modelo, consiste em extensões ao modelo Relacional ou consiste em um sistema orientado a objetos.

	ENCORE	EXODUS	GemStone	IRIS / OpenODB	O2	Object Store	ONTOS	ORION	OZ+	Postgres
Ger. de Versões	Sim	Sim	Sim	Sim	Limitado	Sim	Sim	Sim	Sim	Sim
Recuperação	Baseado em reg. de logs	Define páginas de log	??	Através da HP-SQL	Sim	Baseado em reg. de logs	Sim	Combina regs. de logs e paginação	Atualização periódica	Sobre- escrita
Ger. de Transação	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Sim
Objetos Compostos	Sim	Sim	Não	Restritos	Sim	Restritos	Não	Sim	Objetos Complexos	Sim
Herança Múltipla	Sim	Sim	Não	Sim	Sim	Sim	Sim	Sim	Não	Sim
Distribuição		Não	Sim	---	Sim	Não	Sim	Sim	Sim	Não
Evolução dinâmica do Esquema	Sim	---	Sim (limitada)	Sim (limitada)	Sim (limitada)	Sim (limitada)	Sim	Sim	---	---
Multimídia	Sim	Sim	Sim	Não	Sim	Sim	Não	Sim	Não	Sim
Linguagens	C	E (extensão do C++)	C, C++, OPAL, Parc Place, Smalltalk	C, LISP, OSQL	C	C, C++	C++	LISP, C	OZ+	C, C+, LISP, PostQuel
Ext. Relacional / OO	Orientado a Objetos	Orientado a Objetos	Orientado a Objetos	Ext. Relacional	Orientado a Objetos	Orientado a Objetos	Orientado a Objetos	Orientado a Objetos	Ext. Relacional	Ext. Relacional

Tabela 1 - Principais Características do Modelos de Dados Orientado a Objeto.

4.3.LINGUAGENS DE CONSULTA ORIENTADA A OBJETOS.

Neste item será descrita uma proposta completa para uma linguagem de consulta orientada a objetos, fundamentada no cálculo de predicados orientado a objetos definido por Bertino [Bertino_92]. Nesta proposta são cobertos alguns dos principais aspectos referentes a uma linguagem de consulta orientada a objetos, a saber:

- *Acesso aos objetos*, seguindo o princípio de navegação pelo Esquema da aplicação;
- Suporte a *hierarquias de agregação* e de *herança*;
- Definição da operação de junção (“*join*”);
- Suporte a consultas recursivas e a presença de métodos;
- Consistência teórica na definição do resultado de uma consulta.

4.3.1. Modelo Orientado a Objetos de Referência.

Os conceitos de orientação a objetos utilizados na formulação do Cálculo de Predicados Orientado a Objetos (OOPC) [Bertino_92] definem um modelo básico de referência. Neste modelo um **objeto** possui estado e comportamento associados. O *estado de um objeto* é definido como o valor das propriedades (atributos ou variáveis de instâncias), podendo estas serem primitivas ou objetos não primitivos, representando a definição de objetos complexos, através da abstração de agregação. O *comportamento* do objeto é especificado através de métodos que operam sobre seus estados. Os objetos com mesmo conjunto de propriedades e métodos são agrupados em classes, sendo que um objeto pode ser instância de uma única classe, ou de várias, desde que sejam definidos elementos de qualificação da classe a qual pertencerá em determinado estado do banco. As classes são organizadas através de hierarquias que modelam a semântica do relacionamento ‘IS-A’ entre elas. Uma classe é especificada indicando-se o nome,

propriedades, métodos e o nome de suas super-classes, a partir das quais é definida a herança de informação (propriedades e métodos). Propriedades da classe que sejam multivaloradas são definidas através com construtor de tipo *coleção*.

Uma expressão no OOPC apresenta a seguinte estrutura:

{Objetivo, Abrangência, Qualificação}

1. *Objetivo*: Indica as informações a serem recuperadas a partir do BD, apresentando o seguinte formato:

- v ;
- $v.A_i$;

onde, v é uma variável representando um objeto (*variável objeto*) pertencente a uma determinada classe, e A_i indica o nome de uma propriedade da classe de v . Indicando-se como objetivo da consulta apenas a variável objeto (v), será recuperado o identificador do objeto (OID), apresentando o significado de que este objeto, juntamente com suas propriedades e métodos, torna-se disponível para operações posteriores que possam ser ativadas sobre o resultado da consulta. Especificando a propriedade da classe ($v.A_i$), o resultado consiste no conjunto de objetos que pertencem ao domínio da classe da propriedade.

2. *Abrangência*: Define as classes de objetos sobre as quais serão retiradas as informações descritas no Objetivo da consulta com o seguinte formato:

- v/C

onde v é o nome da variável objeto e C a indicação da classe. Através da ligação da variável à classe, define-se: a) o seu tipo: métodos e propriedades válidos para acesso, e;

b) a abrangência da variável: objetos que serão atribuídos a partir do conjunto de instâncias pertencentes à classe.

3. *Qualificação*: Consiste de uma fórmula bem-formada, definida como uma expressão lógica composta por predicados combinados através de operadores lógicos ($\wedge, \vee, \neg$). Um predicado pode apresentar os seguintes formatos:

a. Atômico:

- Verdadeiro;
- Falso;
- $t_i \theta t_j$: onde t pode ser uma constante, uma variável de objeto ou um termo $v.A_i$, e θ um operador de comparação;

b. Qualificado:

- Θ *Abrangência (Qualificação)*: onde Θ pode ser um qualificador universal ($\forall$) ou existencial ($\exists$);

Figura 1 - Esquema da Aplicação Exemplo para o OOPC.

A Erro! A origem da referência não foi encontrada. apresenta o diagrama com o esquema do BD de controle de projetos, o qual será utilizado na formulação dos exemplos ao longo deste item, a menos de uma referência em contrário. Neste diagrama, o símbolo *, presente ao lado de alguns atributos, indicam que estes são multivalorados. Assm, considere a seguinte consulta inicial, que demonstra as características básicas de uma expressão no OOPC:

Exemplo 1: Recupere o nome de todos os empregados cujo salário anual seja maior que \$100.000.

```
{v.e_nome; v/Empregado; v.Salario > 100.000}
```

Neste exemplo, o objetivo da consulta consiste em objetos que pertencem ao domínio da classe e_nome, definida como atributo dos objetos da classe Empregado (que consiste na abrangência da consulta). A qualificação dos objetos é indicada pelo

predicado atômico de comparação entre os valores de um objeto (Salário) com uma constante (100.000).

Inicialmente, o OOPC admite apenas uma classe na cláusula de abrangência e no máximo uma propriedade como objetivo final da consulta. Assim o resultado da consulta consiste em objetos que pertencem a uma única classe básica, definida no esquema da aplicação.

4.4.2. Predicados de Igualdade.

Nas linguagens de consulta relacional, a igualdade é definida normalmente entre os valores dos atributos das relações. Para que seja identificada a presença de duas entidades iguais, as suas tuplas devem possuir valores iguais para os seus atributos chave. Nas BD Orientada a Objetos é importante distinguir entre tipos distintos de igualdade.

A) Igualdade de Identidade de Objetos: Dois objetos O e O' são iguais por identidade se eles forem o mesmo objeto, ou seja, possuírem o mesmo OID;

B) Igualdade por Valor de Objetos:

- a) Dois objetos primitivos são iguais por valor se eles apresentarem o mesmo valor, por exemplo, $5 = 5$, $'A' = 'A'$, $"Casa" = "Casa"$;
- b) Dois objetos não primitivos O e O' são iguais por valor se: i) O e O' possuírem o mesmo número de propriedades, e: ii) Considere P_i de O , então existe a propriedade P_j de O' , tal que ambas apresentem o mesmo nome e as propriedades P_i e P_j sejam iguais por valor. Como exemplo, uma condição de qualificação de uma expressão do OOPC, onde o endereço do instituto deva ser o mesmo de um determinado empregado, ou seja, $i.Localizacao =$

e. *E_endereco*, será verdadeira se obedecer à definição de igualdade por valor de objetos. Neste exemplo, *i* e *e* representam dois objetos não primitivos, pertencentes respectivamente às classes Instituto e Empregado.

C) Igualdade por valor de Propriedades: Considere duas propriedades P_i do objeto O e P_j de O' , elas são iguais por valor se: i) a estrutura de definição do tipo para P_i seja igual a de P_j , e; ii) se a propriedade P_i é monovalorada, então $O.P_i$ é igual por valor a $O'.P_j$. Do contrário se P_i for multivalorada, devem ser verificadas as seguintes condições:

- cardinalidade de $O.P_i$ é igual à cardinalidade de $O.P_j$;
- para cada objeto O_k pertencente a $O.P_i$, existe um objeto O'_k pertencente a $O.P_j$, tal que O_k e O'_k sejam iguais por valor.

Este tipo de igualdade é que será aplicada para a validação da expressão de comparação entre os endereços dos institutos e seus empregados, indicado anteriormente (*i.Localizacao = e.E_endereco*).

D) Igualdade por identidade de Propriedades: Considere duas propriedades P_i do objeto O e P_j de O' , elas são iguais por valor se: i) a estrutura de definição de tipo para P_i seja igual a de P_j , e; ii) se a propriedade P_i é monovalorada, então $O.P_i$ é igual por identidade a $O'.P_j$. Do contrário se P_i for multivalorada, então devem ser verificadas as seguintes condições:

- cardinalidade de $O.P_i$ é igual à cardinalidade de $O.P_j$;
- para cada objeto O_k pertencente a $O.P_i$, existe uma objeto O'_k pertencente a $O.P_j$, tal que O_k e O'_k sejam iguais por identidade.

Esta última definição de igualdade de predicados, difere da anterior (igualdade por valor), apenas pelo fato de que os objetos pertencentes à propriedade sejam iguais por valor ou por identidade.

Em função destas categorias de igualdade no OOPC são definidos dois tipos de operadores: i) igualdade por identidade “==”; e ii) igualdade por valor “=“.

4.3.3. Predicados sobre as Propriedades dos Objetos.

Os objetos podem apresentar propriedades que sejam multivaloradas ou definidas a partir da estrutura de outro objeto, definindo um objeto complexo. O OOPC suporta a manipulação deste tipo de propriedade através da definição de alguns predicados específicos, realizada de forma dependente do construtor de tipo adotado, neste caso, o construtor genérico, denominado coleção. Define-se os seguintes predicados para coleção de objetos:

- a) Predicado de Pertinência: $y \in x.p$, onde y e x são variáveis objetos e p é um predicado multivalorado do objeto indicado por x , sendo avaliado como verdadeiro se y for um componente da coleção de valores assumidos pela propriedade;
- b) Predicado de Inclusão: $y.p \subseteq x.p$, sendo avaliado como verdadeiro se os valores da propriedade p do objeto y estejam incluídos na coleção de valores da propriedade p' do objeto x .

Como exemplo de consulta utilizando estes predicados, considere:

Exemplo 2: Recupere todos os empregados que participem como pesquisadores do grupo denominado DB.

$\{x.e_nome; x/Empregado; \exists t/Grupo$

$(x.status="Pesquisador" \wedge x \in t.componentes \wedge t.t_nome="DB")\}$

4.3.4. Quantificadores Restritos.

A existência de propriedades multivaloradas permite formular condições que envolvam todos os objetos da propriedade ou sobre a existência de alguns. Isto implica a definição de predicados envolvendo restrições sobre os quantificadores, com os seguintes formatos:

- Quantificador universal restrito: $\forall x / y.p$;
- Quantificador existencial restrito: $\exists x / y.p$;

Através destes quantificadores os objetos indicados por x pertencem à classe de domínio da propriedade multivalorada p do objeto y , devendo satisfazer a restrição indicada na cláusula de qualificação, como demonstrado na consulta a seguir:

Exemplo 3: Recupere o nome de todos os grupos, nos quais todos os empregados apresentem um salário maior que \$20.000,00.

$$\{y.t_nome; y/Grupo; \forall x / y.Grupo (x.salario > 20.000)\}$$

4.3.5. Navegação.

Um requisito importante para qualquer linguagem de consulta OO é a sua capacidade de navegar através das estruturas dos objetos que formam a hierarquia de classes no esquema de uma aplicação. A navegação é definida através da função *ponto*, informalmente apresentada nas consultas anteriores, interpretada como: dado uma variável objeto x e uma propriedade monovalorada p , $x.p$ retorna o objeto que seja valor da propriedade p do objeto x . Como o elemento de retorno consiste de um objeto, esta construção pode ser encadeada em vários níveis, definindo uma expressão para o *caminho* de navegação.

Exemplo 4: Recupere o nome dos projetos que apresentaram relatórios entregues no dia 5 de cada mês.

$$\{t.p_nome; t/Projeto; t.relatorio.d_data.dia = 5\}$$

A navegação se torna mais complexa quando uma das propriedades existentes em um determinado nível for multivalorada. Neste caso pode ser útil manipular coleções de objetos em um nível intermediário da expressão de navegação, produzindo as *expressões do caminho quantificadas*. Seja um objeto x de uma classe C e uma propriedade p de C , cujo domínio é a classe C' , a consulta:

$$\{x.p; x/C; x.\Theta p \text{ termo}\},$$

onde, Θ é um quantificador, Θ é um operador de comparação e *termo* é um operando do predicado, apresenta a seguinte semântica:

- se p é monovalorado então $\{x.p; x/C; \exists y / C'(x.p == y \wedge y \Theta \text{ termo})\}$
- se p é multivalorado então $\{x.p; x/C; \Theta y / x.p(y \Theta \text{ termo})\}$

Considerando o esquema da **Erro! A origem da referência não foi encontrada.**, para recuperar o nome de todos os projetos, tal que todos os grupos participantes apresentam um empregado com a função de “Gerente”, pode-se formular a seguinte expressão:

$$\text{Exemplo 5: } \{x.p_nome; x/Projeto; x.\forall \text{participantes}.\exists \text{componente.status} = \text{“Gerente”}\}$$

4.3.6. Métodos.

Uma linguagem de consulta orientada a objetos deve suportar a presença de métodos em seus comandos, representando o aspecto comportamental dos elementos envolvidos. A execução de um método M em um determinado objeto O é denotado por $O.M(O_1, O_2, \dots, O_n)$, onde $O_1, O_2, \dots, O_n$ consistem nos parâmetros de entrada, podendo

ser um objeto primitivo, uma variável objeto ou uma expressão de navegação. Uma vez que o método retorna como parâmetro de saída um objeto, a sua chamada pode ser inserida em qualquer local que um objeto ou propriedade possa ser utilizado. Em uma expressão de navegação, se o método retornar um objeto primitivo, este pode aparecer apenas no final da consulta. Caso um método que retorne como saída um objeto não primitivo seja utilizado no final da expressão, o único operador permitido é a comparação de igualdade.

Como exemplo considere a definição dos seguintes métodos para a classe Projeto:

- *valor()* $\rightarrow$ *integer*: calcula o custo total do equipamento associado ao objeto;
- *CarroDisponivel (string)* $\rightarrow$ *Automóvel*: determina o carro disponível para uma viagem a um determinado destino fornecido.

Exemplo 6: Recupere todos os projetos nos quais foram gastos mais que \$50.000,00 em equipamentos para Banco de Dados.

$$\{x.p_nome; x/Projeto; x.valor > \$50.000 \wedge x.area = \text{"Banco de Dados"}\}$$

Exemplo 7: Recupere todos os projetos que utilizaram um "Ford" para as viagens até São Paulo.

$$\{x.p_nome; x/Projeto; x.CarroDisponivel(\text{"São Paulo"}).fabricante = \text{"Ford"}\}$$

Neste último exemplo a chamada ao método aparece no interior da expressão de caminho, retornando como saída um objeto da classe Automóvel, sobre o qual é aplicado a propriedade *fabricante*.

O método a ser utilizado em consultas não deve apresentar efeitos colaterais, ou seja, a sua execução não poderá modificar o estado de outros objetos presentes no BD.

Caso contrário, a avaliação de uma consulta terá o seu resultado dependente da ordem de execução dos métodos envolvidos. De uma forma geral os métodos que não apresentam efeitos colaterais realizam apenas operações de consulta.

4.3.7. Hierarquia de Classes.

O primeiro aspecto a ser considerado na presença de uma hierarquia de classes é se a consulta sobre uma classe se aplica apenas a esta ou também às suas subclasses. Assim o OOPC é estendido, permitindo a união de classes na definição da abrangência de uma consulta. Se o escopo de uma variável deve ser restrito apenas à classe especificada, este fato é indicado através do acréscimo do símbolo ‘#’ ao nome da classe. Esta extensão requer a característica de *late binding* no processamento da consulta. Seguindo esta regra a consulta

$$\{t.rg; t/Pessoa; t.idade > 20\}$$

recupera todas as pessoas pertencentes à hierarquia de classes com raiz na classe Pessoa que apresentam idade superior a 20 anos. Para restringir a abrangência da consulta apenas à classe Pessoa, sem pesquisar nas instâncias de suas subclasses, esta consulta é rescrita como:

$$\{t.rg; t/Pessoa\#; t.idade > 20\}$$

Para auxiliar na definição do tipo de classe associada a uma variável objeto tem-se o operador CLASS_OF, que reconhece a classe de um objeto em tempo de execução. Quando aplicado ao nome de uma variável objeto, este operador retorna a classe a que pertence. Uma vez que as subclasses apresentam propriedades distintas, pode-se estabelecer condições diferentes aos elementos da hierarquia, formando o *predicado alternativo* no OOPC, definido como:

$$\text{CLASS_OF}(x) = [C_1 : \text{Pred}_1 ; C_2 : \text{Pred}_2 ; \dots ; C_n : \text{Pred}_n]$$

onde, x consiste em uma variável objeto pertencente à hierarquia $C_1, C_2, \dots, C_n$, e $\text{Pred}_1, \text{Pred}_2, \dots, \text{Pred}_n$ são predicados estabelecidos sobre as propriedades das classes $C_1, C_2, \dots, C_n$, respectivamente, e o par $C_i : \text{Pred}_i$ é denominado *domínio alternativo* para a classe C_i .

A semântica desta expressão é definida como:

$$(\text{CLASS_OF}(x) = C_1 \wedge \text{Pred}_1) \vee \dots \vee (\text{CLASS_OF}(x) = C_n \wedge \text{Pred}_n)$$

Para a correta avaliação desta expressão, o predicado alternativo “*pred*” deve ser completo, ou seja, para cada classe C_i da variável de objeto x , existe em “*pred*” um domínio alternativo para a classe C_i . Além disso, a avaliação de um predicado sobre um objeto pertencente a uma classe da hierarquia na qual a propriedade não se aplica irá retornar um valor indefinido (Ω), e $\Omega \wedge \text{FALSE} = \text{FALSE}$. Como exemplo, considere a consulta a seguir, formulada sobre o esquema da Figura 2, a qual apresenta a especialização da classe Pessoa nas classes Empregado e Estudante, e esta em Graduação e Pós-Graduação.

Figura 2 - Exemplo de uma Hierarquia de Classes.

Exemplo 8:

Recupere o número de identidade de todas as pessoas que apresentam salário maior que 20.000 se forem empregados, ou que não tenham se matriculado no ano de 1994, caso

sejam estudantes.

$\{x.RG; x/Pessoa; CLASS_OF(x) = [Empregado: x.salario > 20.000;$

$Estudante: x.ano_matricula \neq 1994]\}$

A definição do operador CLASS_OF fornece uma solução para o problema de avaliação de predicados sobre objetos que possuem as propriedades indicadas. Seja a seguinte consulta $\{x.RG; x/Pessoa; x.idade < 20 \wedge x.ano_matricula = 1994\}$. Esta expressão pode ser correta se todas as pessoas com menos de 20 anos sejam estudantes, o que não é o caso. Isto significa que um erro pode ocorrer, se a propriedade “ano_matricula” for aplicada sobre um objeto da classe Pessoa que não pertença à classe Estudante. Soluções para este problema consistem em: i) tratar a situação como um erro de formulação da consulta; ii) permitir apenas a existência de predicados alternativos completos em uma hierarquia de classes, garantindo a evolução de pelo menos um

predicado, ou; iii) retornar como resultado um valor indefinido com o significado semântico de “não é aplicável”.

Um problema mais complexo ocorre quando a propriedade avaliada é multivalorada. Isto significa que cada valor da propriedade pode ser um objeto pertencente a uma das classes da hierarquia, com raiz na classe de domínio da propriedade. Assim, pode-se formular consultas estabelecendo condições sobre os valores de uma propriedade, utilizando-se dos quantificadores restritos. Como exemplo, considere a recuperação de todos os clubes nos quais pelo menos um de seus associados seja empregado com salário maior que \$100.000,00. A consulta é expressa como:

Exemplo 9: $\{x.e_nome; x/Clube;$

$$\exists y/x.Associado (CLASS_OF(y) = [Empregado; y.salario > 100.000])\}$$

Com o tratamento de hierarquia de classes, a restrição de uma única classe na cláusula de abrangência foi flexibilizada, permitindo a existência de um conjunto de classes. Consequentemente, pode-se estabelecer consultas sobre um subconjunto de classes de uma hierarquia.

Exemplo 10: Recupere todos os empregados e estudantes que sejam maiores de 20 anos.

$$\{t.RG; t(Empregado\# \wedge Estudante\#); t.idade > 20\}$$

Problemas semelhantes ocorrem na utilização de métodos $O.M(O_1, O_2, \dots, O_n)$, onde O ou um dos objetos O_i são expressões de navegação representando uma propriedade com domínio alternativo. Assim, define-se a *execução alternativa de métodos*, na qual um método é ativado para cada classe da hierarquia.

4.3.8. Operações de *Join* e Projeções Genéricas.

Com a restrição de selecionar apenas uma classe (ou sua árvore de hierarquia) na abrangência da consulta e no máximo uma propriedade a ser recuperada, a utilização de outras classes só é possível através dos elementos de ligação entre os objetos, indicados nas expressões do caminho de navegação. No entanto, não é possível formular consultas como: recupere o nome de todos os empregados que moram na mesma cidade onde se localiza o instituto de pesquisa em que trabalham. Isto ocorre pois não existe um caminho de navegação ligando as duas informações.

Para formular tal consulta torna-se necessário o operador *join*. A definição de sua semântica ocorre permitindo-se a presença de mais de uma classe e propriedade nas cláusulas de objetivo e abrangência, a qual corresponde ao produto cartesiano dos objetos descritos na abrangência da consulta. Os dois tipos de predicados de igualdade (identidade e valor) podem ser utilizados na definição do *join*, e duas variáveis objeto ou propriedades são válidas para o *join* se os seus domínios forem compatíveis, ou seja, o domínio de uma propriedade é idêntico ao da outra, ou consiste em uma subclasse ou superclasse desta. A consulta proposta inicialmente neste item é formulada como:

Exemplo 11: {*e.e_nome; e/Empregado, i/Instituto;*

$$e \in i.componentes \wedge e.e_endereco.cidade = i.localização.cidade \}$$

Outra extensão importante é permitir a projeção sobre mais de uma propriedade, como objetivo final da consulta, sendo possível a utilização de expressões de navegação para este fim.

Exemplo 12: Recupere o nome, a cidade de residência e o cargo de todos os empregados cujo salário seja maior de \$100.000,00.

$$\{e.e_nome, e.e_endereco.cidade, e.cargo; e/Empregado; e.salario > 100.000\}$$

Observe que o resultado desta consulta corresponde a objetos que pertencem a classe de Empregados. No entanto, no caso geral a cláusula de abrangência pode conter classes distintas, e o resultado da expressão pode não pertencer a uma classe em particular. Esta situação será definida com detalhes nos próximos itens.

4.3.9. Consultas Recursivas.

A definição de consultas recursivas é suportada pela linguagem em consequência da existência de ciclos na modelagem. O operador de recursividade no OOPC é representado pelo símbolo ρ . Dado um objeto O instância de uma classe C , e uma propriedade p de O , $O.\rho(p)$ define um conjunto de objetos que são ou valores da propriedade p do objeto O , ou valores da propriedade p de algum objeto O' , o qual é valor da propriedade p do objeto O , e assim por diante.

Exemplo 13: Recupere todos os empregados tal que ao menos um de seus gerentes (em qualquer nível) receba \$10.000 como salário mensal.

$$\{e.e_nome; e/Empregado; \exists y/Empregado (y \in x.\rho(Gerente) \wedge y.salario > 10.000)\}$$

O operador de recursividade pode ser utilizado para especificar um conjunto de domínio para uma variável de objeto. A consulta anterior é rescrita como:

Exemplo: $\{e.e_nome; e/Empregado; \exists y/x.\rho(Gerente) \wedge (y.salario > 10.000)\}$

4.3.10. Análise dos Resultados das Consultas.

Conforme mencionado anteriormente, uma consulta que envolva mais de uma classe na definição de sua abrangência e várias propriedades como objetivo final pode apresentar como resultado um objeto que não pertença a nenhuma das classes existentes

no domínio da aplicação. Uma solução genérica para este problema é considerar o resultado de uma consulta como uma nova classe, com as seguintes propriedades:

- a) um novo identificador de objetos é criado para cada objeto pertencente ao resultado;
- b) a estrutura de tipo da nova classe é determinada pelas seguintes propriedades:
 - b.1) as propriedades da classe são as que aparecem listadas no objetivo da consulta;
 - b.2) a nova classe é criada como subclasse da classe principal (“TOP_CLASS”) do esquema da aplicação.

Com esta definição a linguagem se torna consistente no tratamento fornecido às classes e aos seus objetos. Além disso, o resultado de uma consulta pode ser claramente utilizada na formulação de outras, como se fosse uma nova classe existente no esquema da aplicação. No entanto, estas considerações provocam uma queda de desempenho do sistema, uma vez que, para as consultas nas quais os objetos resultantes pertencem às classes definidas no esquema não é necessário criar uma nova. Neste caso, estas regras podem ser flexibilizadas de forma a suportar a criação de novas classes apenas quando não for possível classificar o resultado da consulta em uma classe já definida.

4.3.11. Comparação com a Linguagem Relacional.

O próprio modelo de dados Orientado a Objetos apresenta construtores semânticos mais poderosos do que os disponíveis no modelo Relacional. Neste item serão apresentados alguns elementos que possibilitam a comparação das principais características das linguagens de consultas [Bertino_92]. Os tópicos de 1 a 6 se referem às diferenças que ocorrem em decorrência de características do modelo, e de 7 a 9, diferenças inerentes à própria linguagem.

- 1) Identificadores: o modelo relacional não suporta o conceito de identificação única dos elementos do mundo real na forma de tupla. Esta noção é implementada através de atributos, de forma contrária ao que ocorre no Modelo Orientado a Objetos (OO);
- 2) Atributos Multivalorados: nos modelos OO são utilizados para a construção de tipos complexos, e no modelo relacional só são obtidos através da criação de uma nova relação, o que implica a utilização de consultas contendo o operador de *join* para selecionar um atributo;
- 3) Relacionamentos: no modelo OO a definição do tipo básico de um objeto permite a inclusão de outro, representando a presença de um relacionamento entre eles. A linguagem de consulta OO suporta este conceito através de uma navegação pelos elementos do esquema. No modelo relacional o relacionamento é representado com a criação de uma nova relação ou com a incorporação de um atributo chave estrangeira. A recuperação da informação é realizada na consulta com o operador de *join*;
- 4) Métodos: não são suportados pelos modelos relacionais, e consistem em um importante recurso para se definir novas propriedades, derivação de dados e aumento da capacidade semântica da linguagem;
- 5) Dados Derivados: nos modelos OO são obtidos através da definição de métodos associados às suas classes, e no relacional tal conceito é implementado com o mecanismo de visões. No entanto, visões são consultas, o que limita a sua flexibilidade em relação aos métodos, mas, em ambos os modelos, na linguagem estes elementos são tratados como atributos a serem manipulados;

- 6) Hierarquia de Classes: esta noção não é suportada no modelo relacional. O escopo de abrangência de uma consulta está limitado ao conjunto de relações listadas na cláusula FROM, contrário ao modelo OO que pode se estender para toda a hierarquia;
- 7) Predicados de Join: as linguagens de consultas orientadas a objetos realizam o *join* relacional através da igualdade de valor e de identidade sobre classes primitivas e não primitivas, sendo que este último tipo não é possível em consulta relacional;
- 8) Recursividade: a linguagem de consulta relacional não suporta a avaliação de uma consulta com recursividade em qualquer nível;
- 9) Extensibilidade da Linguagem: os operadores básicos da linguagem relacional não podem ser expandidos, enquanto que na linguagem Orientada a Objetos pode-se utilizar os métodos para implementar novos operadores.

4.5. CONSIDERAÇÕES FINAIS.

Neste capítulo foram apresentados os principais conceitos que caracterizam um banco de dados orientado a objetos. A proposta de um cálculo de predicados orientado a objetos realizada por Bertino foi apresentada de forma completa, por ser de importância fundamental na comprovação de alguns resultados decorrentes da definição de uma álgebra para a manipulação de imagens, conforme será demonstrado no capítulo 6. No próximo capítulo, será apresentado o modelo de dados orientado a objetos (SIRIUS) utilizado na incorporação do tipo de dado imagem, como um novo elemento de modelagem. As abordagens e conceitos apresentados neste capítulo serão rediscutidos sob a perspectiva do modelo SIRIUS.

SIRIUS - MODELO DE BANCO DE DADOS ORIENTADO A OBJETOS

5.1. CONSIDERAÇÕES INICIAIS.

No capítulo anterior foram apresentados os conceitos básicos referentes a um Banco de Dados Orientado a Objetos e os aspectos que tornam estes modelos os mais adequados ao desenvolvimento de Sistemas de Informação com Suporte a Imagens.

Neste capítulo é descrito o modelo de dados orientado a objetos denominado SIRIUS [Biajiz_96b], desenvolvido para suportar construtores com maior capacidade semântica, capazes de promover uma ampla diversidade de recursos na produção da modelagem de uma aplicação. Este modelo foi elaborado como uma evolução do Modelo de Representação de Objetos (MRO) [Traina_86] [Traina_92] e de seu SGBD Orientado a Objetos, denominado Gerenciador de Objetos (GEO) [Traina_91a], através de um processo de formalização de sua conceituação e dos construtores que se encontravam disponíveis [Biajiz_92]. Estes são definidos utilizando-se as três abstrações consideradas fundamentais [Biajiz_96a] [Traina_94]: *abstração de classificação*, *abstração de associação* e *abstração de generalização*. A abstração de associação é definida através de duas especializações, denominadas de *abstração de composição* e de *agregação*.

5.2. CONCEITUAÇÃO GERAL DO MODELO SIRIUS.

Nesta seção são apresentados os conceitos de objetos, tipos de objetos, atributos, características de atributos e tipos de atributos. As estruturas dos dados manipulados podem ser sofisticadas, pois o modelo SIRIUS procura representar com o título de objetos apenas os elementos do mundo real que podem ser identificados pelo usuário. Assim, objeto em SIRIUS é algo significativo para um usuário do sistema, e não um conceito abstrato. Com isso, os objetos tendem a ser estruturas mais sofisticadas, que incluem um conjunto variado de construções a ele associado. É importante notar que a complexidade que o objeto pode assumir em uma modelagem torna adequada a existência de múltiplas maneiras de representá-lo, cada qual destacando um aspecto particular de como este possa ser interpretado no esquema da aplicação.

5.2.1. Objetos e Tipos de Objetos.

Dentro da terminologia usual da área de orientação a objetos, o termo objeto é genérico, e assim em SIRIUS esse termo é utilizado em duas situações. Primeiro, quando o próprio sistema manipula os objetos, estes apresentam uma característica mais voltada à implementação; nesse caso o termo utilizado é *objeto primário*. Segundo, quando o usuário reconhece um elemento do mundo real, e, portanto, representa os conceitos ou entidades presentes que possam ser identificados pelo usuário. Neste caso o termo é utilizado sem um qualificador específico.

Todos os objetos, primários ou não, possuem um Identificador Interno (OID), controlado pelo sistema, unívoco durante toda a sua existência no BD. Já um objeto (não primário) pode ser identificado também pelo usuário através de um “nome”, que o caracteriza de forma única em relação aos demais. Além do nome, um

objeto tem um tipo e atributos a ele associados. Deve-se observar

Figura 1 - Representação de Objetos.

que todo objeto tem exatamente

um tipo, o qual, por sua vez, como qualquer conceito do SIRIUS, é também um objeto. A Figura 1 mostra a representação gráfica dos objetos no SIRIUS. Em a) é indicada a notação de um objeto, em que na parte superior esquerda do seu retângulo de representação, se encontra o nome do tipo desse objeto e, na parte inferior, o nome do objeto. A parte superior direita é utilizada para dados da abstração de composição, apresentada no item a seguir. O tipo de um objeto pode ser omitido, quando seu tipo for o próprio meta-tipo *tipo-de-objeto*, que é reconhecido e controlado pelo sistema. Em b) representa-se um objeto denominado BD-Fundamentos, cujo tipo é Livro, e um objeto denominado Livro de tipo *tipo-de-objeto*.

5.2.2. Atributos, Tipos de Atributos e Características de Tipos de Atributos.

Atributos são dados do mundo real que estabelecem informações sobre os objetos. Eles são constituídos por um tipo e um valor, e podem ser monovalorados ou multivalorados. Os atributos são representados graficamente por setas, posicionadas conforme indicado na Figura 2. Uma seta simples indica um atributo monovalorado,

e dupla, multivalorado.

Na Figura 2.a estão indicados estes tipos de atributos.

Atributos instanciados com valores são representados como em

b). Em c) há um

exemplo em que o

Figura 2 - Exemplo de Representação de Atributos.

objeto BD-Fundamentos, do tipo Livro, que possui o tipo de atributo não instanciado EmprestadoPara, representando o fato de que este livro não se encontra emprestado, os atributos monovalorados ISBN e Edição instanciados com os valores 13115 e 1o., respectivamente, e o atributo multivalorado Autores, com a lista de valores formada por Navathe e Elmasri.

Usualmente, em modelos de dados, os objetos possuem uma parte estática (normalmente chamada de atributos do objeto) e uma parte dinâmica (métodos do objeto). Em SIRIUS, o termo atributo designa qualquer elemento que se agrega a um objeto. Objetos, por sua vez, possuem não duas, mas quatro partes: estática,

dinâmica, estrutural e de interface. Todos os conceitos do SIRIUS que se referem a atributos aplicam-se igualmente a qualquer uma dessas partes.

A cada tipo de atributo estão associados uma característica e um tipo de dado. Modelos de dados em geral permitem associar atributos a objetos, definindo-lhes um tipo de dado e um domínio. O modelo SIRIUS expande e ao mesmo tempo restringe esses conceitos. Restringe ao modelar como atributos tudo o que se encontra associado a objetos. Expande ao admitir que o suporte de operações oferecidas pelo modelo para atributos especializa-se em diversas formas, de acordo com a “característica” do atributo. Todo atributo possui um tipo e um conjunto não vazio de valores. A cada tipo de atributo estão associados uma característica e o tipo de dado que ele contém. A característica do atributo de um determinado tipo define o conjunto de operações que o Gerenciador de Dados pode efetuar sobre esse atributo, ou seja, estabelecem o comportamento junto ao objeto ao qual esse atributo está associado. Em SIRIUS, as características são divididas em: **estáticas**, abrangendo cadeia, comentário e número; **dinâmicas**, abrangendo regras e procedimento; de **interface**, abrangendo visualização, som, partitura, imagem e gráfico; e **estruturais**, consistindo em estruturas de dados, tempo, tupla e relacionamento.

O significado de cada característica consiste em:

- **Número:** são atributos usados para armazenar números, podendo estes serem inteiros ou de ponto flutuante;
- **Cadeia:** são textos curtos, usualmente representando domínios finitos de pequena cardinalidade;

- **Comentário:** corresponde a campos longos, ou seja, informações para o usuário tratadas apenas de forma textual, que o SGBD permite manipular como linhas, parágrafos, etc;
- **Regra:** são atributos que possuem uma condição e uma ação. A condição permite que a ação seja executada sempre que for feita alguma operação do SGBD sobre o objeto (gatilho), tal como consulta ou atualização, ou quando for recebida alguma mensagem pelo objeto, originária da execução de outra ação ou procedimento no banco de dados;
- **Procedimentos:** são atividades que devem ser exercidas de uma forma pré-determinada pelo sistema, solicitadas pelo aplicativo para execução imediata ou postergada;
- **Tempo:** indicação de tempo, absoluto ou relativo, atribuído aos objetos e atributos. Permite que se registre o tempo absoluto de operações no banco, ou o tempo do mundo real (ou de validade) definido externamente ao BD;
- **Estruturas de Dados:** são agrupamentos de informações com um significado completo que normalmente são representados em estruturas de arranjos de dados (vetores e matrizes), ou para as quais possam ser definidas operações próprias de conjuntos, tais como união e interseção, ou ordenação e tratamento de repetição de valores (listas, filas, pilhas e montes).
- **Tupla:** indica que o atributo em questão agrega atributos diferentes em uma estrutura de tupla;
- **Relacionamento:** indica que o atributo em questão associa um objeto a outros objetos;

- **Gráfico:** corresponde à informação vetorial de localização no espaço, podendo indicar um ponto, uma função, um conjunto de segmentos de retas, uma operação de transformação, ou qualquer combinação destes;
- **Som:** corresponde a sons digitalizados armazenados no banco de dados;
- **Partitura:** informação musical, que pode ser reproduzida por síntese a partir dos dados armazenados;
- **Visualização:** são atributos que podem definir como um elemento de um modelo pode ser visualizado em uma interface de entrada e saída do sistema, para permitir a interação de uma pessoa com um aplicativo que manipule o objeto ou o relacionamento ao qual o atributo está agregado.
- **Imagem:** permite a incorporação de informações na forma de imagens digitais, e consiste no tema central deste trabalho.

Qualquer atributo estático pode ser usado para identificar um objeto, embora usualmente números e cadeias prestem-se excepcionalmente bem para isso. Identificar um objeto significa que não podem existir dois objetos no mesmo contexto com o mesmo valor para esse atributo. Numa dada modelagem, alguns tipos de atributos são identificadores de objetos, ou seja, têm a propriedade adicional de identificar um objeto de maneira significativa para o usuário. Assim, durante uma instanciação, esse tipo torna-se um identificador do objeto instanciado. Havendo identificadores associados, deve-se indicar um deles para ser o principal dos objetos desse tipo. Esse aspecto também é uma característica de atributo, porém, no lugar de utilizar um símbolo distinto para essa característica, sua identificação é efetuada integrada a da abstração de classificação.

SIRIUS, além de apresentar aspectos estáticos e dinâmicos como de qualquer outro modelo para os seus tipos de objetos, estes possuem uma *aparência* (como são vistos externamente, tanto pelos demais componentes do sistema além do Gerenciador de Banco de Dados, quanto por usuários finais), e uma *estrutura* (como seus demais componentes organizam-se, temporal, espacialmente, etc). Isso significa que a descrição do modelo, quando se refere a atributos de um objeto, considera os aspectos estáticos, dinâmicos, estruturais e de interface.

5.2.3. Abstração de Agregação.

Figura 3 - Notação utilizada na agregação de atributos.

Os atributos agregam-se para caracterizar um objeto do mundo real, ou para constituir outros atributos. Os atributos também estabelecem

relacionamentos entre objetos, ou seja, dois ou mais objetos são agregados através de um objeto primário, formando relacionamentos.

A agregação de atributos pode ocorrer através da associação de atributos em tuplas, vetores, listas ou conjuntos sem repetição. Estes elementos consistem de estruturas de dados que organizam a forma como os atributos são agregados. A partir

dessas estruturas pode-se modelar diretamente um atributo multivalorado, estruturando-o conforme necessário. A notação utilizada na agregação de atributos em suas diversas variações está ilustrada na Figura 3. No quadro d) e h) da Figura 3 é representado o conceito de atributo de atributos, representando a caracterização de um aspecto específico do objeto. No caso, o atributo `NroPassaporte` tem por sua vez como atributos a `Data-de-Emissão` e o `Orgão-Expeditor`. Apesar do conceito atributo de atributo sobrecarregar o modelo, ele apresenta aplicações imediatas em modelagem de elementos não convencionais, como exemplo, as informações relacionadas a imagens, conforme será descrito no capítulo seguinte.

A associação entre objetos pode acontecer de duas maneiras. A primeira (mais dependente do tratamento dado a ela pelos aplicativos) consiste em referenciar, a partir de um dos objetos envolvidos, os demais objetos. Essa referência pode ou não apresentar uma equivalente (oposta apenas no caso de serem dois objetos envolvidos) nos demais objetos. Em ambos os casos, a consistência dos dados que implementam essa maneira de representar a associação é de responsabilidade dos aplicativos, com pouco suporte podendo ser oferecido pelo gerenciador. Essa maneira de indicar a associação entre os objetos, não acrescenta nenhum conceito novo ao modelo, mas caracteriza apenas o fato de que o valor de um atributo pode ser outro objeto.

A segunda forma define o conceito de relacionamento considera a associação entre objetos como um objeto primário (o objeto relacionamento), tratado internamente pelo próprio gerenciador. Nesse caso, os objetos envolvidos têm: como atributo uma referência para o objeto relacionamento; o tipo do relacionamento caracterizado pelo tipo de atributo, nesse caso em particular denominado de

“modalidade”; e a característica desse tipo de atributo é denominada de característica “relacionamento”. Essa maneira de indicar a associação entre objetos traz ao modelo a capacidade de suportar relacionamentos, de uma maneira semelhante mas não igual ao conceito suportado nos modelos semânticos.

Um atributo com característica de relacionamento também pode ser um atributo simples, uma tupla, ou um conjunto. O modelo não define restrições para essa ligação, ou seja, um objeto pode estar agregado a vários outros objetos através de diversos relacionamentos. Além disso, um relacionamento pode ter atributos

associados, os quais são denominados atributos de relacionamentos.

A notação utilizada na agregação de objetos através de relacionamentos está ilustrada na Figura 4.

Os relacionamentos são caracterizados por atributos com características de

Figura 4 - Notação utilizada na agregação de objetos.

relacionamento nos objetos envolvidos. Em a) representa-se um relacionamento entre dois objetos, com a forma de indicar de seus atributos, presente em b). Na Figura 4.c está representado um objeto como valor de um atributo.

Figura 5 - Exemplo de relacionamento Triplo.

Os relacionamentos ternários, quaternários etc, são também representados. Em SIRIUS define-se um relacionamento particular denominado *relacionamento triplo* (Figura 4.d), em que a denominação “triplo” não

provém do número de objetos participantes, mas sim por identificar três maneiras de um objeto poder participar do relacionamento [Biajiz_92]. Um relacionamento triplo envolve mais de dois objetos, no qual dois destes possuem a propriedade de identificar o relacionamento através de um par de atributos com característica de relacionamento denominado *relacionamento identificador*. Os demais objetos se relacionam formando os *relacionamentos de conjunto*. Um exemplo pode ser visto na Figura 5, onde o objeto Projeto relaciona-se com os objetos Empregado, Máquina, Ferramenta e Bancada. Apesar de cinco tipos de objetos participarem do relacionamento, esse não é um relacionamento “quinário”. O objeto Projeto relaciona-se com o objeto Empregado constituindo o relacionamento identificador, o objeto Projeto associa-se aos objetos Máquina, Ferramenta e Bancada estabelecendo o relacionamento de conjunto. Semanticamente, um objeto Projeto é desenvolvido por um Empregado usando especificamente os objetos Máquina, Ferramenta e Bancada. Projeto ser desenvolvido por Empregado é identificador, pois as Máquinas, Ferramentas e Bancadas usadas são

identificadas a partir desse relacionamento. Por outro lado, o Projeto em si é que usa tais Máquinas, Ferramentas e Bancadas. Além disso, considerando que se outro Empregado participasse do desenvolvimento desse Projeto, as Máquinas, Ferramentas e Bancadas utilizadas poderiam ser distintas. Pode-se definir que a existência dos relacionamentos de conjunto e do relacionamento de identificação geram semanticamente três divisões D1, D2 e D3. Com essa classificação em divisões, o objeto Projeto pertence à divisão D1, os objetos Máquinas, Ferramentas e Bancadas pertencem à divisão D2, e o objeto Empregado pertence à divisão D3.

5.2.4. Abstração de Classificação.

A abstração de classificação é obrigatória em qualquer modelo de dados, porém o SIRIUS a trata de uma forma mais abrangente, cobrindo aspectos que não são tratados por outros modelos [Traina_94]. Usualmente, modelos de dados admitem a ocorrência de apenas um nível de classificação, que corresponde à definição dos tipos e às instanciações de objetos. Alguns gerenciadores suportam ainda que a definição dos próprios tipos sejam objetos modelados por meta-tipos, e, nesse caso, há o aumento de um nível na classificação. Porém, isso não constitui um novo construtor para o modelo, e não estende sua capacidade de representação. O SIRIUS, ao contrário, incorpora a Abstração de Classificação como um construtor semântico, permitindo a construção de uma *hierarquia de classificação*, dentro de uma modelagem, tal como qualquer outro construtor.

A abstração de classificação é representada pela indicação do nome do tipo do objeto e pelo próprio nome do objeto. A maneira como um objeto é instanciado é

estabelecida através de um subconjunto dos atributos deste objeto que é tipo, os quais são denominados “Atributos de Classificação”, definindo o “padrão” de como modelar cada instância desse tipo. Em um diagrama, os atributos de classificação são representados acima do objeto, enquanto que os atributos do próprio objeto, chamados de “Atributos de Instância”, são representados abaixo deste.

O fato de um atributo ser de classificação indica que ele não é instanciado no objeto tipo ao qual está ligado, mas será atribuído às instâncias desse objeto. Instanciar um objeto significa criar um objeto instância do objeto tipo. Na instanciação do objeto de um determinado tipo, os atributos do tipo são utilizados para controlar esse processo, de forma que a instância seja criada adequadamente. Os

Figura 6 - Abstração de classificação e seus atributos de Classificação.

quantos níveis, no mínimo, a hierarquia terá, assim como quais atributos instanciam-se nos diversos níveis. Isso é indicado com um número precedendo o nome do atributo. O número 1 significa que esse atributo será instanciado no primeiro nível, o número 2 significa que esse atributo será instanciado na primeira instanciação de um

atributos já instanciados não tomam parte desse processo.

Uma propriedade dos atributos de classificação é a indicação de

objeto instância desse tipo, e assim por diante. A Figura 6 ilustra esses conceitos, descrevendo o fato de que os títulos de livros são classificados em livros, e volumes são classificados pelos seus tipos, ou, de forma contrária, livros são instanciados em títulos, e estes em Volumes (Exemplares).

Além destes aspectos, qualquer objeto pode ter atributos “extras”, não previstos em seu tipo, que sejam de instância ou de classificação. Isso é fundamental, pois pode-se inserir novos atributos em qualquer nível da hierarquia.

5.2.5. Abstração de Generalização.

A abstração de generalização é caracterizada pelo fato de um tipo de objeto poder acrescentar mais detalhes (especializar) a um outro tipo já existente. Note que conceitualmente não existe restrição para que qualquer objeto possa ser especializado ou generalizado. Porém, quando o objeto não é um tipo, especializá-lo ou generalizá-lo recai no caso trivial de inserção ou eliminação de atributos, e portanto, não é necessário que faça parte do modelo em si.

O SIRIUS impõe, para uma generalização, que sejam definidos um tipo que será especializado, um conjunto de tipos especializados (sub-tipos), um critério de especialização, restrições para especialização e um conjunto de regras de inicialização para cada tipo especializado. O critério de generalização/especialização dá-se de duas maneiras: pelo valor de um único atributo cujo domínio seja discreto e pré-definido e pelo resultado da avaliação de um predicado sobre os valores de um ou mais atributos. Existem ainda duas restrições associadas a uma abstração de generalização:

- a) **Sobreposição:** indica se um objeto do tipo genérico pode ser de apenas um (disjunção) ou mais de um tipo específico (sobreposição);
- b) **Participação:** indica se todo objeto do tipo genérico deve necessariamente ser também de um tipo específico (participação total) ou não (participação parcial);
- c) **Categorização:** indica se todo objeto de um tipo específico deve necessariamente ser também do tipo genérico (categoria total ou subtipo) ou não (categoria parcial).

Um exemplo de cada situação é apresentado na Figura 7.

5.2.6. Abstração de Composição.

A Abstração de Composição, tal como a Abstração de Agregação, é uma especialização da abstração de associação, porém representa um significado diferente, mais específico: cada instância da Abstração de Composição determina a existência de um objeto composto como o objeto abstrato e um conjunto de objetos

Figura 7 - Exemplos de ocorrências de abstração de Generalização.

que são “*parte de*” esse objeto, como seu detalhe. Ao contrário da agregação - onde os objetos detalhe em geral são primários e representam atributos, sendo portanto não reconhecíveis pelos usuários como objetos - os objetos que são *parte de*

Figura 8 - Representação de Composição (colônia de objetos).

um objeto composto são objetos de fato, definidos e reconhecidos pelo usuário.

A Figura 8 mostra o relacionamento *é parte de* que existe entre um objeto composto e suas partes. É comum que esses objetos formem um conjunto que têm propriedades distintas, úteis para o gerenciamento do banco. Assim, o SIRIUS define o conceito de Colônias como o conjunto de objetos que compõem um objeto composto segundo um determinado aspecto. A idéia de aspecto representa o fato de um objeto poder ser visto como composto de diferentes maneiras, disjuntas entre si. Por exemplo, uma placa de circuito impresso pode ser composta por uma coleção de

componentes ou pelos elementos do circuito que constitui o diagrama que a representa. A Figura 8.b mostra os relacionamentos implícitos que ocorrem entre os objetos envolvidos em uma ocorrência de uma Abstração de Composição: um objeto composto restringe uma colônia, onde *habitam* os objetos que são parte do objeto composto. Colônias determinam objetos primários que, como qualquer outro, possuem tipo. Assim, toda colônia tem um tipo.

Uma colônia pode representar uma composição lógica ou física de objetos. Impõe-se a restrição de que todo objeto deve habitar fisicamente exatamente uma colônia, e esse vínculo estabelece uma relação de dependência existencial entre o objeto e a colônia em que fisicamente ele habita: se a colônia deixar de existir, todos os objetos que a habitam deixam de existir também. Por outro lado, um objeto pode habitar logicamente qualquer quantidade de colônias, sem causar nenhum tipo de dependência existencial.

Existe uma colônia física de tipo denominado “Global”, da qual pode haver apenas uma instância. A partir dela é definida a hierarquia de composição, a qual estabelece o contexto em que os objetos estão sendo compostos.

5.3. SIRIUS COMO MODELO CONCEITUAL PARA REPRESENTAÇÃO DE IMAGENS.

O modelo SIRIUS tal como apresentado neste capítulo, foi resultado da evolução de outro modelo de dados Orientado a Objeto, denominado MRO - Modelo de Representação de Objetos [Traina_86] [Traina_92]. O desenvolvimento de vários aspectos conceituais do modelo SIRIUS sofreu influências dos requisitos apresentados para a representação de imagens. Dentre estes, pode-se ressaltar, a conceito de atributos de atributos e o mecanismo de instanciação de objetos,

juntamente com a flexibilidade da inserção de atributos específicos em qualquer nível do processo de instanciação. Assim, a definição do conjunto de construtores de SIRIUS, se completou a partir da conclusão do trabalho atual, e de outros que colaboram neste sentido.

Procedimento semelhante será aplicado para o desenvolvimento do Sistema Gerenciador de Banco de Dados. O MRO possui o seu gerenciador próprio, denominado de Gerenciador de Objetos (GEO) [Traina_91a]. Para SIRIUS, a construção de seu gerenciador se dará, através de uma evolução do GEO, com as adaptações necessárias para fornecer suporte aos construtores semânticos contidos no novo modelo.

5.4. CONSIDERAÇÕES FINAIS.

Neste capítulo foi descrito o modelo Orientado a Objetos denominado SIRIUS. Consiste de um modelo com grande capacidade de representação da semântica envolvida nas informações presentes na modelagem de uma determinada aplicação, sendo fundamentado no conceito de abstrações. O conjunto de construtores disponível torna este modelo adequado para a modelagem de sistemas aplicativos não convencionais, como, por exemplo, o suporte à definição de imagens em um Sistema de Informação. Assim, conceitos, como o meta-esquema de representação de dados, abstrações de instanciação de objetos e especialização, características de tipos de atributos, atributos de atributos serão fundamentais na conceituação do tipo de atributo imagem. No capítulo seguinte serão apresentados estes detalhes conceituais que referem ao tratamento de imagens e da informação presente em seu conteúdo.

IMAGEM: CONCEITOS E MODELAGEM EM UM BANCO DE DADOS ORIENTADO A OBJETOS

6.1. CONSIDERAÇÕES INICIAIS.

Neste capítulo descreve-se os aspectos conceituais envolvidos na incorporação do tipo de dado imagem como um novo elemento a ser utilizado na Modelagem de alguma aplicação, na qual este se faça presente. O objetivo principal deste capítulo é a conceituação de imagens como uma característica de atributo, e a identificação dos elementos necessários à sua representação, de forma a proporcionar um tratamento uniforme aos demais conceitos disponíveis no modelo. Além disso, será apresentada a metodologia para a incorporação de imagens na modelagem de uma aplicação. Complementando a conceituação, será descrita a linguagem de definição e manipulação de imagens, assim como a formalização de seus operadores através de uma linguagem formal.

6.2. REPRESENTAÇÃO DE IMAGENS NO SIRIUS.

Neste item será apresentado um conjunto de conceitos que possibilite a definição do esquema de uma aplicação que envolva a manipulação de imagens, utilizando-se do Modelo SIRIUS.

6.2.1. Constantes Imagens.

Nas linguagens de programação as constantes representam informações com um determinado valor fixo e com uma semântica bem definida, que não pode sofrer alterações além de sua criação e eliminação. Pode-se definir constantes para representar valores numéricos, tais como 5 (cinco), 3.141592 (representação do número PI), ou que simbolizam sequências de caracteres, tais como “Banco de Dados”. Estas constantes numéricas ou alfanuméricas são definidas com o objetivo de serem padrões para futuras operações de comparação ou manipulação envolvendo os valores representados.

Um Banco de Dados (BD) para armazenar imagens e permitir a realização de operações de recuperação ou manipulação, deve apresentar algum mecanismo através do qual o usuário possa expressá-las, diretamente como tal, sem considerar a utilização de outra informação que não seja as imagens. Dentre formas alternativas, mencionadas nos capítulos anteriores, se incluem a associação de itens descritivos na forma de texto para representar o conteúdo das imagens ou elementos simbólicos (ícones).

Uma **constante imagem** consiste de alguma imagem, indicada como um padrão para futuras operações a serem formuladas sobre outras imagens, que definem o conteúdo de um Banco de Imagens. As constantes imagens representam as que são

consideradas significativas na descrição do conjunto de imagens a serem manipuladas ou presentes em um domínio de aplicação. Uma diferença entre uma constante imagem e uma constante numérica em uma linguagem de consulta é a facilidade com que se pode expressar por exemplo uma constante numérica, o que não ocorre para se expressar uma constante imagem, obrigando-se à existência de recursos extras para expressá-las. No entanto, o seu significado e seu papel sintático na linguagem devem ser iguais. Por exemplo, o médico pode obter uma tomografia na qual apareça a imagem do coração significativa para a descrição de alguma determinada patologia, e que possa ser utilizada como parâmetro para futuras buscas por exames que apresentem uma imagem do coração com características semelhantes à fornecida pelo exame inicial. Portanto esta imagem pode ser definida no sistema como um padrão de comparação com as imagens posteriores a serem inseridas no BD, que contenha exames de tomografia computadorizada da região do tórax. Para uma mesma categoria de imagens, o conjunto de constantes selecionado pode não ser adequado. No exemplo anterior, se as imagens fossem referentes a exames de tomografia da região craniana, uma tomografia do coração não é significativa para descrever aquelas que se espera obter, sendo necessária a indicação de outras para caracterizar este domínio da aplicação.

A mesma consideração é válida para domínios de aplicação completamente distintos tais como gerenciamento de Marcas e Patentes, Controle de Obras em Museus, reconhecimento de suspeitos por fotos em uma aplicação policial, Controle de Projetos de Engenharia e outros. Para cada domínio o conceito de constante

imagem representa as imagens de determinação de seu escopo de abrangência, através da identificação de suas características mais marcantes.

Com o conceito de constante imagem definem-se duas categorias de imagens para o BD, a saber: (a) as imagens pertencentes ao domínio da aplicação representando o Banco de Imagens (BI), e (b) as constantes imagens, consistindo em um subconjunto do BI, as quais podem ser indicadas como elementos básicos de comparações com as demais. As constantes imagens formam um conjunto a ser utilizado para a recuperação de imagens que sejam semelhantes em algum aspecto ao conjunto de informações representada no interior da imagem que define a constante. Este conceito consiste em um avanço sobre a utilização de ícones ou de elementos descritivos na forma de atributos do tipo texto na especificação do conteúdo da imagem.

A utilização de ícones sugere que estes sejam definidos com a segurança de se poder representar, sob um aspecto bem abrangente, o conteúdo de um conjunto amplo de imagens que possam ser inseridas no BD. Isto não ocorre, pois uma imagem passa a ser indicada através de um elemento simbólico [Lee_89] [Chang_91] [Cardenas_93], que não representa toda a informação nela presente. Tem-se, portanto, um processo de “compactação” com perdas elevadas da informação originalmente fornecida ao BD para armazenamento. Além disso, a flexibilidade em se representar diferentes imagens com padrões diversos e com um grau variado de complexidade de seu conteúdo não é atingido de forma satisfatória. Imagens com formas que não sejam claramente definidas não são facilmente simbolizadas através de ícones em consequência da grande perda de informação originada deste processo.

Em domínios de aplicação nos quais as imagens apresentam objetos sólidos, com a clara definição de volume, há um grau maior de complexidade a ser tratado para a especificação do posicionamento relativo dos objetos na imagem [Costagliola_92].

A presença de atributos com informações textuais provoca uma perda maior de informação, pois uma imagem pode ser descrita de diferentes formas, dependendo da abordagem, objetivo e metodologia utilizadas. Esta informação é normalmente obtida através de um especialista no domínio da aplicação, responsável pelo reconhecimento e descrição dos elementos presentes no interior da imagem. Além disso, a riqueza de detalhes obtidos depende do nível de conhecimento apresentado pelo especialista e na disponibilidade de ferramentas adequadas para o processamento das imagens. Finalizado o processo descritivo do conteúdo da imagem, este texto irá constituir no único meio de acesso às informações contidas em seu interior, ou seja, o usuário apresenta uma descrição textual da informação a ser recuperada ao gerenciador. Para que a comparação desta descrição com as contidas no BD seja efetiva, ambas devem ser realizadas utilizando-se de um mesmo vocabulário básico seguindo regras sintáticas e semânticas bem definidas. Com esta padronização, cada usuário passa a descrever as imagens de forma semelhante, o que permite a implementação de operações de comparação, que podem envolver, inclusive, a utilização de técnicas de processamento de linguagem natural. No entanto, a associação de elementos textuais para representar o conteúdo de uma imagem apresenta problemas maiores de perda de informação, ou seja, por mais completa que seja a descrição não consegue revelar o seu conteúdo com riqueza e precisão de detalhes.

Com as considerações anteriores pode-se notar que um mecanismo mais adequado de indicar uma imagem a ser consultada é apresentar uma semelhante, na qual esteja presente a informação procurada. Ou seja, para localizar um determinado elemento em um BD de fichas policiais, apresente a foto de uma pessoa e retorne todas as que sejam semelhantes, ao contrário de se indicar um “retrato falado” (ícone representativo) ou uma descrição textual das características da pessoa procurada. A foto, uma constante imagem, apresenta a informação de uma maneira mais clara e precisa. Ou ainda, para uma aplicação em um BD com exames médicos computadorizados, apresente um no qual uma patologia semelhante tenha sido identificada, e será retornado todos os casos médicos nos quais enfermidades semelhantes tenham sido localizadas.

As constantes imagens podem representar uma imagem como um todo ou um objeto ou cena claramente identificável que possa estar presente no interior de outra, definindo uma sub-imagem. Desta forma qualquer imagem, independente de sua complexidade, pode ser indicada como constante imagem, uma vez que seja válida e significativa na caracterização do domínio da aplicação.

6.2.2. Sumarizadores.

Qualquer informação inserida em um BD é realizada com o objetivo de ser recuperada posteriormente, seja no interior de outra operação ou através de uma consulta para a sua busca. Com as imagens ocorre o mesmo, ou seja, estas devem ser manipuladas e tratadas de forma semelhante aos demais tipos de dados presentes no BD, obedecendo aos aspectos que lhes sejam particulares.

Para afirmar que uma determinada imagem contém alguma das constantes indicadas como elementos de comparação, torna-se necessário definir uma série de aspectos, relacionados especificamente com as características gráficas de imagens, tais como forma, textura, histograma etc., que serão utilizados para extrair os parâmetros básicos necessários à realização de operações de comparação. Por exemplo, uma imagem contém a constante coração, se nesta imagem estiver presente algum elemento com forma, textura, etc semelhantes aos apresentados pelo coração. Portanto, estes elementos consistem nos **sumarizadores**, que representam operações possíveis de serem aplicadas sobre o domínio das imagens no BD. Os sumarizadores representam características que permitem a comparação entre duas imagens, ou seja, pode-se, por exemplo, comparar o conteúdo de uma imagem com outra, apresentando como base a forma dos objetos que estas contem. Neste caso, a característica da imagem utilizada para a sua descrição consiste na forma ou padrão dos objetos.

Como cada sumarizador representa uma manipulação sobre o dado imagem, este pode retornar diferentes valores decorrentes do processamento realizado. Por exemplo, considere um sumarizador que represente a busca de padrão, na qual são fornecidas duas imagens: uma indicando o padrão de busca (constante) e outra o alvo da operação. Esta operação pode retornar a sua localização na imagem alvo em termos de coordenadas e a porcentagem de semelhança do elemento encontrado com o parâmetro da busca. Um outro sumarizador, como por exemplo o histograma com a distribuição de cores, pode retornar uma informação que consiste no próprio histograma, acompanhado da média, variância e desvio padrão da distribuição de tons de cores. Cada sumarizador definido representa um processamento que retorna

uma série de parâmetros dependentes da própria operação. Portanto, os **parâmetros** ou **variáveis descritoras** de um sumariizador representam o conjunto de valores obtidos a partir da aplicação deste sobre uma determinada imagem. Estes elementos descrevem o sumariizador em seu aspecto estrutural, através do conjunto de informações que possa ser extraído das imagens. Estas informações são representadas através de elementos numéricos, portanto, mensuráveis, o que permite definir operações de recuperação das imagens através da comparação de seus valores. O valor representado pelo parâmetro indica o **nível de similaridade** das imagens envolvidas na operação.

A comparação e definição de ordem relativa entre imagens para um determinado aspecto é possível, se este representar uma característica que possa ser descrita através de valores mensuráveis. As características tais como forma, textura, histograma se enquadram nesta categoria. Assim, ao se descrever o conteúdo de imagem, utilizando-se uma das características para a recuperação a partir do Banco de Dados, esta operação será avaliada, comparando-se os valores resultantes da aplicação dos operadores correspondentes. As operações de comparação entre as imagens são realizadas de forma indireta através dos valores que descrevem de forma mensurável o grau de semelhança de cada imagem presente no BD com relação às constantes imagens.

6.2.3. Instâncias das Variáveis Descritoras.

A aplicação de um sumariizador em uma nova imagem inserida no BD, resulta em elementos numéricos que definem a instância da sua variável de descrição. As variáveis assumem valores que podem ser de diferentes tipos de dados, a saber: um

elemento numérico, um par ordenado, uma tripla ou um conjunto ordenado de valores. A semântica associada a cada um destes tipos diferentes de valores é determinada através da característica representada pelo sumarizador e da constante imagem. Além disso a sua definição se torna mais precisa na medida que envolve valores mensuráveis.

Em alguns sumarizadores, o resultado representa naturalmente uma comparação entre dois elementos do tipo imagem, sendo que neste caso o valor indica o grau ou nível de semelhança entre as imagens. Como exemplo deste sumarizador, considere o que realiza a busca de padrão, onde uma das variáveis de descrição indica a semelhança entre as duas imagens. O valor assumido pela variável “porcentagem de semelhança” será o resultado da comparação entre a constante imagem, à qual se aplica, com uma nova imagem a ser inserida no BD. Considere um exame de tomografia de uma secção do corpo humano que contenha elementos gráficos, a partir dos quais se possa afirmar que na imagem está presente o coração com uma determinada probabilidade, representando assim a similaridade entre o coração indicado por uma constante imagem e o coração presente na imagem inserida no BD. Neste caso, a probabilidade indicada pode ser representada através de um valor numérico, por exemplo 80. Observe que deve ser definido para cada sumarizador, além do tipo de valor que possa ser retornado, o domínio do qual estes serão extraídos. No caso anterior, consistem de números inteiros pertencentes ao intervalo fechado de 0 a 100.

Para outros sumarizadores as suas variáveis de descrição não representam o resultado de uma comparação entre duas imagens, por ser esta operação

semanticamente inválida. Neste caso, o valor assumido pela variável de descrição é o resultado da aplicação do sumarizador na nova imagem criada no BD. Se enquadra nesta categoria o sumarizador histograma, que pode ser caracterizado pela própria distribuição dos níveis de cores, ou pela média, desvio padrão e variância, os quais consistem de fatores que se referem à imagem na qual foi aplicado. Estes podem ser posteriormente utilizados para se comparar o histograma de duas imagens, obtendo-se o grau de similaridade entre estas, de acordo com sua distribuição de cores. Mesmo para o sumarizador busca de padrão pode-se definir algumas variáveis que não representam uma comparação explícita entre imagens. Uma possível variável seriam as coordenadas de localização de um padrão no interior de outra imagem. Este conjunto de valores pode ser aplicado na realização de outros tipos de consultas, envolvendo a distribuição espacial dos objetos no interior de uma imagem.

6.2.4. Limite de Aceitação.

Observe que a aplicação dos sumarizadores resulta em valores em suas variáveis que raramente possibilitam uma comparação precisa do tipo verdadeiro ou falso entre os parâmetros da expressão. Para as imagens não é possível estabelecer um resultado lógico preciso nas operações de comparação com as constantes imagens, a menos que as duas envolvidas sejam exatamente as mesmas, sem nenhum fator que as diferenciem. No caso genérico, as imagens possuem um volume elevado de informações que tornam complexa e com alto custo de processamento a realização desta operação. A aplicação dos sumarizadores permitem que a comparação seja efetuada dentro de outro critério que não seja o de verdadeiro ou falso.

O “sucesso” (ou o resultado verdadeiro) da avaliação de um sumariizador depende se o seu resultado encontrar dentro de um **limite de aceitação** (ou **tolerância**) indicado. Desta forma, todas as consultas a serem realizadas sobre o BD com imagens, passa por uma etapa inicial de definição do contexto de aceitação do resultado esperado, que indica as diferentes necessidades do usuário em cada instante. Este conceito permite uma generalização maior da semântica associada à expressão de definição da condição da consulta, uma vez que esta pode ser definida livremente. Torna-se portanto possível expressar o que significa uma imagem “parecida” ou “semelhante” com um determinado padrão de busca apresentado. Estes termos passam a apresentar uma precisão em seu significado na medida em que as características representadas pelos sumariizadores sejam quantificadas. O “parecido” pode ser definido como valor para uma busca de padrão que resulte em um nível de semelhança entre 80% e 100%.

Além do limite de aceitação definido para a construção do resultado de uma consulta, este valor é utilizado como critério de criação da entrada na estrutura de acesso a ser construída para cada característica suportada. Nem toda imagem inserida no BD possui uma chave de acesso formada pelo valor resultante da aplicação de um sumariizador em relação a alguma constante. Sem esta definição, qualquer imagem iria possuir uma chave de acesso em cada estrutura, mesmo que não apresente um resultado significativo para o sumariizador aplicado, aumentando de forma desnecessária o volume de informações a serem consideradas na localização de um determinado elemento do tipo imagem. Portanto, cada estrutura de acesso apresenta entradas apenas para as imagens semanticamente relacionadas entre si.

6.2.5. Modificadores.

Além da elaboração do conjunto de sumarizadores para as constantes imagens, torna-se necessário a representação de outro conjunto de parâmetros que possam ser aplicados ou não a todas as constantes. Estes parâmetros se referem a alterações no aspecto comportamental que o sistema poderá apresentar para as operações realizadas sobre cada constante. Por exemplo, uma constante do tipo coração pode permanecer sendo identificada dentro de outra imagem com relação a uma busca de padrão, se a mesma aparecer rotacionada, em relação à original, por 30 graus, ou em uma escala diferente. Estas alterações consistem em modificações na forma de apresentação da constante imagem resultante de transformações geométricas, ângulos de visualização, contrastes, e assim mesmo continuar sendo identificada em uma nova imagem inserida no BD. A alteração no comportamento de um sumarizador, definindo uma nova forma de atuar no reconhecimento de uma determinada característica para uma constante imagem, é denominada de **modificador**.

Estes parâmetros procuram fornecer um mecanismo para que as diversas formas de apresentação de uma imagem possam ser representadas e reconhecidas, tal que não comprometam o processo de aplicação de um determinado sumarizador. Como mencionado no exemplo anterior, uma imagem, ao sofrer uma operação de busca de padrão por uma constante representativa do coração, não pode resultar em falha se esta encontrar rotacionada em relação à constante, caso este aspecto de diferenciação não seja relevante. Esta definição implica em um incremento na complexidade dos métodos de processamento de imagens aplicados, no entanto,

fornece a flexibilidade necessária à manipulação do conjunto de imagens com o objetivo de formular consultas baseadas em seu conteúdo gráfico.

Os modificadores, além de não se aplicarem necessariamente a todos os sumarizadores, podem implicar em comportamento distinto em cada conjunto de constantes imagens. Como mencionado na definição das constantes imagens, algumas representam imagens complexas, com a presença de vários elementos identificáveis, como por exemplo, uma tomografia de uma região do crânio. Sendo estas imagens definidas como uma das constantes no BD, os sumarizadores a serem aplicados apresentarão o seguinte comportamento: os sumarizadores juntamente com seus possíveis modificadores atuam considerando as imagens como um todo, ou seja, como um único elemento, sem envolver a identificação própria de cada um dos objetos presentes. No entanto, algumas constantes representam apenas alguns detalhes que podem estar presentes como um dos elementos significativos no interior da imagem. Como exemplo, o detalhe do coração definido como constante e uma imagem contendo o coração; uma constante imagem sobre a característica de um determinado tumor a ser utilizada como elemento de comparação para imagens de diferentes partes do corpo que possam contê-lo. Neste caso, a atuação do modificador deverá possibilitar a identificação do elemento representado pela constante no interior da imagem, de acordo com o sumarizador selecionado.

Observe o ponto de diferenciação entre estes dois tipos de constantes: (a) uma constante imagem é tratada como uma imagem completa a ser comparada com uma outra imagem como um todo; e, (b) a constante representando um elemento que possa ser localizado no interior de outra imagem. Esta diferença leva a definições

distintas da semântica relacionada com as operações a serem aplicadas entre a constante e as imagens solicitadas. No caso da constante do tipo (a) as operações são aplicadas sobre a imagem global, levando em consideração os modificadores que possam estar definidos. No caso da constante do tipo (b) a semântica definida para cada operação pode implicar na realização de uma busca no interior de uma imagem pela presença do elemento representado.

6.3. MODELAGEM DO CONCEITO DE IMAGEM NO SIRIUS.

No item anterior foi apresentada a conceituação de imagens e dos elementos envolvidos na realização do seu suporte e tratamento dentro de uma aplicação que seja desenvolvida utilizando-se o Modelo SIRIUS. Neste tópico, estes elementos serão modelados em função dos conceitos suportados pelo SIRIUS, o que fornece um significado semântico mais preciso sobre a definição da imagem e a representação da informação presente no interior de cada uma.

Uma imagem inserida em um BD definido através do SIRIUS apresenta o seu conjunto de informações descrito por constantes imagens e sumarizadores utilizados em sua identificação. As constantes imagens representam o conjunto de informações, na forma de objetos ou imagens complexas, que podem ser utilizadas como elementos de descrição do conteúdo das imagens associadas a algum tipo de objeto ou tipo de relacionamento do BD. Esta descrição é obtida através dos sumarizadores que se encontram associados a cada constante imagem. Os sumarizadores representam as características que serão utilizadas como base de comparação entre a informação contida na constante imagem com as imagens no BD. O resultado da operação representada pelo sumariador consiste em um valor mensurável, o qual é

indicado através dos parâmetros ou variáveis de descrição. Além destes elementos, pode-se especificar uma alteração no comportamento do sumariizador aplicado à constante imagem, através dos modificadores, representando um aspecto diferente através do qual foi obtido a descrição do conteúdo da imagem.

A modelagem destes elementos aparece no esquema de uma aplicação no SIRIUS como objetos habitantes da **colônia α -imagem**. Nesta colônia habitam todos os objetos que representam os conceitos necessários ao suporte de imagens e das informações contidas em seu interior, definindo os meta-tipos de dados para imagens, sendo portanto controlados pelo gerenciador. Como consequência, o suporte a imagens se torna nativo do Sistema, restando ao projetista da aplicação o trabalho de especificação das características inerentes ao domínio da aplicação. A Figura 1 apresenta o Meta-Eschema do Atributo com Característica de Imagem no SIRIUS, representando a semântica da informação identificada neste atributo. Consiste de um esquema em que os conceitos são representados de uma forma genérica.

Figura 1 - Meta Esquema para o Atributo com Característica de Imagem.

Uma informação que seja do tipo de dado imagem é modelada no SIRIUS como uma especialização do objeto Tipo Atributo, com a característica contendo o valor "Imagem". Este objeto representa o meta-tipo denominado AtributoImagem, ou, de acordo com a terminologia do SIRIUS, Atributo com Característica de Imagem. Uma vez definida como característica de atributo, uma imagem pode ser inserida na modelagem como elemento que descreva um Tipo de Objeto ou um Atributo Tipo Relacionamento no Esquema da Aplicação. O objeto Tipo AtributoImagem apresenta como atributos, o Nome_Tipo_Atributo, identificando-o quando uma instância deste for definida no Esquema da Aplicação, e o Nome_Instance, que identifica cada valor do atributo existente no Banco. Como o valor deste tipo de Atributo é uma imagem, o Nome_Instance identifica esta

instância em particular. Além da identificação da imagem, esta possui um conjunto de informações próprio de cada instância, ou seja, cada imagem existente no BD, representado como atributos do atributo Nome_Instância. São estes:

- Objeto_Imagem (imagem): conjunto de bytes representando a matriz com os “pixel” de formação da imagem em si;
- Tamanho: número de bytes total da imagem;
- Dim_X: dimensão da imagem no eixo x, fornecida em número de “pixel”;
- Dim_Y: dimensão da imagem no eixo y, fornecida em número de “pixel”;
- Formato: nome do formato de arquivo gráfico com a imagem.

Pode-se notar que a partir da modelagem do objeto Atributo com Característica de Imagem, o seu conjunto de atributos define os critérios para a classificação/instanciação de ocorrências deste tipo. Os números apresentados ao lado do nome de cada atributo indica qual será instanciado em cada nível da hierarquia. O atributo Nome_Tipo_Atributo será instanciado no primeiro nível da hierarquia, com o valor representando o nome deste atributo em uma determinada porção da modelagem. Isto ocorre quando da criação de um atributo com característica de imagem em algum tipo de objeto ou relacionamento presente no Esquema da Aplicação. O atributo Nome_Instância, por identificar cada imagem do BD, recebe o seu valor no segundo nível da hierarquia de classificação, quando um novo valor for atribuído para o atributo do tipo Nome_Tipo_Atributo, ou seja, a inserção de uma nova imagem no BD. O mesmo ocorre com os demais atributos indicados para este nível da hierarquia, responsáveis por descrever cada imagem existente no Banco.

As imagens que delimitam as características do domínio da aplicação definidas como constantes imagens são modeladas através do objeto com nome do tipo `Constante_Imagem`. Os atributos deste tipo de objeto consistem em:

- **Nome:** identificador para a primeira instância, significando o nome fornecido para a constante imagem;
- **Tipo:** informa se a constante imagem representa uma imagem que possa ser localizada através de algum critério no interior de outra (subimagem), ou se esta consiste em uma imagem complexa;
- **Descrição:** uma vez que a constante imagem é também uma imagem, esta apresenta os mesmos atributos de descrição de uma instância do atributo com característica de imagem, definidos anteriormente, ou seja, `Objeto_Imagem`, `Tamanho`, `DimX`, `DimY` e `Formato`. Além destes atributos fundamentais para a definição da constante imagem, caso seja relevante a um determinado domínio de aplicação, pode-se acrescentar atributos que permitem complementar sua descrição, tais como meio de aquisição, comentários gerais sobre a imagem representada e outros.

O objeto tipo `Constante_Imagem` apresenta uma associação com o objeto tipo `Atributo com Característica Imagem` indicado pelo atributo relacionamento `Descrito_Por/Descreve`, definindo o tipo de informação que possa estar presente no interior da imagem. Este relacionamento apresenta, em ambos os sentidos, multiplicidade (1,N) significando que uma instância do atributo de característica imagem pode conter vários objetos (constante imagem) em seu interior, e estes podem ser identificados em diversas ocorrências do atributo.

As características que possam ser extraídas de uma imagem são representadas através do objeto tipo Sumarizador. Os seus atributos são o Nome do sumarizador, que irá identificá-lo nas expressões de acesso ao BD, o Tipo de operação a ser realizada, que consiste no identificador do tipo da característica representada, ou seja, manipulação de histograma, busca de padrão, e outros, e o atributo Alteração, indicando se o sumarizador admite modificação no aspecto comportamental da operação representada. Os objetos do tipo Variável_Descritora representam o conjunto de valores utilizados na descrição das características suportadas pelo gerenciador. Seus atributos são o Nome de identificação da variável, a Categoria a que pertence, ou seja, qual tipo de dado a ser representado por esta variável para o sumarizador associado, e o Limite_Aceitação para definir o valor limite na criação de uma entrada em sua estrutura de acesso. Esta associação é modelada através do atributo relacionamento Descrito_Por entre o objeto do tipo Sumarizador e o objeto do tipo Variável_Descritora. Além disso, associado à criação de uma instância do relacionamento Descrito_Por, define-se a *regra de integridade de descrição* que determina a categoria de variável que possa participar desta associação.

Com esta restrição é garantida a associação de variáveis que sejam válidas na descrição de um sumarizador, baseando-se na característica representada. Através deste Esquema, pode-se criar tipos diferentes de sumarizadores para indicar a mesma característica através de elementos descritores diversos, de acordo com a necessidade da aplicação. No entanto, os descritores devem ser semanticamente significativos na descrição da característica representada no sumarizador. Como exemplo, variáveis

que indicam as coordenadas de localização de uma constante em uma imagem são significativas quando associadas a um sumariador de busca de padrão, e não ao de manipulação de histograma.

Em SIRIUS, o domínio para o atributo Tipo do objeto Sumarizador consiste nos objetos HISTOGRAMA e MATCHING, do tipo Operador, que representam os métodos de processamento de imagens a serem aplicados. A condificação destes métodos é realizada em uma linguagem de programação qualquer, e definida como a instância de um atributo com a característica de procedimento (Método) do objeto tipo Operador. Estes tipos de objetos são descritos por um conjunto de variáveis internas, as quais representam o conjunto de parâmetros retornados por cada método associado ao Operador. Estas variáveis internas, descritas pela tupla $\langle \text{Tipo_de_Dado}, \text{Variavel_Interna} \rangle$, podem ser das seguintes categorias:

- a) HISTOGRAMA: além do conjunto do vetor representando a distribuição de cores da imagem, este pode ser também descrito pelas seguintes variáveis descritoras:
 - a.1) Média da distribuição de cores;
 - a.2) Valor do nível de variância;
 - a.3) e o seu desvio padrão;
- b) MATCHING: os sumariadores de busca de padrão podem retornar informações que consistem em:
 - b.1) Coordenadas da posição de *matching* do padrão no interior da imagem;
 - b.2) Quantidades de pontos semelhantes ou porcentagem de semelhança.

Os modificadores são representados através de uma especialização do objeto tipo Sumarizador, quando este admitir modificação em seu comportamento padrão.

Atributos dos modificadores consistem nos parâmetros específicos que representam a sua atuação. Sendo o objeto tipo Modificador um subtipo do objeto Sumarizador, é possível que o primeiro apresente um conjunto de variáveis próprias que o descreva, além das que já se encontram definidas em consequência do mecanismo de herança. Na modelagem os atributos específicos consistem em:

- `Tipo_Modificador`: indica o tipo de operação suportada pelo modificador, como por exemplo, rotação ou translação para o caso de modificadores do sumarizador busca de padrão (*matching*);
- `Valor_Médio`: definição do valor médio de variação dos elementos associados a cada sumarizador para que o resultado de sua aplicação possa ser considerado como um elemento válido na descrição do conteúdo da imagem;
- `Limite_Aceitação`: valor máximo definido para que o resultado da aplicação do modificador possa ser considerado como um elemento válido na descrição do conteúdo da imagem.

A informação que possa ser extraída a partir de uma imagem é indicada através do relacionamento triplo envolvendo os objetos do tipo `Constante_Imagem`, `Sumarizador` e `Variável_Descritora`. O objeto tipo `Constante_Imagem` relaciona-se com o objeto tipo `Sumarizador` constituindo no *relacionamento identificador* e o objeto tipo `Sumarizador` associa-se ao objeto tipo `Variável_Descritora` estabelecendo o *relacionamento de conjunto*. Semanticamente, um sumarizador caracteriza um determinado aspecto da constante imagem, através de uma série de parâmetros representados pelas variáveis descritoras. O relacionamento `Sumarizador` caracterizando a `Constante_Imagem`

é *identificador*, pois as variáveis descritoras utilizadas neste processo são identificadas a partir de instâncias deste relacionamento. Por outro lado, o Sumarizador em si é descrito pelas Variáveis_Descritoras, podendo definir outro sumarizador que seja descrito por algumas das variáveis, caracterizando a mesma constante imagem, ou uma outra qualquer. Como atributos deste relacionamento tem-se o Valor_Cte_Imagem, que armazena o valor da variável descritora que descreve o sumarizador aplicado à constante imagem, caso este resultado tenha significado semântico. O outro atributo Valor_Imagem representa o conjunto de valores de cada variável descritora, resultantes da aplicação do Sumarizador na extração de uma característica da constante imagem comparada a uma imagem inserida no BD.

6.3.1. Definição do Esquema de Imagens para uma Aplicação.

A definição do Esquema de uma aplicação que envolva a utilização de imagens, tem início com a criação de um objeto do tipo Atributo com Característica de Imagem na colônia Esquema. A etapa seguinte consiste na construção do esquema de caracterização do domínio da aplicação, ou seja, tratamento de imagens médicas via tomografia ou radiografia, sistema de informações geográficas, fotos, quadros em museus, etc, a qual é realizada obedecendo o Meta-Esquema existente na colônia α -imagem. Esta caracterização ocorre através da instanciação dos objetos tipo Constantes Imagens adequadas para representar as informações que se espera encontrar nas imagens a serem posteriormente inseridas e manipuladas. Outro elemento, consiste na inicialização do conjunto de características que descrevem o aspecto de comparação de cada constante imagem, ou seja, quais sumarizadores

estarão ativos. Uma vez definidos os sumariantores, deve-se associar as variáveis de descrição obedecendo a restrição semântica definida no Esquema para sua validação. Em consequência da característica de associação das variáveis para descrever o sumariantor, pode-se, como mencionado no item anterior, criar vários tipos de sumariantores distintos para uma mesma característica de imagem (constante imagem). O significado imediato da definição anterior, está, por exemplo, na criação de dois sumariantores do tipo busca de padrão: (a) um no qual o interesse seja apenas o nível de semelhança entre o padrão de busca e a imagem alvo; e (b) outro sumariantor em que, além do nível de semelhança, sejam consideradas as coordenadas de localização do padrão de busca.

Na definição dos sumariantores que possam apresentar alterações em seu comportamento, cria-se as instâncias de seus subtipos que sejam mais adequados, ou seja, os modificadores, e associa-se outras variáveis que possam ser específicas à característica representada. Pelo Meta-Esquema presente na colônia α -imagem, define-se o número de constantes imagens que seja apropriado na representação de uma aplicação, assim como das características mais adequadas na descrição da semântica de uma imagem.

Finalizando a especificação do esquema da aplicação define o conjunto de informações que se pretende encontrar em cada atributo com característica imagem. Esta informação é representada através do objeto relacionamento Descrito-*Por*, entre o objeto do tipo Atributo com Característica de Imagem e com as instâncias dos objetos do tipo Constante_Imagem que possam ser identificadas. Esta associação caracteriza o “domínio” de cada imagem presente no BD, dentro do contexto da

aplicação. Se a aplicação manipula um conjunto de imagens que se referem a vários temas ou categorias, esta classificação já pode ser realizada inicialmente a nível do Esquema, ou seja, indica-se previamente o tipo de informação que se espera encontrar em cada imagem.

A caracterização final da semântica, ou seja a descrição do conjunto de informações contidas no interior de uma imagem, é obtida a partir dos valores resultantes da aplicação dos sumarizadores associados a cada constante que se espera encontrar. Como os sumarizadores representam a característica de cada constante que pode ou não estar presente em uma determinada imagem, os valores das variáveis descritoras indicam o seu grau de similaridade. Através deste conjunto de valores, referente a cada aspecto das características (sumarizadores) associadas às constantes, tem-se a descrição sobre o conteúdo da imagem. Estes elementos serão posteriormente utilizados como parâmetros na realização de consultas, representando as características da imagem que se deseja recuperar.

Os valores assumidos pelas variáveis de descrição dos sumarizadores são obtidos com a inserção de uma nova imagem no BD. A partir deste instante são aplicadas as operações representadas pelos sumarizadores associados às constantes de caracterização do domínio do atributo imagem. Isto significa que todo o processamento das imagens para a extração de informações do seu conteúdo é realizado apenas neste ponto. Durante a avaliação das operações de consulta, os sumarizadores não serão mais ativados, uma vez que o seu resultado já se encontra armazenado no BD. Portanto, a operação de inserção apresentará um desempenho dependente do processamento a ser realizado sobre a imagem. A avaliação da

consulta se torna mais eficiente uma vez que as operações caras ao sistema já foram realizadas, restando apenas a definição de uma estrutura de acesso que não venha a prejudicar o seu desempenho.

6.3.2 Exemplo da Modelagem de uma Aplicação com Imagens Médicas no SIRIUS.

A Figura 2 apresenta o exemplo de modelagem das imagens presentes em um domínio de exames de tomografia por ressonância magnética de algumas regiões do corpo humano, em particular referenciando-se à caixa craniana e à região do tórax. Nesta aplicação define-se os objetos do tipo *Médico* e *Paciente*, e um relacionamento entre eles através dos atributos *Consulta* e *Consultado_Por*. Para caracterizar uma aplicação com a presença de imagens no BD, cria-se associado a um dos tipos de objetos do esquema, um atributo com a característica de imagem. Este consiste nos atributos *ExameTomografia* e *TomografiaCrânio*, do objeto *Paciente*, representando os tipos de exames que este tenha realizado, criados como valores de instância do objeto tipo Atributo com Característica de Imagem. Observe que os atributos *ExameTomografia* e *TomografiaCrânio* apresentam um conjunto de atributos adicionais, além dos que se encontram definidos no Meta-Esquema da aplicação (Figura 1), no caso *Anotação*, *DataRealização* e *Laboratório*. Estes atributos são considerados específicos desta modelagem em particular do domínio da aplicação.

Figura 2 - Esquema de Exames médicos realizados em Pacientes.

Uma vez definida a presença de imagens na aplicação de BD a ser desenvolvida, a etapa seguinte consiste na delimitação do seu domínio de abrangência ou o escopo da aplicação, criando-se instâncias do objeto tipo Constante Imagem. Em se tratando de uma aplicação com o domínio nas imagens médicas, os atributos com característica de imagem irão apresentar valores, cujos conteúdos possam ser descrito pelos objetos tipo Constante Imagem *Coração*, *Pulmão*, *Medula*, *Hipotálamo*, e *Tálamo* (Figura 3 e Figura 4). Observe que todas as constantes imagens apresentam o atributo *Tipo* com o valor “*Subimagem*”, indicando que estas possam ser localizadas, através de um determinado critério, no interior de uma imagem. O nome atribuído a cada objeto do Tipo Constante Imagem é significativo com relação ao conteúdo apresentado pela matriz de pixel que consiste no valor do

atributo *Objeto_Imagem*. Assim pode-se formalizar que as instâncias do atributo tipo *ExameTomografia* poderão ser identificadas em seu conteúdo, imagens semelhantes às representadas pelos objetos do tipo *Coração*, *Pulmão* e *Medula* (Figura 3). O mesmo ocorrendo com o atributo tipo *TomografiaCrânio*, o qual apresentará o conteúdo de suas instâncias descritas em função de comparações realizadas com as imagens presentes nos objetos do tipo *Hipotálamo* e *Tálamo* (Figura 4). Esta descrição é realizada com a criação de instâncias do atributo tipo relacionamento *Descrito_Por*.

Figura 3 - Descrição do Conteúdo do Atributo Exame de Tomografia.

Os critérios para a comparação entre as constantes imagens com os valores dos atributos com característica imagem são definidos através dos objetos Tipo Sumarizador. No esquema os objetos *Padrão*, *BuscaPadrão* e *Histo*

Figura 4 - Descrição do Conteúdo do Atributo Tomografia do Crânio.

Figura 5 - Definição dos Operadores.

representam, os dois primeiros, as características de busca de padrão (*matching*) e o último, a característica de determinação do Histograma de uma imagem. Esta informação é indicada através dos valores assumidos pelo atributo *Tipo*, conforme é observado

na Figura 5. O objeto *Padrão* admite uma especialização em sua definição, permitindo a modelagem de uma alteração em seu comportamento. Esta alteração é indicada pelos objetos do tipo Modificador, *Rotação* e *Trans*. O modificador *Rotação* representa o fato de uma constante imagem ser reconhecida no interior de uma determinada instância do BD, através do sumarizador *Padrão*, mesmo que esta apareça rotacionada em relação à sua posição original. O limite máximo da rotação suportada consiste no valor assumido pelo atributo *Valor_Médio*, no caso é de 30 graus da posição original. De forma semelhante o modificador *Trans* representa a operação de translação, proporcionando o reconhecimento das constantes imagens que se encontrem deslocadas da posição original ao longo do eixos. Definição semelhante é apresentada para o sumarizador *BuscaPadrão*, no qual a alteração em suas ações é modelada pelo modificador *Girar*, do tipo *ROTACIONA*, no entanto a tolerância para a sua aplicação consiste de 35 graus em relação à posição original.

Figura 6 - Descrição das Constantes Imagens: Medula e Tálamo.

Para finalizar, a caracterização dos sumarizadores é obtida com a definição

Figura 7 - Descrição das Constantes Imagens - Coração e Pulmão

dos objetos do tipo Variável Descritora. As que são necessárias para a descrição do esquema da aplicação são representadas pelos objetos *PosiçãoX*, *PosiçãoY*, *Semelhança*, *Media*, *DesvioPadrão*, *Variância* e *Distribuição*. A formalização da informação de comparação contida em cada constante imagem é realizada com a instanciação do relacionamento triplo *Caracteriza/Descrito_Por*. Os esquemas destas associações estão descritos nas figuras 5, 6, 7 e 8. Considerando o esquema da Figura 8, semanticamente o objeto *Histo* caracteriza um aspecto da informação contida no objeto constante imagem *Coração*, ou seja, sua distribuição de cores, descrevendo esta informação especificamente através das Variáveis Descritoras *Media*, *DesvioPadrão*, *Variância* e *Distribuição*. O mesmo ocorre para o objeto *Padrão*, o qual descreve outro aspecto da informação presente no interior na constante imagem *Coração*, que consiste na comparação de padrões a serem realizados com as instâncias dos atributos em que possa ser localizada (Figura 7). Esta comparação é descrita através das variáveis descritoras *Semelhança*, *PosiçãoX* e *PosiçãoY*. Uma interpretação semântica semelhante pode ser realizada para os demais esquemas de descrição das constantes imagens. Importante notar que a representação precisa do conhecimento referente ao domínio da aplicação é obtida com a criação de instâncias para o atributo relacionamento triplo *Caracteriza/Descrito_Por*, envolvendo os objetos do tipo constante imagem, sumarizador e variável descritora.

A modelagem através de construtores genéricos (tipos de objetos e atributos) dos conceitos de suporte a imagens no SIRIUS permite que sejam

Figura 8 - Descrição da Constante Imagem: Hipotálamo

Figura 9 - Descrição da Distribuição de Cores da Constante Coração.

definidos de uma forma natural várias instâncias de objetos representando características distintas e/ou específicas presentes na aplicação. Esta flexibilidade pode ser observada com a definição de instâncias distintas do objeto Sumarizador,

Padrão e *BuscaPadrão*, com o mesmo valor para o atributo tipo, *MATCHING*

(<<figura instância dos objetos tipo operador>>) Apesar de semanticamente representarem a operação de comparação de padrões, estes são definidos de forma distinta, aparecendo no Esquema da Aplicação descrevendo aspectos diversos para cada constante imagem.

A descrição precisa do conteúdo de uma imagem pode ser obtida através de uma análise do esquema da aplicação. Considerando apenas uma instância para cada tipo de relacionamento presente no Esquema apresentado nas figuras mencionadas neste item, pode-se exemplificar este aspecto. O atributo do tipo *ExameTomografia* apresenta o seu conteúdo descrito pelas constantes imagens *Coração*, *Pulmão* e *Medula*, sendo portanto os únicos órgãos a serem utilizados na extração de qualquer informação presente em uma imagem que seja instância deste atributo. No caso da constante imagem *Coração*, as informações serão extraídas de cada instância, através da comparação dos histogramas (*Histo*), em todos os seus aspectos (*Média*, *DesvioPadrão*, *Variança* e *Distribuição*), e dos padrões (*Padrão*) em função dos aspectos *Semelhança*, *PosiçãoX* e *PosiçãoY*. Estes objetos representam a modelagem do conhecimento sobre o conteúdo de uma imagem de exames de tomografia. Interpretação semelhante pode ser fornecida para cada constante imagem associada aos tipos de atributos com característica de imagem.

A Figura 10 apresenta uma porção do banco de dados com a representação de instâncias do objeto tipo Paciente, cujo esquema se encontra descrito na Figura 2. O paciente “José Silva” apresenta em uma pasta de documentação, dois exames identificados por *ExTomografia* e *ExCranio*, que são instâncias de um atributo com

Figura 10 - Representação de Instâncias para o Atributo com Característica de Imagem.

característica de imagem. Portanto, para obter uma descrição do conteúdo presente nestas imagens, é necessário verificar as porções do esquema que o descreve. Observe que o atributo *ExTomografia* apresenta alguns atributos que são específicos a esta modelagem, ou seja, *Anotação*, *Laboratório* e *DataRealização*, os quais não foram especificados no esquema de um atributo com característica de imagem. Este consiste de um recurso próprio do modelo: a inserção de novos atributos em níveis quaisquer de instanciação.

6.4. ESTRUTURAS DE ACESSO A IMAGENS.

A especificação de uma estrutura de acesso e indexação de imagens consiste em um importante ponto de pesquisa. Como mencionado no capítulo 2, as estruturas de indexação propostas são mais eficientes no tratamento de informações espaciais, sendo, portanto, adequadas a outros tipos de sistemas tais como processamento

geográfico (GIS). Em função dos conceitos apresentados para o tratamento de imagens, a definição das estruturas de acesso deve permitir a avaliação de consultas baseadas em seu conteúdo. A nível das estruturas de acesso, o conteúdo de uma imagem consiste do conjunto de valores assumidos pelas variáveis de descrição dos sumarizadores.

O conteúdo de cada entrada na estrutura de índices é portanto formado pelo grau de similaridade representado nas variáveis de descrição de um sumariizador, e com o identificador da imagem no BD no qual foi obtido, e o tipo do atributo com característica de imagem. A cada imagem do BD é definido, a nível de Esquema da aplicação, o conjunto de constantes que descreve o seu domínio. Estas, independentes de estar ou não ligadas a uma imagem, possuem o conjunto de sumarizadores representativos de suas características, as quais são medidas por valores assumidos pelas variáveis de descrição. Portanto, deve existir uma estrutura de acesso para cada variável associada a cada sumariizador de descrição da constante. Ou definindo de outra forma, o sistema gerencia uma estrutura de acesso para cada instância do relacionamento de modalidade Descrito_Por, presente em cada instância da tripla com modalidade Caracteriza/Descrito_Por existente no BD.

Um fator importante para a definição desta estrutura de acesso, consiste nas características representadas através dos sumarizadores. Estas retornam valores que são mensuráveis e, conseqüentemente, possibilitam definir uma relação de ordem entre eles. Portanto é possível classificar as imagens presentes no BD através da

ordem crescente de uma característica, como, por exemplo, o nível de semelhança entre duas imagens, fornecido em porcentagem.

Ao associar um sumariizador a uma constante imagem, cria-se uma nova instância do relacionamento de modalidade Descrito_Por. No exemplo das figuras 5 e 6, o sumariizador BuscaPadrão se encontra associado, respectivamente, às constantes imagens Tálamo e Pulmão, apresentando, portanto, duas instâncias do relacionamento Descrito_Por. Cada uma destas instâncias apresentará como atributo uma entrada para sua estrutura de acesso. Os valores presentes nesta estrutura de acesso consistem de uma entrada para cada imagem do BD, cujo resultado se encontra dentro do seu limite de aceitação. Desta forma, para um mesmo sumariizador, uma imagem pode se encontrar na estrutura de acesso de uma de suas variáveis e não estar presente em outra, dependendo do limite de aceitação que seja definido. No exemplo, as instâncias dos atributos do tipo ExameTomografia e TomografiaCrânio, ou seja, as várias imagens, serão avaliadas com relação a presença das constantes Pulmão e Tálamo. A avaliação será realizada em função dos modificadores que se encontram associados. Se o resultado estiver no limite de aceitação definido será criada uma entrada na estrutura de índice correspondente à variável descritora.

6.5. LINGUAGEM DE ACESSO A IMAGENS NO SIRIUS.

6.5.1. A Abordagem da Linguagem.

A abordagem adotada para a construção da linguagem de acesso às imagens em um Banco de Dados definido através do modelo SIRIUS consiste em incorporar

extensões à linguagem de consulta relacional (SQL). Apesar de ser uma linguagem definida para BD Relacionais, pequenas alterações fornecidas à semântica de cada uma das construções a torna adequada à manipulação de um Banco de Dados Orientado a Objetos. Além disso, a sua sintaxe é bem difundida na comunidade de BD, o que facilita o aprendizado por parte do usuário final e do desenvolvedor de aplicações, dos comandos e construções que sejam específicos à manipulação de imagens.

6.5.2. Requisitos para uma linguagem de acesso a imagens.

Uma linguagem de consulta que faça acesso a um Banco de Imagens, realizando o tratamento da informação contida na imagem como dados gráficos, deve apresentar uma série de requisitos, dentre os quais podem ser citados:

- a) O tipo de dado imagem deve ser manipulado naturalmente pela linguagem, sem a necessidade do usuário desta se preocupar com o tratamento interno oferecido pelo sistema ou com complexidade da informação gráfica representada;
- b) A consulta de um conjunto de imagens deve ser realizada fornecendo-se uma descrição gráfica do conteúdo da informação a ser recuperada. Isto implica em uma manipulação da imagem como uma imagem, e não se utilizar apenas da descrição textual resultante da atividade de análise e processamento, ou de outras características de atributos associadas a esta.
- c) Possibilidade da indicação de elementos significativos que possam estar presentes nas cenas representadas através da imagem. Estes elementos serão utilizados como construtores da informação a ser recuperada. Não consiste em realizar a criação de

ícones representativos como ocorre nas linguagens de consulta espaciais, mas na apresentação de uma imagem como elemento de comparação.

- d) A transparência na manipulação de todos os elementos gráficos que compõe a linguagem implica a presença de uma interface adequada à formulação das consultas, com características de interface gráfica.
- e) A manipulação de informações textuais e gráficas deve ser realizada de maneira uniforme pela linguagem, sem a diferenciação através de uma sintaxe, observando apenas as respectivas distinções dos tipos de dados suportados e da interpretação semântica oferecida.
- f) A linguagem deve permitir diversas formas de apresentação do resultado de uma consulta que envolva a recuperação de informações através de imagens. A visualização destes dados deve ser flexível, uma vez que o resultado pode conter informações que sejam gráficas (histogramas da imagem), a própria imagem e textos que as descrevam. Além disso, com relação às imagens o usuário pode estar interessado na visualização de aspectos distintos, tais como borda, geometria dos objetos, padrões e outros.
- g) A utilização de resultados de consultas anteriores em novas consultas, possibilitando uma interação dinâmica com o sistema na busca da informação mais adequada em decorrência da pesquisa realizada.

A linguagem de definição e manipulação do atributo com característica de imagem proposta para o SIRIUS procura cobrir todos estes aspectos considerados fundamentais para que este tipo de dado seja utilizado com transparência pelo

usuário da aplicação. Nos itens a seguir, serão apresentados os comandos propostos para definição de uma linguagem com as características apontadas.

6.5.3. Especificação da Linguagem de Acesso a Imagens.

Neste tópico serão apresentados os conjuntos de comandos e elementos da sintaxe da linguagem que permitem a manipulação do atributo com característica de imagem, conforme os conceitos contidos na modelagem deste elemento no SIRIUS (Figura 1). A linguagem de acesso a imagens é dividida em duas partes, refletindo as etapas de desenvolvimento e utilização de uma aplicação:

- a) Linguagem de Definição de Imagens (LDI): apresenta os comandos que possibilitam a definição do esquema de uma aplicação, envolvendo a presença de imagens e a descrição de seu conteúdo;
- b) Linguagem de Manipulação de Imagens (LMI): apresenta as construções necessárias à formulação de consultas ao banco de imagens, descrito através da LDI.

6.5.3.1. Linguagem de Definição de Imagens no SIRIUS.

O Esquema de uma aplicação que envolve a presença de imagens como uma das características de atributos de seus tipos de objetos e relacionamentos deve incluir a definição dos elementos envolvidos em sua manipulação. O conjunto de comandos da linguagem de definição de imagens permite a descrição das constantes imagens que sejam significativas, assim como dos sumarizadores e modificadores que possam ser aplicados. O conjunto de sumarizadores suportado consiste de uma

característica pré-definida pelo gerenciador, não possibilitando alterações, a menos que este sofra modificações em sua estrutura interna.

Nos tópicos seguintes serão apresentados os comandos que permitem a construção de esquemas para aplicações, envolvendo a manipulação de imagens através de comandos para criação, associação, alteração e eliminação dos tipos de objetos. Cada construção válida para a linguagem será apresentada na forma de uma sintaxe na qual são demonstrados os seus principais elementos sintáticos, sendo seguido de um exemplo ilustrativo da sua utilização. Todos os exemplos serão desenvolvidos sobre o esquema da aplicação de BD com imagens descrito no item 6.3.2, a menos de que seja indicado o contrário.

6.5.3.1.1. Criação dos Objetos de Descrição do Domínio da Aplicação.

A. Definição do Atributo com Característica de Imagem.

O primeiro passo na direção de se utilizar o conceito de manipulação de imagens por um BD modelado com as construções semânticas do SIRIUS consiste na criação do atributo com característica de imagem em algum tipo de objeto ou atributo com característica de relacionamento do Esquema desta aplicação. Esta operação é realizada através do seguinte comando:

```
CREATE IMAGE <nome_tipo_imagem> ON ( <nome_tipo_objeto> |  
                                     <nome_tipo_objeto>`.`<nome_relacionamento>  
) `;`
```

Descrição: realiza a criação do atributo com característica imagem com o nome <nome_tipo_imagem>, em um objeto do tipo <nome_tipo_objeto> ou em um atributo com característica de relacionamento <nome_relacionamento>

pertencente ao objeto <nome_tipo_objeto>. O atributo imagem é criado na mesma colônia na qual habita o tipo de objeto ou atributo relacionamento ao qual se encontrará associado.

Considerando o banco de dados contendo informações sobre exames clínicos dos pacientes de um determinado Hospital presentes na Figura 2, pode-se indicar a associação dos atributos com característica de imagem ao objeto *Paciente* previamente definido no Esquema, da seguinte forma:

```
CREATE IMAGE TomografiaCranio, ExameTomografia  
ON Paciente;
```

Se para cada consulta realizada pelo Paciente com um determinado Medico seja produzido um exame computadorizado, então pode-se definir atributo TomografiaCranio como um atributo do relacionamento existente entre os objetos. Observe a necessidade de indicar o objeto origem do relacionamento no qual a imagem será associada, uma vez que no SIRIUS o conceito de relacionamento é definido como uma característica de atributo, conforme apresentado no exemplo a seguir.

```
CREATE IMAGE TomografiaCranio ON Paciente.ConsultadoPor;
```

B. Definição das Constantes Imagens.

A etapa seguinte consiste na criação das constantes imagens. A sintaxe para este comando é apresentada a seguir.

```
CREATE [SUB] IMAGECONST <NomeConst> AS <Localizador> `;'
```

Descrição: O parâmetro <NomeConst> representa o nome de identificação do objeto tipo constante imagem a ser utilizado sempre que este for referenciado em algum

outro comando ou expressão da linguagem. Os demais atributos que descrevem características específicas da imagem indicada como constante, ou seja, o objeto imagem: informações sobre o formato, não são especificados explicitamente no comando de criação, uma vez que estes recebem um tratamento direto a partir do gerenciador com este suporte. A instanciação dos valores para este conjunto de atributos ocorre quando da correta interpretação de um comando de criação da constante imagem. A cláusula opcional [SUB] é utilizada para definir que uma constante imagem possa ser localizada no interior de uma imagem do BD. A sua omissão do comando significa que a constante imagem representa uma imagem complexa que será comparada integralmente com as demais existentes no BD.

A cláusula <Localizador> consiste de uma forma genérica de se indicar o local onde se encontra o alvo do comando, no caso, a própria imagem de definição da constante. Pode indicar tanto um arquivo externo, quanto uma imagem que já se encontra armazenada no BD. No primeiro caso a sua sintaxe fica:

<tipo_arquivo> <caminho>

onde, <tipo_arquivo> indica um arquivo externo ao Banco de Dados através de uma das seguintes palavras chaves: FILE, para indicar um arquivo de dados; TEXT, para arquivos de textos; ou BINARY, para arquivos executáveis.

A sintaxe a seguir, consiste na indicação de alguma imagem que se encontra no Banco de Dados:

<Caract_Atributo> <Tipo_Obj> <Nome_Obj> <Atributo>
<Estrutura_Atrib>

onde, <Caract_Atributo> indica a característica do atributo <Atributo> referenciado, através das palavras chaves: NUMBER, PROCEDURE ou IMAGE. Neste caso em particular, a palavra IMAGE será sempre utilizada, uma vez que o atributo será da característica de imagem. Por exemplo, os comandos a seguir criam as constantes imagens Coração a partir do disco, e a constante imagem Pulmão, a partir de uma imagem presente no BD, armazenada como a segunda imagem do atributo Exame_RM_Torax do Paciente João.

```
CREATE IMAGECONST Coracao AS FILE ~image/heart.tiff;  
CREATE IMAGECONST Pulmao AS IMAGE Paciente João  
Exame_RM_Torax[2];
```

A execução do comando no formato anterior implica na criação de constantes imagens que não possam ser localizadas no interior de uma outra imagem. Normalmente, nas aplicações médicas, as estruturas biológicas representadas através das constantes indicadas anteriormente são localizadas como um elemento de composição da imagem de um exame de tomografia. Para que esta característica possa ser suportada, então a cláusula SUB deve ser acrescentada ao comando anterior, ou seja:

```
CREATE SUB IMAGECONST Talamo AS IMAGE Paciente Bob  
RM_Cerebro[5];
```

B. Definição do Conjunto de Sumarizadores.

O conjunto de sumarizadores deve ser previamente desenvolvido em uma linguagem de programação, como um método para o Processamento de Imagens.

```
CREATE SUMMARIZER <nome_summ> AS <Localizador>  
[<restrição>] `;`
```

Descrição: o parâmetro <nome_summ> contém uma lista de identificadores utilizados para se referenciar aos sumarizadores em expressões da linguagem. O <Localizador> indica o método que implementa a operação de processamento de imagens. Este método consiste no atributo do objeto do tipo Operador, cujas instâncias apresentam a implementação das primitivas de processamento de imagens aplicadas na extração de características da imagem.

As instâncias dos objetos do tipo operador, definidas para o sistema, consistem em HISTOGRAMA e MATCHING, os quais representam respectivamente a determinação do histograma de uma imagem e a operação de comparação de padrão. Estes consistem nos tipos assumidos pelos sumarizadores que são atualmente suportados no gerenciador. Os métodos são armazenados como atributo Img_Sumarizador dos objetos Histograma e Matching, respectivamente. No entanto, o conjunto de sumarizadores pode ser facilmente estendido, sendo necessário para isto a incorporação do suporte a um tipo em particular de operação.

No Esquema da aplicação de exames médicos de tomografia do item 6.3.2, é relevante para a aplicação a comparação de padrão da imagem e a distribuição dos níveis de cores. O comando a seguir realiza a criação dos objetos tipo sumarizador

Histo e Padrão:

```
CREATE SUMMARIZER Padrao AS  
    PROCEDURE Operador Matching Img_Sumarizador;  
CREATE SUMMARIZER Histo AS  
    PROCEDURE Operador Histograma Img_Sumarizador;
```

Apesar de ser criado apenas um objeto sumarizador para cada tipo básico suportado pelo gerenciador neste Esquema de aplicação, pode-se criar outros para um

mesmo tipo. Considere que o projetista da aplicação represente um sumarizador busca de padrão adicional, denominado de *BuscaPadrão*:

```
CREATE SUMMARIZER BuscaPadrao AS  
    PROCEDURE Operador Matching Img_Sumarizador;
```

A justificativa, a nível de modelagem, para criar no Esquema mais do que um tipo de sumarizador consiste em possibilitar ao seu projetista a definição de comportamentos distintos, tanto para as características medidas pelo sumarizador (variáveis de descrição), quanto na flexibilidade em sua extração (modificadores).

A cláusula opcional <Restrição> permite ao usuário controlar a execução do método de processamento de imagem, atribuindo uma série de parâmetros, denominados variáveis internas. Além de definir os valores a serem retornados pelos métodos, as variáveis internas constituirão no tipo de dado para cada variável descritora inserida no Esquema da Aplicação. A sintaxe para a especificação das variáveis internas consiste em:

<Tipo_Dado> <Nome_Var_Interna>

O operador Matching é especificado a partir das seguintes variáveis internas:

- INT COORDENADAX: valor da coordenada X, onde foi localizado o padrão no interior de uma imagem.
- INT COORDENADAY: valor da coordenada Y, onde foi localizado o padrão no interior de uma imagem;
- INT SEMELHANÇA: representa o número de pontos semelhantes existentes entre os padrões de comparação.

O tipo de dado para o armazenamento interno dos valores assumidos por variáveis descritoras que sejam de algum dos tipos indicados anteriormente é um número inteiro. No entanto, os tipos definidos para as variáveis descritoras possuem um significado semântico mais preciso, uma vez que permite ao gerenciador validar a sua eventual ligação com algum sumarizador.

Para a descrição do operador Histograma, o gerenciador apresenta suporte para as seguintes variáveis internas:

- INT DISTRIBUIÇÃO[]: representa o vetor contendo a distribuição de cores (ou para imagens médicas, os níveis de tons de cinza) da imagem;
- INT MEDIA: indica o valor médio da distribuição de cores;
- INT DESVIOPADRÃO: indica o desvio padrão apresentado pela distribuição de cores;
- INT VARIÂNCIA: indica o valor da variância da distribuição de cores.

C. Definição das Variáveis Descritoras.

A especificação de um sumarizador não se completa com indicação de quais serão utilizados. O aspecto comportamental do sumarizador é descrito através das variáveis ou parâmetros representando o conjunto de valores retornados por este, quando da aplicação em uma imagem. A definição destas variáveis é realizada através do seguinte comando:

```
CREATE IMAGEPAR <nome_tipo> <lista_variaveis>  
[RANGE <valor> | FROM <valor> TO <valor>] `;`
```

- Descrição: insere um conjunto de objetos do tipo Variável Descritora no Esquema da aplicação identificados por <lista_variaveis>, sendo o domínio dos

valores assumidos pelas variáveis pertencentes ao tipo de dado `<nome_tipo>`. Os tipos de dados válidos para as variáveis descritoras consistem nos que foram definidos internamente pelo gerenciador, através das variáveis internas.

Com o conceito de estruturas de dados suportado pelo SIRIUS, conjuntos de valores, listas, cadeias de caracteres, e outras organizações de dados são construídas sobre cada um dos tipos básicos, permitindo que o tipo de uma variável descritora tenha como domínio algum destes tipos construídos.

A cláusula `RANGE <valor> | FROM <valor> TO <valor>` indica o limite de tolerância para que instâncias do atributo com característica de imagem possam apresentar uma entrada na estrutura de acesso definida para este objeto tipo variável descritora, em associação a um determinado objeto tipo sumarizador. A definição `RANGE <valor>` é utilizada para variáveis, nas quais seja adequado definir o limite de tolerância em função de um intervalo dado pela instância assumida pela variável `+/- o <valor>`. Isto é adequado para variáveis que armazenam o valor da aplicação do sumarizador sobre a constante imagem, como no caso do sumarizador tipo Histograma. A cláusula `FROM <valor> TO <valor>` aplica-se para a definição explícita do intervalo a partir dos valores contidos. Do contrário, esta cláusula é adequada para sumarizadores que armazenam o resultado direto de uma comparação, como, por exemplo, o sumarizador do tipo Matching.

No exemplo indicado, considere a criação dos objetos do tipo Variável Descritora presentes no Esquema.

a) Criação do objeto tipo Variável Descritora *Semelhança*, no qual o intervalo de aceitação de valores é de 60% a 100%:

```
CREATE IMAGEPAR SEMELHANCA Semelhanca FROM 60 TO 100;
```

b) Criação de objetos do tipo Variável Descritora nos quais não é necessário estabelecer um intervalo de aceitação de valores:

```
CREATE IMAGEPAR DISTRIBUICAO Distribuicao;  
CREATE IMAGEPAR COORDENADAX PosicaoX;  
CREATE IMAGEPAR COORDENADAY PosicaoY;
```

c) Criação de objetos do tipo Variável Descritora nos quais o intervalo de validação dos valores seja uma variação em torno de um determinado valor:

```
CREATE IMAGEPAR MEDIA Media RANGE 10,  
                DESVIOPADRAO DesvioPadrao RANGE 50,  
                VARIANCIA Variancia RANGE 50;
```

D. Descrição dos Modificadores dos Sumarizadores.

Outro elemento que complementa a descrição do comportamento de um sumarizador são os modificadores, os quais são definidos a partir de um conjunto pré-definido suportado pelo gerenciador. Especialmente os sumarizadores relacionados com a operação de busca de padrão podem ter o seu comportamento alterado, em função da aplicação dos seguintes modificadores:

a) **ROTAÇÃO**: permite que um dos parâmetros do sumarizador, ao qual o modificador esteja associado, possa aparecer rotacionado em relação ao original, representado pela constante imagem, em um determinado ângulo;

b) **TRANSLAÇÃO**: permite que um dos parâmetros do sumarizador, ao qual o modificador esteja associado, possa aparecer transladado em relação ao original, representado pela constante imagem, por uma determinada diferença de coordenadas ao longo dos eixos;

c) ESCALA: permite que um dos parâmetros do sumariador, ao qual o modificador está associado, possa aparecer em uma escala diferente do original representado pela constante imagem.

Pela própria característica de cada modificador, nota-se que a associação a um determinado sumariador é altamente dependente da semântica destes dois elementos. O modificador ROTAÇÃO só apresenta um significado preciso se este for associado ao sumariador que representa a busca de padrões em imagens (MATCHING).

A sintaxe de definição dos modificadores é mostrada a seguir.

```
CREATE MODIFIER <nome_modificador> ON <nome_sumariador>  
AS ( <restricao_mod> );
```

Descrição: o parâmetro <nome_modificador> indica o nome do objeto tipo Modificador, o qual será subtipo do objeto tipo Sumariador <nome_sumariador>. As características do modificador são indicadas através da cláusula <restricao_mod>, que possui a seguinte forma:

<identificador> <tipo_modificador> <variacao>

onde, <identificador> indica o nome para a instância do modificador criada para o <tipo_modificador>, o qual deve ser um dos tipos básicos suportados descritos anteriormente, ou seja, ROTAÇÃO, TRANSLAÇÃO ou ESCALA. O parâmetro <variacao> representa a variação máxima que cada tipo de modificador pode aceitar, especificada como:

a) RANGE <valor>, para os tipos de modificadores cuja tolerância para a sua aplicação pode ser indicada através da variação sobre um único valor;

b) FROM <valor> TO <valor>, para o tipo de modificador cuja tolerância para a sua aplicação é indicada através de um intervalo delimitado de valores.

O significado deste valor é estabelecido em função da operação representada. Para o caso do modificador ROTAÇÃO pode-se atribuir o valor 30, indicando quantos graus a constante pode aparecer rotacionada na imagem de busca. Note que este valor pode apresentar diferentes tipos de dados, em consequência do tipo de informação manipulada pelo modificador, como observa-se no exemplo a seguir:

```
CREATE MODIFIER Rotacao ON Padrao
      AS (rot ROTACAO RANGE 30);
CREATE MODIFIER Trans ON Padrao
      AS (tr_10 TRANSLACAO RANGE (10,10));
```

onde o valor de inicialização para o modificador TRANSLAÇÃO consiste na variação das coordenadas nos eixos X e Y respectivamente, quando da aplicação do sumarizador *Padrao* sobre uma constante imagem. A definição de uma especialização no comportamento do sumarizador *BuscaPadrão*, considerando apenas o aspecto de rotação de uma constante imagem, é realizada por:

```
CREATE MODIFIER Rotacao ON BuscaPadrao
      AS (rot ROTACAO RANGE 35);
```

Pode-se definir um modificador a partir de uma composição de movimentos, ou seja:

```
CREATE MODIFIER Match_Parcial ON BuscaPadrao
      AS (rot ROTACAO RANGE 35,
          trans TRANSLACAO RANGE (15,15),
          esc ESCALA RANGE 5 );
```

Com a definição dos modificadores, a linguagem de manipulação de imagens, insere elementos que permite alterar o comportamento de um determinado sumarizador dentro dos limites aceitáveis por este. Com a aplicação desta alteração a

atuação de cada sumariizador torna-se mais ampla, no sentido de que não apenas a característica por este representado será extraída de uma imagem no BD, tal qual apareça em uma constante, mas pode-se considerar também suas possíveis variações.

6.5.3.1.2. Associação entre os Objetos do Esquema da Aplicação.

Este conjunto de comandos permite definir os relacionamentos entre os objetos criados anteriormente, de forma a precisar a semântica envolvida na aplicação e caracterizar o conteúdo das imagens presentes no BD.

A. Descrição das Constantes Imagens.

A descrição da constante imagem indicando o conjunto de características a serem utilizadas como elementos de comparação com as imagens inseridas no BD é realizada definindo-se o relacionamento triplo entre os objetos dos tipos Constante Imagem, Sumariizador e Variável Descritora. O comando da linguagem de definição de imagens para definir esta associação é o seguinte:

```
SET COMPARISON <nome_caract> IMAGECONST <nome_const>  
    CHARACTERIZED BY <nome_sumariizador>  
    DESCRIBED BY <lista_variaveis>';'
```

Descrição: o comando descreve o objeto tipo constante imagem <nome_const>, através do sumariizador que possa ser aplicado (<nome_sumariizador>) e das variáveis de descrição que o caracteriza (<lista_variaveis>). A cláusula <nome_caract> permite associar um nome de identificação à característica descrita. O comando apresenta uma parte indicada pela cláusula CHARACTERIZED BY, no qual é criado o relacionamento identificador *Caracteriza/Caracterizado_Por* entre o objeto constante imagem e sumariizador. A outra identificada pela cláusula

DESCRIBED BY define o relacionamento de conjunto *Descrito_Por* entre o sumarizador e as suas variáveis descritoras.

A caracterização do objeto tipo constante imagem *Coração* através dos sumarizadores *Histo* e *BuscaPadrão* é realizada com a seguinte sequência de comandos:

```
SET COMPARISON Histograma_do_Coracao IMAGECONST Coracao
    CHARACTERIZED BY      Histo
    DESCRIBED BY          Media, DesvioPadrao, Variancia,
                        Distribuicao;
```

```
SET COMPARISON Posicao_Coracao IMAGECONST Coracao
    CHARACTERIZED BY      BuscaPadrao
    DESCRIBED BY          Semelhanca, PosicaoX, PosicaoY;
```

Observe que, com o conceito de tipos de sumarizadores, pode-se definir para uma aplicação uma lista de sumarizadores do mesmo tipo, mas com características diferentes com relação a sua descrição. Isto oferece uma maior flexibilidade na definição dos elementos de manipulação de imagens, no sentido de que o aspecto mais importante na comparação entre as imagens seja descrito com precisão, para a necessidade atual da aplicação. Assim o objeto tipo Sumarizador *Histo* descrevendo outra constante imagem através de um conjunto de características distintas é obtido com o seguinte comando:

```
SET COMPARISON Dist_Hipo IMAGECONST Hipotalamo
    CHARACTERIZED BY      Histo
    DESCRIBED BY          Distribuicao;
```

Neste último caso apenas a característica *Distribuicao* é relevante na descrição do sumarizador *Histo*, quando associada à constante imagem *Hipotálamo*.

B. Descrição dos Atributos com Característica de Imagem.

A descrição do conteúdo que se espera encontrar em uma instância do atributo com característica imagem é realizada através de sua associação com as constantes imagens que o descreva. A sintaxe da linguagem para esta finalidade consiste em:

```
SET      IMAGE      <nome_tipo_imagem>      DESCRIBED      BY
<lista_constantes>`;'
```

Descrição: Indica que as imagens que pertencem ao atributo tipo <nome_tipo_imagem> apresentarão o seu conteúdo descrito pelas constantes imagens presentes na lista indicada por <lista_constantes>. As suas informações serão descritas em função do conjunto de associações definidas para cada objeto tipo constante imagem presente na lista.

Como exemplo, considere a descrição do conteúdo dos exames modelados no Esquema (Figura 4) através dos atributos *ExameTomografia* e *TomografiaCrânio*:

```
SET IMAGE ExameTomografia DESCRIBED BY Coracao, Pulmao,
Medula;
SET IMAGE TomografiaCranio DESCRIBED BY Hipotalamo, Talamo;
```

Com a execução deste comando, fica mais precisa a semântica da informação contida em cada imagem, ou seja, as instâncias do atributo com característica de imagem *ExameTomografia* serão descritas em função dos objetos constante imagem *Coração*, *Pulmão* e *Medula*, enquanto que *TomografiaCrânio* será descrito pelas características presentes nos objetos *Hipotálamo* e *Tálamo*.

6.5.3.1.3. Alteração da Descrição dos Objetos no Esquema.

Após a criação do conjunto de tipos de objetos e relacionamentos que formam o Esquema da aplicação com imagens, estes podem apresentar a sua definição alterada, através, basicamente, da remoção das associações existentes entre elementos que os caracterizam. Portanto, este conjunto de comandos são aplicados para remover relacionamentos entre os objetos do esquema, o que significa a eliminação de características semânticas na descrição de uma determinada informação presente na imagem. A inclusão de novos elementos no Esquema pode ser realizada através dos comandos de criação apresentados na seção anterior, conforme será esclarecido a seguir.

A. Alterar a descrição do Atributo com Característica de Imagem.

Os tipos de atributos com característica de imagem que foram definidos para o esquema da aplicação podem apresentar a sua definição sobre o conteúdo gráfico da informação alterados, através da eliminação de constantes imagens.

```
ALTER IMAGE <nome_tipo_imagem>  
      DROP IMAGECONST <nome_const> ';' 
```

Descrição: as instâncias do atributo com característica de imagem <nome_tipo_imagem> não serão descritas pela informação contida no objeto constante imagem <nome_const>. A exclusão desta associação, implica a eliminação de todas as ocorrências das instâncias deste atributo na estrutura de acesso da constante imagem.

Como exemplo, considere que a partir de um determinado instante de utilização do BD, não seja mais significativo a definição da constante imagem

Medula para descrever o conteúdo do *ExameTomografia* dos pacientes. Para a validação desta restrição, deve-se eliminar a associação existente:

```
ALTER IMAGE ExameTomografia DROP IMAGECONST Medula;
```

A inserção de uma nova constante imagem a um determinado tipo de atributo pode ser realizada criando-se uma nova associação, ou seja, utilizando-se os comandos de definição de Esquema. Para a aplicação, considere que, após utilizar o BD, seja inserida uma nova alteração no Esquema, de forma que o *ExameTomografia* possa ser descrito através das características presentes em uma imagem apresentando um câncer na medula. Antes de criar esta nova associação, deve inserir a descrição das características desta constante imagem, que é realizada com a seguinte sequência de comandos:

```
CREATE IMAGECONST CancerMedula;  
SET IMAGECONST CancerMedula CHARACTERIZED BY Padrao  
DESCRIBED BY Semelhanca;
```

Uma vez definida a constante imagem e suas características, pode-se criar a associação indicada, através do comando a seguir:

```
SET IMAGE ExameTomografia DESCRIBED BY CancerMedula;
```

Este comando indica ao gerenciador que a partir deste instante a constante imagem *CancerMedula* descreve o conteúdo das instâncias do atributo *ExameTomografia*.

B. Alterar a descrição das Constantes Imagens.

A alteração da descrição de uma constante imagem ocorre sob dois aspectos considerando os objetos associados através do relacionamento triplo, ou seja, sumariizador e variáveis descritoras, realizando-se a eliminação de suas associações.

Apenas para o relacionamento *Descrito_Por* pode-se inserir uma nova instância, significando a presença de uma novo aspecto do sumariizador para a descrição da constante imagem.

```
(i) ALTER IMAGECONST <nome_const> CHARACTERIZED BY
<NomeSumm>
      ( ADD IMAGEPAR <lista_var_descritora> |
        DROP IMAGEPAR <lista_var_descritora> ) `;`
(ii) ALTER IMAGECONST <nome_const> DROP SUMMARIZER
<NomeSumm> `;`
```

Descrição: com o primeiro comando (i) realiza-se a inserção, através da cláusula ADD, ou remoção (DROP) dos objetos tipo variável descritora indicados em <lista_var_descritora> do relacionamento triplo identificado pela constante imagem <nome_const> e sumariizador <nome_op>. O comando (ii) permite que seja eliminado o relacionamento identificador da tripla formado pela constante imagem <nome_const> com o sumariizador <NomeSumm>; como consequência, a instância do relacionamento triplo é eliminado.

A remoção de um sumariizador da descrição da constante imagem elimina um aspecto sob o qual um conjunto de imagens poderia ser descrito. Com esta operação as estruturas de acesso presentes para cada variável de descrição do sumariizador para a constante imagem que se encontrava associado são eliminadas do BD, não possibilitando desta forma a construção de expressões de consulta que envolvam esta descrição da imagem. Note que este sumariizador pode constituir em um elemento de extração de informação para outra constante imagem e, conseqüentemente, possuir a sua própria estrutura de acesso. No entanto, esta não será eliminada, uma vez que foi removida a conexão do sumariizador com outra constante. Apenas quando o sumariizador for eliminado do BD, é que as suas estruturas de acesso serão totalmente

excluídas. Consideração semelhante é válida quando se elimina um relacionamento de conjunto a partir de um objeto do tipo variável descritora. A estrutura de acesso definida em consequência desta associação é eliminada do BD.

Considere a eliminação das variáveis de descrição *DesvioPadrão* e *Variancia* do sumariizador *Histo* associado à constante imagem *Coração*:

```
ALTER IMAGECONST Coracao CHARACTERIZED BY Histo
DROP IMAGEPAR DesvioPadrao, Variancia;
```

Para eliminar a caracterização realizada pelo sumariizador *BuscaPadrão* da constante imagem *Coração*, o seguinte comando deve ser avaliado:

```
ALTER IMAGECONST Coracao DROP SUMMARIZER BuscaPadrao;
```

Uma vez que o comando anterior elimina o relacionamento de identificação, os relacionamentos de conjunto existente com os objetos *Semelhança*, *PosiçãoX* e *PosiçãoY* são também eliminados, assim como as estruturas de acesso.

A inserção de uma nova variável descritora ou de um novo sumariizador a uma constante imagem significa que, a partir deste instante, tem-se um aspecto adicional a ser considerado para a avaliação das futuras imagens. As instâncias dos atributos com característica de imagem, que se encontram no BD descritos por esta constante, não serão reavaliadas com relação a esta nova característica. Caso esta seja a necessidade da aplicação em seu estado atual, um conjunto de instâncias que possam apresentar a constante imagem alterada deve ser selecionado para se obter uma outra avaliação segundo o novo sumariizador.

Para inserir um novo aspecto ao sumariizador *Histo* associado à constante imagem *Hipotálamo*, utiliza-se o comando a seguir:

```
ALTER IMAGECONST Hipotalamo CHARACTERIZED BY Histo
ADD IMAGEPAR Media;
```

A inserção de um novo sumariizador é realizada através dos comandos de definição do Esquema da aplicação, ou seja, definir o sumariizador *BuscaPadrão* válido para a constante *Hipotálamo*, da seguinte forma;

```
SET IMAGECONST Hipotalamo CHARACTERIZED BY BuscaPadrao  
DESCRIBED BY Semelhanca;
```

6.5.3.1.4. Eliminação dos Objetos no Esquema.

A linguagem proposta para a consulta e tratamento de imagens permite que o Esquema definido para uma aplicação seja alterado após a sua definição inicial. Estas alterações podem ser realizadas através dos comandos descritos na seção anterior, ou simplesmente eliminando a presença de alguns elementos de caracterização da aplicação. Os comandos para esta operação são semelhantes, diferenciando no comportamento apresentado, de acordo com o tipo de cada objeto a ser aplicado. O formato geral consiste em:

```
DROP <Nome_Tipo_Objeto> <Nome_Objeto> `;`
```

A. Eliminar uma Variável Descritora.

```
DROP IMAGEPAR <nome_var> `;`
```

Descrição: elimina a variável <nome_var> do Esquema da aplicação e, conseqüentemente, todas as instâncias de relacionamentos de conjunto do qual esta esteja participando. Uma vez que os relacionamentos são eliminados, as estruturas de acesso existentes são também extraídas do BD.

B. Eliminar um Sumarizador.

```
DROP SUMMARIZER <NomeSumm> `;`
```

Descrição: elimina o sumarizador <NomeSumm> do esquema da aplicação e todas as suas associações existentes com as constantes imagens e as variáveis de descrição.

De outra forma, como o objeto do tipo sumariizador participa do relacionamento identificador da tripla, todos os relacionamentos de conjunto são também eliminados. Isto significa que as estruturas de acesso, criadas em consequência da existência destes relacionamentos, são eliminadas. Se para este sumariizador foi definido algum subtipo, ou seja, um modificador, todos estes serão excluídos do Esquema.

Como exemplo, o comando a seguir realiza a eliminação do sumariizador *BuscaPadrão* do Esquema, e como consequência imediata, o modificador *Rotação* definido como seu subtipo:

```
DROP SUMMARIZER BuscaPadrao;
```

C. Eliminar um Modificador.

```
DROP MODIFIER <nome_modificador> ON <nome_sumarizador> `;'
```

Descrição: elimina a definição do modificador <nome_modificador> do objeto tipo sumariizador <nome_sumarizador> presente no Esquema da aplicação.

D. Eliminar uma Constante Imagem.

```
DROP IMAGECONST <nome_const>;
```

Descrição: através deste comando a constante identificada com o nome <nome_const> é eliminada do Esquema. Esta operação implica em eliminar a imagem que representa a constante do conteúdo do BD. De forma semelhante à eliminação do objeto do tipo Sumariizador, uma constante imagem participa do relacionamento identificador da tripla, implicando na eliminação dos relacionamentos de conjunto desta instância com as variáveis descritoras, e das estruturas de acesso. O relacionamento apresentado pela constante imagem de descrição do atributo com característica de imagem é também removido. Isto

significa que, as imagens descritas em função desta constante passam a não apresentá-la como um elemento de representação do seu conteúdo.

E. Eliminar um Tipo de Atributo com Característica de Imagem.

```
DROP IMAGE <nome_img> `;`
```

Descrição: a eliminação de um Tipo de atributo com característica de imagem do Esquema, provoca a remoção de todas as instâncias de imagens deste Tipo da Base. Além disso, as associações existentes com as constantes imagens da aplicação são eliminadas.

Por exemplo, eliminar a existência do atributo *ExameTomografia* é obtida com:

```
DROP IMAGE ExameTomografia;
```

6.5.3.1.5. Definição de uma Instância para um Tipo de Atributo com Característica de Imagem.

Conforme os aspectos de modelagem descritos para o atributo com característica de imagem no SIRIUS, este possui um conjunto de atributos que apresentam dois níveis de instanciação. O primeiro nível consiste na indicação do nome do tipo do atributo imagem e sua associação com algum outro tipo de objeto do esquema da aplicação. Esta definição é realizada através de comandos da linguagem de Definição de Imagens, com a criação e descrição dos tipos de dados que sejam apropriados. O nível seguinte consiste na definição de uma instância para este tipo, ou seja, na inserção de uma nova imagem no banco de dados. Para esta operação utiliza-se o comando a seguir:

```
INSERT INTO <Nome_Atr_Imagem>  
    [ON ( <nome_instancia_obj> )]  
VALUES <nome_arquivo_imagem> `;`
```


Descrição: realiza-se a inserção de uma instância para o atributo com característica de imagem denominado por <Nome_Atr_Imagem>, no objeto <nome_instancia_obj>, instância de um tipo de objeto descrito no Esquema da aplicação por este tipo de atributo. A instância do atributo com a característica de imagem se encontra no arquivo do sistema indicado por <nome_arquivo_imagem>.

Como exemplo, considere a inserção de um novo exame de tomografia para o Paciente “José da Silva”, o qual foi concluído e se encontra no arquivo do sistema denominado de “\local\exames\ex01tm08.pac”:

```
INSERT INTO ExameTomografia ON "José da Silva"  
VALUES "\local\exames\ex01tm08.pac"
```

Este consiste no comando que uma vez executado irá utilizar-se do Esquema da Aplicação para ativar os métodos apropriados à descrição do conteúdo da imagem. O exemplo anterior, consiste em obter as constantes imagens para que as primitivas de definição do comportamento de cada sumariizador que se encontra associado sejam aplicadas sobre a nova instância da imagem inserida no Banco.

6.5.3.2. Linguagem de Manipulação de Imagens (LMI).

O Banco de Imagens pode sofrer uma série de operações de acesso ao seu conteúdo, que consistem em consultas, inserções ou remoções de imagens, ou aplicar operações específicas do processamento de imagem com o objetivo de melhorar sua qualidade ou realçar algum detalhe. A linguagem de acesso às imagens oferece um conjunto de construções sintáticas que possibilita expressar em alto nível o interesse do usuário com relação ao conteúdo do BD.

6.5.3.2.1. Definição do Contexto para os Sumarizadores.

Um aspecto importante na elaboração de consultas ao BD consiste na definição do contexto no qual a expressão de condição será válida. O contexto de validação da operação pode ser indicado para cada elemento que a defina. Desta forma o conjunto de parâmetros ou variáveis de descrição dos sumarizadores podem ter os seus valores de validação estabelecidos como contexto de uma determinada consulta. O mesmo se aplica aos modificadores, quando estes forem definidos para o sumarizador em questão. Assim, uma mesma consulta pode apresentar resultados diferentes, se esta for avaliada em contextos distintos.

A definição do contexto envolve portanto dois aspectos: a) Variáveis descritoras, indicando os valores que possam se válidos, e b) Modificadores: indicando quais devem ser considerados juntamente com seu nível de tolerância.

A sintaxe para a definição do contexto para as variáveis descritoras consiste em:

```
ACCEPT IMAGEPAR <nome_variavel>  
    [RANGE <valor> | FROM <valor> TO <valor>]  
    ON <nome_constante> CHARACTERIZED BY <nome_sumarizador>
```

Descrição: o contexto será definido para a variável <nome_variavel>, cujos valores serão considerados válidos em uma condição da consulta, se estiverem na variação definida pela cláusula RANGE <valor> ou no intervalo indicado por FROM <valor> TO <valor>. A definição deste contexto se aplica à variável <nome_variavel>, cuja instância é identificada pelo relacionamento <nome_constante> descrito por <nome_sumarizador>.

Para definir o contexto de validação sobre a variável descritora *Semelhança*, para o sumariizador *Padrão* associado à constante imagem *Coração*, em um intervalo de 80% a 100%, deve ser escrito o seguinte comando:

```
ACCEPT IMAGEPAR Semelhanca FROM 80 TO 100
      ON Coracao CHARACTERIZED BY Padrao;
```

Com este comando significa que, em sendo utilizada a variável descritora *Semelhança* em uma expressão de validação da consulta, esta condição será verdadeira apenas para as instâncias dos atributos com característica de imagem que apresentar um valor para este aspecto, que se encontre no intervalo indicado pelo contexto.

Outro contexto a ser definido se refere aos modificadores de cada sumariizador, realizado pelo comando a seguir:

```
ACCEPT MODIFIER <nome_modificador>
      [RANGE <valor> | FROM <valor> TO
<valor>]
      ON <nome_sumarizador> `;`
```

Descrição: o contexto é definido para o modificador <nome_modificador>, cuja avaliação será considerada verdadeira, em uma determinada consulta, para as instâncias dos atributos com característica imagem descrita pelo sumariizador <nome_sumarizador> (através de uma constante imagem) que apresentem valores na abrangência definida pela cláusula RANGE <valor>, ou no intervalo indicado por FROM <valor> TO <valor>.

Considere a definição do contexto para o modificador do tipo ROTACIONA criado como especialização do objeto *BuscaPadrão*, de forma a ser válido apenas o reconhecimento de uma determinada constante imagem que apareça rotacionada por um valor máximo de 15 graus. Esta especificação é realizada pelo seguinte comando:

ACCEPT MODIFIER Rotacao RANGE 15 ON BuscaPadrao;

Uma vez estabelecido este contexto para o modificador, implica que em cada ocorrência deste na condição de uma consulta ao BD, o mesmo será avaliado como verdadeiro apenas para as imagens nas quais o valor para o modificador *Rotação* seja menor que 15 graus.

A definição do contexto permanecerá ativa até que seja realizada uma nova especificação sobre o elemento considerado. O nível de aceitação (contexto) dos elementos de descrição do sumariador é importante para se definir o mecanismo de validação do resultado a ser apresentado após a realização de uma determinada operação. O sumariador de comparação que estará presente nesta operação será validado de acordo com o contexto estabelecido.

6.5.3.2.2. Definição da Imagem Padrão.

Todas as imagens presentes no BD são descritas através de duas categorias de atributos, a saber, os textuais com informações genéricas sobre as características da imagem, e os gráficos, que são os sumariadores que representam uma descrição gráfica do conteúdo da imagem.

A descrição da imagem a ser recuperada é obtida com a indicação das características relevantes à sua especialização dentro do conjunto existente no BD. A especificação do conteúdo gráfico é realizada através das constantes imagens, sumariadores e variáveis descritoras que representam as características presentes na imagem. Os vários aspectos que descrevem uma imagem são identificados na expressão da condição de formação do resultado da consulta.

A sintaxe geral de um comando de recuperação e acesso a imagens no Banco segue o padrão utilizado na linguagem de consulta SQL (“Structured Query Language”), com alterações nos elementos utilizados para referenciar dados no Banco, de forma a permitir o suporte aos conceitos de orientação a objetos. Portanto, uma consulta na linguagem de acesso a imagens apresenta as seguintes cláusulas:

```
SELECT      <lista_de_atributos>
FROM        <lista_de_objetos>
[ WHERE     <condição> ]';'
```

onde:

- <lista_de_atributos> é uma lista contendo os nomes de atributos pertencentes às características que permitem a sua visualização. No SIRIUS, todas as características de atributos podem ter o seu valor exibido em algum meio de saída, com exceção realizada aos atributos com característica de tempo e estruturas de dados. Nesta lista os atributos podem ser identificados diretamente através de seus nomes, significando que consiste de um atributo do objeto do tipo indicado na cláusula FROM, referenciados através de uma expressão da forma *ob.Atr*, sendo *ob* a identificação de uma instância do tipo de objeto, e *Atr* o seu atributo. Outro elemento válido na lista de atributos consiste simplesmente na indicação do identificador do objeto, significando que o resultado da consulta consistirá no código de identificação interna do objeto, tornando-o disponível às próximas operações que sejam realizadas.
- <lista_de_objetos> é uma lista contendo os tipos de objetos dos quais são retirados os valores dos atributos ou as instâncias dos objetos resultantes da consulta.

- <condição> a qual consiste de uma fórmula bem formada em que os valores dos atributos podem ser combinados através de conectivos lógicos (AND, OR e NOT) e com a presença de operadores de comparação (<, >, =, !=). Nestas expressões é permitida a formulação de caminhos de navegação através das instâncias do BD, elaborada como *ob.N*, sendo *N* um nome de atributo ou tipo de objeto, ou uma outra expressão de navegação com a forma *ob.N*.

No restante deste item serão apresentados uma série de exemplos de consulta a um Banco de Imagens. Para a formulação de cada expressão de consulta será considerado o Esquema da aplicação de exames tomografia apresentado na Figura 2.

A. Consultas Básicas.

Para ilustrar os aspectos básicos da linguagem de consulta ao Banco de Imagens e o significado associado de cada comando, será apresentada algumas consultas a seguir.

Consulta 1: Recuperar o nome de todos os pacientes cadastrados no BD.

```
SELECT P.Nome FROM PACIENTE P;
```

A cláusula `SELECT` indica que o resultado da consulta consiste no valor apresentado pelo atributo *nome*, presente em cada instância de objeto *P*. O tipo do objeto *P* é indicado na cláusula `FROM`, definindo-o como sendo do tipo *Paciente*. Esta consiste em uma consulta básica, uma vez que a informação a ser recuperada é um atributo obtido a partir de uma busca em um único tipo de objeto.

Para recuperar informações que estejam presentes em outros objetos, a expressão da condição pode ser definida através de uma navegação pelos tipos de objetos do Esquema, até localizar a informação desejada. A representação ocorre de

forma semelhante à determinação dos atributos, ou seja, com o símbolo “.” entre os dois tipos de objeto. Para que a navegação seja possível é necessário que exista um atributo relacionamento entre os objetos que definem a semântica desta ligação. Esta necessidade é indicada na consulta a seguir.

Consulta 2: Recuperar o nome de todos os pacientes atendidos pelo médico “Dr. Pedro”.

```
SELECT P.Nome FROM PACIENTE P
WHERE P.ConsultadoPor.Nome = "Dr.Pedro";
```

Esta consulta apresenta a sua formulação semelhante à anterior, apenas com a inclusão da cláusula WHERE. A condição da consulta consiste em uma operação de comparação entre os valores assumidos pelo atributo *Nome* do objeto *Médico*, o qual é identificado a partir do atributo relacionamento *AtendidoPor* apresentado pelo objeto *Paciente*. Como a partir da modelagem o relacionamento possui multiplicidade N:M, uma instância do objeto *Paciente* pode se apresentar associada a várias instâncias do objeto *Médico*, a condição de comparação será avaliada para cada uma destas. Para o objeto do tipo *Médico* que venha tornar a condição verdadeira, será retornado o valor do atributo *Nome* do objeto *Paciente* relacionado com esta instância.

B. Consultas para Verificação de Instâncias.

Um usuário da aplicação pode formular consultas que envolvam apenas a verificação da existência de valores para um determinado atributo, ou, de outra forma, se há instância para objetos de um tipo do Esquema. Para estas consultas define-se o operador HASVALUE, o qual aplicado a um tipo de atributo ou objeto

resulta em um valor verdadeiro se este apresentar um valor ou existir uma instância para este tipo. Assim considere a consulta descrita no próximo exemplo.

Consulta 3: Recupere o nome de todos os pacientes que tenham realizado exames de tomografia por RN.

```
SELECT P.Nome FROM PACIENTE P
WHERE HASVALUE P.ExameTomografia;
```

Esta consulta apresentará como resultado os valores do atributo *Nome* para os objetos do tipo *Paciente* que possuírem valores para outro atributo, ou seja, *ExameTomografia*. O operador *HASVALUE* se aplica sobre o atributo *ExameTomografia*, sendo válido quando este apresentar algum valor. Este operador pode ser utilizado para verificar valores ou a existência de instâncias de atributos em uma expressão de navegação pelo esquema do BD.

Consulta 4: Recupere todos os pacientes que tenham realizado exames de tomografia nos quais possam aparecer o coração.

```
SELECT P.Nome FROM PACIENTE P
WHERE HASVALUE P.ExameTomografia.Coracao
```

O objetivo da consulta consiste em determinar se os exames de tomografia realizados pelos pacientes do hospital podem apresentar o coração. De outra forma, isto significa verificar se o atributo com característica de imagem *ExameTomografia* é descrito pela constante imagem *Coração* neste domínio da aplicação. Esta consulta será válida para todos os objetos do tipo *Paciente* no BD, uma vez que a consideração anterior é verdadeira a partir do esquema. O mesmo não seria verdadeiro se a consulta fosse realizada considerando a descrição do atributo *TomografiaCrânio*, uma vez que não se espera encontrar no conteúdo destes exames uma imagem do coração. Além disso, pode-se observar que a expressão de

navegação pelo esquema do BD foi formulada de uma forma diferente, uma vez que o nome do tipo do objeto destino é que se encontra mencionado, ao contrário do atributo tipo relacionamento existente entre eles. Esta formulação é possível uma vez que, entre os objetos que formam o esquema de descrição do conteúdo gráfico de uma imagem, existe um único tipo de relacionamento, e ao mesmo tempo que a expressão resultante é mais significativa em relação ao fato que representa.

A expressão de verificação de instâncias de objetos pode aparecer combinada com outras, definindo uma condição mais restrita à formação do resultado da consulta, como no exemplo a seguir.

Consulta 5: Recupere todos os pacientes que tenham realizado exames de tomografia do coração e que sejam maiores de 40 anos.

```
SELECT P.Nome FROM PACIENTE P
WHERE HASVALUE P.ExameTomografia.Coracao
      AND P.Idade > 40;
```

Este tipo de consulta consiste em uma verificação do Esquema, através do conteúdo esperado para cada atributo com característica de imagem. A verificação da presença de uma determinada característica em uma instância desse atributo é realizada através dos modelos de consultas apresentadas a seguir.

C. Descrição do Conteúdo da Imagem.

A descrição do conteúdo de uma instância do atributo com característica de imagem é realizada utilizando-se os elementos existentes no Esquema que realizam este tratamento semântico. Uma vez que é definida através de tipos de objetos e relacionamentos, a descrição do conteúdo de uma imagem será apresentada como uma expressão de navegação pelo BD, na condição de validação da consulta.

Consulta 6: Recupere todos os pacientes que realizaram exames de tomografia

nos quais apareçam imagens com o formato do Coração e apresentem idade superior a 40 anos.

```
SELECT P.Nome FROM PACIENTE P
WHERE HASVALUE P.ExameTomografia.Coracao.Padrao
      AND P.Idade > 40;
```

O resultado da consulta é validado a partir da descrição de uma imagem que contém um exame de tomografia no qual seja possível identificar o coração, representado pela constante imagem *Coração*, e que seja semelhante a esta, ou a característica de comparação consiste na verificação dos padrões, modelado pelo objeto *BuscaPadrão*. A presença do operador HASVALUE é necessária, uma vez que não foi mencionado o critério através do qual possa ser definido com precisão a presença de uma imagem com o formato do coração em um determinado exame.

Caso seja relevante a apresentação da imagem representando o exame de tomografia como resultado da consulta, é suficiente acrescentar este atributo como um elemento na lista de seleção. Uma vez que cada instância do atributo com característica de imagem possui os seus próprios atributos, dentre eles o objeto imagem, para efeito de sua utilização na linguagem de consulta pode-se identificá-los com o uso da expressão de navegação. Portanto, na consulta anterior para retornar a imagem do exame realizado esta deve ser alterada como demonstrado a seguir.

Consulta 7: Recupere todos os pacientes com idade superior a 40 anos e seus exames de tomografia nos quais apareçam imagens com o formato do Coração.

```
SELECT P.Nome, P.ExameTomografia.imagem FROM PACIENTE P
WHERE HASVALUE P.ExameTomografia.Coracao.Padrao
      AND P.Idade > 40;
```

Pode-se estabelecer condições sobre os atributos de cada instância do atributo com característica imagem, conforme a consulta a seguir.

Consulta 8: Recupere todos os exames de tomografia de todos os pacientes, que tenham sido realizados no mês de janeiro de 1997, em que tenha sido detectada a presença de um tumor no coração em pacientes com idade entre 40 e 50 anos.

```
SELECT P.Nome, P.ExameTomografia.imagem FROM PACIENTE P
WHERE HASVALUE P.ExameTomografia.Coracao.Padrao
      AND P.ExameTomografia.DataRealizacao >= "01-jan-97"
      AND P.ExameTomografia.DataRealizacao <= "31-jan-97"
      AND P.Idade > 40 AND P.Idade < 50;
```

O atributo *DataRealização* caracteriza cada instância contendo a imagem com o exame de tomografia, indicando a sua data de realização.

D. Consultas baseadas em Contexto.

Nos exemplos anteriores era necessário a presença do operador HASVALUE na condição da consulta como consequência de não ser indicado o aspecto de cada sumariador a ser utilizado para a comparação. Para que isto não ocorra, deve-se utilizar as variáveis descritoras e, neste caso, definir o contexto de validação para a condição presente na consulta. Assim, considere o exemplo a seguir.

Consulta 9: Recupere todos os pacientes e seus exames de tomografia, nos quais apareçam imagens com uma semelhança no formato de no mínimo 80% com o Coração.

```
ACCEPT Semelhanca FROM 80 TO 100
      ON Coracao CHARACTERIZED BY Padrao;
SELECT P.Nome, P.ExameTomografia.imagem FROM Paciente P
WHERE P.ExameTomografia.Coracao.Padrao.Semelhanca;
```

Neste caso, a formulação da consulta menciona explicitamente que, além da comparação de padrão entre a informação representada pela constante imagem com

as imagens do BD, este deve ser realizado considerando a *semelhança* existente entre elas, o que implica diretamente na utilização das variáveis descritoras. Assim, antes da consulta ser submetida para avaliação, foi definido o contexto de validação do seu resultado, para o objeto tipo variável descritora *Semelhança* realizada pelo comando ACCEPT. Com esta definição, apenas as imagens que apresentarem um valor para o aspecto *semelhança*, da característica busca de padrão, da constante *imagem coração*, no intervalo indicado, irão pertencer ao resultado final. Destes objetos, são requisitados apenas os atributos *Nome* e a *imagem* do *ExameTomografia* realizado pelo *Paciente*. A expressão presente na condição da consulta em conjunto com o contexto definido representa uma descrição sintaticamente precisa do conteúdo da imagem que se deseja recuperar a partir do BD. A interpretação semântica é obtida a partir do significado associado a cada elemento utilizado pelo modelo na caracterização das imagens, e principalmente pelo objeto *imagem* que define o conteúdo da própria constante *imagem* utilizada como base de comparação. A consulta realizada anteriormente se refere à interpretação do conteúdo da *imagem* representada como um valor para o objeto tipo constante *imagem Coração*.

Conforme mencionado anteriormente pode-se definir o contexto para os Modificadores, no caso dos objetos do tipo Sumarizador que admitem especialização em seu comportamento.

Consulta 10: Recupere o nome de cada médico cardiologista e dos seus pacientes, com idade superior a 25 anos, que apresentaram uma patologia semelhante em no mínimo de 80% à indicada no exame de tomografia do Coração, admitindo alterações em sua apresentação.

A definição de contexto para esta consulta deve apresentar, além da formalização do intervalo de aceitação da variável descritora, a especificação para o modificador *Rotação*, definido como um subtipo do sumarizador *BuscaPadrão*, que consiste na forma de comparação a ser realizada para as imagens do BD. Este contexto é especificado como se segue:

```
ACCEPT IMAGEDVAR Semelhanca FROM 80 TO 100
      ON Coracao CHARACTERIZED BY Padrao;
ACCEPT MODIFIER Rotacao RANGE 15
      ON BuscaPadrao;
```

Após a execução destes comandos contendo a preparação do contexto, a consulta pode ser submetida.

```
SELECT M.Nome, M.Atende.Nome, M.Atende.ExameTomografia.imagem
FROM Medico M
WHERE M.Consulta.idade > 25 AND
      M.Consulta.ExameTomografia.Coracao.Padrao.Semelhanca
      AND
      M.Especialidade = "Cardiologista"
```

Nesta formulação, o ponto de início do processo de navegação ocorre a partir do objeto *Médico* responsável por atender os pacientes com as características descritas para o conteúdo de seus exames de tomografia. Esta consulta pode ser escrita de forma alternativa considerando a especialização definida para o objeto *BuscaPadrão*, o que implica em substituir a referência a este objeto pelo modificador *Rotação*:

```
SELECT M.Nome, M.Atende.Nome, M.Atende.ExameTomografia.imagem
FROM Medico M
WHERE M.Consulta.idade > 25 AND
      M.Consulta.ExameTomografia.Coracao.Rotacao.Semelhanca
      AND M.Especialidade = "Cardiologista"
```

E. Consultas com o Operador de Junção.

Nos exemplos anteriores as imagens a serem recuperadas são comparadas com as características extraídas das constantes imagens definidas para o Esquema da Aplicação. No entanto, em algumas aplicações pode-se formular consultas nas quais a imagem fornecida como parâmetro de comparação consiste de outra além das constantes imagens. Um exemplo consiste na consulta a seguir:

Consulta 11: Recupere todos os pacientes com uma patologia no tórax semelhante à apresentada pelo paciente “Pedro”.

No tipo de consulta formulado anteriormente, o critério para a validação de seu resultado consiste em um conjunto de imagens que seja semelhante à do exame de tomografia do paciente “Pedro”. No entanto, a partir da definição do atributo com característica de imagem, a delimitação do domínio de uma aplicação ocorre através de suas constantes imagens e de todos os objetos que se encontram associados. Isto significa que as comparações serão realizadas considerando as informações presentes no BD, que descrevem o conteúdo da imagem com o exame de tomografia do paciente “Pedro”. Para o esquema exemplo da aplicação este atributo é descrito pelos objetos constantes imagens *Coração*, *Pulmão* e *Medula*, e estes por suas variáveis descritoras e modificadores. Os valores que esta instância do atributo *ExameTomografia* apresenta nas estruturas de acesso consiste na base de comparação com as demais imagens no BD. Portanto, o resultado da consulta será composto por todas as imagens que apresentarem uma descrição semelhante à da instância do atributo *ExameTomografia* do objeto “Pedro”, com relação às características descritas no Esquema. Ou seja, se o exame de tomografia do paciente “Pedro”

apresenta uma patologia semelhante (80%) à presente na constante imagem coração, e outro exame de tomografia do BD também o possui, então ambos os exames são semelhantes, com relação à constante imagem coração.

Formulando a consulta em comandos da linguagem de manipulação de imagens, tem-se:

```
SELECT P.Nome FROM PACIENTE P, S
WHERE S.Nome = "Pedro" AND
      P.ExameTomografia = S.ExameTomografia;
```

Se a consulta mencionar um aspecto mais restritivo, representado pela presença de alguma constante imagem, como, por exemplo, o *Coração*, pode-se escrever a consulta anterior assim:

```
SELECT P.Nome FROM PACIENTE P, S
WHERE S.Nome = "Pedro" AND
      P.ExameTomografia.Coracao = S.ExameTomografia.Coracao;
```

Esta estratégia de avaliação da consulta pode apresentar resultados que sejam insatisfatórios, uma vez que as imagens são comparadas indiretamente entre si, através dos objetos de delimitação do domínio da aplicação, ou seja, constantes imagens, variáveis descritoras e modificadores. Para o que resultado possa ser mais efetivo, considerando-se a informação alvo da consulta, a imagem que será o elemento de comparação, no exemplo da consulta 11 uma instância do *ExameTomografia*, deveria ser re-indexada como uma constante imagem para esta aplicação. Assim sendo, todas as demais imagens do BD apresentarão o seu conteúdo descrito em função da informação presente na instância do atributo *ExameTomografia*, tornando as consultas formuladas anteriormente mais eficientes e precisas. No entanto, o tempo necessário à operação de reindexação e a provável taxa

de comparação com diferentes imagens elevada, torna adequado o recurso simbolizado pela consulta 11.

6.6. ÁLGEBRA PARA A MANIPULAÇÃO DE BANCO DE IMAGENS.

A interpretação semântica da modelagem do atributo com característica imagem, apresentada na Figura 1 é de fundamental importância para definir o significado de cada comando descrito através da linguagem de manipulação. Nesta modelagem foram utilizados os conceitos típicos de Orientação a Objetos, na qual a informação que define o conteúdo gráfico de uma imagem é representada explicitamente através de associações com os elementos (constantes imagens) que caracterizam cada domínio de aplicação da imagem, e definição dos seus aspectos gráficos de descrição (sumarizadores e variáveis descritoras). Todos estes elementos foram modelados através de tipos de objetos presentes no esquema da aplicação, proporcionando assim uma uniformidade em sua representação.

A modelagem com o conceito de objetos proporciona a descrição dos conceitos através de qualquer linguagem com suporte à manipulação de objetos. Assim, a formalização da linguagem de acesso a imagem com a definição de uma álgebra de imagens não é necessária, uma vez que é suficiente a existência de operadores para a manipulação dos objetos no Banco. Como demonstração deste aspecto, a proposta do cálculo de predicados orientado a objetos (OOPC) [Bertino_92] apresenta uma série de operadores e construções semânticas que permite definir expressões lógicas com o objetivo de representar o comportamento de uma linguagem de consulta a um BD Orientado a Objetos. Este cálculo de predicados pode ser utilizado como uma ferramenta de formalização dos conceitos de

manipulação de imagens e descrição do comportamento das operações que se encontram disponíveis.

As demais propostas existentes para a especificação de uma álgebra de imagens incorporam na linguagem elementos específicos deste tipo de dado, não proporcionando um encapsulamento da complexidade inerente à informação presente na imagem. Algumas abordagens procuram simplificar a semântica dos sumarizadores, considerando a imagem como um conjunto de objetos ou elementos simbólicos bem definidos, e como consequência se utiliza da teoria de conjuntos na formulação de suas consultas [Racine_92] [Schene_92]. No entanto o resultado obtido não se revela muito adequado para a maioria das aplicações, uma vez que apenas a presença ou não de um determinado objeto em uma imagem não consiste de um requisito suficiente para a formulação de consultas. Outras abordagens se revelam específicas à solução de problemas de uma determinada área, como, por exemplo, em banco de dados com manipulação de informações espaciais. Neste domínio, os operadores da álgebra de imagem incorporam os aspectos de raciocínio espacial, importante na definição das posições relativas dos objetos nas imagens nesta aplicação [Jungert_93]. No entanto este procedimento irá provocar um aumento no conjunto básico de operadores existentes na linguagem de consulta de um Banco de Dados.

O aspecto positivo da abordagem apresentada pelo presente trabalho é o tratamento da complexidade da informação contida na imagem a nível de modelagem da aplicação, não sendo necessária a definição de operadores adicionais, mas apenas o estabelecimento do significado semântico preciso de cada expressão, o qual é

originário da própria conceituação de orientação a objetos e dos construtores do modelo. No item a seguir será apresentada uma série de exemplos de consultas formuladas através de expressões do OOPC, que comprovam a força semântica da modelagem pelo SIRIUS, e sua adequação à solução do problema de consultas a imagens por conteúdo em um Banco de Dados Orientado a Objetos.

6.6.1. Definindo Predicados para Imagens.

De acordo com a descrição realizada no capítulo 3, uma expressão do cálculo de predicados orientado a objetos (OOPC) apresenta a seguinte estrutura:

{Objetivo; Abrangência; Qualificação}

A cláusula que define o *Objetivo* da consulta indica quais informações devem ser recuperadas a partir do BD, na aplicação do banco de imagens, podem constituir em atributos (ou propriedades) que sejam não gráficas (texto, números, e outros), referenciando ao objeto ou relacionamento ao qual a imagem se encontra associada, ou gráficos, correspondendo à própria imagem, ou atributos desta, tais como data de aquisição, formato e dimensões. A cláusula de *Abrangência* define os tipos de objetos sobre os quais serão retiradas informações descritas no Objetivo da consulta. Conforme estabelecido na definição genérica do OOPC, a abrangência de uma consulta pode envolver mais do que um tipo de objeto distinto. As propriedades indicadas como objetivo devem descrever os objetos dos tipos indicados em sua abrangência.

A qualificação da consulta consiste em uma fórmula bem formada, definindo uma expressão lógica com uma combinação dos predicados através dos operadores lógicos ($\wedge, \vee, \neg$). Os predicados podem estabelecer condições que envolvam atributos

gráficos e não gráficos. A perfeita combinação destes atributos permite descrever com uma melhor precisão o conteúdo gráfico da imagem a ser recuperada. Com estas considerações pode-se formular consultas básicas sobre um banco de dados com imagens, obedecendo ao esquema apresentado na Figura 1. Para um melhor efeito de comparação, as consultas serão apresentadas na mesma sequência em que foram formuladas no seção 5 deste capítulo.

6.6.2. Expressões de Consultas ao BD de Imagens.

A. Consultas Básicas.

Consulta 1: Recuperar o nome de todos os pacientes cadastrados no BD.

```
SELECT P.Nome FROM PACIENTE P;
```

A expressão do OOPC que representa a mesma consulta fica formulado como indicado a seguir:

```
{p.Nome; p/Paciente};
```

Uma vez que não há teste a ser realizado, a qualificação da consulta foi omitida, com o significado de que todas as instâncias dos objetos do tipo Paciente são válidas, constituindo como resultado da consulta o atributo *Nome* de cada uma das instâncias. O conceito de igualdade de objetos e predicados em seus dois aspectos (igualdade de valor e identidade) não sofre alterações para sua aplicação no contexto de um BD com imagens. Ao realizar a comparação das características de uma imagem, esta é definida sobre os valores dos atributos utilizados em sua representação. Assim a consulta a seguir é escrita no OOPC utilizando-se a comparação de valores de predicados.

Consulta 2: Recuperar o nome de todos os pacientes atendidos pelo médico “Dr. Pedro”.

```
SELECT P.Nome FROM PACIENTE P
WHERE P.ConsultadoPor.Nome = "Dr.Pedro";
```

Consulta no OOPC:

```
{p.Nome; p/Paciente; p.ConsultadoPor.Nome = "Dr.Pedro"};
```

A função *ponto* definida no OOPC permite a navegação pelo Esquema da aplicação através das associações (relacionamentos) definidas entre os tipos de objetos, formando uma expressão do caminho de navegação até o objeto ou atributo desejado. Na consulta 2, a expressão *p.ConsultadoPor.Nome* se referencia à propriedade *Nome* do objeto *Médico* que se encontra associado ao objeto *Paciente* pelo atributo de relacionamento *ConsultadoPor*. Esta expressão de navegação formada com a função *ponto*, representa a junção (“*join*”) implícita entre os objetos de um esquema de um BDOO, derivado a partir dos atributos tipo relacionamento.

B. Consultas para Verificação de Instâncias.

O próximo grupo de consultas se refere à verificação da existência de valores para os tipos de objetos ou atributos. Estas modalidades de consultas são expressas na linguagem do cálculo de predicado orientado a objetos, utilizando-se dos quantificadores universais, apesar destes apresentarem um significado mais amplo do que o operador HASVALUE da linguagem de acesso a imagens.

Consulta 3: Recupere o nome de todos os pacientes que tenham realizado exames de tomografia por RN.

```
SELECT P.Nome FROM PACIENTE P
WHERE HASVALUE P.ExameTomografia;
```

Esta consulta simples pode ser formulada de duas maneiras seguindo a sintaxe do OOPC, considerando que o paciente possa ter realizado nenhum ou vários exames de tomografia, sendo portanto um atributo multivalorado.

a) Utilizando o conceito de “expressão de caminho quantificada”:

$\{p.\text{Nome}; p/\text{Paciente}; p.\exists\text{ExameTomografia}\};$

b) Utilizando o conceito de restrição de quantificadores:

$\{p.\text{Nome}; p/\text{Paciente}; \exists ex/p.\text{ExameTomografia}\};$

A cláusula de qualificação presente em ambas expressões consiste de predicados atômicos, cujo resultado será verdadeiro ou falso, caso exista algum objeto do tipo *ExameTomografia*, descrevendo uma característica do paciente.

Consulta 4: Recupere todos os pacientes que tenham realizado exames de tomografia nos quais possam aparecer o coração.

```
SELECT P.Nome FROM PACIENTE P
WHERE HASVALUE P.ExameTomografia.Coracao
```

Consulta no OOPC:

$\{p.\text{Nome}; p/\text{Paciente}; p.\exists\text{ExameTomografia}.\exists\text{Coração}\};$

De forma semelhante à consulta anterior, a sua qualificação é formada por um predicado atômico, formulado com a utilização do quantificador existencial ao longo do caminho indicado. O valor verdadeiro depende da existência de alguma instância do objeto do tipo *ExameTomografia* associado ao objeto do tipo *Paciente*, no qual tenha-se identificado o coração, segundo algum critério qualquer. Em função da presença do quantificador existencial, a condição é validada com a verificação de instâncias de objetos no BD, definindo o significado associado à expressão na linguagem de consulta.

A combinação de expressões de comparação definindo um predicado composto é suportada na linguagem do OOPC, conforme pode ser verificado no exemplo a seguir.

Consulta 5: Recupere todos os pacientes que tenham realizado exames de tomografia do coração e que sejam maiores de 40 anos.

```
SELECT P.Nome FROM PACIENTE P
WHERE HASVALUE P.ExameTomografia.Coracao
      AND P.Idade > 40;
```

Consulta no OOPC:

```
{p.Nome;      p/Paciente;      p.∃ExameTomografia.∃Coração      ^
p.Idade>40};
```

Esta consulta apresenta um predicado que, além de verificar a existência de instâncias de objetos no BD, irá validar o valor de um atributo (propriedade) do objeto tipo *Paciente*, com o objeto atômico 40. Consiste na elaboração de uma consulta envolvendo atributos que representam características gráficas e textuais, na descrição da informação a ser recuperada. Com esta expressão pode ser observado o tratamento homogêneo obtido na manipulação de conceitos com as semânticas distintas como atributos textuais e gráficos. Isto é possível em consequência de sua modelagem através de tipos de objetos.

C. Descrição do Conteúdo da Imagem.

A descrição do conteúdo gráfico de um atributo do tipo imagem é realizada no OOPC também através de uma expressão de caminho quantificada, uma vez que envolve a verificação da presença ou não de uma característica no interior da imagem. Assim, as consultas envolvendo a descrição do conteúdo da imagem pode ser expressa no cálculo de predicados como se segue.

Consulta 6: Recupere todos os pacientes que realizaram exames de tomografia nos quais apareçam imagens com um formato do Coração e apresentem idade superior a 40 anos.

```
SELECT P.Nome FROM PACIENTE P
WHERE HASVALUE P.ExameTomografia.Coracao.Padrao
      AND P.Idade > 40;
```

Consulta no OOPC:

```
{p.Nome; p/Paciente;
  p.∃ExameTomografia.∃Coração.∃Padrão ∧ p.Idade>40};
```

A expressão do OOPC preserva o significado associado à descrição da informação presente em uma imagem. No exemplo, esta consiste em exames de tomografia nos quais podem ser identificados objetos com o padrão representado pela constante imagem *Coração*.

De forma semelhante, a cláusula de objetivo do OOPC permite uma lista de atributos do objeto do tipo indicado em abrangência, possibilitando a representação da próxima consulta.

Consulta 7: Recupere todos os pacientes e seus exames de tomografia nos quais apareçam imagens com um formato do Coração e apresentem idade superior a 40 anos.

```
SELECT P.Nome, P.ExameTomografia.imagem FROM PACIENTE P
WHERE HASVALUE P.ExameTomografia.Coracao.Padrao
      AND P.Idade > 40;
```

Consulta no OOPC:

```
{p.Nome, p.ExameTomografia.imagem; p/Paciente;
  p.∃ExameTomografia.∃Coração.∃Padrão ∧ p.Idade>40};
```

Neste caso, a lista de atributos apresenta uma expressão de navegação cujo objeto de destino consiste na imagem do exame de tomografia. Esta consiste de uma

generalização do formato originalmente apresentado para esta cláusula, uma vez que o modelo de referência utilizado pelo OOPC não suporta o conceito de atributos de atributos. Uma vez que este se encontra presente no SIRIUS, a sua representação no OOPC deve ser realizada conforme demonstrado anteriormente, admitindo a existência de uma expressão de navegação, na cláusula de objetivo da consulta. No entanto, esta construção não invalida o significado da expressão da cláusula de abrangência, uma vez que a lista de atributos descrevem o tipo de objeto Paciente. Considerando o tratamento uniforme fornecido aos atributos de atributos, estes podem ser livremente utilizados, inclusive para formulação de condições de teste:

Consulta 8: Recupere todos os exames de tomografia de todos os pacientes, que tenham sido realizados no mês de janeiro de 1997, e que tenha sido detectada a presença de um tumor no coração em pacientes com idade entre 40 e 50 anos.

```
SELECT P.Nome, P.ExameTomografia.imagem FROM PACIENTE P
WHERE HASVALUE P.ExameTomografia.Coracao.Padiao
      AND P.ExameTomografia.DataRealizacao >= "01-jan-97"
      AND P.ExameTomografia.DataRealizacao <= "31-jan-97"
      AND P.Idade > 40 AND P.Idade < 50;
```

Consulta no OOPC:

```
{p.Nome, p.ExameTomografia.imagem; p/Paciente;
  p.∃ExameTomografia.∃Coração.∃Padrão ∧
  p.ExameTomografia.DataRealizacao >= "01-jan-97" ∧
  p.ExameTomografia.DataRealizacao <= "31-jan-97" ∧
  p.Idade > 40 ∧ p.Idade < 50};
```

Para recuperar a descrição do conteúdo de uma instância de um atributo com característica de imagem, é necessário indicar como objetivo da consulta o objeto que representa este atributo. Para demonstrar este fato, considere a consulta 7 formulada anteriormente, na qual seja retornada as informações completas referentes aos exames de tomografia realizados pelos pacientes:

Consulta 7a: Recupere todos os pacientes com mais de 40 anos e a descrição de seus exames de tomografia, nos quais apareçam imagens com um formato do Coração.

```
SELECT P.Nome, P.ExameTomografia FROM PACIENTE P
WHERE HASVALUE P.ExameTomografia.Coracao.Padrao
AND P.Idade > 40;
```

Consulta no OOPC:

```
{p.Nome, p.ExameTomografia; p/Paciente;
p.∃ExameTomografia.∃Coração.∃Padrão ∧ p.Idade>40};
```

O atributo *p.ExameTomografia* representa um objeto cujos atributos são os que se encontram definidos no Esquema da aplicação. Uma vez com o identificador deste objeto disponível, pode-se recuperar o seu atributo que seja relevante. Isto é possível através do encadeamento de consultas, realizado com a formalização de seu resultado definindo um novo tipo de objeto, conforme apresentado no item E a seguir.

D. Consultas baseadas em Contexto.

A definição do contexto para a validação de resultados de consultas não é diretamente representado em expressões do OOPC, da forma como definido pela linguagem de manipulação de imagens. O contexto de variáveis de descrição, que seja estabelecido como um intervalo de valores, pode ser representado no OOPC através da combinação de predicados que delimitem este conjunto de validação.

Consulta 9: Recupere todos os pacientes e seus exames de tomografia, nos quais apareçam imagens com uma semelhança no formato de no mínimo 80% com o Coração.

```
ACCEPT Semelhanca FROM 80 TO 100
ON Coracao CHARACTERIZED BY Padrao;
```

```
SELECT P.Nome, P.ExameTomografia.imagem FROM Paciente
P
WHERE P.ExameTomografia.Coracao.Padrao.Semelhanca;
```

Consulta no OOPC:

```
{p.Nome, p.ExameTomografia.imagem; p/Paciente;
  p.ExameTomografia.Coracao.Padrão.Semelhança > 80 ^
  p.ExameTomografia.Coracao.Padrão.Semelhança < 100 };
```

Para as variáveis descritoras, nas quais o contexto é definido pela variação com referência a um valor base, o contexto é estabelecido com a indicação na expressão do OOPC, na forma $v[\text{valor}]$, onde v consiste na variável descritora. Assim, considere uma reformulação da consulta 9, onde a semelhança com a imagem do coração seja definida pela média, em torno de 5, da sua distribuição de cores:

```
ACCEPT Media RANGE 5
ON Coracao CHARACTERIZED BY Histo;
SELECT P.Nome, P.ExameTomografia.imagem FROM Paciente P
WHERE P.ExameTomografia.Coracao.Histo.Media;
```

Consulta no OOPC:

```
{p.Nome, p.ExameTomografia.imagem; p/Paciente;
  p.ExameTomografia.Coracao.Histo.Media[5] };
```

A definição de contexto para os modificadores segue uma definição semelhante à apresentada para as variáveis descritoras, quando for estabelecida como uma variação em referência a um valor base. Na expressão $m[\text{valor}]$, m indica o nome do modificador para o qual será definido o seu contexto, e valor representa a amplitude do intervalo de variação. Na consulta 10, a seguir, é utilizado o modificador Rotação, com estas características.

Consulta 10: Recupere o nome de cada médico cardiologista e dos seus pacientes, com idade superior a 25 anos, que apresentaram uma patologia semelhante em, no

mínimo, 80% à indicada no exame de tomografia do Coração, admitindo alterações em sua apresentação.

```
ACCEPT IMAGEDVAR Semelhanca FROM 80 TO 100
      ON Coracao CHARACTERIZED BY BuscaPadrao;
ACCEPT MODIFIER Rotacao RANGE 15
      ON BuscaPadrao;
```

```
SELECT M.Nome, M.Atende.Nome, M.Atende.ExameTomografia.imagem
FROM Medico M
WHERE M.Atende.idade > 25 AND
      M.Atende.ExameTomografia.Coracao.Rotacao.Semelhanca
      AND M.Especialidade = "Cardiologista"
```

Sua representação no OOPC fica:

Consulta no OOPC:

```
{m.Nome, m.Atende.Nome, m.Atende.ExameTomografia.imagem;
m/Medico;
  m.Atende.idade > 25 ^
^ m.Atende.ExameTomografia.Coracao.Rotacao[15].Semelhanca>80
^ .Atende.ExameTomografia.Coracao.Rotacao[15].Semelhanca<100
^ m.Especialidade = "Cardiologista" };
```

Na consulta anterior também é demonstrada a representação do contexto para variáveis descritoras, definido a partir de um intervalo de variação pela cláusula FROM ... TO ... Não é acrescentada nenhuma representação especial, apenas a formulação da expressão para a validação de valores pertencentes a um intervalo fechado.

E. Consultas com o Operador de Junção.

A linguagem de manipulação de imagens apresentou uma construção que permite formular consultas envolvendo a utilização do operador de *join* entre atributos imagens, que não sejam os elementos definidos no esquema para as

comparações. Estas consultas podem ser expressas no OOPC, utilizando-se uma construção ligeiramente diferente.

Consulta 11: Recupere todos os pacientes com uma patologia no tórax semelhante à apresentada pelo paciente “Pedro”.

```
SELECT P.Nome FROM PACIENTE P, S
WHERE S.Nome = "Pedro" AND
      P.ExameTomografia = S.ExameTomografia;
```

Consulta no OOPC:

```
{p.Nome; p/Paciente; ∃s/Paciente (s.Nome = "Pedro" ∧
                                   s.ExameTomografia = p.ExameTomografia)}
```

A imagem que define a base da comparação, é indicada na cláusula com a condição da consulta. Os aspectos mais restritivos da consulta, representado pela presença de uma alguma constante imagem, são representados no OOPC através das expressões de navegação pelas instâncias do BD. Na consulta anterior, considerando apenas a comparação para o coração, então tem-se:

```
SELECT P.Nome FROM PACIENTE P, S
WHERE S.Nome = "Pedro" AND
      P.ExameTomografia.Coracao = S.ExameTomografia.Coracao;
```

Consulta no OOPC:

```
{p.Nome; p/Paciente; ∃s/Paciente (s.Nome = "Pedro" ∧
                                   s.ExameTomografia.Coracao = p.ExameTomografia.Coracao)}
```

F. Definindo os Resultados de uma Consulta.

O OOPC apresenta a restrição inicial de que apenas um tipo de objeto é permitido na cláusula de abrangência. As propriedades (atributos) que definem o objetivo da consulta devem descrever este tipo de objeto. Como consequência, as expressões definidas no OOPC são fechadas, uma vez que o resultado da consulta consiste de uma coleção de objetos, que são instâncias de um mesmo tipo de objeto.

A solução para o problema com a categoria de consultas nas quais a abrangência deve conter objetos de tipos distintos, não sendo possível definir o resultado como pertencente a um determinado tipo, consiste em definir uma nova classe para estas consultas.

Esta definição é perfeitamente aplicada em um esquema com suporte a imagens, permitindo que as consultas formuladas através de sua linguagem de consulta apresentem como resultado instâncias de objetos que pertencem a um determinado tipo existente no Esquema ou a um novo tipo criado a partir da especificação do objetivo e abrangência da consulta.

Consulta 10a: Recupere o nome de cada médico cardiologista e dos seus pacientes, com idade superior a 25 anos, que apresentaram uma patologia semelhante à indicada no exame de tomografia do Coração:

```
{m.Nome,  m.Paciente.Nome,  p.Paciente.ExameTomografia.imagem;
  m/Medico;
    m.Paciente.Idade > 25
  ^ m.Paciente. ∃ExameTomografia. ∃Coracao. ∃Histo
  ^ m.Especialidade = "Cardiologista" }
```

Na consulta formulada anteriormente, a instância do objeto resultante não pertence ao tipo *Medico*, invalidando a conceituação inicial para expressões no OOPC. Com a flexibilidade inserida a esta definição, para esta consulta será definido um novo tipo (classe) cujos atributos são os indicados na cláusula de objetivo, apresentando o mesmo domínio presente nos tipos base. Desta forma, como estabelecido pelo OOPC, o nome dos atributos podem ser representados por sua expressão de navegação, ficando a definição do tipo como segue:

```
Nome do Tipo:  Exemplo10a
Atributos: ( m.Nome: string
             m.Paciente.Nome: string
```

```
m.Paciente.ExameTomografia.imagem:
obj_imagem)
```

A partir desta definição, a linguagem torna-se mais poderosa, inclusive com a possibilidade de utilizar o resultado de uma consulta em outra, tal como demonstrado no exemplo a seguir.

Consulta 10b: Para os pacientes que apresentaram problemas cardíacos conforme definido pela consulta 10a, recupere apenas os que também sofreram de alguma patologia no pulmão.

```
{p.Paciente.Nome; p/Consulta10a;
  ∃t/Paciente(t.Nome = p.Paciente.Nome)
  ∧ t.∃ExameTomografia.∃Pulmao.∃Padrao )
```

Esta consulta foi formulada utilizando o conceito de quantificador restrito, uma vez que mesmo sendo criado um novo tipo de objeto, suas instâncias não apresentam os relacionamentos definidos no Esquema da aplicação. Assim, o predicado é formulado considerando os objetos do tipo presente no Esquema, no exemplo instâncias de *Paciente*, sendo acrescentada uma condição de *join* baseada em valores de atributos pertencentes a objetos de dois tipos distintos. A condição é perfeitamente válida, uma vez que os domínios dos atributos são compatíveis, ou, de outra forma, idênticos.

Um exemplo mais simples pode ser apresentado, considerando a consulta 7a, selecionando-se para cada exame recuperado o objeto imagem, a sua data de realização e as anotações feitas pelos especialistas:

Consulta 7b:

```
{e.ExameTomografia.imagem,
e.ExameTomografia.DataRealizacao,
e.ExameTomografia.Anotacao; e/Consulta7a )
```

6.7. CONSIDERAÇÕES FINAIS.

Neste capítulo foi apresentada a conceituação básica necessária à incorporação de imagens na modelagem de uma aplicação, com o objetivo de permitir que esta apresente o seu conteúdo descrito a partir de outras imagens definidas como constantes imagens. Também observou-se as características extraídas neste processo, representadas pelos sumarizadores e variáveis descritoras, além dos modificadores modelando as possíveis alterações em seu comportamento. A imagem, enquanto característica de atributo no SIRIUS, pertence ao grupo de atributos denominado de **interface**, que abrange, além de imagens, visualização, som, partitura e gráfico. Consistem dos atributos, no qual o aspecto de apresentação da informação por eles contida é dependente do modelo de interface definido para a aplicação.

No entanto, analisando o Esquema proposto para imagens, este é representado na colônia α -imagem como um tipo de objeto, que apresenta uma série de atributos. Um objeto no SIRIUS pode ser descrito utilizando os aspectos estáticos, dinâmicos, estruturais e de interface. Assim sendo, a informação imagem, modelada como um tipo de objeto, pode ser descrita utilizando estes aspectos:

- *Estático*: conjunto de atributos que descrevem a imagem, correspondendo a aspectos gerais, e também ao objeto imagem, ou seja, formato, tamanho, dimensões e outros.
- *Dinâmico*: se referem aos métodos associados a cada sumarizador para a extração de informação gráfica presente na imagem;
- *Estrutural*: corresponde à forma de armazenamento interno e estruturas de acesso disponíveis à informação extraída de cada instância do atributo imagem;

- *Interface*: mecanismos de apresentação da informação contida em cada instância do objeto imagem e de acesso ao conjunto complementar de atributos de descrição da imagem.

No próximo capítulo, toda a conceituação envolvendo o atributo com característica de imagem será novamente descrita em relação aos módulos que compõem o sistema de consultas baseadas em conteúdo a um Banco de Imagens Orientado a Objetos.

SISTEMA DE GERENCIAMENTO DE IMAGENS

7.1. CONSIDERAÇÕES INICIAIS.

Nos capítulos anteriores foram apresentados o conjunto de conceitos e a fundamentação teórica referentes à incorporação do tipo de dado imagens na modelagem de uma aplicação, formulados utilizando-se do modelo SIRIUS e de seus construtores semânticos. Neste capítulo será realizada a descrição completa dos módulos que definem a arquitetura do Sistema de Gerenciamento de Imagens (SGI), desenvolvido para a demonstração da abordagem proposta para o tratamento do tipo de dado imagem. Os principais componentes do SGI são:

- *Sistema de Gerenciamento de Interface do Usuário*: fornece aos grupos de usuários do sistema acesso ao conjunto de informações de origem textual e gráfica e elementos para a formulação das consultas;
- *Sistema de Processamento de Consultas*: define os planos de realização das consultas baseadas em conteúdo de imagens;
- *Sistema de Gerenciamento do Esquema de Imagens*: possibilita a definição dos elementos básicos de descrição do modelo de conhecimento da aplicação, através das constantes imagens, sumarizadores e variáveis descritoras;

- *Sistema de Gerenciamento de Imagens*: permite a inserção do tipo de dado imagem, com a descrição do seu conteúdo baseando-se no esquema representativo do domínio de conhecimento da aplicação;
- *Sistema de Análise de Imagens (SisMatch)*: extração do conjunto de informações de descrição do conteúdo da imagem;
- *Sistema de Controle de Acesso a Imagens*: realiza o gerenciamento dos mecanismos de acesso e recuperação das imagens a partir do Banco de Dados.

Todos estes módulos, assim como a abordagem de implementação e as decisões de projeto, serão descritos com detalhes no restante deste capítulo, que se apresenta com a seguinte organização: inicialmente serão apresentadas as principais alternativas de projeto disponíveis para o desenvolvimento do SGI, considerando a evolução atual das pesquisas envolvendo o Grupo de Banco de Dados e Processamento de Imagens do ICMSC, USP-São Carlos. Na seção 3, serão discutidos os aspectos referentes ao projeto conceitual, lógico e o modelo de implementação para o SGI. A seguir, na seção 4 será descrita a arquitetura do SGI e o fluxo de informação entre os módulos que a compõem. Encerrando o capítulo, é apresentada a construção de uma aplicação sobre o protótipo implementado.

7.2. ABORDAGENS PARA O PROJETO DO SGI.

Conforme mencionado no capítulo 5, o modelo SIRIUS utilizado para o desenvolvimento da conceituação necessária à incorporação de imagens em uma aplicação consiste na evolução do Modelo de Representação de Objetos (MRO) [Traina_92]. Para o MRO, foram realizados uma série de trabalhos de pesquisa e

implementações de seus respectivos projetos que resultaram na definição de um Sistema Gerenciador de Objetos, denominado de GEO [Traina_91a].

Com a evolução do modelo MRO e a consequente formalização do SIRIUS, iniciou-se um processo semelhante para o sistema gerenciador GEO. Este trabalho inclui o desenvolvimento de um novo modelo de implementação para o núcleo de gerenciamento de objetos, integrado aos conceitos do SIRIUS. No entanto, esta etapa não se encontra totalmente concluída, sendo desenvolvida em paralelo com as atividades de pesquisa tema do presente trabalho sobre o tipo de dado imagem.

O desenvolvimento do Sistema de Gerenciamento de Imagens compreende a construção do seu núcleo sobre o Gerenciador de Objetos (GEO), disponível para suporte ao MRO. Apesar das diferenças conceituais existentes entre os modelos, o mapeamento do esquema de imagens para o MRO não iria representar perdas de características semânticas. No entanto, uma vez que o Gerenciador se encontra em um processo de reestruturação completa, esta abordagem para o desenvolvimento do SGI não se revelou a mais adequada.

A construção sobre o Gerenciador do SIRIUS está descartada porque este não se encontra concluído e uma série de alternativas de projeto devem ser analisadas e validadas até a sua complementação final. Com este cenário de transição entre modelos e implementações de Gerenciadores, resta a alternativa de definir um protótipo do SGI que se utilizasse de outro SGBD. Este protótipo deve permitir a demonstração dos conceitos mencionados neste trabalho para o tratamento de imagens, assim como o teste de uma aplicação que represente um domínio real, como o gerenciamento de imagens médicas. A seleção consiste de um SGBD Relacional,

com suporte ao padrão SQL, possibilitando a incorporação do protótipo, para testes, em um ambiente hospitalar, facilitando sua integração com os demais elementos do Sistema de Informação Hospitalar. No entanto, para o desenvolvimento deste protótipo, torna-se necessária a formulação do projeto do esquema relacional que represente os conceitos de imagens definidos no SIRIUS. A formulação deste projeto se encontra descrita a seguir.

7.3. PROJETO DO SISTEMA DE GERENCIAMENTO DE IMAGENS.

A etapa inicial no desenvolvimento do protótipo consiste no mapeamento para o Modelo Entidade-Relacionamento (ME-R), da modelagem dos conceitos de representação de imagens e seu conteúdo, como atributo com característica de imagem no SIRIUS (Figura 1). A Figura 2 apresenta o diagrama entidade-relacionamento (DE-R) obtido como resultado final do processo de mapeamento.

Figura 1 - Esquema do Atributo com Característica de Imagens no SIRIUS.

Alguns aspectos conceituais presentes no SIRIUS não são diretamente suportados pelo ME-R. Inicialmente, consistem na base conceitual de ambos os modelos: no SIRIUS, os elementos significativos do mundo real são representados como objetos, ao passo que no ME-R o conceito similar consiste em entidades. Isto significa que a abordagem de encapsulamento de informações e representação do comportamento de cada objeto, indicado de forma natural no SIRIUS, não possuem a mesma representação no ME-R. Assim, o conceito de atributo com característica de imagem e os demais elementos envolvidos na descrição de seu conteúdo gráfico são modelados como entidades no ME-R.

Figura 2 -DE-R do Atributo com Característica de Imagem.

O atributo com característica de imagem, modelado através do objeto tipo `AtributoImagem`, possui um conjunto de atributos que formam uma hierarquia de instanciação para os seus valores. Esta característica é representada no DE-R utilizando-se duas construções:

- O relacionamento `Entidade_Aplicação` `Possui` `Atributo_Carac_Imagem` representa o primeiro nível de instanciação, que consiste na definição de um atributo com característica de imagem para uma entidade do domínio da aplicação.
- A entidade `Imagem` modela cada instância do atributo com característica de imagem inserida no Banco de Dados, ou seja, o segundo nível de instanciação dos

atributos definidos para o objeto `AtributoImagem`, da modelagem através do SIRIUS. O relacionamento “são instâncias” com a entidade `Atributo_Imagem`, formada através da agregação das entidades `Paciente` e `Atributo_Carac_Imagem`, representa a semântica originária do conceito de atributos do atributo disponível no modelo SIRIUS.

O mapeamento do relacionamento triplo existente na modelagem a partir do SIRIUS, representando o conteúdo da informação em uma imagem, é realizado com a definição de duas entidades através da abstração de agregação:

- `Característica_Imagem`: representa o relacionamento de identificação do triplo, indicando os sumarizadores de caracterização de cada constante imagem;
- `Componente_Imagem`: representa o relacionamento de conjunto existente entre as variáveis descritoras, para cada sumarizador de descrição de uma característica da constante imagem.

A informação presente no conteúdo do atributo com característica de imagem é representado no relacionamento `Atributo_Carac_Imagem Formada_Por Constante_Imagem`. Os valores que descrevem precisamente a informação presente em uma imagem no BD, segundo o esquema da aplicação, são modelados como atributo do relacionamento `Imagem Descrita_Por Componente_Imagem`.

Após o mapeamento para o modelo entidade-relacionamento, pode-se produzir o modelo Relacional, compondo o conjunto de relações que definem o Banco de Imagens. A matriz de pixel que define a imagem é armazenada no BD relacional como um atributo do tipo BLOB (*binary-large object*), que para efeito de

melhor de melhor desempenho do sistema, é armazenada em uma relação própria, contendo como campos além da imagem, um código de identificação da instância.

7.4. ARQUITETURA DO SGI.

Figura 3 - Arquitetura do Sistema de Gerenciamento de Imagens.

A arquitetura do Sistema de Gerenciamento de Imagens está definida na Figura 3. O módulo que se apresenta aos usuários do SGI consiste no **Sistema de Gerenciamento de Interface**. Este permite a formulação de consultas que possam envolver elementos textuais (descrição da imagem e aspectos sobre os demais atributos, tais como, data de aquisição, unidade médica, e anotações complementares realizadas pelo profissional) e elementos gráficos (fornecendo as constantes imagens

para obter as que sejam semelhantes em relação aos aspectos que o usuário eleger como significativos).

Um sistema que permite a manipulação e o acesso a um conjunto de imagens apresenta a seus possíveis usuários um conjunto diversificado de funcionalidade. Um aspecto importante para a caracterização da forma de utilização do sistema e, conseqüentemente, dos elementos significados presentes no Gerenciador de Interface consiste em fornecer uma classificação aos grupos de usuários. Esta descrição é realizada em concordância com os tipos de acessos e operações que sejam permitas.

Os usuários podem ser divididos em três grandes grupos: o de **desenvolvimento**, **produção** e os **clientes**. Os usuários do **grupo de desenvolvimento** são responsáveis pela análise e projeto de uma aplicação pertencentes a um determinado domínio, como, por exemplo, uma aplicação para o sistema de PACS. Este grupo de usuários é composto pelos analistas de suporte, responsáveis pela análise de requisitos, delimitação da abrangência da aplicação e, principalmente, pelo detalhamento dos tipos e características das imagens que serão manipuladas. Este detalhamento é realizado com o desenvolvimento dos algoritmos de processamento de imagens, definindo os sumarizadores e suas variáveis descritoras. Os algoritmos são implementados como métodos e associados aos objetos no Banco de Dados como atributos com a característica de “procedimento”. No entanto, em função da abordagem de construção do protótipo sobre o Gerenciador Relacional, o trabalho do usuário de desenvolvimento é realizado através das ferramentas disponíveis ao Gerente de Banco de Dados, com conhecimento sobre a estrutura da aplicação.

O **grupo de produção** consiste nos usuários que irão configurar a aplicação e definir o esquema do Banco de Imagens (BI), preparando-o para a sua utilização pelos usuários clientes. Este grupo seleciona as constantes imagens e métodos de processamento de imagens adequados a serem utilizados como sumarizadores, para cada atributo com característica de imagem, associado aos objetos da aplicação. Os usuários deste grupo podem ser classificados em:

A. *Gerente do Banco de Imagens (DBA)*: grupo de usuários que trabalham em conjunto com o grupo de desenvolvimento e os especialistas do domínio da aplicação (médicos, por exemplo), sendo responsáveis pela construção do Banco de Imagens através da definição do esquema para a aplicação, que consiste na indicação do conjunto de imagens a serem apresentadas como base de comparação (constantes imagens) e delimitação do conjunto de características que se aplicará a cada constante imagem, assim como dos parâmetros de descrição destas, ou seja, variáveis e modificadores;

B. *Gerente da Aplicação*: trabalha em colaboração com o Gerente de Banco de Imagens para definir os planos de recuperação e manipulação de imagens que possam satisfazer os requisitos dos tipos de consultas a serem submetidas ao BD.

Por fim, tem-se o **grupo dos clientes**, consistindo nos usuários finais que submetem consultas sobre as informações contidas no BI. As consultas são descritas de acordo com o plano de pesquisa estabelecido para cada uma, através do grupo de gerência da aplicação.

O **Sistema de Gerenciamento de Imagens** contém os procedimentos de incorporação das imagens no Banco de Dados a partir dos equipamentos de exames

computadorizados disponíveis no ambiente médico/hospitalar, como por exemplo, o aparelho de tomografia por ressonância magnética (RM). Este equipamento fornece imagens digitais, nas quais podem ser identificados elementos que definem porções do tecido mole, desempenhando um papel importante na realização de diagnósticos [Panepucci_87] [Tannús_94]. Os aparelhos de tomografia por RM fornecem imagens que são em sua maioria geradas em um formato proprietário de armazenamento em arquivos no disco. A consequência deste fato é que para a utilização destas imagens torna-se necessária a sua conversão para algum formato que seja mais adequado à realização de operações de processamento da imagem.

No ambiente de tomografia, disponível no Instituto de Física de São Carlos (IFSC - USP), o arquivo produzido se encontra em um formato particular, que no entanto é bem simples, denominado PAC. Neste formato, além da matriz de pixel da imagem, é armazenada em seu cabeçalho a informação sobre as suas dimensões, em número de linhas e colunas. A incorporação da imagem é realizada através de sua conversão para o formato de trabalho interno do módulo de análise e processamento de imagens, que suporta arquivos PCX.

Além do processo de conversão das imagens obtidas a partir do tomógrafo, o Sistema de Gerenciamento de Imagens permite que se realize anotações sobre o conteúdo da imagem, através da inicialização de valores para os atributos do atributo com característica de imagem. Esta consiste em uma descrição básica, a qual será complementada com a avaliação do seu conteúdo, de acordo com os elementos descritos no esquema da aplicação.

A definição dos objetos que descrevem as características fundamentais das imagens presentes no domínio da aplicação é realizada pelo **Sistema de Gerenciamento de Esquema de Imagens**. Cada conjunto de imagens apresenta alguns elementos próprios que as descrevam e sejam significativos na formulação das consultas. Com o Gerenciador de Esquemas, o sistema torna-se flexível, se adaptando aos diversos tipos de imagens do ambiente médico. Portanto, o conjunto de relações que representam os objetos do tipo constante imagem, sumarizador e variável descritora possuem seu acesso gerenciado pelos procedimentos que definem este sistema. O Gerenciador de Esquema realiza uma interação com todos os demais módulos do SGI, uma vez que as operações de acesso às imagens possuem a sua formulação validada a partir do esquema especificado para a aplicação.

A análise das imagens inseridas no Banco é realizada pelo **Sistema SisMatch**, o qual implementa os sumarizadores para as operações de *matching* entre as imagens [Senzako_96]. Na versão atual do módulo de análise são utilizados os seguintes métodos:

- a) Correlação Cruzada (CorCruz): este módulo realiza o *matching* entre duas imagens seguindo o método da correlação cruzada, retornando como elementos descritivos as coordenadas (x, y) da melhor posição de *matching* e o coeficiente de correlação. Estes consistem, portanto, nas variáveis descritoras válidas para o sumarizador do tipo *matching* por correlação cruzada.
- b) Deteccção Sequencial de Similaridade (SSDA): este módulo retorna como elementos descritivos da operação de *matching* informações sobre as coordenadas (x, y) da melhor posição de localização do padrão, o nível de similaridade da

imagem e o seu coeficiente de correlação, o qual define as variáveis descritoras válidas para o sumariador do tipo *matching* por detecção sequencial de similaridade.

- c) Preservação do Momento (PresMom): este módulo retorna como elementos descritivos da operação de *matching* informações sobre as coordenadas (x, y) da melhor posição de localização do padrão, e a quantidade de pontos iguais entre a imagem e o padrão de busca, o qual define as variáveis descritoras válidas para o sumariador do tipo *matching* por preservação de momento.

O **Sistema de Processamento de Consultas** realiza o gerenciamento dos diversos tipos de consultas, textuais ou por conteúdo de imagens, que sejam submetidas ao BD. De acordo com o conjunto de informações solicitadas pela consulta, são selecionados os planos adequados para o seu processamento. Esta seleção é realizada com uma análise do Esquema da aplicação, resultando em uma delimitação das características a serem pesquisadas e comparadas com a condição definida. Além disso, o módulo de processamento de consultas é também responsável pela definição do contexto de validação das consultas, formulado a partir de informações obtidas junto ao grupo de usuários gerente da aplicação. A especificação da consulta é realizada através de uma interface gráfica, obedecendo às características e funcionalidades presentes no módulo de gerenciamento de interface.

O controle de acesso às imagens presentes no BD é realizado a partir do próprio núcleo de gerenciamento de acesso disponível no SGBD Relacional. Isto ocorre como uma consequência direta do processo de modelagem da informação presente em uma imagem e do seu mapeamento para o modelo lógico de

implementação (modelo relacional). Neste modelo, as informações que conceitualmente estariam armazenadas em uma estrutura de acesso com suporte a chaves multidimensionais ficam armazenadas como tuplas em uma relação. Nesta relação, as chaves de acesso consistem nos identificadores dos elementos de descrição da imagem, ou seja, constantes imagens, sumarizadores e variáveis descritoras. Portanto, a indexação é obtida a partir da estrutura de índices fornecida pelo Gerenciador Relacional, validando o conjunto resposta da consulta de acordo com o contexto previamente definido.

7.5. MODELO DE CONSTRUÇÃO DE UMA APLICAÇÃO.

O Sistema de Gerenciamento de Imagens foi definido procurando satisfazer o requisito de facilidade de uso, através da formulação de uma interface intuitiva para o usuário. O apêndice B apresenta uma descrição completa com a construção do Esquema de uma aplicação sobre o SGI implementado, exemplificando as características de seu modelo operacional. Nesta implementação do SGI, a interação do usuário é realizada através de uma interface gráfica, na qual os elementos para a construção do Esquema da aplicação são selecionados de maneira natural e direta.

O fator de maior influência no desempenho do SGI consiste no processo de extração das informações de descrição do conteúdo de cada imagem, realizada no instante de sua inserção. Esta consiste na operação que apresenta o maior custo computacional, uma vez que os algoritmos de processamento de imagens, referentes aos sumarizadores definidos para cada constante imagem, serão aplicados sobre a nova instância inserida no Banco de Dados. Para a definição do conjunto resultante das

consultas submetidas ao Banco, o Sistema irá trabalhar com as informações que descrevem o conteúdo da imagem. Estas informações, que já foram previamente preparadas, se encontram na relação que representa a estrutura de acesso às imagens, ficando o custo computacional referente à operação de recuperação tuplas em uma relação no Banco de Dados. O custo computacional necessário para realizar o pré-processamento das imagens é pago assim que uma consulta baseada em conteúdo seja submetida ao BD. Se esta abordagem não fosse aplicada, a cada novo tipo de consulta a ser formulado, os métodos de implementação dos sumarizadores seriam ativados, para a produção do resultado final da consulta.

7.6. CONSIDERAÇÕES FINAIS.

Neste capítulo foram descritos os módulos que definem o protótipo do Sistema de Gerenciamento de Imagens (SGI), implementado para a validação do conjunto de conceitos formulados neste trabalho, com suporte à realização de consultas baseadas no conteúdo das imagens. Apesar do modelo de dados utilizado na implementação (Modelo Relacional) não apresentar suporte semântico às construções presentes na formulação de um Esquema para o tipo de dado Imagens, a implementação resultante demonstrou a importância da representação do conhecimento sobre o domínio da aplicação a nível do Esquema do Banco de Dados.

Capítulo 8

CONCLUSÕES E PROPOSTAS DE FUTURAS PESQUISAS

8.1. CONSIDERAÇÕES INICIAIS.

Neste capítulo são descritos os possíveis trabalhos e projetos de pesquisas que podem ser formulados em decorrência da complementação desta etapa. Também são realizadas as conclusões finais sobre o projeto ressaltando-se as suas contribuições tanto para o desenvolvimento do Modelo Orientado a Objetos SIRIUS, quanto para a área dos Sistemas de Armazenamento e Comunicação de Imagens Médicas (PACS), como para o desenvolvimento de um Banco de Imagens e da representação do tipo de dado imagem.

8.2. CONTRIBUIÇÕES INOVADORAS DO TRABALHO.

Neste trabalho é realizada a proposta da incorporação ao tratamento do tipo de dado imagem como um elemento integrante em um modelo de dados. O principal objetivo do projeto consiste na formulação de consultas baseadas na descrição do conteúdo gráfico ou visual interno de cada imagem. Para solucionar este problema, abordagens como a descrição textual e utilização de ícones com seus relacionamentos espaciais não se revelaram apropriados ao domínio de imagens médicas por

ressonância magnética. A conceituação de constantes imagens em conjunto com os sumarizadores, representando as características de comparação, proporciona ao usuário da aplicação elementos mais significativos na descrição do domínio da aplicação em questão. Esta abordagem para a descrição do conteúdo da imagem pode ser também utilizada em cooperação com as formas de busca mais tradicionais, como os atributos textuais, complementando o significado semântico presente na consulta.

Além disso, estes conceitos são utilizados como uma forma de se realizar um pré-processamento das imagens, com o objetivo de aumentar o desempenho nas consultas. O processo de recuperação da imagem a partir do BD apresenta um melhor desempenho, uma vez que a análise de cada imagem é realizada no instante de sua inserção. O resultado desta atividade de análise da imagem é armazenado no BD, juntamente com toda a imagem, de forma que a operação com maior custo de processamento é realizada apenas uma única vez. Portanto, consultas sobre uma grande coleção de imagens apresentam um baixo custo de processamento, uma vez que as operações mais pesadas já foram realizadas. A presença dos objetos tipo variável descritora e sua indicação direta nas expressões de consulta permitem ao Gerenciador de Imagens selecionar, inicialmente, o conjunto adequado das estruturas de acesso que serão percorridas para a composição do resultado final.

A modelagem apresentada para o tipo de dado imagem como uma característica de atributo, o qual possui o conteúdo da informação gráfica representado por tipos de objetos existentes no Esquema da aplicação, proporciona uma uniformidade na manipulação de todas as informações que possam ser extraídas de uma imagem, sejam estas textuais, representadas na forma de atributos do atributo

imagem, ou gráficos, em associações definidas com as constantes imagens, sumarizadores e variáveis descritoras. Este aspecto fornece um mecanismo adequado para a representação do conhecimento sobre o conjunto de imagens a serem manipulados pelo Gerenciador. Com esta abordagem a validação de algumas características da imagem é realizada diretamente a nível do Esquema, como validação das regras de conhecimento neste inserido. Através do Esquema proposto para a representação de imagens, o modelo suporta perfeitamente o tipo de dado imagem, permitindo a sua utilização em qualquer domínio, em que seja possível definir claramente um conjunto de padrões representativos para a aplicação. Com a presença do esquema conceitual para o atributo com característica de imagem, é realizado um encapsulamento de toda a complexidade naturalmente presente na manipulação deste tipo de dado.

A conceituação de objetos do tipo constante imagem, sumarizadores e variáveis realizada no Esquema do Modelo proporciona uma flexibilidade na definição de qualquer elemento que possa ser utilizado como uma característica de descrição do conteúdo visual de uma imagem. Isto ocorre, pois, se um novo aspecto é definido como significativo para a comparação entre duas imagens, este é incorporado ao BD através da modelagem dos elementos que o descreva. Ou seja, define-se um novo objeto do tipo sumariizador e quantos objetos forem necessários do tipo variável descritora para representar os itens que a operação irá extrair da imagem. Além disso, com o conceito de orientação a objetos, o algoritmo de implementação do sumariizador é definido como um método desse tipo de objeto. Desta forma, a conceituação proposta permite inclusive que novas definições para a

implementação de cada sumariador possam ser incorporadas na etapa de construção do esquema ou projeto da aplicação.

O tratamento uniforme proporcionado pela modelagem apresentou também como consequência dois aspectos importantes: a estrutura da linguagem e a formalização dos seus operadores através de uma álgebra com suporte a orientação a objetos. A sintaxe da linguagem de definição e manipulação de imagens consistiu em uma extensão à Linguagem de Consulta Estruturada (SQL), e os seus comandos não apresentam diferenças sensíveis na sintaxe quando da utilização de um atributo com característica de imagem. Cada comando possui um significado semântico preciso, o que permite a sua análise automática.

Os elementos presentes no Esquema do atributo com característica de imagem e seus correspondentes significados semânticos, possibilitaram a especificação da operação de *join* entre imagens. Esta operação foi definida seguindo um princípio distinto da existência de um atributo de ligação entre as imagens, como ocorre no *join* presente no modelo Relacional. Neste modelo, uma relação deve apresentar um atributo, que consiste em uma chave estrangeira, através do qual possa combinar os valores das tuplas em ambas as relações. Este processo ocorre no tipo de dado imagem, considerando a presença de características contendo um domínio comum, e a comparação é realizada admitindo a existência de um limite de tolerância para a sua validação, o qual consiste no próprio contexto de validação da consulta.

Para a formalização dos operadores de manipulação de imagens, não foi necessário definir uma álgebra específica para este fim. A estrutura conceitual presente no Esquema de imagens e sua modelagem através de tipos de objetos

possibilitaram que qualquer linguagem com suporte à orientação objetos, desde que permita o suporte à definição de características de atributo, manipule imagens conforme descrito neste trabalho. Este aspecto foi demonstrado utilizando-se o cálculo de predicados orientado a objetos (OOPC) proposto como uma especificação para uma linguagem de consulta a um Banco de Dados Orientado a Objetos. Verificou-se que as consultas formuladas na linguagem de acesso a imagens foram perfeitamente rescritas como expressões do OOPC.

8.3. PROPOSTAS DE FUTURAS PESQUISAS.

A partir da conclusão desta etapa inicial, consistindo na especificação dos conceitos envolvendo o tratamento de imagens, uma série de projetos adicionais serão inicializados, com o objetivo de concluir um Sistema Gerenciador de Imagens que possa ser perfeitamente integrado a um ambiente hospitalar, considerando todos os seus demais módulos de gerenciamento. Assim, para o projeto final, serão adotadas duas estratégias: (a) Prosseguir com o desenvolvimento e a implementação do protótipo atual, utilizando um Gerenciador de Banco de Dados Relacional (com suporte ao padrão SQL), incorporando as novas técnicas resultantes das atividades de pesquisa; (b) desenvolvimento do núcleo de gerenciamento de imagens sobre o Gerenciador de Dados proposto para o SIRIUS, denominado de SIRIUS/GO com total suporte aos conceitos de orientação a objetos inseridos neste modelo.

O prosseguimento no desenvolvimento do protótipo apresenta como objetivo a validação, junto a um ambiente hospitalar, dos novos conceitos e propostas que podem ser formulados. Dentro deste domínio, a incorporação se torna mais efetiva, uma vez que não é necessário solucionar aspectos que se referem à implementação

do Gerenciador. Com a evolução do desenvolvimento do Gerenciador de Banco de Dados para o SIRIUS, os recursos de modelagem definidos para o tipo de dado imagem passam a ser suportados de forma adequada pelo modelo de aplicação proposto. Desta forma, não ocorre perda de informação semântica, como a proporcionada pelo mapeamento do Esquema de imagens para o modelo relacional.

Com relação ao conjunto de características de descrição do conteúdo da imagem, ou seja, os sumarizadores, a sua evolução também será realizada em dois campos. O inicial se refere às pesquisas na área de Processamento de Imagem, com a definição e implementação de técnicas de *matching* mais sofisticadas, que permitem a incorporação de modificadores para os sumarizadores, aspecto não suportado na versão atual do protótipo. Isto implica no desenvolvimento de algoritmos que possibilitem o reconhecimento de padrões que possam aparecer alterados por algum fator, representado através dos modificadores. Além da definição destes algoritmos, as técnicas atuais devem ser refinadas, com o objetivo de incorporar procedimentos que possam ser apropriados e mais eficientes na detecção de elementos específicos que envolvam uma determinada patologia. Esta abordagem, provoca uma ampliação no conjunto de sumarizadores, incorporando aspectos específicos do domínio de conhecimento da aplicação. Estes tipos de sumarizadores provocam uma especialização do suporte oferecido pelo Gerenciador de Imagens, na construção de uma determinada aplicação. A definição desta categoria de sumarizadores não elimina o aspecto geral obtido com a representação do conhecimento contido em uma imagem, através de construtores do modelo de dados. Neste sentido, outros sumarizadores de abordagem geral, representando operações distintas de

Processamento de Imagem, tais como a manipulação de histogramas, serão também incorporados ao sistema.

O aspecto de desempenho e eficiência consiste em um requisito fundamental no desenvolvimento destes algoritmos, uma vez que as operações de inserção de uma nova instância do atributo imagem no Banco de Imagens apresenta o seu nível máximo de processamento, na etapa de extração das informações de descrição do seu conteúdo. Portanto, tanto as implementações atuais para os sumarizadores quanto as novas a serem definidas, dentro das abordagens propostas, devem considerar o item desempenho como um fator determinante no tempo de processamento das consultas submetidas ao Banco de Imagens. O aspecto eficiência deve ser aprimorado, com o objetivo de eliminar as ocorrências de imagens não adequadas aos objetivos das consultas submetidas. Este objetivo pode ser alcançado por outras técnicas, além da melhoria nos algoritmos de Processamento de Imagens. Como, por exemplo, a presença de elementos com a representação de conhecimento do domínio da aplicação, associados às imagens, a nível do esquema da aplicação e podendo os mesmos serem indicados na expressão de formulação da consulta.

Além do suporte a sumarizadores distintos, o sistema deverá permitir a definição destes de forma dinâmica, ou seja, através da inserção de métodos adicionais aos objetos do tipo sumariizador existente no Esquema da aplicação. Como resultado, o sistema passará a apresentar um conjunto de ferramentas dedicadas às funções desempenhadas pelos usuários do Grupo de Desenvolvimento. Com o desenvolvimento do Gerenciador de Objetos com suporte ao modelo SIRIUS, este objetivo é naturalmente atingido com os mecanismos para a definição de métodos

dos objetos. A existência de uma nova implementação para um sumarizador pode ser inserida no Banco de Dados como um novo método do objeto do tipo atributo com característica de imagem.

Outro aspecto importante consiste na especificação da estrutura de acesso para as chaves multidimensionais. Neste caso, o estudo das árvores-R e suas variações indicam uma possível direção para esta atividade de pesquisa. Com a incorporação de uma estrutura de acesso com estas características, a definição das variáveis descritoras para os sumarizadores passa a representar cada dimensão da chave. A definição do relacionamento entre as variáveis descritoras e os sumarizadores apresenta como significado a seleção das dimensões que sejam válidas, dentro do domínio de conhecimento da aplicação. O contexto da consulta define uma região no espaço n -dimensional, no qual se encontram presentes as imagens que poderão ter os seus demais atributos comparados, de acordo com a condição de validação. Portanto, as modificações propostas para os algoritmos da árvore-R (ou suas variações) devem suportar a seleção prévia de dimensões na formulação da consulta e a redução do espaço de busca, em função da existência do conceito de contexto. Parte desta proposta de trabalho já teve o seu início como pesquisa a ser desenvolvida a nível de mestrado, a qual deverá resultar em uma análise das estruturas para chaves multidimensionais, definindo a mais apropriada, e possíveis alterações ou modificações em sua conceituação, que a torne adequada à aplicação de imagens médicas.

A indexação de um Banco de Imagens, e sua consequente política de acesso, pode ser tratada de uma forma genérica com a resolução do problema a seguir

descrito. A estrutura de índices criada para o acesso às imagens em um BD, é organizada de acordo com a característica da imagem utilizada na classificação. Como exemplo, considere a característica de comparação de padrões entre imagens. Uma vez que sejam definidas as constantes imagens descrevendo o padrão de busca, o sumariizador representando a operação de análise da imagem e as variáveis descritoras resultantes, a estrutura de índices irá constituir em uma forma de acesso para apenas esta característica. Qualquer elemento distinto não será detectado pelo sistema ou classificado como tal. No entanto as imagens podem sofrer alterações, através de operações de processamento.

As operações de convolução no geral, realizam a transformação de uma imagem I em outra I' qualquer, formada seguindo o aspecto representado pela operação aplicada, ou seja, realce de bordas, definição de contrastes, eliminação de

ruídos, pseudo-colorização e outros. Com a operação de convolução, define-se o mapeamento de um espaço de imagens para outro, obtido de acordo com o

Figura 1 - Transformação na Estrutura de Indexação.

significado contido no operador. Uma vez que o primeiro espaço de imagens se encontra representado no BD e, associado a uma estrutura de acesso I_x , uma abordagem a ser pesquisada consiste em definir um mapeamento desta estrutura de índices que represente a convolução realizada nas imagens, para se obter um novo elemento I_x' (Figura 1). Esta nova estrutura poderá ser utilizada para acesso às imagens pertencentes ao espaço resultante da operação de

convolução. Note que se a imagem obtida a partir da convolução fosse submetida ao BD, esta seria classificada de acordo com o critério presente na estrutura de índices I_x . Mas, uma vez realizada uma convolução sobre a estrutura de índices, esta poderá classificar corretamente a nova imagem I' , precisamente de acordo com o critério utilizado na sua formação. Desta forma, a transformação T sobre a estrutura de índices irá realizar o mapeamento de uma estrutura atual para outra, representando as alterações sobre a imagem, eliminando a necessidade de ativar todos os sumarizadores para a descrição da nova imagem. Esta será obtida diretamente a partir da modificação na estrutura de acesso.

Com a formalização deste mapeamento, obtém-se um BD flexível que suporte a manipulação e acesso a qualquer tipo de imagem de diversas características, podendo ser estas resultantes de operações de convolução distintas. Além disso, esta abordagem provoca uma redução na carga de processamento necessária, uma vez que pode-se obter a nova estrutura de acesso para este conjunto de imagens, sem a necessidade de aplicar os algoritmos de processamento de imagens. Dentre estas operações, pode ser incluído o conjunto de modificadores dos sumarizadores. Estes poderão ser definidos como operações que alteram a estrutura de uma imagem. Desta forma, a estrutura de acesso, considerando a atuação de um modificador, passará a ser definida como um mapeamento da estrutura já existente para o seu sumarizador.

Com a evolução das técnicas de análise de imagens e das estruturas de indexação, estas podem ser também realizadas através da incorporação de contribuições originárias do campo de redes neurais [Badal_94] [Stafylopatis_92]. As redes neurais podem ser utilizadas no processo de classificação das imagens, de

acordo com as constantes imagens e sumarizadores definidos no esquema. Além disso, a definição da própria estrutura de acesso para cada constante imagem pode ser obtida a partir de uma rede neural apropriada. Assim, esta organização passa a representar a evolução natural do conhecimento e das informações presentes em um ambiente hospitalar, à medida que novos exames são realizados, e patologias distintas são identificadas e inseridas no BD, característica esta inerente à definição de uma Rede Neural e do seu mecanismo de aprendizado.

Um dos aspectos definidos na conceituação atual, e que no entanto não se encontra contemplada na implementação atual, consiste na linguagem de definição e manipulação de imagens. Neste campo, alguns trabalhos a serem propostos consistem em definir uma evolução em seu conjunto de comandos, com o objetivo a se tornar a linguagem de definição de esquemas de BD baseando-se no Modelo SIRIUS. A sua sintaxe semelhante à linguagem SQL torna mais rápido o processo de familiarização com os seus comandos por parte dos desenvolvedores de aplicação para Banco de Dados. Dentro do objetivo de se obter maior desempenho no processamento das consultas em um BD com Imagens, alguns aspectos referentes especificamente à abordagem da linguagem, como seu plano de avaliação de consulta, serão objetivos de pesquisa, considerando as características de orientação a objetos [Straube_95].

8.4. CONSIDERAÇÕES FINAIS.

Com a conclusão deste projeto, o seu objetivo principal, de realizar o suporte à formulação de consultas a um banco de dados com imagens, através da indicação de seu conteúdo gráfico, foi plenamente atingido. O modelo de dados orientado a

objetos, SIRIUS demonstrou-se possuir um conjunto de construções semânticas que tornaram possível a modelagem da complexidade de informações que possam estar contidas em uma imagem. Além disso, os conceitos de orientação a objetos, torna natural o processo de refinamento teórico do projeto atual, nas linhas de pesquisas indicadas anteriormente.

Alguns resultados deste trabalho se encontram disponíveis na literatura, como consequência de sua apresentação em alguns dos principais congressos da área a nível nacional [Santos_96a] [Santos_96b] e internacional [Traina_97].

Referências Bibliográficas

- [Atkinson_89] *M. Atkinson; et al.*: **"The Object Oriented DataBase System Manifesto"**, Technical Report GIP ALTAIR - INRIA, 1989.
- [Badal_94] *Badal, D.Z.*: **"Image Indexing and Neural Networks"**, Int. Journal on Neural and Mass-Parallel Comp. and Inf. Systems, vol 3, no 1, feb 94.
- [Beckmann_90] *Beckmann, Norbert; Kriegel, H. Peter; Schneider, R.; Seeger, B.*: **"The R*-tree: an Efficient and Robust Access Method for Points and Rectangles"**, Proc of ACM SIGMOD Int. Conf. on Management of Data, p. 322-331, may 1990.
- [Bertino_91] *Bertino, E., Martino, L.*: **"Object-Oriented Database Management Systems: Concepts and Issues"**, IEEE Computer, vol ??, nro ??, p. 33-47, april 1991.
- [Bertino_92] *Bertino, E., Negri, M., Pelagatti, G., Sbattella, L.*: **"Object-Oriented Query Languages: the Notion and the Issues"**, IEEE Trans. on Knowledge and Data Enginnering, vol 4, nro 3, p. 223-237, June-1992.
- [Bertino_93] *Bertino, E.; Martino, L.*: **"Object-Oriented Databases Systems - Concepts and Architectures"**, Ed. Addison-Wesley Pub. Co., 1993.
- [Bertino_95] *Bertino, Elisa; Foscoli, Paola*: **"Index Organizations for Object-Oriented Database Systems"**, IEEE Trans. on Knowledge and Data Enginnering, vol 7, nro 2, p. 193-209, April-1995.
- [Biajiz_92] *Biajiz, Mauro*: **"MRO* - Uma Atualização do Modelo de Representação de Objetos e uma Abordagem Formal"**, Dissertação de Mestrado em Ciências da Computação - ICMSC-USP, São Carlos, SP, julho de 1992.
- [Biajiz_96a] *Biajiz, Mauro*: **"Representação de Modelos de Dados Orientados a Objetos através de Parametrização de Abstrações"**, Tese de Doutorado, IFSC-USP- São Carlos, setembro 1996.
- [Biajiz_96b] *Biajiz, M.; Traina Jr, C.; Vieira, M.T.P.*: **"SIRIUS - Modelo de Dados Orientado a Objetos e Baseado em Abstrações de Dados"**, in Anais do XI Simp. Bras. de Banco de Dados, São Carlos - SP, p. 338-352, out. 1996.
- [Bimbo_93] *Bimbo, Alberto Del; Campanai, M., Nesi, P.*: **"A Three-Dimensional Iconic Environment for Image Database Querying"**, IEEE Trans on Software Engineering, vol 19, nro 10, p. 997-1011, Oct-1993.
- [Bimbo_95] *Bimbo, Alberto Del; Vicario, Enrico; Zingoni, Daniele*: **"Symbolic Description and Visual Querying of Image Sequences Using Spatio-Temporal Logic"**, IEEE Trans. on Knowledge and Data Enginnering, vol 7, nro 4, p. 609-622, Aug-1995.
- [Brink_95] *Brink, Anne; Marcus, Sherry; Subrahmanian, V.S.*: **"Heterogeneous Multimedia Reasoning"**, IEEE Computer, vol 28, nro 9, p. 33-39, september 1995.
- [Brinkhoff_93] *Brinkhoff, T.; Kriegel, H.P.; and Seeger, B.*: **"Efficient Processing of Spatial Joins Using R-trees"**, Proc. ACM SIGMOD Int. Conf. on Management of Data, May 1993, p. 237-246.

- [Bretl_89] *Bretl, Robert, et al.*: **"The GemStone Data Management System"**, in: "Object Oriented Concepts, Databases, and Applications" (Kim, W.; Lochovsky, F., eds) p. 263-308, Addison-Wesley, 1989.
- [Caelli_93] *Caelli, Terry; Reye, David*: **"On the Classification of Image Regions by Colour, Texture and Shape"**, Pattern Recognition, vol 26, nro 4, p. 461-470, 1993.
- [Cardenas_93] *Cardenas, Alfonso F., et al.*: **"The Knowledge-Based Object-Oriented PICQUERY+ Language"**, IEEE Trans. on Knowledge and Data Engineering, vol 5, nro 4, p. 644-656, aug - 1993.
- [Carey_89] *Carey, M.J. et al.*: **"Storage Management for Objects in EXODUS"**, in: "Object-Oriented Concepts, Databases, and Applications" (Kim, W.; Lochovsky, F., eds) p. 341-369, Addison-Wesley, 1989.
- [Cattell_94] *Cattell, R.G.G.*: **"Object Data Management"**, Addison Wesley Publishing Co., 1994.
- [Chang_81] *Chang, N.S.; Fu, K.S.*: **"Picture Query Languages for Pictorial Data Base Systems"**, IEEE Computer, vol 14, nro 11, p. 23-33, Nov-1981.
- [Chang_87] *Chang, S.K., Shi, Q.Y, and Yan, C.W.*: **"Iconic Indexing by 2D Strings"**, IEEE Trans. Pattern Analysis and Machine Intelligence, vol 9, nro 3, p. 413-428, may 1987.
- [Chang_88] *Chang, Shi-Kuo; Yan, C.W.; Dimitroff, Donald C.; Arndt, Timothy*: **"An Intelligent Image Database System"**, IEEE Trans on Software Engineering, vol 14, nro 5, p. 681-688, may - 1988.
- [Chang_91] *Chang, Shi-Kuo; Kunii, Tosiya L.*: **"Pictorial Data-Base Systems"** IEEE Computer, vol 24, nro 11, p. 13-21, Nov-1991.
- [Chang_92a] *Chang, Chin-Chen; Wu, Tzong-Chen*: **"Retrieving the most Similar Symbolic Pictures from Pictorial Databases"**, Information Processing & Management, vol 28, nro 5, p. 581-588, 1992.
- [Chang_92b] *Chang, Shi-Kuo; Hsu, Arding*: **"Image Information Systems: Where do we go from Here?"**, IEEE Trans. on Knowledge and Data Engineering, vol 4, nro 5, p. 431-442, october 1992.
- [Chiueh_94] *Chiueh, Tzi-cker*: **"Content-Based Image Indexing"**, in Proc. of the 20th VLDB Conference, Santiago, Chile, p. 582-593, 1994.
- [Costagliola_92] *Costagliola, G.; Tortora, G.; Arndt, T.*: **"A Unifying Approach to Iconic Indexing for 2-d and 3-d Scenes"**, IEEE Trans. on Knowledge and Data Engineering, vol 4, nro 3, p. 205-222, june-1992.
- [Costagliola_95] *Costagliola, Gennaro; Ferrucci, Filomena*: **"Non-Redundant 2D Strings"**, IEEE Trans. on Knowledge and Data Engineering, vol 7, nro 2, p. 347-350, april-1995.
- [DeWitt_94] *DeWitt, David J. et al.*: **"Client-Server Paradise"**, in Proc. of the 20th VLDB Conference, Santiago, Chile, p. 558-569, 1994.
- [Dimitrova_95] *Dimitrova, N.; Golshani, F.*: **"Motion Recovery for Video Content Classification"**, ACM Trans. on Information Systems, vol 13, nro 4, p. 408-439, Oct-1995.

- [Dutton_90] *Dutton, Gail Finlayson* : **"The Current State of PACS"**, Applied Radiology, p. 15-19, august 1990.
- [Egenhofer_94] *Egenhofer, Max J.*: **"Spatial SQL: A Query and Presentation Language"**, IEEE Trans. on Knowledge and Data Engineering, vol 6, nro 1, p. 86-95, Feb-1994.
- [Faloutsos_87] *Faloutsos, C.; Sellis, T., and Roussopoulos, N.*: **"Analysis of Object Oriented Spatial Access Methods"**, in Proc. ACM SIGMOD, p. 27-29, may-1987.
- [Faloutsos_93] *C. Faloutsos et al.*: **"Efficient and Effective Querying by Image Content"**, Tech. Report RJ 9453, IBM Research Division, San Jose, Calif., aug 1993.
- [Fishman_89] *Fishman, D.H. et al.*: **"Overview of the Iris DBMS"**, in: "Object-Oriented Concepts, Databases, and Applications" (Kim, W.; Lochovsky, F., eds) p. 219-250, Addison-Wesley, 1989.
- [Flickner_95] *Myron Flickner et al.*: **"Query by Image and Video Content: The QBIC System"**, IEEE Computer, vol 28, nro 9, p. 23-32, september 1995.
- [Gonzalez_87] *Gonzalez, Rafael C.*: **"Digital Image Processing"**, Addison-Wesley, 1987.
- [Gudivada_95a] *Gudivada, Venkat N.; Raghavan, Vijay V.*: **"Design and Evaluation of Algorithms for Image Retrieval by Spatial Similarity"**, ACM Trans on Information Systems, vol 13, no 2, p. 115-144, April-1995.
- [Gudivada_95b] *Gudivada, Venkat N.; Raghavan, Vijay V.*: **"Content-Based Image Retrieval Systems"**, IEEE Computer, vol 28, nro 9, p. 18-22, september 1995.
- [Günther_90] *Günther, O.; Buchmann, A.*: **"Research Issues in Spatial Databases"**, ACM Sigmod Record, vol 19, nro 4, p. 61-68, Dec-1990.
- [Günther_91] *Günther, Oliver; Bilmes, Jeff.* **"Tree-Based Access Methods for Spatial DataBases: Implementation and Performance Evaluation"**, IEEE Trans. on Knowledge and Data Engineering, vol 3, nro 3, p. 342-356, Sep-1991.
- [Guttman_84] *Guttman, Antonin*: **"R-Trees: A Dynamic Index Structure for Spatial Searching"**, Proc. of ACM SIGMOD Int. Conf. on Management of Data, p. 47-57, 1984.
- [Grosky_89] *William I. Grosky, R. Mehrotra*: **"Image DataBase Management"**, IEEE Computer, vol 22, nro 12, 1989.
- [Herot_80] *Herot, Christopher F.*: **"Spatial Management of Data"**, ACM Trans. on Database Systems, vol 5, nro 4, p. 493-514, Dec-1980.
- [Hoel_92] *Hoel, E.G.; Samet, H.*: **"A Qualitative Comparison Study of Data Structures for Large Line Segment Databases"**, Proc. of ACM SIGMOD Int. Conf. on Management of Data, San Diego, p. 205-214, may 1992.
- [Hofstede_96] *Hofstede, A.H.M. ter; Proper, H.A.; van der Weide, Th.P.*: **"Query Formulation as an Information Retrieval Problem"**, The Computer Journal, vol.39, no 3, p. 255-274, 1996.
- [Holmes_92] *Holmes, P.; Jungert, E.*: **"Symbolic and Geometric Connectivity Graph Methods for route Planning in Digitized Maps"**, IEEE Trans. on Pattern Analysis and Machine Intelligence, vol 14, no. 5, p. 549-565, may 1992.

- [Hsu_96] Hsu, Chih-Cheng; Chu, Wesley C.; Taira, Ricky K.: **"A Knowledge-Based Approach for Retrieving Images by Content"**, IEEE Trans. on Knowledge and Data Engineering, vol 8, nro 4, p. 522-532, aug-1996.
- [Hunter_79] Hunter, G.M.; Steiglitz, K.: **"Operations on Images Using Quad Trees"**, IEEE Trans. on Pattern Analysis and Machine Intelligence, vol 1, nro 2, p. 145-153, april 1979.
- [Joseph_88] Joseph, Thomas; Cardenas, Alfonso F.: **"PICQUERY: A High Level Query Language for Pictorial DataBase Management"**, IEEE Trans on Software Engineering, vol 14, nro 5, p. 630-638, may - 1988.
- [Jungert_93] Jungert, E.; Chang, S.K.: **"An Image Algebra for Pictorial Data Manipulation"**, CVGIP: Image Understanding, vol 58, nro 2, p. 147-160, sep-1993.
- [Kamel_92] Kamel, I.; Faloutsos, C.: **"Parallel R-Trees"**, Proc. of ACM SIGMOD Int. Conf. on Management of Data, San Diego, p. 195-204, may 1992.
- [Kankanhalli_96] M.S. Kankanhalli, B.M. Mehtre, J.K. Wu: **"Cluster-Based Color Matching for Image Retrieval"**, Pattern Recognition, vol 29, nro 4, p. 701-708, 1996.
- [Kasturi_89] R. Kasturi, et al.: **"Map Data Processing in Geographic Systems"**, IEEE Computer, vol 22, nro 12, 1989.
- [Kim_89a] Kim, Won et al.: **"Features of the ORION Object-Oriented Database System"**, in: "Object Oriented Concepts, Databases, and Applications" (Kim, W.; Lochovsky, F., eds) p. 251-282, Addison-Wesley, 1989.
- [Kim_89b] Kim, Won; Kim, Kyung-Chang; Dale, Alfred: **"Indexing Techniques for Object-Oriented Databases"**, in: "Object Oriented Concepts, Databases, and Applications" (Kim, W.; Lochovsky, F., eds) p. 371-394, Addison-Wesley, 1989.
- [Kim_90a] Kim, Won: **"Introduction to Object-Oriented Databases"**, The MIT Press, 1990.
- [Kim_90b] kim, Won: **"Object Oriented Databases: Definition and Research Directions"**, IEEE Trans on Knowledge and Data Engineering, vol 2, nro 3, p. 327-341, sep 1990.
- [Lee_89] Lee, S.; Shan, M.; Yang, W.: **"Similarity Retrieval of Iconic Image Database"**, Pattern Recognition, vol 22, nro 6, p. 675-682, june 1989.
- [Mascarenhas_89] Mascarenhas, N.D.A.; Velasco, F.R.D.: **"Processamento Digital de Imagens"**, IV EBAI, Argentina, Janeiro 1989.
- [McLeod_91] McLeod, D.: **"Perspective on Object Databases"**, Information and Software Technology, vol 33, nro 1, p. 13-21, jan-feb 1991.
- [Mehrotra_95] Mehrotra, Rajiv; Gary, James E.: **"Similar-Shape Retrieval in Shape Data Management"**, vol 28, nro 9, p. 57-62, september 1995.
- [Nabil_96] Nabil, M.; Ngu, Anne H.H.; Shepard, John: **"Picture Similarity Retrieval Using the 2D Projection Interval Representation"**, IEEE Trans. on Knowledge and Data Engineering, vol 8, nro 4, p. 533-539, august, 1996.
- [Niblack_93] W. Niblack et al.: **"The QBIC Project: Querying Images by Content Using Color, Texture, and Shape"**, Tech. Report RJ 9203, IBM Research Division, San Jose, Calif., fev 1993.

- [Nierstrasz_89] Oscar Nierstrasz: **"A Survey of Object-Oriented Concepts, DataBases, and Applications"**, Edited by W. Kim & F.H. Lochovsky; ACM Press, 1989.
- [Nievergelt_84] Nievergelt, J.; Hinteberger, H.; Sevcik, K.C.: **"The Grid File: an Adaptable, Symmetric Multikey File Structure"**, ACM Trans. on Database Systems, vol 9, nro 1, p. 38-71, March 1984.
- [Ogle_95] Ogle, Virginia E, Stonebraker, M.: **"Chabot: Retrieval from a Relational Database of Images"**, IEEE Computer, vol 28, nro 9, p. 40-48, september 1995.
- [Oliveira_96] Oliveira, J.Lopes; Medeiros, C. Bauzer : **"User Interface, Architectures, Languages and Models in Geographic Databases"**, in Anais do XI Simp. Bras. de Banco de Dados, São Carlos - SP, p. 20-39, out. 1996.
- [Orenstein_86] Orenstein, J.A.: **"Spatial Query Processing in an Object-Oriented Database System"**, Proc of ACM SIGMOD Int. Conf. on Management of Data, Washington, p. 326-333, may 1986.
- [Orenstein_88] Orenstein, J.A.; Manola, F. A.: **"PROBE Spatial Data Modeling and Query Processing in an Image Database Application"**, IEEE Trans. on Software Engineering, vol 14, no 5, p. 611-629, may-1988.
- [Orenstein_89] Orenstein, J.A.: **"Redundancy in Spatial Databases"**, Proc. of ACM SIGMOD Int. Conf. on Management of Data, Portland, p. 294-305, may 1989.
- [Orenstein_90] Orenstein, J.A.: **"A Comparison of Spatial Query Processing Techniques for Native and Parameter Spaces"**, Proc. of ACM SIGMOD Int. Conf. on Management of Data, p. 343-352, may 1990.
- [Panepucci_87] H. Panepucci; N. Beckmann; A. Tannús; T. Bonagamba: **"NMR Imaging In: Recent Advances in Organic NMR Spectroscopy"**, editores J.B. Lambert e Roberto Rittner, Norell Press, (USA) 1987, cap 13.
- [Papadias_95] Papadias, D.; Theodoridis, Y.; Sellis, T.; Egenhofer, M.J.: **"Topological Relations in the World of Minimum Bounding Rectangles: A Study with R-trees"**, Proc. of ACM SIGMOD Int. Conf. on Management of Data, San Jose, p. 92-103, may 1995.
- [Racine_92] Racine, M.; Badal, D.Z.: **"An Algebra for the Access and Manipulation of Image Database"**, Tech. Rep. EAS-CS-92-7, University of Colorado at Colorado Springs, july 1992.
- [Ramaswamy_95] Ramaswamy, Sridhar; Kanellakis, Paris C.: **"OODB Indexing by Class-Division"**, Proc. of ACM SIGMOD Int. Conf. Man. of Data, p. 139-149, 1995.
- [Robinson_81] Robinson, J.T.: **"The k-d-B-Tree: A Search Structure for Large Multidimensional Dynamic Indexes"**, in Proc. of ACM SIGMOD, p. 10-18, 1981.
- [Roussopoulos_88] Roussopoulos, Nick; Faloutsos, Christos; Sellis, Timos: **"An Efficient Pictorial DataBase System for PSQL"**, IEEE Trans. on Software Engineering, vol 14, nro 5, p. 639-650, may - 1988.
- [Roussopoulos_95] Roussopoulos, N.; Kelley, S.; Vincent, F.: **"Nearest Neighbor Queries"**, Proc. of ACM SIGMOD Int. Conf. on Data Management, p. 71-79, may-1995.
- [Samet_84] Samet, Hanan: **"The Quadrees and Related Hierarchical Data Structures"**, ACM Comp. Surveys, vol.16, no2, p. 187-260, June 1984.

- [Santos_96a] Santos, Rildo R.; Agma.J.M., Traina Jr.C.: **"Uma Linguagem de Definição e Recuperação de Imagens Baseada em Conteúdo em uma Base de Dados Orientada a Objetos"**, in Anais do XI Simp. Bras. de Banco de Dados, São Carlos - SP, p. 127-142, out. 1996.
- [Santos_96b] Santos, R.R; Oliveira,M.R.; Traina,Agma.J.M., Traina Jr.,C.: **"Banco de Imagens Médicas com Suporte a Consultas Baseadas em Conteúdo Gráfico"**, in Anais do III Forum Nacional de Ciência e Tecnologia em Saúde, VI Simp. Bras. de Informática em Saúde, Campos do Jordão, p. 661-662, out. 1996.
- [Schalkoff_89] Robert J. Schalkoff. **"Digital Image Processing and Computer Vision"**, edited by John Wiley & Sons, Inc., 1989.
- [Schene_92] Schene,V.; Badal,D.Z.: **"Definition of Image Algebra"**, Tech. Rep. EAS-CS-92-6, University of Colorado at Colorado Springs, july 1992.
- [Senzako_96] Senzako, Edna Y.: **"SisMatch- Matching para Imagens de Tomografia por Ressonância Magnética"**, Dissertação de Mestrado, ICMSC - USP, São Carlos, 1996.
- [Sistla_94] Sistla, A. Prasad, Yu, C., Haddad, R.: **"Reasoning about Spatial Relationships in Picture Retrieval Systems"**, in Proc. of the 20th VLDB Conference, Santiago, Chile, p. 570-581, 1994.
- [Sproull_91] Sproull,Robert.F.: **"Refinements to Nearest-Neighbor Searching in k-Dimensional Trees"**, Algorithmica, vol 6, p. 579-589, 1991
- [Srihari_95] Rohini K., Srihari : **"Automatic Indexing and Content-Based Retrieval of Captioned Images"**, IEEE Computer, vol 28, nro 9, p. 49-56, september1995.
- [Stafylopatis_92] Stafylopatis, Andreas; Likas, Aristidis: **"Pictorial Information Retrieval Using the Random Neural Network"**, IEEE Trans. on Software Engineering, vol 18, nro 7, p. 590-600, july-1992.
- [Stonebraker_90] M. Stonebraker; L. A. Rowe; M. Hirohama : **"The Implementation of POSTGRES"**, IEEE Trans. on Knowledge and Data Engineering, vol 2, Nro 1, March 1990, p. 125-142.
- [Straube_95] Straube, Dave D.; Öznu, Tamer. **"Query Optimization and Execution Plan Generation in Object-Oriented Data Management Systems"**, IEEE Trans. on Knowledge and Data Enginnering, vol 7, nro 2, p. 210-227, April-1995.
- [Tanaka_88] Tanaka,M.; Ichikawa,T.: **"A Visual User Interface for Map Information Retrieval based on Semantic Significance"**, IEEE Trans. on Software Engineering, vol 14, nro 5, p. 666-670, may 1988
- [Tannús_94] A. Tannús et al.: **"Architecture of 2.0 Tesla NMR Tomograph"**, in 9o. International Society of Magnetic Resonance, vol III, p. 1101-1109, S. Francisco, USA, aug 1194.
- [Traina_86] Caetano Traina Jr.: **"Máquina e Modelo de Dados Dedicados para aplicações de Engenharia"**, Tese de Doutorado, IFQSC-USP, dezembro 1986.
- [Traina_91a] Caetano Traina Jr.: **"GEO: Um Sistema de Gerenciamento de Bases de Dados Orientados a Objetos - Estado Atual de Desenvolvimento e**

- Implementação"**, in Anais do VI Simpósio Brasileiro de Banco de Dados, Manaus, 1991.
- [Traina_92] *Caeatano Traina Jr., Jan Frans Willen Slaets* - **"MRO - Um Modelo de Representação de Objetos"**, in Notas Internas, nro 104, ICMSC-USP, fevereiro 1992.
- [Traina_94] *Traina Jr., C.; Traina, A.J.M., Biajiz, M.*:"**O Papel das Abstração de Instanciação em um Meta-Modelo de Abstrações para BDOO**", in Anais do IX Simp. Bras. de Banco de Dados, p. 173-187, São Carlos, set-1994.
- [Traina_97] *Traina Jr., C.; Traina, Agma J.M; Santos, R.R.; Senzako, E.Y.*: **"Content-Based Medical Images Retrieval in Object Oriented Database"**, in Proc. of 10th IEEE Symposium on Computer-Based Medical Systems, vol 2, p. 67-72, june 11-13th, Maribor - Slovenia, 1997.
- [Unnikrishnan_88] *Unnikrishnan, A.;Shankar, Priti; Venkatesh, Y.V.*: **"Threaded Linear Hierarquical Quadrees for Computation of Geometric Properties of Binary Images"**, IEEE Trans. on Software Engineering, vol 14, nro 5, p. 659-665, may 1988
- [Vassilakopoulos_95] *Vassilakopoulos, M.; Manolopoulos, Y.*: **"Dynamic Inverted Quadtree: A Structure for Pictorial Database"**, Information Systems, vol 20, nro 6, p. 483-500, 1995.
- [Wang_91] *Wang, Fangju*: **"Relational-Linear Quadtree Approach for Two-dimensional Spatial Representation and Manipulation"**, IEEE Trans. on Knowledge and Data Enginnering, vol 3, nro 1, p. 118-122, Mar-1991.
- [Weiss_95] *Weiss, R.; Duda,A.; Gifford,D.K.*: **"Composition and Search with a Video Algebra"**, IEEE Multimidea, p. 12-25, may-1995.
- [Willians_92] *Willians, J.; Badal, B.Z.*: **"Image Database Access Method"**, Tech. Rep. EAS-CS-92-8, University of Colorado at Colorado Springs, july 1992.
- [Woelk_86] *Woelk, D.; Kim, W.; Luther, W.*: **"An Object-Oriented Approach to Multimedia Databases"**, Proc. of ACM SIGMOD Int. Conf. Man. of Data, p. 311-325, 1986.
- [Zand_95] *Zand, M.; Collins,V.;Caviness,D.*: **"A Survey of Current Object-Oriented Databases"**, Data Base Advances, vol 26, no 1, p. 14-29, feb 1995.

Apêndice A

Sumário da Linguagem de Acesso a Imagens

Neste apêndice se encontra a sintaxe de todos os comandos presentes na Linguagem de Definição de Imagens (LDI) e da Linguagem de Manipulação de Imagens (LMI).

1. LINGUAGEM DE DEFINIÇÃO DE IMAGENS NO SIRIUS.

1.1. Criação dos Objetos de Descrição do Domínio da Aplicação.

```
CREATE IMAGE <nome_tipo_imagem> ON ( <nome_tipo_objeto> |  
    <nome_tipo_objeto>`.`<nome_relacionamento> ) `;`
```

```
CREATE [SUB] IMAGECONST <NomeConst> AS <Localizador> `;`
```

```
CREATE SUMMARIZER <nome_summ> AS <Localizador>  
    [<restrição>] `;`
```

```
CREATE IMAGEPAR <nome_tipo> <lista_variaveis>  
    [RANGE <valor> | FROM <valor> TO <valor>] `;`
```

```
CREATE MODIFIER <nome_modificador> ON <nome_sumarizador>  
    AS ( <restricao_mod> );
```

1.2. Associação entre os Objetos do Esquema da Aplicação.

```
SET COMPARISON <nome_caract> IMAGECONST <nome_const>  
    CHARACTERIZED BY <nome_sumarizador>  
    DESCRIBED BY <lista_variaveis> `;`
```

```
SET IMAGE <nome_tipo_imagem> DESCRIBED BY  
    <lista_constantes> `;`
```

1.3. Alteração da Descrição dos Objetos no Esquema.

```
ALTER IMAGE <nome_tipo_imagem>
```

```
DROP IMAGECONST <nome_const> ';' 
```

```
ALTER IMAGECONST <nome_const> CHARACTERIZED BY <NomeSumm>  
    ( ADD IMAGEPAR <lista_var_descritora> |  
      DROP IMAGEPAR <lista_var_descritora> ) ';' 
```

```
ALTER IMAGECONST <nome_const> DROP SUMMARIZER <NomeSumm> ';' 
```

1.4. Eliminação dos Objetos no Esquema.

```
DROP (IMAGE | IMAGECONST | SUMMARIZER | IMAGEPAR)  
    <nome_item> ';' 
```

```
DROP MODIFIER <nome_modificador> ON <nome_sumarizador> ';' 
```

1.5. Definição de uma Instância para um Tipo de Atributo com Característica de Imagem.

```
INSERT INTO <Nome_Atr_Imagem>  
    [ON ( <nome_instancia_obj> )]  
VALUES <nome_arquivo_imagem> ';' 
```

2. LINGUAGEM DE MANIPULAÇÃO DE IMAGENS.

```
ACCEPT IMAGEPAR <nome_variavel>  
    [RANGE <valor> | FROM <valor> TO <valor>]  
    ON <nome_constante> CHARACTERIZED BY <nome_sumarizador> 
```

```
ACCEPT MODIFIER <nome_modificador>  
    [RANGE <valor> | FROM <valor> TO <valor>]  
    ON <nome_sumarizador> ';' 
```

```
SELECT <lista_atributos>  
FROM <lista_objetos>  
WHERE <condicao> ';' 
```

Modelo de Construção de um Aplicativo com Imagens

O protótipo desenvolvido para o Sistema de Gerenciamento de Imagens permite a formulação do Esquema de uma aplicação com o domínio de imagens originárias a partir de exames de tomografia computadorizada. Neste apêndice é apresentado uma navegação através do sistema, exemplificando o seu modelo operacional e a forma de interação com os diversos usuários do sistema.

1. APRESENTAÇÃO DO SISTEMA.

O sistema apresenta uma interface baseada em menus, onde são apresentados aos usuários todas as opções disponíveis, para a definição do Esquema da aplicação e interação com o Banco de Imagens, através dos formulários de consulta.

A configuração da barra de menus principal do Sistema é determinada pela categoria do usuário que realiza acesso. A validação do usuário é realizada através de um mecanismo de senha, que irá validar a visão do banco de dados que se encontrará disponível (Figura 1). Consistindo de usuários pertencentes aos grupos de desenvolvimento e produção, estes possuem permissão de acesso a todos os aspectos funcionais do sistema (Figura 1). Os usuários do grupo de clientes, apresentam acesso apenas às operações de inserção de novas imagens, instâncias dos atributos com

característica de imagem, e consultas ao Banco de Imagens. A interface do Sistema é semelhante à apresentada na Figura 1, não apresentando no entanto as opções de definição dos Objetos e Relacionamentos para o esquema da aplicação.

Figura 1 - Interface do Sistema para os Grupos de Desenvolvimento e Produção.

2. INTERFACES DOS GRUPOS DE DESENVOLVIMENTO E PRODUÇÃO.

2.1. Definindo um Atributo com Característica de Imagem.

A definição de um atributo com característica de imagem é realizada selecionando-se a opção Atributo Imagem do item de menu denominado Objetos. A partir desta seleção é apresentada a interface da Figura 2, em que o usuário indica o nome de identificação do atributo com característica de imagem, indicando também a qual tipo de objeto do esquema da aplicação que este atributo se encontrará associado.

Figura 2 - Definindo um Atributo com Característica de Imagem.

2.2. Definindo as Constantes Imagens.

As constantes imagens são inseridas no Esquema da aplicação, através da interface apresenta na Figura 3, a partir da seleção da opção Constante Imagem do item de menu Objetos. A implementação atual define a localização da imagem a ser inserida, como um arquivo localizado em um diretório do disco. Através desta interface, o usuário também poderá descrever características adicionais referentes à constante imagem, tais como anotações gerais que julgar adequado. Sendo definido uma constante imagem do tipo sub-imagem, as coordenadas presentes, indicam o canto superior esquerdo do retângulo de delimitação. A sub-imagem apresenta um tamanho padrão de uma matriz de 32x32 pixel, em consequência da funcionalidade do módulo de Processamento de Imagens.

Figura 3 - Definindo uma Constante Imagem.

2.3. Definindo os Sumarizadores.

Cada sumarizador que seja válido para o Esquema da aplicação é definido através da interface presente na Figura 4, a partir da seleção da opção Sumarizador do item de menu Objetos. O operador que define o módulo de processamento de imagens do sumarizador é selecionado a partir de um conjunto pré-definido no sistema em sua implementação atual. Esta seleção está indicada no campo “Tipo do Sumarizador”, cujos valores são obtidos a partir da relação que contém a definição dos operadores.

Em conjunto a definição dos sumarizadores, é sinalizada quais são os possíveis modificadores válidos, assim como de seus elementos descritivos.

Manager [Sumarizadores do Esquema da Aplicação]

File Edit Cadastro Objeto Relacionamento Contexto Consulta Database Window Help

Sumarizadores: Registro 3 de 3

Nome do Sumarizador: PadraoPM

Tipo do Sumarizador: PRES_MOMENTO

Alteração no Comportamento?

☒ SIM

☐ NÃO

Modificadores do Comportamento:

Nome	Tipo	Valor Médio	Limite Aproximado
------	------	-------------	-------------------

OK Cancel Help

Figura 4 - Definindo os Sumarizadores do Esquema da Aplicação.

2.4. Definindo as Variáveis Descritoras.

As variáveis descritoras presentes no esquema da aplicação são definidas através da interface presente na Figura 5, a partir da seleção da opção Variável Descritora do item de menu Objetos. O tipo da variável descritora é selecionado a partir do conjunto de tipos suportado pelo sistema, ou seja, a partir dos tipos de variáveis internas definidas para cada operador pré-definido no sistema.

Figura 5 - Definindo as Variáveis Descritoras do Esquema.

2.5. Associação entre os Objetos do Esquema.

As interfaces para a definição dos relacionamentos entre os objetos do Esquema, de forma a consolidar a representação do conhecimento sobre o domínio da aplicação, apresentam uma funcionalidade semelhante. Todos os campos a serem preenchidos são implementados através de elementos de escolha, cujos valores são obtidos a partir dos que já se encontram definidos no Banco. Além das restrições de integridade referencial já existentes no Banco, esta característica impede a definição de qualquer associação que conceitualmente seja inválida.

2.5.1. Descrição do Conteúdo do Atributo com Característica de Imagem.

A descrição do conteúdo de cada instância do atributo com característica de imagem é realizada com a interface apresentada na Figura 6, a partir da seleção da opção “Descreve Atributo Imagem” do item de menu Relacionamento.

Figura 6 - Descreve o Conteúdo do Atributo com Característica de Imagem.

2.5.2. Descrição das Constantes Imagens.

A descrição das características a serem extraídas de cada constante imagem, descrevendo-as em função dos sumarizadores e de suas variáveis descritoras, é realizada com a interface apresentada na Figura 7, a partir da seleção da opção “Descreve Constante Imagem” do item de menu Relacionamento. Com a seleção do Sumarizador, apenas as variáveis descritoras que sejam válidas para este tipo, tornam-se disponíveis como opção para escolha, no campo “Variáveis Descritoras” da interface.

Figura 7 - Descrevendo as Constantes Imagens do Esquema.

Todas estas interfaces são utilizadas também nas operações de alteração de definições no Esquema ou mesmo sua eliminação. Através da barra de rolagem presente em cada janela, é possível navegar pelos elementos presentes no Banco de Dados, recuperando a informação desejada. Uma vez posicionado neste registro, pode-se alterar a sua definição, através de uma atualização, modificação ou eliminação de elementos do Esquema. Com esta funcionalidade, o Sistema passa a

apresentar uma interface uniforme para as diferentes atividades a serem realizadas por seus usuários.

3. INTERFACES DO GRUPO CLIENTES.

3.1. Inserindo Imagens no Banco de Dados.

A inserção de uma nova imagem, como instância de uma atributo com característica de imagem, é realizada com a seleção do objeto do Esquema da aplicação que possui este elemento definido como um de seus atributos. Na aplicação de exames médicos, consiste em selecionar o “Paciente” que realizou um novo exame computadorizado (Figura 8) que será cadastrado no Banco de Imagens.

The screenshot shows a window titled 'imaget' with a menu bar (File, Edit, Cadastro, Objetos, Relacionamento, Contexto, Consulta, Database, Window, Help) and a toolbar. Below the menu is a sub-window titled 'Pacientes do Hospital'. The main form area is titled 'Pacientes do Hospital:' and contains the following fields and controls:

- Nome do Paciente:** Text field containing 'José Antônio da Silva'.
- Número de Identidade:** Text field containing '1234'.
- Endereço:** Text field containing 'Av. das Alamedas, 25'.
- Sexo:** Radio button group with 'Masculino' selected and 'Feminino' unselected.
- Data de Nascimento:** Text field containing '12/02/70'.
- Plano de Saúde:** Text field containing 'UNIMED'.
- Patologia:** Text field containing 'Tumor Benigno'.
- Observação:** Text area containing 'Realizará Exames de Tomografia'.
- Pesquisas:** A section containing three buttons: 'Exames Computadorizados', 'Consulta', and 'Novo'.

At the bottom of the window are three buttons: 'OK', 'Cancel', and 'Help'.

Figura 8 - Instâncias do Tipo de Objeto Paciente.

Uma vez selecionado o Paciente, o formulário para a inserção de um novo exame é apresentado (Figura 9). O tipo deste exame é selecionado a partir dos atributos com característica de imagem definido no Esquema. Esta interface é semelhante à disponível para a definição de constantes imagens, permitindo que usuário realize as associações relevantes sobre este exame, que consistem nos atributos textuais da imagem. Uma vez obtida a imagem a partir de um arquivo em disco, o seu conteúdo será descrito de acordo com as características associadas com o atributo imagem.

Figura 9 - Inserindo Imagens no Banco de Dados.

3.2. Formulando Consultas ao Banco de Imagens.

3.2.1. Definição do Contexto de Validação da Consulta.

A definição do contexto de validação de uma consulta é realizada selecionando-se uma das opções do item de menu “Contexto”, para as variáveis descritoras (Figura 10) e para os modificadores (Figura 11). De forma semelhante às demais interfaces do sistema, os campos consistem de itens de seleção, cujos valores são obtidos a partir do Banco de Dados, obedecendo os relacionamentos existentes entre os elementos do Esquema.

Figura 10 - Definindo o Contexto das Variáveis Descritoras.

Figura 11 - Definindo Contexto dos Modificadores.

3.2.2. Definindo a Imagem de Busca.

A formulação da consulta ao Banco de Imagens é realizada através do formulário padrão presente na Figura 12. Neste, o usuário pode realizar consultas que envolvam cada um dos elementos apresentados: atributo com característica de imagem, constante imagem, sumariizador, ou variável descritora. O usuário não é obrigado a selecionar valores para todos os elementos. Uma vez indicado o conjunto que considera válido para o seu objetivo, pode-se submeter a consulta para o Banco de Imagens.

Figura 12 - Formulando uma Consulta ao Banco de Imagens.

3.3. Examinando Resultados das Consultas.

O resultado das consultas submetidas ao Banco de Imagens, pode ser analisada através do formulário que se encontra na Figura 13. Se para algum exame o usuário deseja examinar algum detalhe, sobre os atributos não textuais associados à imagem, esta tarefa pode ser realizada, através da opção “Detalhe”, onde será apresentado um formulário semelhante de cadastro dos exames computadorizados (Figura 9).

Figura 13 - Resultados das Consultas ao Bando de Imagens.

Este protótipo foi utilizado para medir o tempo de inserção e busca por imagens em um pequeno banco de dados, contendo aproximadamente 50 imagens em cada atributo. O tempo de busca no BD através de consultas envolvendo predicados com 5 comparações sobre um atributo pode ser considerado não mensurável. A recuperação de cada imagem identificada no conjunto de resposta é realizada em aproximadamente 0.3 segundos, em uma máquina Pentium 133Mhz com 64 Mbytes de memória principal, executando o servidor de Banco de Dados Oracle, versão 7.3, no ambiente do Windows NT 4.0. A maior parte do tempo de processamento decorre do armazenamento da imagem no banco de dados, e este tempo está diretamente relacionado com o número de constantes imagens que descreve o conteúdo do atributo imagem instanciado, e o número de operadores (grupos de comparação) associados a cada constante imagem. No entanto, este tempo pode ser considerado independente do número de imagens já armazenadas no BD. Em um caso típico envolvendo o operador de Correlação-Cruzada considerando um conjunto de quatro constantes imagens, o sistema leva aproximadamente 8 minutos para armazenar cada imagem no Banco de Dados. Deve ser observado que se os conceitos de descrição do conteúdo das imagens não forem utilizados, o tempo de armazenamento de cada imagem não é significativo. No entanto, se a descrição das imagens não for realizada durante a sua inserção, cada consulta levaria por volta de dois minutos para processar cada imagem armazenada, considerando cada predicado de comparação que se encontre definido no Esquema da Aplicação. Portanto, em para avaliar uma consulta sobre um banco de dados com 50 imagens, envolvendo a comparação com 5 predicados, o sistema irá gastar por volta de 8h:20minutos.