

**Universidade Federal de São Carlos**  
**Centro de Ciências Exatas e de Tecnologia**  
**Departamento de Computação**  
**Programa de Pós-Graduação em Ciência da Computação**

**UM SISTEMA PARA RECUPERAÇÃO DE IMAGENS  
COM BASE EM CARACTERÍSTICAS  
GEOMÉTRICAS, CONJUNTOS NEBULOSOS E  
INDEXAÇÃO MÉTRICA**

Alan Gavioli

São Carlos – SP  
Maio/2005

**Ficha catalográfica elaborada pelo DePT da  
Biblioteca Comunitária da UFSCar**

G283sr

Gavioli, Alan.

Um sistema para recuperação de imagens com base em características geométricas, conjuntos nebulosos e indexação métrica / Alan Gavioli. -- São Carlos : UFSCar, 2006.

113 p.

Dissertação (Mestrado) -- Universidade Federal de São Carlos, 2005.

1. Banco de dados. 2. Busca por conteúdo. 3. Métodos de acesso métrico. 4. Conjuntos difusos. 5. Funções de distância métricas. I. Título.

CDD: 005.74 (20<sup>a</sup>)

# Universidade Federal de São Carlos

Centro de Ciências Exatas e de Tecnologia

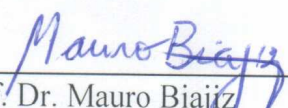
Programa de Pós-Graduação em Ciência da Computação

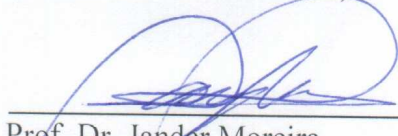
## *“Um Sistema para Recuperação de Imagens com base em Características Geométricas, Conjuntos Nebulosos e Indexação Métrica”*


ALAN GAVIOLI

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

### **Membros da Banca:**

  
\_\_\_\_\_  
Prof. Dr. Mauro Biajiz  
(Orientador - DC/UFSCar)

  
\_\_\_\_\_  
Prof. Dr. Jander Moreira  
(Co-Orientador - DC/UFSCar)

  
\_\_\_\_\_  
Profª. Dra. Denise Guliato  
(UFU)

## **DEDICATÓRIA**

Aos meus pais, Luiz e Nadir, que jamais mediram esforços para me incentivar em cada etapa de minha vida. Devo tudo o que sou a vocês.

## **AGRADECIMENTOS**

A Deus, por sua enorme generosidade em todos os momentos de minha vida. A Ele agradeço principalmente por ter saúde e paz, que possibilitam que eu alcance os meus objetivos.

Ao meu orientador, Prof. Mauro Biajiz, pela grande dedicação a este projeto de mestrado, pela confiança, pela paciência, pela amizade e por sua participação fundamental na minha formação como pesquisador.

Ao meu co-orientador, Prof. Jander Moreira, pela importante participação neste projeto de mestrado e pela disposição em colaborar com sugestões de grande valor.

Aos colegas André Oriani e Marcos Rodrigues Vieira, pelas importantes contribuições que forneceram.

Aos colegas do Grupo de Banco de Dados, pela cooperação e pela amizade.

Ao amigo Marcos Quiles, pelas sugestões relevantes e pela amizade de vários anos.

Aos demais professores e amigos que de alguma forma contribuíram para a realização deste trabalho.

À CAPES, que me auxiliou financeiramente durante o desenvolvimento deste trabalho.

GAVIOLI, A. **Um Sistema para Recuperação de Imagens com base em Características Geométricas, Conjuntos Nebulosos e Indexação Métrica**. 2005. 113p. Dissertação (Mestrado em Ciência da Computação) – Departamento de Computação, Universidade Federal de São Carlos, São Carlos, 2005.

## RESUMO

Neste trabalho apresenta-se o sistema denominado *MIGFIR* (*Metric Indexing and Geometric Features-based Image Retrieval*), que foi desenvolvido principalmente para permitir que determinados tipos de imagens (como de faces humanas, por exemplo) sejam indexados e recuperados com eficiência e precisão usando suas medidas geométricas, duas novas funções de similaridade e o método de acesso métrico *Slim-tree*. Esse sistema também permite que os usuários configurem parâmetros relevantes durante a definição de consultas, e possibilita ainda aplicar conjuntos nebulosos e conceitos relacionados para que a linguagem empregada nas expressões de busca de uma das classes de consulta suportadas se aproxime consideravelmente da linguagem normalmente utilizada por seres humanos. A eficiência e a precisão do sistema na execução de consultas sobre vários conjuntos reais e sintéticos de imagens de faces humanas também são apresentadas. Com relação à eficiência, os resultados obtidos mostram que o uso de *MIGFIR* pode promover uma redução bastante considerável do tempo de execução de consultas em comparação com uma abordagem similar que aplica varredura seqüencial dos dados. Já em termos de precisão, os resultados mostram que o desempenho é satisfatório na recuperação de faces humanas similares a determinados tipos de imagens-exemplo.

GAVIOLI, A. **A Metric Indexing, Fuzzy Sets and Geometric Features-based Image Retrieval System**. 2005. 113p. Master Thesis (Master's Degree in Computer Science) – Department of Computer Science, Federal University of Sao Carlos, Sao Carlos, 2005.

## ABSTRACT

In this work, the *MIGFIR* (Metric Indexing and Geometric Features-based Image Retrieval) system is presented. This system was developed mainly to make possible to index and retrieve certain image classes (like human face images, for example) with efficiency and accuracy using geometric measures, two new similarity functions and the metric access method called *Slim-tree*. This system also allows configuring important parameters during the definition of queries and makes possible to apply fuzzy sets and related concepts so that the language used in search expressions of one of the query classes that are supported be considerably close of the language usually applied by human beings. Efficiency and accuracy of the system at the execution of queries on several real and synthetic human face image sets are also presented. About efficiency, the obtained results show that *MIGFIR* can promote a quite considerable reduction of the execution time of queries in comparison with a similar approach that applies sequential scanning of the data. About accuracy, the results show that performance is satisfactory during the retrieval of human faces similar to certain example-images.

## LISTA DE FIGURAS

Figura 2.1 – Representação gráfica da função de pertinência difusa $\mu_F$ .....	19
Figura 2.2 – Diferença na representação do conceito <i>adolescente</i> : (a) função característica do conjunto clássico <i>adolescente</i> ; (b) função de pertinência do conjunto nebuloso <i>adolescente</i> .	19
Figura 2.3 – Representação gráfica das funções de pertinência nebulosas que definem os termos lingüísticos da variável <i>Velocidade</i> .....	21
Figura 2.4 – Possível função de pertinência para o predicado nebuloso <i>Alto</i> .....	22
Figura 2.5 – Obtenção do valor-verdade do predicado nebuloso "Face larga": (a) face humana com duas medidas destacadas; (b) possível função de pertinência para determinar o valor-verdade do predicado. ....	23
Figura 3.1 – <i>FQ-tree</i> usando a métrica <i>Hamming</i> para indexar cadeias de caracteres. ....	31
Figura 3.2 – Representação gráfica de uma <i>M-tree</i> : (a) estrutura hierárquica; (b) distribuição ilustrativa dos dados no espaço. ....	32
Figura 3.3 – Ilustração das estruturas dos nós da <i>Slim-tree</i> : (a) representação dos dois tipos de nós; (b) armazenando sete palavras da língua inglesa. Fonte: [TRAINA JR. et al., 2001].....	36
Figura 3.4 – Representação de uma <i>Slim-tree</i> : (a) estrutura hierárquica; (b) distribuição ilustrativa dos dados no espaço. Fonte: [SANTOS, 2003]. ....	37
Figura 4.1 – Tela do sistema <i>QBIC</i> exibindo imagens do resultado de uma consulta. Fonte: [FLICKNER et al., 1995].....	43
Figura 4.2 – Interface de consulta do sistema <i>Blobworld</i> . Fonte: [CARSON et al., 1999].....	47
Figura 5.1 – Arquitetura básica do sistema <i>MIGFIR</i> . ....	54
Figura 5.2 – Tela inicial do sistema <i>MIGFIR</i> .....	55
Figura 5.3 – Tela para elaboração de expressões de busca para consultas baseadas em imagens-exemplo. ....	58
Figura 5.4 – Tela para exibição das imagens que satisfazem uma consulta baseada em imagens-exemplo. ....	60
Figura 5.5 – Tela inicial da interface para consultas nebulosas. ....	61
Figura 5.6 – As possíveis composições da tela inicial: (a) para expressões <i>tipo 3</i> ; (b) para expressões <i>tipo 4</i> .....	63
Figura 5.7 – Tela para elaboração de expressões de busca para consultas baseadas em predicados nebulosos. ....	64



Figura 5.8 – Tela para exibição das imagens que satisfazem uma consulta baseada em predicados nebulosos.....	68
Figura 5.9 – Arquitetura da biblioteca de estruturas de indexação <i>Arboretum</i> .....	69
Figura 5.10 – Classes utilizadas para representar imagens de faces: (a) Classe <i>TFaceImage0</i> ; (b) Classe <i>TFaceImage1</i> ; (c) Classe <i>TFaceImage2</i> ; (d) Classe <i>TFaceImage3</i> .....	70
Figura 5.11 – Classes utilizadas para instanciar as duas funções de similaridade consideradas: (a) Classe correspondente à métrica <i>MGF</i> ; (b) Classe correspondente à métrica <i>MGF2</i> .....	71
Figura 5.12 – Três esboços simples de faces humanas. ....	78
Figura 5.13 – Esboço de uma face humana com 5 medidas geométricas em destaque.....	78
Figura 5.14 – Algoritmo <i>Search_Result_Images</i> . ....	81
Figura 6.1 – Gráficos apresentando o tempo consumido durante a execução de $F_1$ para cada configuração de consulta por similaridade e cada conjunto de dados, considerando-se a execução em <i>MIGFIR</i> e em <i>SeqScan</i> . ....	93
Figura 6.2 – Gráficos apresentando o tempo consumido durante a execução de $F_2$ para cada configuração de consulta por similaridade e cada conjunto de dados, considerando-se a execução em <i>MIGFIR</i> e em <i>SeqScan</i> . ....	94
Figura 6.3 – Gráficos apresentando o tempo consumido durante a execução de $F_3$ para cada configuração de consulta por similaridade e cada conjunto de dados, considerando-se a execução em <i>MIGFIR</i> e em <i>SeqScan</i> . ....	96
Figura 6.4 – Gráficos apresentando o tempo consumido durante a execução de $F_4$ para cada configuração de consulta por similaridade e cada conjunto de dados, considerando-se a execução em <i>MIGFIR</i> e em <i>SeqScan</i> . ....	98
Figura 6.5 – Cinco imagens usadas nas consultas baseadas em exemplos que foram analisadas.....	99
Figura 6.6 – Gráficos apresentando o tempo consumido durante a execução de $E_1$ para cada configuração de consulta por similaridade e cada conjunto de dados, considerando-se a execução em <i>MIGFIR</i> e em <i>SeqScan</i> . ....	100
Figura 6.7 – Gráficos apresentando o tempo consumido durante a execução de $E_2$ para cada configuração de consulta por similaridade e cada conjunto de dados, considerando-se a execução em <i>MIGFIR</i> e em <i>SeqScan</i> . ....	102
Figura 6.8 – Gráficos apresentando o tempo consumido durante a execução de $E_3$ para cada configuração de consulta por similaridade e cada conjunto de dados, considerando-se a execução em <i>MIGFIR</i> e em <i>SeqScan</i> . ....	103

## LISTA DE TABELAS

Tabela 5.1 – Características extraídas de imagens e os possíveis termos lingüísticos.....	67
Tabela 5.2 – Valores de características geométricas referentes aos três esboços de faces considerados.....	78
Tabela 5.3 – Predicados nebulosos válidos e respectivos valores-verdade referentes aos três esboços de faces considerados.....	79
Tabela 6.1 – Quatro configurações de parâmetros para a execução da consulta nebulosa $F_1$ .....	92
Tabela 6.2 – Quatro configurações de parâmetros para a execução da consulta nebulosa $F_2$ .....	93
Tabela 6.3 – Quatro configurações de parâmetros para a execução da consulta nebulosa $F_3$ .....	95
Tabela 6.4 – Quatro configurações de parâmetros para a execução da consulta nebulosa $F_4$ .....	97
Tabela 6.5 – Três configurações de parâmetros para a execução da consulta baseada em exemplo $E_1$ .....	99
Tabela 6.6 – Três configurações de parâmetros para a execução da consulta baseada em exemplos $E_2$ .....	101
Tabela 6.7 – Duas configurações de parâmetros para a execução da consulta baseada em exemplos $E_3$ .....	103

## LISTA DE SIGLAS

CBIR	<i>Content-based Image Retrieval</i>
FC	<i>Feature Contrast</i>
FFC	<i>Fuzzy Features Contrast</i>
K-NNQ	<i>K-Nearest Neighbor Query</i>
MAM	Método de Acesso Métrico
MGF	<i>Metric for Geometric Features</i>
RQ	<i>Range Query</i>

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO .....</b>	<b>12</b>
1.1	CONSIDERAÇÕES INICIAIS E MOTIVAÇÃO .....	12
1.2	OBJETIVOS DA DISSERTAÇÃO .....	14
1.3	ORGANIZAÇÃO DA DISSERTAÇÃO .....	15
<b>2</b>	<b>CONJUNTOS NEBULOSOS .....</b>	<b>16</b>
2.1	CONSIDERAÇÕES INICIAIS .....	16
2.2	CONJUNTOS CLÁSSICOS E CONJUNTOS NEBULOSOS .....	16
2.3	OPERAÇÕES BÁSICAS SOBRE CONJUNTOS NEBULOSOS .....	20
2.4	VARIÁVEIS LINGÜÍSTICAS .....	21
2.5	PREDICADOS NEBULOSOS .....	22
2.6	MODELO FUZZY FEATURES CONTRAST .....	23
2.7	CONSIDERAÇÕES FINAIS .....	25
<b>3</b>	<b>MÉTODOS DE ACESSO MÉTRICOS .....</b>	<b>27</b>
3.1	CONSIDERAÇÕES INICIAIS .....	27
3.2	ESPAÇOS MÉTRICOS .....	27
3.3	TIPOS DE CONSULTAS .....	29
3.4	PRINCIPAIS MÉTODOS .....	30
3.4.1	Slim-tree .....	34
3.5	CONSIDERAÇÕES FINAIS .....	40
<b>4</b>	<b>SISTEMAS PARA RECUPERAÇÃO DE IMAGENS POR CONTEÚDO.....</b>	<b>42</b>
4.1	CONSIDERAÇÕES INICIAIS .....	42
4.2	PRINCIPAIS SISTEMAS .....	42

4.2.1	QBIC .....	42
4.2.2	Garlic .....	44
4.2.3	MARS.....	45
4.2.4	VisualSEEk .....	46
4.2.5	Blobworld.....	47
4.2.6	Sistema de Berretti, Del Bimbo, e Pala .....	48
4.2.7	RETIN .....	49
4.2.8	CbPACS .....	50
4.3	CONSIDERAÇÕES FINAIS .....	51
<b>5</b>	<b>O SISTEMA MIGFIR.....</b>	<b>52</b>
5.1	CONSIDERAÇÕES INICIAIS .....	52
5.2	ARQUITETURA DO SISTEMA .....	53
5.3	MÓDULO MIGFIR_PREDICATEGENERATOR.....	54
5.4	MÓDULO MIGFIR_QUERYINTERFACES .....	56
5.4.1	Interface para Consultas baseadas em Imagens-exemplo.....	56
5.4.2	Interface para Consultas baseadas em Predicados Nebulosos.....	61
5.5	MÓDULO MIGFIR_QUERYIMPLEMENTATION .....	68
5.5.1	MIGFIR_QueryImplementation_Data .....	69
5.5.2	MIGFIR_QueryImplementation_Application.....	72
5.5.2.1	Leitura dos dados das imagens .....	73
5.5.2.2	Criação de objetos para representar imagens .....	73
5.5.2.3	Criação de instâncias da Slim-tree.....	74
5.5.2.4	Inserção dos objetos nas instâncias da Slim-tree.....	75
5.5.2.5	Recuperação das imagens similares a cada componente da expressão .....	80
5.5.2.6	Execução de operações lógicas entre componentes.....	82
5.5.2.7	Geração do resultado final da consulta.....	83
5.6	CONSIDERAÇÕES FINAIS .....	85
<b>6</b>	<b>AVALIAÇÃO EXPERIMENTAL .....</b>	<b>87</b>
6.1	CONSIDERAÇÕES INICIAIS .....	87
6.2	AVALIAÇÃO DA PRECISÃO NA EXECUÇÃO DE CONSULTAS .....	88
6.3	AVALIAÇÃO DA EFICIÊNCIA NA EXECUÇÃO DE CONSULTAS .....	91
6.4	CONSIDERAÇÕES FINAIS .....	104

<b>7</b>	<b>CONCLUSÕES.....</b>	<b>105</b>
7.1	CONSIDERAÇÕES FINAIS E RESULTADOS .....	105
7.2	PRINCIPAIS CONTRIBUIÇÕES.....	106
7.3	PROPOSTAS PARA TRABALHOS FUTUROS .....	107
<b>8</b>	<b>REFERÊNCIAS BIBLIOGRÁFICAS .....</b>	<b>109</b>

# 1 INTRODUÇÃO

## 1.1 CONSIDERAÇÕES INICIAIS E MOTIVAÇÃO

Há vários anos, pesquisadores estão dedicando bastante atenção ao gerenciamento de dados multimídia, como por exemplo imagem, vídeo e som. Em particular, a literatura permite notar que há um grande interesse relacionado à indexação e à recuperação apropriadas de imagens, motivado fortemente por dois fatores: o número de bancos de dados que armazenam grandes quantidades desse tipo de mídia tem aumentado consideravelmente; e a manutenção desses bancos tem se mostrado cada vez mais importante para diversos setores, tais como investigações criminais, Medicina, detecção de registros de marcas, agências de notícias, serviços na Internet, órgãos governamentais, dentre outros.

Os estudos relacionados ao gerenciamento de grandes conjuntos de imagens conduziram ao interesse por métodos para a recuperação de imagens baseada em conteúdo (em inglês: *CBIR*), isto é, baseada em características como cores, medidas geométricas, texturas, formas, dentre outras. Alguns trabalhos da área relatam que a pesquisa sobre esses métodos começou a ser realizada na década de 80, sob influência considerável de uma publicação de Chang e Fu [CHANG; FU, 1980], mas se tornou realmente intensa na década de 90. Essa pesquisa tem sido responsável pela produção de um grande número de trabalhos, tanto teóricos quanto na forma de sistemas computadorizados.

Com o propósito de tornar possível recuperar imagens por conteúdo em uma base de dados, antes de armazená-las é necessário analisá-las, de modo que as representações precisas do conteúdo sejam extraídas e em seguida armazenadas junto com os próprios objetos na base. Fazendo-se isto, essas representações podem ser usadas durante a realização das consultas. Além disso, o funcionamento de sistemas para a recuperação de imagens baseada em conteúdo freqüentemente depende também do emprego de estruturas

de indexação (também chamadas de métodos de acesso) adequadas e dos tipos de consultas possíveis.

Consultas cujos critérios de seleção são baseados em igualdade ou em ordem geralmente são inadequadas para sistemas que operam sobre bancos que armazenam imagens, já que a estrutura desse tipo de dado é consideravelmente mais complexa que a de tipos tradicionais, como dados numéricos e textuais, e a recuperação por conteúdo costuma não ser um processo exato. Para esses sistemas, consultas por similaridade são mais apropriadas [TRAINA JR. et al., 2001]. Em um exemplo típico, o critério de busca na base de dados deve ser aproximado tal que todos os objetos que tenham grau de similaridade com um dado objeto-exemplo dentro de um limiar pré-definido devem ser recuperados.

Como as consultas por similaridade envolvendo imagens normalmente dependem da existência de funções métricas (conceito definido no capítulo 3) adequadas, o projeto desse tipo de função costuma ser uma tarefa muito importante para a recuperação baseada em conteúdo.

Com relação aos métodos de acesso, a varredura seqüencial é uma maneira óbvia de buscar objetos em uma base de dados; no entanto, ela pode fazer com que as recuperações sejam bastante lentas. Por isso, é muito importante que sistemas para *CBIR* empreguem estruturas de indexação que sejam mais rápidas que a varredura seqüencial. É relevante que essas estruturas suportem a execução de consultas por similaridade.

Apesar de existir uma grande quantidade de métodos de acesso tradicionais [ELMASRI; NAVATHE, 2001] e espaciais [GAEDE; GÜNTHER, 1998], de acordo com Traina Jr. *et al.* [TRAINA JR. et al., 2000] a abordagem mais recomendada para a indexação de imagens, de modo que possam ser executadas consultas por similaridade, é utilizar estruturas de indexação métricas (discutidas no capítulo 3). Essas estruturas aplicam funções métricas para determinar a similaridade entre cada par de objetos.

Como a tarefa de recuperação de imagens por conteúdo pode ser baseada em diversos tipos de características, como por exemplo cores, texturas, medidas geométricas e formas, e também em diferentes abordagens, atualmente pode-se encontrar na literatura uma grande quantidade de ferramentas que realizam esse tipo de tarefa (no capítulo 4 serão dadas referências para vários trabalhos).



No entanto, quando se deseja identificar ferramentas para *CBIR* com características consideravelmente específicas, geralmente a quantidade de trabalhos encontrada é muito pequena. Por exemplo, alguns trabalhos relevantes na literatura destacam: a) o valor da indexação adequada e da recuperação de faces humanas para determinadas aplicações (por exemplo, para identificação de criminosos e cadastro de pessoas desaparecidas) [ANTANI; KASTURI; JAIN, 2002], [CHUA; ZHAO; KANKANHALLI, 2002]; b) a relevância da aplicação de conceitos relacionados a conjuntos nebulosos (esses conceitos serão definidos no capítulo 2) na linguagem usada para definir consultas, visando aproximar mais essas definições da linguagem normalmente utilizada pelas pessoas [RUI; HUANG; CHANG, 1999], [LOSADA et al., 2004]; c) a importância de medidas geométricas presentes em imagens (por exemplo, nas de faces humanas) durante a avaliação de similaridade efetuada por humanos [SANTINI; JAIN, 1999], [WORRING et al., 2002]. Mas apesar desses três aspectos serem considerados importantes, durante os levantamentos bibliográficos realizados não foi encontrado um sistema para recuperação de imagens por conteúdo que levasse em consideração todos esses aspectos.

## 1.2 OBJETIVOS DA DISSERTAÇÃO

Com base no que foi exposto até aqui, o principal objetivo desta dissertação é apresentar (no capítulo 5) o sistema denominado *MIGFIR* (*Metric Indexing and Geometric Features-based Image Retrieval*), que foi desenvolvido com os seguintes propósitos:

- possibilitar que alguns tipos de imagens, como por exemplo de faces humanas, sejam indexados e recuperados de maneira apropriada, empregando medidas geométricas das imagens, duas novas funções de similaridade que serão apresentadas e uma estrutura de indexação métrica;
- oferecer aos usuários uma grande variedade de recursos para a elaboração de consultas com diferentes níveis de complexidade, incluindo a possibilidade de uso de conceitos relacionados a conjuntos nebulosos.

Esta dissertação também apresentará a avaliação experimental do sistema desenvolvido, com o objetivo de comparar os resultados obtidos com o seu uso na execução de consultas aos resultados proporcionados por outras abordagens importantes.

### 1.3 ORGANIZAÇÃO DA DISSERTAÇÃO

O restante desta dissertação está organizado da seguinte forma:

- no capítulo 2 serão discutidos o conceito de conjunto nebuloso e conceitos relacionados, que foram importantes para o desenvolvimento do sistema *MIGFIR*;
- no capítulo 3 serão abordados aspectos relativos a estruturas de indexação métricas e serão apresentadas as estruturas que foram mais relevantes para o desenvolvimento do trabalho de mestrado;
- no capítulo 4 serão abordadas as principais ferramentas para recuperação de imagens por conteúdo que estão relacionadas ao *MIGFIR*;
- no capítulo 5 será apresentado de modo detalhado o sistema *MIGFIR*;
- no capítulo 6 serão mostrados e comparados os resultados dos experimentos que avaliaram o desempenho do sistema desenvolvido na recuperação de imagens que satisfazem consultas com diferentes graus de complexidade;
- por fim, no capítulo 7 serão apresentadas as considerações finais e as principais contribuições do projeto de mestrado desenvolvido, além de propostas para trabalhos futuros.

## 2 CONJUNTOS NEBULOSOS

### 2.1 CONSIDERAÇÕES INICIAIS

No mundo real, é comum encontrar situações em que não é possível afirmar que algo é estritamente verdadeiro ou falso, como exige a lógica bivalorada ou clássica proposta por Aristóteles. Por exemplo, considere um conjunto  $E$  constituído por valores de estaturas de homens altos. Logo, não se pode afirmar com precisão típica da lógica clássica que 1,80 metro é um valor de estatura que pertence ou não a  $E$ , já que esse conjunto de estaturas, assim como muitos outros, não possui um critério de pertinência precisamente definido.

Para atuar sobre esse tipo de situação, Lotfi A. Zadeh introduziu em [ZADEH, 1965] uma teoria sobre conjuntos em que a pertinência não é uma questão de afirmação ou negação, mas sim uma questão de atribuição de grau de pertinência. Esses conjuntos foram denominados conjuntos nebulosos ou difusos (*fuzzy sets*).

As próximas seções deste capítulo discutem o conceito de conjunto nebuloso e conceitos relacionados, que foram utilizados durante o desenvolvimento do sistema que será apresentado no capítulo 5.

### 2.2 CONJUNTOS CLÁSSICOS E CONJUNTOS NEBULOSOS

Na Matemática, um subconjunto qualquer  $S$  de um dado conjunto de referência  $C$  (também denominado *universo de discurso*) pode ser descrito basicamente de três formas:

- $S$  é descrito através da enumeração de todos os seus elementos, desde que seja um conjunto finito; ou seja, se  $s_1, s_2, \dots, s_n$  são os elementos de  $S$ , então:

$$S = \{s_1, s_2, \dots, s_n\}.$$

- $S$  é definido por meio de uma propriedade  $P$  (uma regra) satisfeita pelos seus elementos; ou seja, se para qualquer  $x \in C$ ,  $P(x)$  é uma proposição verdadeira ou

falsa, então  $S$  é definido como o conjunto composto por todos os elementos de  $C$  para os quais a proposição  $P(x)$  é verdadeira:

$$S = \{x / P(x)\}.$$

- $S$  é descrito por uma função geralmente chamada de função característica, que declara quais elementos do conjunto  $C$  são membros de  $S$  e quais não são; assim, se  $\mu_S$  denota a função característica de  $S$ , então:

$$\mu_S(x) = \begin{cases} 1 & \text{para } x \text{ pertencente a } S \\ 0 & \text{para } x \text{ não pertencente a } S \end{cases}$$

Observe que a função característica mapeia os elementos de  $C$  para elementos do conjunto  $\{0, 1\}$ , o que é formalmente expresso por:

$$\mu_S : C \rightarrow \{0, 1\}.$$

Logo, pode-se afirmar que  $S$  é um conjunto clássico (também denominado *crisp*), pois para cada  $x \in C$ , quando  $x$  é um membro de  $S$  tem-se que  $\mu_S(x) = 1$  e quando  $x$  não é um membro de  $S$  tem-se que  $\mu_S(x) = 0$ .

Pode-se dizer que a função característica de um conjunto clássico associa o valor 1 ou o valor 0 a cada elemento do universo de discurso desse conjunto. Essa função pode ser generalizada de modo que os valores associados aos elementos do universo de discurso pertençam a um intervalo especificado e, conseqüentemente, indiquem o grau de pertinência desses elementos ao subconjunto em questão. Uma função generalizada desta forma é chamada de função de pertinência nebulosa, e o conjunto definido por ela é um conjunto nebuloso.

Em geral, o intervalo de valores de funções de pertinência utilizado é o intervalo de números reais  $[0, 1]$ . Nesse caso, uma função de pertinência define um conjunto nebuloso associando elementos de um dado universo de discurso a valores em  $[0, 1]$ . Pode-se dizer então que quanto mais próximo de 1 está o grau de pertinência, mais o elemento associado a ele pertence ao conjunto nebuloso [BOJADZIEV; BOJADZIEV, 1995].

Note-se que os conjuntos clássicos podem ser vistos como um caso especial dos nebulosos, tal que os graus de pertinência são todos iguais a 1. Outra informação que deve

ser enfatizada é que cada conjunto nebuloso é definido completamente e exclusivamente por uma função de pertinência.

Uma notação freqüentemente empregada na literatura para denotar funções de pertinência difusas é a seguinte:

$$\mu_F : U \rightarrow [0, 1],$$

onde  $\mu_F$  é a função de pertinência difusa,  $F$  é o conjunto difuso obtido a partir de  $\mu_F$ , e  $U$  é o conjunto de referência de  $F$ . Assim, no conjunto  $F$  existe um valor  $\mu_F(x)$  para cada  $x$  pertencente a  $U$ .

Para obter um subconjunto de  $U$  de acordo com um grau de pertinência desejado, é necessário utilizar o conceito de  $\alpha$ -cut e sua variante, *strong  $\alpha$ -cut* [KLIR; YUAN, 1995]. Para um conjunto nebuloso  $F$ , o  $\alpha$ -cut (denotado por  ${}^\alpha F$ ) e o *strong  $\alpha$ -cut* (denotado por  ${}^{\alpha+} F$ ) são os conjuntos clássicos:

$${}^\alpha F = \{x \mid \mu_F(x) \geq \alpha\},$$

$${}^{\alpha+} F = \{x \mid \mu_F(x) > \alpha\},$$

onde  $x$  pertence ao universo de discurso de  $F$ , e  $\alpha$  é um valor pertencente a  $[0, 1]$ .

Considere, por exemplo,  ${}^{0+} F$  definido a partir do conjunto nebuloso  $F$  de números naturais próximos de 12. Logo, uma possível instanciação de  ${}^{0+} F$  seria:

$${}^{0+} F = \{9, 10, 11, 12, 13, 14, 15\},$$

com base na função de pertinência difusa  $\mu_F$  representada na Figura 2.1. Esta figura permite notar que o universo de discurso é discreto.

Contudo, um conjunto nebuloso também pode ser formado a partir de um universo contínuo. Para ilustrar essa possibilidade, considere a Figura 2.2, que exhibe a diferença entre a representação do conceito *adolescente* por meio de um conjunto clássico (Figura 2.2(a)) e por meio de um nebuloso (Figura 2.2(b)).

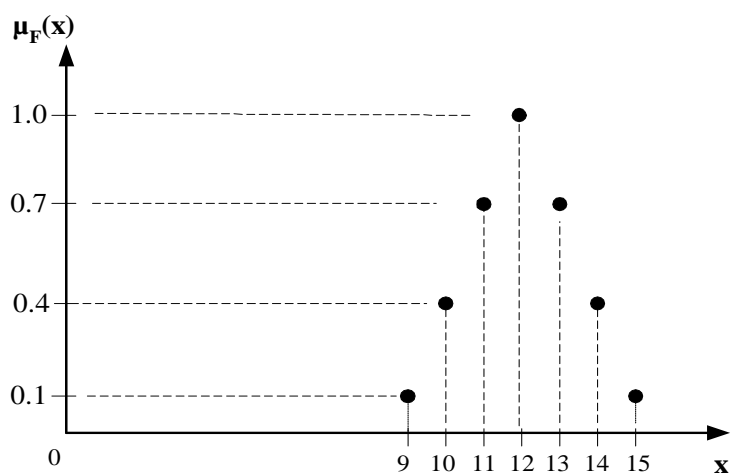


Figura 2.1 – Representação gráfica da função de pertinência difusa  $\mu_F$ .

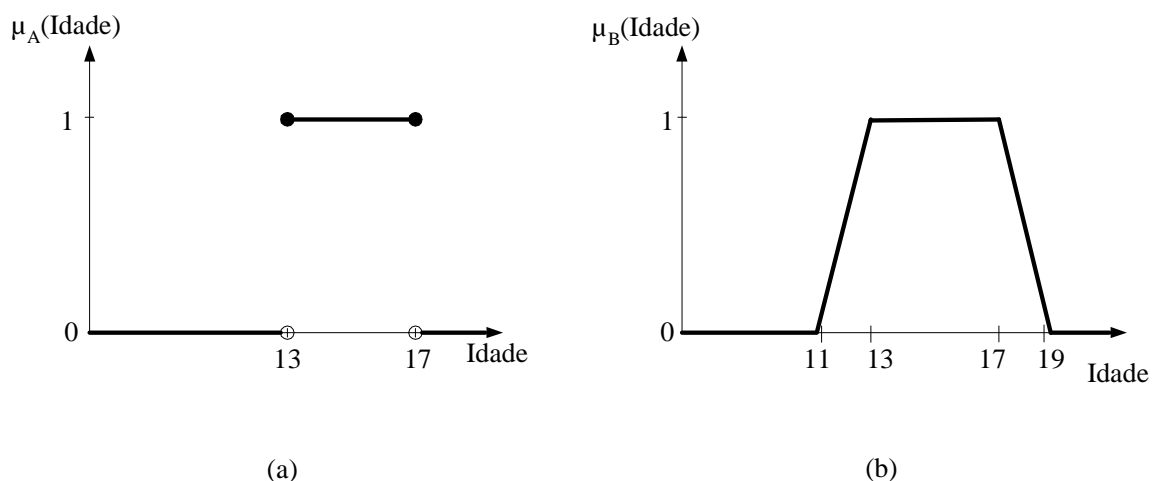


Figura 2.2 – Diferença na representação do conceito *adolescente*: (a) função característica do conjunto clássico *adolescente*; (b) função de pertinência do conjunto nebuloso *adolescente*.

Analisando-se o conjunto *crisp*  $A$  da Figura 2.2(a) e o conjunto nebuloso  $B$  da Figura 2.2(b), nota-se que  $A$  não exprime o conceito de adolescente de forma completa, pois um indivíduo com 12 anos e 11 meses é considerado totalmente incompatível com este conceito; por outro lado,  $B$  possibilita expressar que qualquer indivíduo com idade maior ou igual a 13 e menor ou igual a 17 anos é um adolescente, abaixo de 11 ou acima de 19 anos não é um adolescente, e no intervalo  $[11, 13)$  (respectivamente  $(17, 19]$ ) é considerado

quase um adolescente, com grau de pertinência diretamente proporcional à proximidade dos 13 (respectivamente dos 17) anos.

### 2.3 OPERAÇÕES BÁSICAS SOBRE CONJUNTOS NEBULOSOS

Nesta seção, abordam-se operações e conceitos básicos definidos por Zadeh em [ZADEH, 1965]. Para todos esses conceitos e operações, considere que  $F$  e  $G$  são conjuntos nebulosos e que  $x \in U$ , onde  $U$  é o universo de discurso.

A união é a primeira operação a ser apresentada. Assim, a união de dois conjuntos  $F$  e  $G$  com funções de pertinência  $\mu_F(x)$  e  $\mu_G(x)$ , respectivamente, é definida como um conjunto nebuloso  $R$  denotado por  $R = F \cup G$ , de modo que sua função de pertinência  $\mu_R(x)$  está relacionada a  $\mu_F(x)$  e  $\mu_G(x)$  por:

$$\mu_R(x) = \text{Max}[\mu_F(x), \mu_G(x)], \quad (2.1)$$

onde  $\text{Max}$  é a função que retorna o maior de seus dois argumentos; por exemplo,  $\text{Max}[0,3, 0,8] = 0,8$ .

A intersecção de  $F$  e  $G$ , cujas funções de pertinência são  $\mu_F(x)$  e  $\mu_G(x)$  (respectivamente), é definida como um conjunto nebuloso  $R$  especificado por  $R = F \cap G$ , de modo que sua função de pertinência  $\mu_R(x)$  está relacionada a  $\mu_F(x)$  e  $\mu_G(x)$  por:

$$\mu_R(x) = \text{Min}[\mu_F(x), \mu_G(x)], \quad (2.2)$$

onde  $\text{Min}$  é a função que retorna o menor de seus dois argumentos; por exemplo,  $\text{Min}[0,3, 0,8] = 0,3$ .

Já o conceito de complemento de um conjunto nebuloso  $F$ , denotado por  $F'$ , é definido por:

$$\mu_{F'}(x) = 1 - \mu_F(x). \quad (2.3)$$

A igualdade também é um conceito importante. Logo,  $F$  e  $G$  são considerados iguais, o que se representa por  $F = G$ , se e somente se para todo  $x \in U$ ,  $\mu_F(x) = \mu_G(x)$ .

Outros detalhes referentes às operações sobre conjuntos nebulosos podem ser encontrados, por exemplo, em [KLIR; YUAN, 1995], [TERANO; ASAI; SUGENO, 1991].

## 2.4 VARIÁVEIS LINGÜÍSTICAS

A definição de variável lingüística apresentada em [BOJADZIEV; BOJADZIEV, 1995] diz que se trata de um tipo de variável cujos valores são palavras ou sentenças em linguagem natural.

Já de acordo com Klir e Yuan [KLIR; YUAN, 1995], uma variável lingüística é uma quintupla  $(v, T, X, g, m)$ , onde  $v$  é o seu nome,  $T$  é o conjunto de termos lingüísticos da variável,  $X$  é o universo de discurso considerado,  $g$  é uma regra sintática para criar termos lingüísticos, e  $m$  é uma regra semântica que determina o significado de cada termo pertencente a  $T$  através de um conjunto nebuloso em  $X$ .

Para ilustrar o conceito de variável lingüística, considere a palavra “velocidade”. Pode-se definir que *Velocidade* é uma variável deste tipo, cujos possíveis termos lingüísticos são representados pelos conjuntos nebulosos *Negativa Alta*, *Negativa Baixa*, *Zero*, *Positiva Baixa* e *Positiva Alta*. Como os termos correspondem a conjuntos nebulosos, pode-se afirmar que cada termo é definido por uma função de pertinência apropriada.

A Figura 2.3 representa graficamente a variável lingüística *Velocidade*, mostrando seus termos e as respectivas funções de pertinência. Esta figura também permite notar que o conjunto de referência associado à variável é contínuo, isto é,  $U = [-maxvel, maxvel]$ .

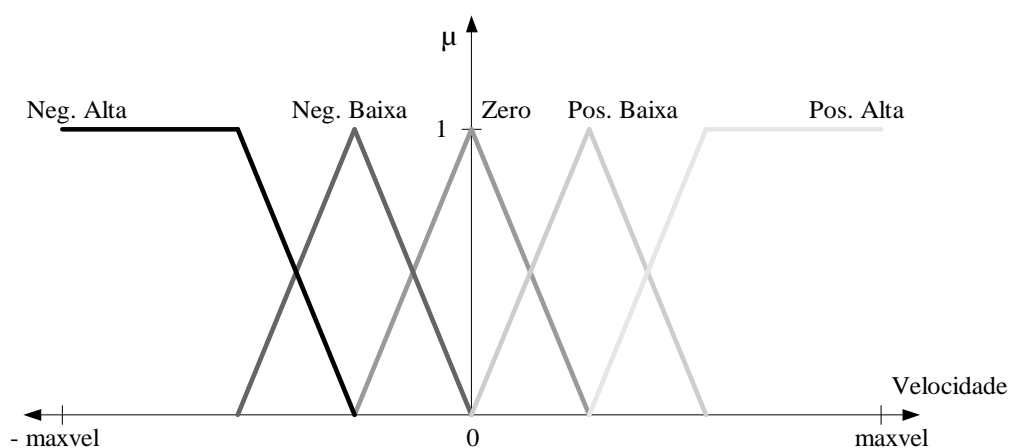


Figura 2.3 – Representação gráfica das funções de pertinência nebulosas que definem os termos lingüísticos da variável *Velocidade*.

Considera-se um conjunto de termos lingüísticos *completo* se para qualquer elemento pertencente ao universo de discurso existe nesse conjunto ao menos um termo que abrange



o elemento. A Figura 2.3 evidencia que  $T = \{Negativa\ Alta, Negativa\ Baixa, Zero, Positiva\ Baixa, Positiva\ Alta\}$  é completo para o universo  $U = [-maxvel, maxvel]$ .

## 2.5 PREDICADOS NEBULOSOS

Predicados nebulosos são elementos qualificadores empregados na composição de proposições nebulosas. Esse tipo de predicado é expresso através do uso de termos lingüísticos, já que ele está relacionado ao conceito de variável lingüística. Por exemplo, na proposição nebulosa *João é um homem alto*, o predicado está relacionado à variável lingüística *estatura de homens* e é expresso pelo termo *alto*. Este termo é definido por uma função de pertinência de modo que o valor da estatura de João é utilizado para obter o grau de pertinência desse indivíduo ao conjunto (esse grau representa o valor-verdade da proposição).

A Figura 2.4 ilustra uma função de pertinência que poderia definir o conjunto nebuloso *Alto*, e que conseqüentemente poderia ser empregada para obter o valor-verdade da proposição *João é um homem alto*. Nesta figura, os valores no eixo  $e$  correspondem a estaturas de indivíduos e os valores no eixo  $\mu$  significam os graus de pertinência (valores-verdade).

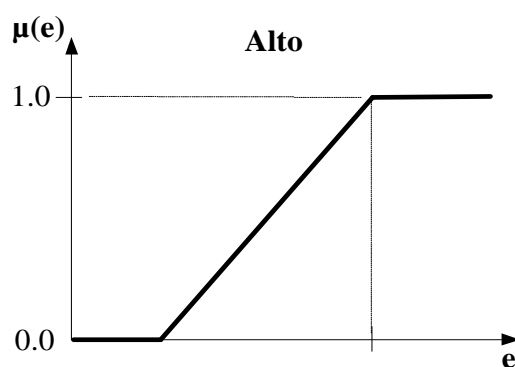


Figura 2.4 – Possível função de pertinência para o predicado nebuloso *Alto*.

O conceito de predicado nebuloso foi muito aplicado durante o desenvolvimento do sistema que será discutido no capítulo 5. O poder de expressão desse tipo de predicado também já foi utilizado em outros trabalhos relacionados à recuperação de imagens por

conteúdo, com ênfase para o modelo *Fuzzy Features Contrast*, proposto por Santini e Jain [SANTINI; JAIN, 1999].

## 2.6 MODELO FUZZY FEATURES CONTRAST

O modelo para medição de similaridade *Fuzzy Features Contrast (FFC)* foi desenvolvido por Santini e Jain [SANTINI; JAIN, 1999] com base em um modelo denominado *Feature Contrast (FC)*, proposto por Tversky em [TVERSKY, 1977]. O desenvolvimento de *FFC* ocorreu a partir da constatação de que predicados nebulosos poderiam ser empregados para estender a função *FC*, o que geraria um novo modelo de medição de similaridade capaz de abordar apropriadamente situações que não eram tratadas pela função *FC*.

Em [SANTINI; JAIN, 1999] foi escolhida a tarefa de avaliar a similaridade entre imagens de faces humanas para explicar a construção do modelo *Fuzzy Features Contrast*. Embora a avaliação de similaridade entre faces geralmente dependa de vários tipos de características, tais como cores, medidas geométricas, texturas, dentre outras, o uso de cinco medidas geométricas faciais foi considerado suficiente para a construção do modelo. Assim, cada uma das imagens do conjunto empregado foi representada por cinco predicados nebulosos, cujos valores-verdade foram determinados a partir das medidas geométricas e de funções de pertinência adequadas.

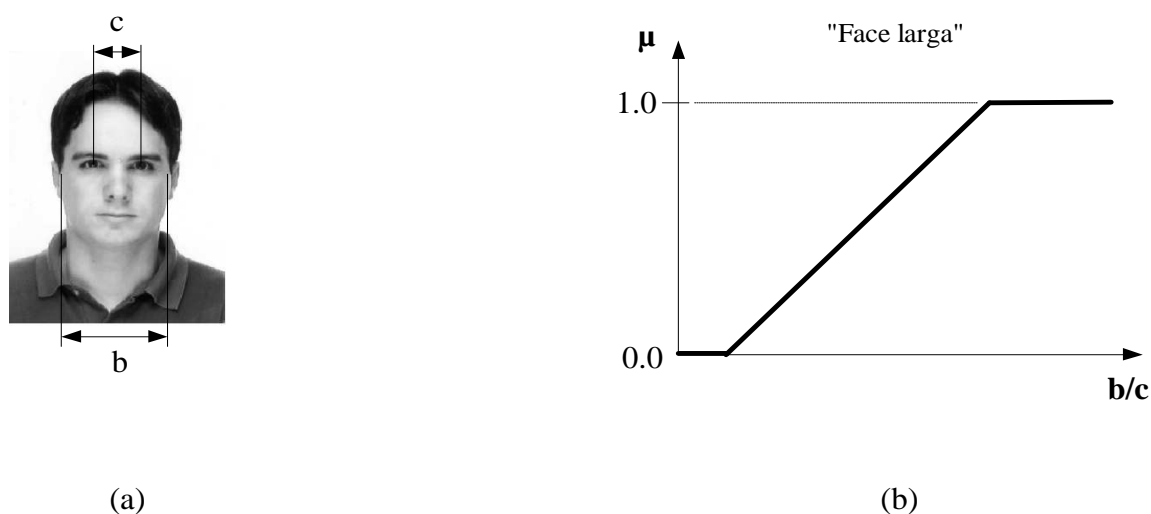


Figura 2.5 – Obtenção do valor-verdade do predicado nebuloso "Face larga": (a) face humana com duas medidas destacadas; (b) possível função de pertinência para determinar o valor-verdade do predicado.

Considere, por exemplo, o predicado nebuloso *Face larga* e a imagem de face humana mostrada na Figura 2.5(a). Para determinar o valor-verdade deste predicado para a imagem exibida, devem ser medidas a largura  $b$  da face e a distância  $c$  entre os olhos, e em seguida deve-se empregar uma função de pertinência como a que é ilustrada na Figura 2.5(b).

Na função de pertinência mostrada na Figura 2.5(b), note que a medida  $b$  é normalizada em relação à medida  $c$ .

Para que a similaridade baseada em medidas geométricas pudesse ser determinada utilizando-se predicados nebulosos, foi estabelecido que o conteúdo de cada imagem deveria ser representado através de um conjunto  $\mu(\lambda)$  cujos elementos seriam os valores-verdade correspondentes aos  $n$  predicados associados à imagem, ou seja:

$$\mu(\lambda) = \{\mu_1(\lambda), \dots, \mu_n(\lambda)\}, \quad (2.4)$$

onde  $\lambda$  representa o conjunto de medidas geométricas de uma dada imagem, e  $\mu_i(\lambda)$  é o valor-verdade do  $i$ -ésimo predicado nebuloso considerado, isto é,  $\mu_i : \lambda \rightarrow [0, 1]$ . Em outras palavras,  $\mu(\lambda)$  é o conjunto nebuloso de predicados verdadeiros (válidos) sobre as medidas de  $\lambda$ .

Para gerar a função *FFC*, também foi utilizada uma função  $f$  de cardinalidade e as definições da intersecção e da diferença entre dois conjuntos nebulosos  $\mu(\lambda)$  e  $\mu(\psi)$  de predicados válidos.

A função  $f$  referente à cardinalidade de um conjunto nebuloso  $\mu = \{\mu_1, \dots, \mu_n\}$  é dada por:

$$f(\mu) = \sum_{i=1}^n \mu_i. \quad (2.5)$$

A intersecção dos conjuntos  $\mu(\lambda)$  e  $\mu(\psi)$  é tradicionalmente dada por:

$$\mu_{\cap}(\lambda, \psi) = \{\min\{\mu_1(\lambda), \mu_1(\psi)\}, \dots, \min\{\mu_n(\lambda), \mu_n(\psi)\}\} \quad (2.6)$$

e a diferença foi estabelecida em [SANTINI; JAIN, 1999] como o conjunto nebuloso  $\mu_{-}(\lambda, \psi)$ , que pode ser determinado da seguinte maneira:

$$\mu_-(\lambda, \psi) = \{\max\{\mu_1(\lambda) - \mu_1(\psi), 0\}, \dots, \max\{\mu_n(\lambda) - \mu_n(\psi), 0\}\}. \quad (2.7)$$

Assim, a função *Fuzzy Features Contrast*, capaz de determinar a similaridade entre dois conjuntos de valores-verdade de predicados  $\mu(\lambda)$  e  $\mu(\psi)$  que representam duas imagens, foi definida como:

$$S(\lambda, \psi) = \sum_{i=1}^n \min\{\mu_i(\lambda), \mu_i(\psi)\} - \alpha \sum_{i=1}^n \max\{\mu_i(\lambda) - \mu_i(\psi), 0\} - \beta \sum_{i=1}^n \max\{\mu_i(\psi) - \mu_i(\lambda), 0\} \quad (2.8)$$

onde  $S(\lambda, \psi)$  indica o valor da similaridade entre as duas imagens e  $\alpha, \beta \geq 0$  são duas constantes obtidas experimentalmente.

Ainda em [SANTINI; JAIN, 1999] foi apresentada uma avaliação experimental envolvendo o modelo *FFC*, a distância *City Block* [DOMINGUES, 1982], a distância Euclideana [DOMINGUES, 1982] e o modelo *Thurstone-Shepard* [SHEPARD, 1987]. Nessa avaliação, em que as quatro funções citadas foram utilizadas para determinar a similaridade entre imagens de faces a partir de medidas geométricas, o modelo *FFC* proporcionou os melhores resultados.

Por fim, é relevante frisar que *FFC* é uma função de similaridade que não satisfaz os denominados axiomas métricos (que serão discutidos no capítulo 3).

## 2.7 CONSIDERAÇÕES FINAIS

O propósito deste capítulo foi apresentar o conceito de conjunto nebuloso e alguns conceitos relacionados, que serão utilizados em outros capítulos desta dissertação, principalmente quando forem discutidas as características do sistema *MIGFIR*.

Na seção 2.2 foram dadas as definições de conjunto clássico e de conjunto nebuloso; além disso, as diferenças entre esses dois conceitos foram estabelecidas.

Operações e conceitos definidos em [ZADEH, 1965] que foram considerados relevantes para este trabalho foram discutidos na seção 2.3.

Na seção 2.4 foram abordados os conceitos de variável lingüística e termos lingüísticos, que foram empregados na seção 2.5 para a definição de proposições e predicados nebulosos.

Finalmente, a seção 2.6 apresentou o modelo *Fuzzy Features Contrast*, que serviu como ponto inicial para o desenvolvimento de duas funções de similaridade que serão discutidas no capítulo 5.

## 3 MÉTODOS DE ACESSO MÉTRICOS

### 3.1 CONSIDERAÇÕES INICIAIS

Para bancos de dados tradicionais, existe uma grande quantidade de métodos de acesso disponíveis, como índices de nível único, índices multinível, *B-tree*, *B<sup>+</sup>-tree*, estruturas baseadas em *hashing*, etc [ELMASRI; NAVATHE, 2001].

No entanto, estruturas de indexação tradicionais geralmente não são adequadas para bases de dados multimídia, que normalmente contêm dados pertencentes a espaços métricos [CIACCIA; PATELLA; ZEZULA, 1997].

O projeto de métodos de acesso eficientes<sup>1</sup> para dados métricos tem sido, há algumas décadas, tema de estudos de diversos pesquisadores. Esses estudos têm gerado uma quantidade bastante considerável de estruturas.

Diante do fato de que uma das diretrizes para o desenvolvimento do sistema *MIGFIR* (detalhado no capítulo 5) foi a utilização de um método de acesso métrico (MAM), os objetivos deste capítulo são discutir aspectos relativos a estruturas de indexação métricas e apresentar as estruturas mais relevantes para o desenvolvimento de *MIGFIR*.

### 3.2 ESPAÇOS MÉTRICOS

Segundo Lima [LIMA, 1977], um espaço métrico  $M$  é um par  $(D, d)$ , tal que  $D \neq \emptyset$  é um conjunto de objetos e  $d$  é uma função de distância métrica sobre  $D$ .

---

<sup>1</sup> Nesta dissertação, quando se fala em eficiência de métodos de acesso a preocupação é com a eficiência de espaço (espaço ocupado pela estrutura de indexação) e de tempo (geralmente baseada na quantidade de acessos ao disco e no número de cálculos de distância entre objetos).

O conjunto  $D$  também é chamado de domínio de valores característicos ou domínio de chaves de indexação. Existe uma grande variedade de possibilidades para os elementos que constituem  $D$ , ou seja, esses elementos podem ser, por exemplo, palavras, pontos ou vetores.

A função de distância  $d: D \times D \rightarrow R^+$  é uma função que associa um número real  $d(a, b) \geq 0$  a cada par de elementos  $(a, b) \in D \times D$ , tal que esse número é acatado como a distância métrica de  $a$  a  $b$ . É importante enfatizar que quanto menor o valor dessa distância, mais similares são os elementos.

Para a função  $d$  ser uma métrica sobre um conjunto  $D$ , ela deve satisfazer para quaisquer  $a, b, c \in D$  os seguintes axiomas, geralmente chamados de axiomas métricos:

- simetria:  $d(a, b) = d(b, a)$ ;
- reflexividade:  $d(a, a) = 0$ ;
- positividade: se  $a \neq b$  então  $d(a, b) > 0$ ;
- desigualdade triangular:  $d(a, c) \leq d(a, b) + d(b, c)$ .

Um exemplo simples de espaço métrico é a reta matemática, ou seja, o conjunto  $R$  dos números reais com  $d: R \times R \rightarrow R^+$  dada por  $d(a, b) = |a - b|$  como função de distância métrica. O espaço Euclidiano  $R^n$  também é um exemplo de espaço métrico, que generaliza a reta.

Em [DOMINGUES, 1982], [NADLER; SMITH, 1993], [BAEZA-YATES et al., 1994], [VELTKAMP, 2001], [GAVIOLI et al., 2004], são abordadas algumas funções de distância e de similaridade (derivadas ou equivalentes a funções de distância) métricas muito úteis.

É possível definir dados métricos como aqueles que podem ser indexados em espaços métricos com a utilização de MAMs, que por sua vez empregam funções de distância ou de similaridade que satisfazem os axiomas apresentados nesta seção.

Uma característica importante desses dados é que em muitos casos não é simples definir sua dimensionalidade [TRAINA JR. et al., 2001].

### 3.3 TIPOS DE CONSULTAS

Consultas cujos critérios de busca são baseados em igualdade ou em ordem quase sempre são inapropriadas para bancos de dados compostos por dados métricos. Para esses bancos, consultas por similaridade normalmente são mais adequadas. Por exemplo, para uma base que armazena imagens, é muito comum um usuário querer obter aquelas que são similares (próximas) a uma determinada imagem segundo um critério específico.

As consultas por similaridade podem ser executadas quando se dispõe de funções apropriadas para determinar valores de similaridade. Em espaços métricos, tais funções devem ser métricas.

Em [CIACCIA; PATELLA; ZEZULA, 1997], [TRAINA JR. et al., 2002] relata-se que os dois tipos de consultas por similaridade executadas com maior frequência em espaços métricos são:

- Consulta por abrangência (*range query*): considere um espaço métrico  $M = (D, d)$ , onde  $D$  é o domínio de elementos e  $d$  é uma função métrica (ou, por simplicidade, *métrica*); dado um elemento de consulta  $e \in D$  e um raio de abrangência  $r(e) \geq 0$ , o objetivo de uma consulta por abrangência é encontrar todo elemento indexado  $e'$  cuja distância em relação a  $e$  é menor ou igual ao raio  $r(e)$ . Assim, uma *range query* ( $RQ$ ) pode ser expressa por:

$$RQ(e) = \{e' \mid d(e', e) \leq r(e)\}.$$

- Consulta aos  $k$  vizinhos mais próximos (*k-nearest neighbor query*): considere um espaço métrico  $M = (D, d)$ , onde  $D$  é o domínio de elementos e  $d$  é uma métrica; dado um elemento de consulta  $e \in D$  e um número inteiro  $k \geq 1$ , o objetivo de uma consulta aos  $k$  vizinhos mais próximos é encontrar os  $k$  elementos indexados que têm as menores distâncias em relação a  $e$ . Portanto, em uma *k-nearest neighbor query* ( $K-NNQ$ ) pretende-se encontrar o subconjunto resposta  $R \subseteq D$  tal que:

$$R = \{x \in D \mid |R| = k \text{ e } \forall x \in R, \forall y \in (D - R), d(e, x) \leq d(e, y)\}.$$



### 3.4 PRINCIPAIS MÉTODOS

As primeiras propostas para indexação em espaços métricos foram apresentadas em [BURKHARD; KELLER, 1973]. Foram estabelecidas três técnicas para tentar reduzir o número de comparações necessárias entre objetos durante buscas. Essas técnicas dividem conjuntos de dados métricos de acordo com um processo recursivo, que é materializado em uma estrutura de árvore.

Em [UHLMANN, 1991] foi proposta uma árvore métrica capaz de dividir os elementos de um conjunto de dados em dois subconjuntos disjuntos utilizando um elemento chamado de representativo. Aplicando a técnica denominada *ball decomposition*, o espaço métrico pode ser dividido em dois tal que metade dos objetos fica dentro do raio de uma bola métrica [DOMINGUES, 1982] e a outra metade fica fora. Uhlmann também definiu outra maneira de dividir o espaço métrico em dois, tomando dois objetos como representativos e em seguida associando cada um dos demais objetos ao representativo mais próximo. Esta forma de decomposição do espaço, chamada de *generalized hyperplane*, caracteriza o método de acesso *Generalized Hyperplane tree (GH-tree)* [UHLMANN, 1991].

Em [YIANILOS, 1993] foi introduzida a estrutura de indexação *VP-tree (Vantage Point tree)* para o problema de encontrar os  $k$  vizinhos mais próximos em espaços métricos. Em uma *VP-tree*, o principal aspecto é que cada nó divide o espaço em duas partes utilizando um representativo (denominado *vantage point*). Desse modo, os objetos com distância em relação ao *vantage point* inferior a um valor  $r$  pertencem ao subespaço interno, ao passo que os elementos com distância superior a  $r$  pertencem ao subespaço externo.

Outro MAM que emprega representativos para agrupar objetos com certa similaridade é a *FQ-tree (Fixed-Queries tree)*, apresentada em [BAEZA-YATES et al., 1994]. O principal objetivo dessa estrutura de indexação é minimizar a quantidade de comparações (cálculos de distância) de objetos e uma característica que a difere das árvores métricas anteriores é que ela utiliza apenas um representativo para todos os nós de um mesmo nível da árvore, tal que cada representativo pode ser ou não um elemento do conjunto indexado.

A Figura 3.1 ilustra um exemplo de *FQ-tree* em que a função métrica *Hamming* [BAEZA-YATES et al., 1994] é empregada para se obter a dissimilaridade (distância) entre

cadeias de caracteres de tamanho 4. Nesta figura, os números associados aos arcos da estrutura de árvore correspondem às distâncias entre os objetos que estão nas folhas e os representativos  $r_1$ ,  $r_2$  e  $r_3$ . Estas distâncias são utilizadas para se determinar quais subárvores são relevantes e quais não são durante as buscas. Neste exemplo considerou-se a capacidade de cada nó folha igual a 2.

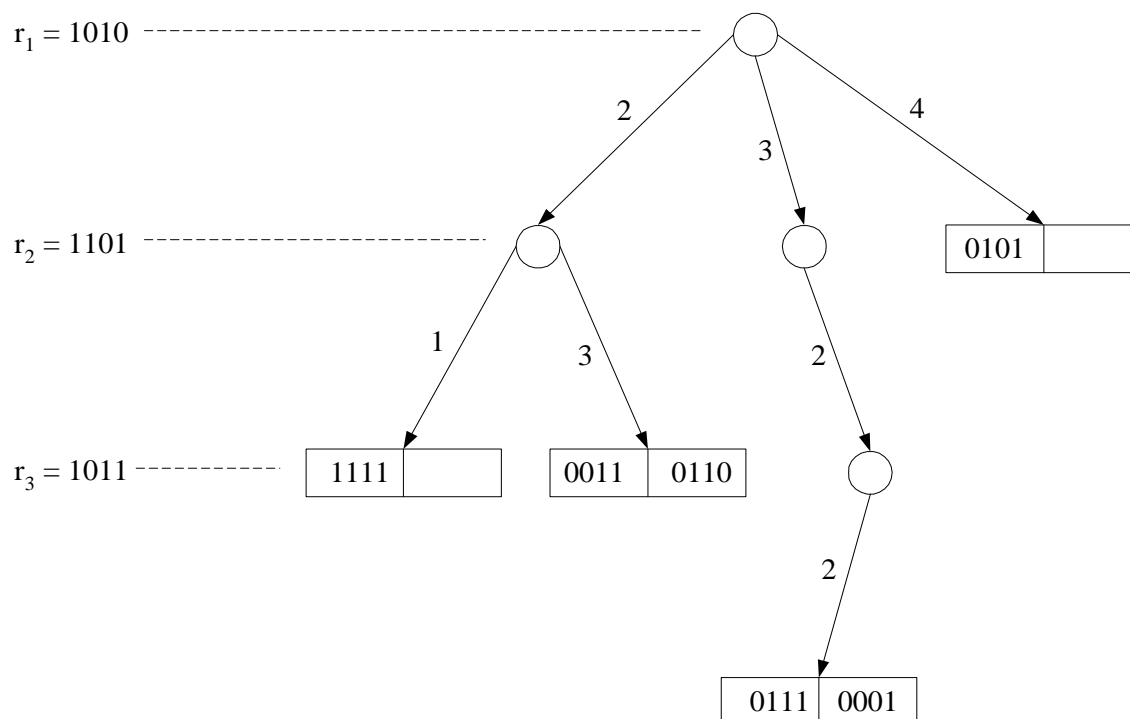


Figura 3.1 – *FQ-tree* usando a métrica *Hamming* para indexar cadeias de caracteres.

A *MVP-tree* (*Multi-Vantage Point tree*), uma estrutura métrica semelhante à *VP-tree*, foi proposta em [BOZKAYA; OZSOYOGLU, 1997]. Essa estrutura também emprega *vantage points* para dividir o espaço métrico, porém utiliza mais de um *vantage point* em cada nó, aumentando o *fanout* dos nós. Outra diferença entre *VP-tree* e *MVP-tree* é que esta última armazena as distâncias entre os objetos que estão nos nós folha e os *vantage points* de níveis mais altos, com o propósito de tentar reduzir os custos associados às buscas.

Até aqui, todos os métodos de acesso apresentados nesta seção são classificados como estáticos, já que uma vez construídos eles não suportam modificações como novas inserções ou exclusões. A árvore métrica proposta em [CIACCIA; PATELLA; ZEZULA, 1997], denominada *M-tree*, supera as limitações intrínsecas das estruturas estáticas.

A *M-tree* é um MAM com estrutura de árvore dinâmica e balanceada, que utiliza uma métrica para particionar os objetos de acordo com suas distâncias relativas. Cada nó tem tamanho fixo e corresponde a uma região limitada do espaço métrico. Existem dois tipos de nós possíveis: *folhas*, que armazenam os objetos de dados, e *internos*, que funcionam como roteadores durante operações de busca.

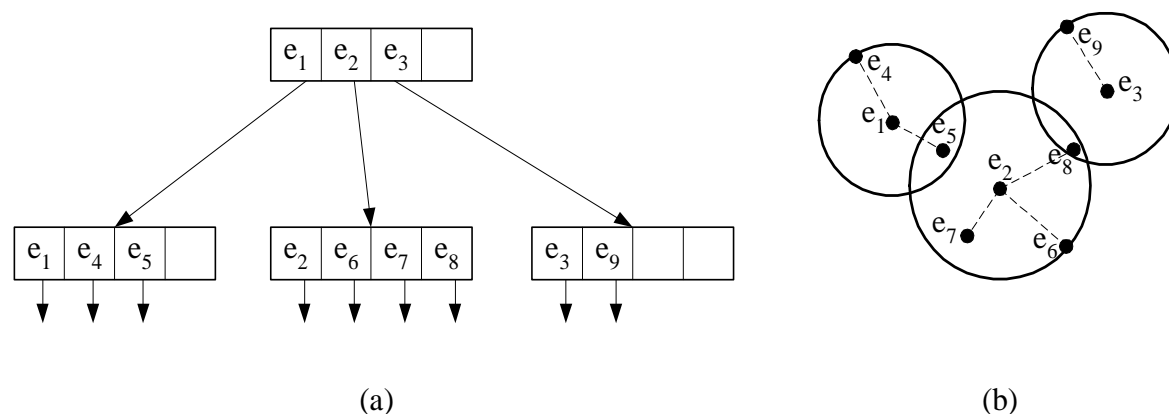


Figura 3.2 – Representação gráfica de uma *M-tree*: (a) estrutura hierárquica; (b) distribuição ilustrativa dos dados no espaço.

A Figura 3.2 ilustra um exemplo de *M-tree*. A estrutura hierárquica com um nó interno, que neste caso também é o nó raiz, e três nós folha é exibida na Figura 3.2(a). Note que cada objeto do nó interno é o representativo de uma folha. Já a Figura 3.2(b) ilustra a distribuição dos objetos indexados no espaço. Como os elementos  $e_5$  e  $e_8$  estão, cada um, sob o raio de abrangência de dois nós, diz-se que há sobreposição de nós nesta *M-tree*.

Os algoritmos da *M-tree* podem ser encontrados em [CIACCIA; PATELLA; ZEZULA, 1997]. O algoritmo de inserção, fundamentado na política denominada *mindist*, merece destaque. Ele parte do nó raiz da árvore e, para alcançar a folha apropriada, a subárvore escolhida em cada nível é aquela cujo representativo tem a menor distância em relação ao objeto a ser inserido.

A  $M^2$ -tree, introduzida em [CIACCIA; PATELLA, 2000], também é uma árvore de indexação dinâmica e balanceada. A característica que a distingue das estruturas anteriores é sua capacidade de atuar em situações para as quais mais de uma função de distância métrica é utilizada, tal que cada métrica é associada a uma das características consideradas dos objetos do conjunto de dados. Apesar da idéia central da  $M^2$ -tree ser bastante

interessante, seus desenvolvedores não apresentaram algoritmos e resultados relativos a esse método de uma forma tão detalhada como foi feito para a *M-tree*.

Com o intuito de indexar objetos pertencentes a espaços métricos e ao mesmo tempo medir e reduzir sobreposições de nós que podem ocasionar perda de desempenho, em [TRAINA JR. et al., 2000] foi apresentada a árvore de indexação denominada *Slim-tree*. Devido à sua enorme importância para o sistema *MIGFIR*, essa árvore será discutida de modo detalhado na subseção 3.4.1.

Em [SANTOS et al., 2001] foi proposto o *conceito OMNI*, cujo objetivo geral era permitir a construção de um MAM sobre outro método já existente. Assim, esse conceito tenta acelerar a execução de consultas por abrangência e consultas aos  $k$  vizinhos mais próximos através da redução do número de cálculos de distância. De acordo com os criadores do *conceito OMNI*, para conseguir tal redução são selecionados alguns objetos do espaço para serem representativos globais, chamados de *foci*, e em seguida são armazenadas as distâncias de cada um dos demais objetos em relação a todos os *foci*. Assim, durante o processamento de uma consulta essas distâncias armazenadas são usadas para “podar” o espaço de busca, isto é, para descartar o acesso a alguns nós.

A estrutura denominada *DF-tree* foi apresentada em [TRAINA JR. et al., 2002]. Ela é uma árvore semelhante à *Slim-tree*, mas que emprega um conjunto de representativos globais além dos representativos locais para aumentar sua “podabilidade”, isto é, para elevar a quantidade de objetos descartados com o uso da desigualdade triangular durante operações de busca.

Em [YAMAMOTO; BIAJIZ; TRAINA JR., 2003] foi proposta a *HCS-tree*, que também é um MAM dinâmico com estrutura de árvore balanceada. Esse método de acesso é fundamentado em um algoritmo de inserção denominado *HCS (Hybrid ChooseSubtree)*, que se baseia no algoritmo *ChooseSubtree* da *Slim-tree* e que aplica conceitos da lógica nebulosa para melhorar a distribuição dos objetos durante a inserção.

O algoritmo *HCS* é utilizado para escolher, para cada nível da árvore, a subárvore em que um objeto deve ser inserido de acordo com as políticas de inserção *mindist* (usada pela *M-tree*) e *minoccup* (empregada pela *Slim-tree*). Assim, esse algoritmo tenta obter a ocupação máxima dos nós e ao mesmo tempo tenta manter os objetos mais próximos no mesmo nó ou em nós próximos.

Duas variáveis lingüísticas, denominadas *Adjacência* ( $A_D$ ) e *Preenchimento* ( $P_R$ ), foram definidas para que o algoritmo *HCS* possa empregar os valores associados a elas na identificação da política de inserção apropriada para inserir um dado objeto em cada nível de uma árvore *HCS-tree*.

A *HCS-tree* é bastante similar à *Slim-tree*, tanto com relação à estrutura de dados como à maioria dos algoritmos. A utilização de conceitos da lógica nebulosa e de dois algoritmos novos apresentados em [YAMAMOTO, 2003] é que tornaram a *HCS-tree* um MAM diferente da *Slim-tree*. O primeiro algoritmo novo é o *HCS* e o outro é um algoritmo para tentar reduzir a cobertura dos nós com o objetivo de diminuir sobreposições, denominado *Reorganize Node*.

Já em [VIEIRA et al., 2004] foi apresentado o MAM dinâmico denominado *DBM-tree* (*Density-Based Metric tree*), cuja principal característica é a capacidade de minimizar a sobreposição entre nós com alta densidade por meio da não obrigatoriedade de que a estrutura seja balanceada. Outro aspecto que desperta atenção sobre essa árvore métrica é que, ao contrário de métodos como a *Slim-tree* e a *M-tree*, nela existe apenas um tipo de nó, ou seja, não há distinção entre nós índice e nós folha.

A avaliação experimental da *DBM-tree* produziu um resultado interessante: apesar das instâncias dessa árvore geralmente terem alturas superiores às de instâncias da *Slim-tree* e da *M-tree* que armazenam os mesmos dados, as árvores do tipo *DBM-tree* realizaram, na média, menos acessos ao disco do que as do tipo *Slim-tree* ou *M-tree*.

Para concluir a discussão sobre MAMs, na subseção a seguir serão apresentados diversos detalhes da *Slim-tree*.

### 3.4.1 Slim-tree

A *Slim-tree* é um MAM dinâmico com estrutura de árvore balanceada que divide o espaço de busca em regiões não disjuntas, correspondentes a cada subárvore, definidas por elementos do conjunto de dados chamados de representativos e por seus raios de cobertura, que estabelecem a abrangência de cada região (subárvore). Logo, a estrutura hierárquica é construída com o uso dos objetos representativos.

A *Slim-tree* cresce segundo o modelo *bottom-up*, isto é, cresce a partir dos nós folha (também denominados nós de dados) em direção ao nó raiz. Do mesmo modo que ocorre

nas árvores  $B^+$ , todos os elementos indexados do conjunto de dados são agrupados nas folhas, ao passo que os nós internos (também chamados de nós índice) são utilizados para direcionar a busca para as folhas apropriadas.

Cada nó pode conter um número máximo de objetos pré-determinado, já que independentemente de ser folha ou índice ele corresponde a uma página do disco de tamanho fixo. Mas as estruturas dos nós folha e índice são distintas.

A estrutura de uma folha é:

$$\text{Nófolha [vetor de } \langle S_i, \text{Oid}_i, d(S_i, \text{Rep}(S_i)) \rangle ],$$

tal que  $\text{Oid}_i$  é o identificador único do elemento  $S_i$  e  $d(S_i, \text{Rep}(S_i))$  é a distância métrica entre  $S_i$  e o elemento representativo  $\text{Rep}(S_i)$  desse nó.

Já a estrutura de um nó índice é:

$$\text{Nóíndice [vetor de } \langle S_i, d(S_i, \text{Rep}(S_i)), \text{Ptr}(TS_i), R_i, \text{NEntries}(\text{Ptr}(TS_i)) \rangle ],$$

onde:  $S_i$  é o representativo da subárvore apontada por  $\text{Ptr}(TS_i)$ ;  $R_i$  é o raio de cobertura de  $S_i$ ;  $d(S_i, \text{Rep}(S_i))$  é a distância métrica entre  $S_i$  e o representativo  $\text{Rep}(S_i)$  deste nó;  $\text{Ptr}(TS_i)$  é o ponteiro para o nó raiz da subárvore cujo representativo é  $S_i$ ; e  $\text{NEntries}(\text{Ptr}(TS_i))$  é a quantidade de entradas no nó apontado por  $\text{Ptr}(TS_i)$ .

A Figura 3.3 mostra a estrutura de cada tipo de nó e um exemplo de *Slim-tree*. Podem-se notar as diferenças entre os dois tipos de nós na Figura 3.3(a), enquanto a Figura 3.3(b) ilustra um exemplo em que os objetos indexados são sete palavras da língua inglesa e a função de distância métrica empregada é a  $L_{edit}$  [BAEZA-YATES et al., 1994].

Já a Figura 3.4 ilustra a *Slim-tree* de uma outra maneira, ou seja, a Figura 3.4(a) representa a estrutura hierárquica constituída por três nós índice e seis nós folha e a Figura 3.4(b) ilustra a disposição dos objetos indexados no espaço. Os elementos representativos, como por exemplo os objetos  $g$ ,  $a$  e  $n$ , também podem ser chamados de centróides. A Figura 3.4(b) também permite notar que há sobreposição de nós.

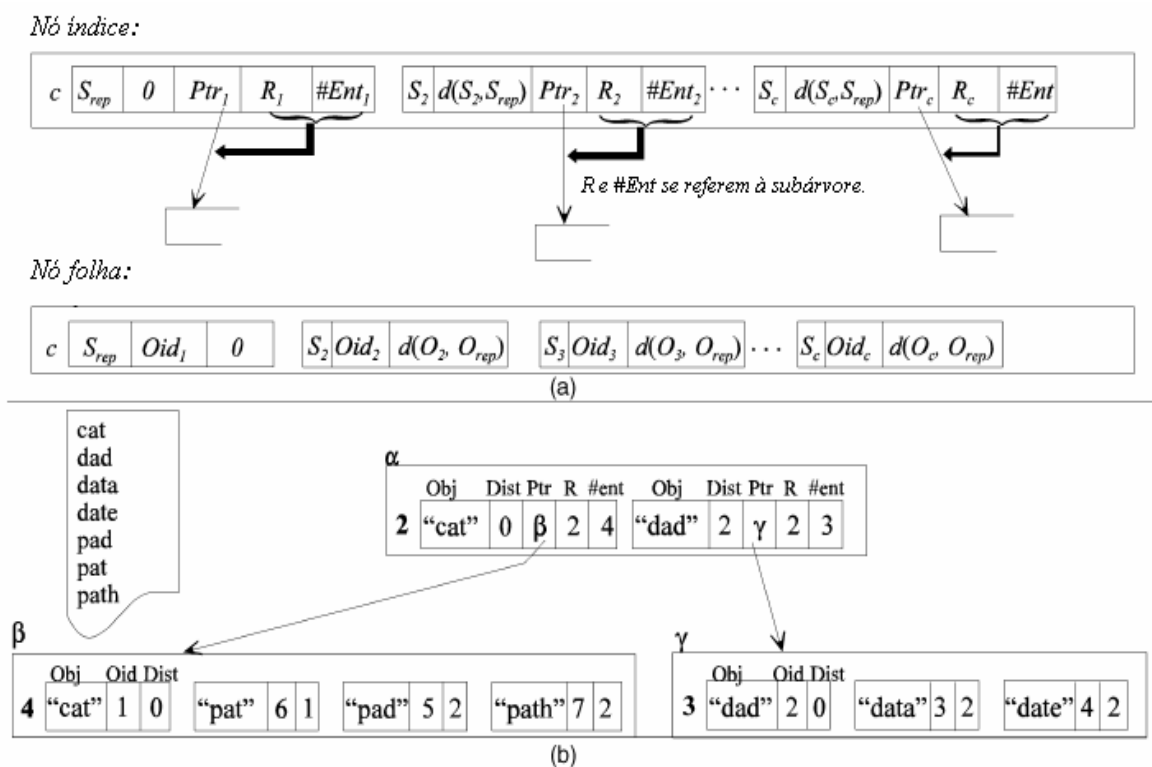


Figura 3.3 – Ilustração das estruturas dos nós da *Slim-tree*: (a) representação dos dois tipos de nós; (b) armazenando sete palavras da língua inglesa. Fonte: [TRAINA JR. et al., 2001].

Comparando a Figura 3.4 à Figura 3.2, pode-se observar que a estrutura da *Slim-tree* é bastante similar à da *M-tree*.

Com relação à inserção em uma *Slim-tree*, não é necessário ter todos os objetos do conjunto de dados no momento da construção da árvore; ou seja, novos objetos podem ser inseridos a qualquer momento.

Para a inserção de um novo objeto em uma *Slim-tree*, o algoritmo parte da raiz e, em cada nível, procura um nó com raio de abrangência que cubra o elemento. Se esse nó não existe, é selecionado aquele cujo centro está mais próximo do novo objeto. No entanto, se existe mais de um nó que pode cobri-lo, é executado um algoritmo chamado de *ChooseSubtree* para escolher um deles.

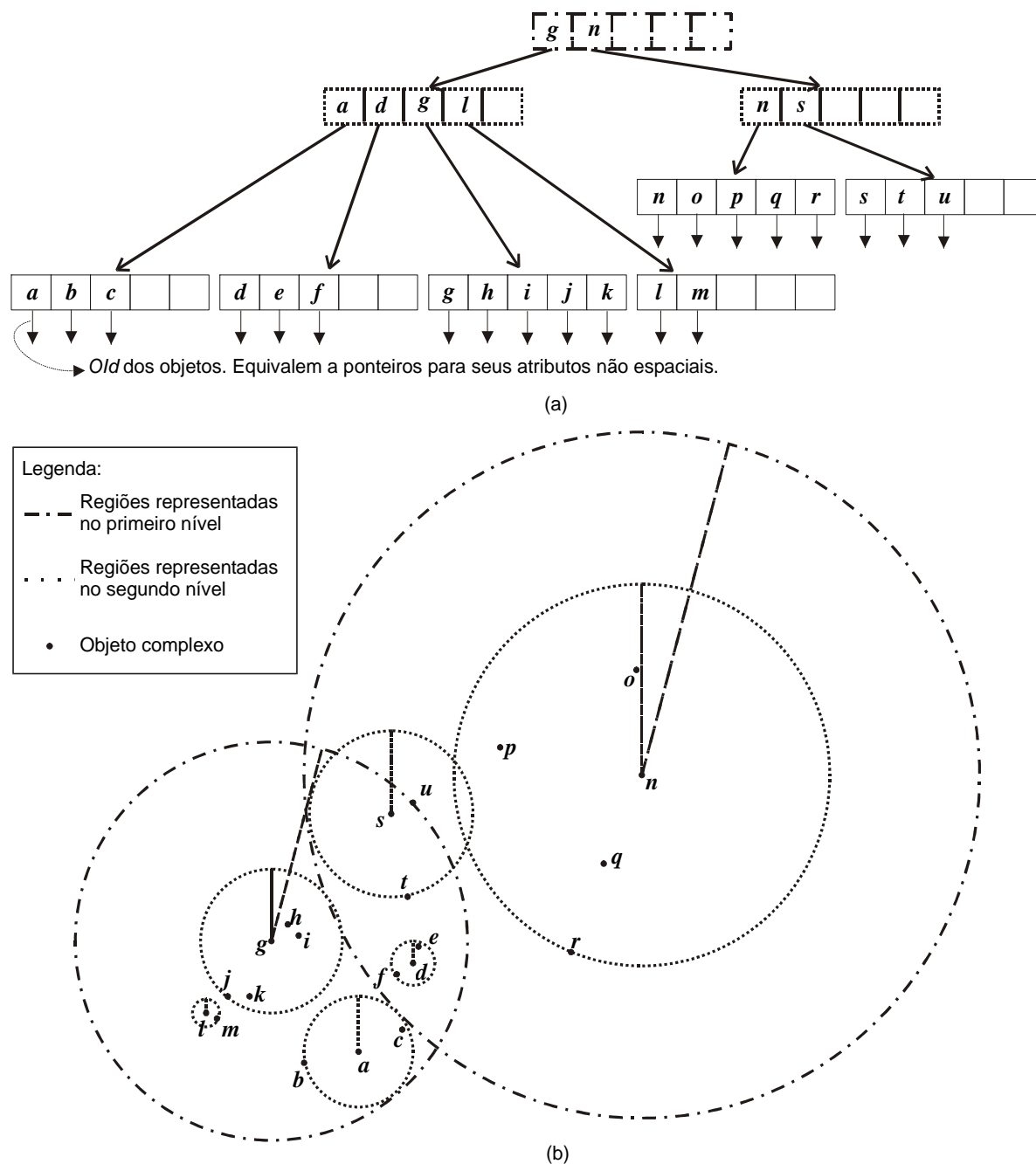


Figura 3.4 – Representação de uma *Slim-tree*: (a) estrutura hierárquica; (b) distribuição ilustrativa dos dados no espaço. Fonte: [SANTOS, 2003].

Existem três possibilidades para o algoritmo *ChooseSubtree*:

- *Distância Mínima (mindist)*: escolhe o nó candidato cujo objeto representativo está mais próximo do novo elemento;
- *Ocupação Mínima (minoccup)*: dentre os candidatos, escolhe o nó que contém a menor quantidade de elementos; é o critério de seleção padrão da *Slim-tree*;



- *Aleatório (random)*: escolhe um dos nós candidatos aleatoriamente; é o critério de seleção mais simples e menos custoso.

Algumas características de uma árvore são bastante dependentes do critério selecionado para o *ChooseSubtree*. Comparando-se os três critérios apresentados, o *minoccup* tende a produzir uma árvore com o maior percentual de ocupação (conseqüentemente mais baixa e com maior grau de sobreposição); o *mindist* tende a gerar uma árvore com o menor percentual de ocupação (conseqüentemente mais alta e com menor grau de sobreposição); e o critério *random* gera, na média, resultados intermediários.

Quando um novo objeto deve ser inserido em um nó que já está com sua capacidade esgotada, um algoritmo de divisão (*splitting*) precisa ser empregado para alocar um novo nó no mesmo nível da árvore e, em seguida, dividir os elementos entre os dois. Para a *Slim-tree*, existem três possíveis algoritmos de divisão de nós:

- *MST*: divide os objetos em dois grupos de acordo com a árvore do caminho mínimo (*Minimal Spanning Tree*) [KRUSKAL, 1956]; é o algoritmo padrão da *Slim-tree* por apresentar bom desempenho;
- *minMax*: a partir dos elementos de um nó cheio, esse algoritmo gera todos os possíveis pares de representativos e escolhe o par com a menor soma dos raios de cobertura; em seguida, distribui os elementos considerando o representativo que está mais próximo; o *minMax* é o algoritmo de divisão padrão da *M-tree*;
- *Aleatório (random)*: escolhe dois novos representativos aleatoriamente com base nos objetos que estão no nó cheio e, em seguida, divide os demais objetos segundo o critério do representativo mais próximo.

Desses três algoritmos de divisão, evidentemente o *random* é o que consome o menor tempo; mas normalmente ele produz os resultados menos satisfatórios. O *minMax* é o que proporciona a obtenção dos melhores nós, em termos de sobreposição mínima; porém, ele é o algoritmo com maior custo computacional, ou seja, é o mais lento. Já o *MST* apresenta custo computacional intermediário e gera resultados que na maioria das vezes são tão bons quanto os produzidos pelo *minMax*.

Com relação às consultas suportadas, a *Slim-tree* é capaz de executar os dois tipos de consultas por similaridade apresentados na seção 3.3 e algumas variações desses dois tipos,

como por exemplo uma forma de consulta por abrangência que também leva em consideração a quantidade de vizinhos mais próximos.

O algoritmo de busca, que requer como parâmetros de entrada um objeto de consulta  $Q$  e os parâmetros da consulta por similaridade escolhida, inicia pelo nó raiz através da comparação entre o objeto  $Q$  e todos os representativos do nível imediatamente inferior. Conseqüentemente, o algoritmo percorre, de forma recursiva, somente as subárvores que abrangem possíveis respostas, até alcançar um nó folha. Quando isto ocorre, o algoritmo obtém os graus de similaridade entre os objetos da folha e  $Q$ , para determinar quais desses elementos poderão estar na resposta da consulta. A execução desse algoritmo de busca é finalizada no momento em que não existem mais ramos contendo objetos candidatos à resposta.

A existência de sobreposições de nós aumenta a quantidade de cálculos de distância necessários para responder uma consulta, pois se percorre um número de caminhos maior que o necessário. Para medir e tentar reduzir as sobreposições de nós em árvores do tipo *Slim-tree*, pode-se usar o *fat-factor*, o *bloat-factor* e o algoritmo *Slim-down* [TRAINA JR. et al., 2000].

O *fat-factor* é uma medida utilizada para expressar a qualidade de uma árvore métrica com base na quantidade de sobreposições entre os seus nós. Assim, uma boa árvore tem pouquíssima ou idealmente nenhuma sobreposição. O valor do *fat-factor* ( $fat(T)$ ) pode ser obtido por meio de:

$$fat(T) = \frac{I_C - H * N}{N} \cdot \frac{1}{M - H} , \quad (3.1)$$

tal que  $T$  é uma *Slim-tree*,  $N$  é a quantidade de objetos indexados,  $H$  é a altura da árvore,  $M$  é o número de nós e  $I_C$  é a quantidade total de acessos ao disco necessários para a obtenção da resposta para uma consulta pontual (consulta por igualdade, relativa a um dado objeto).

O valor dado pela equação (3.1) sempre é um número real pertencente ao intervalo  $[0, 1]$ , de modo que 0 é o melhor valor (pois indica que uma árvore não possui sobreposições) e 1 é o pior (indica que há sobreposições envolvendo todos os nós).

O *bloat-factor* (ou *fat-factor relativo*) é uma medida que permite comparar duas árvores métricas diferentes que indexam os mesmos objetos. O valor do *fat-factor* relativo ( $rfat(T)$ ) pode ser determinado da seguinte maneira:

$$rfat(T) = \frac{I_C - H_{min} * N}{N} * \frac{1}{M_{min} - H_{min}}, \quad (3.2)$$

onde  $T$  é uma *Slim-tree*,  $N$  é o número de objetos indexados,  $H_{min}$  é a altura teórica mínima possível para  $T$ ,  $M_{min}$  é a quantidade teórica mínima de nós e  $I_C$  é a quantidade total de acessos ao disco necessários para a obtenção da resposta para uma consulta pontual.

O valor dado pela equação (3.2) sempre é um número real maior que zero, tal que quanto maior esse valor, pior é a situação da árvore em termos de sobreposições.

Já o algoritmo *Slim-down* pode ser aplicado sobre uma árvore previamente construída para avaliar possíveis sobreposições envolvendo nós folha. Caso haja uma sobreposição entre duas folhas, esse algoritmo compara os objetos desses nós com os dois representativos, com o propósito de determinar quais objetos podem ser transferidos de um nó para o outro para reduzir a sobreposição existente. Após a realização de todas as transferências de elementos possíveis, a execução do *Slim-down* é encerrada.

O *Slim-down* pode melhorar consideravelmente o desempenho das consultas por similaridade. No entanto, deve ser empregado com cautela, devido ao seu custo computacional relativamente alto.

Finalmente, é importante destacar que o algoritmo de consulta da *Slim-tree* se beneficia da estrutura de árvore e da propriedade de desigualdade triangular das funções métricas, com o intuito de descartar ramos inteiros de uma árvore que são irrelevantes para o resultado de uma consulta, minimizando assim as quantidades totais de acessos ao disco e de cálculos de distância.

### 3.5 CONSIDERAÇÕES FINAIS

Na literatura da área de Bancos de Dados encontra-se uma grande quantidade de estruturas de indexação que visam prover acesso eficiente a dados pertencentes a espaços métricos. Os objetivos deste capítulo foram definir conceitos associados a métodos de acesso métricos e discutir as estruturas mais importantes.

Na seção 3.2 foram definidos conceitos básicos como espaços, funções de distância, axiomas e dados métricos. Na seção 3.3 foram apresentados os dois tipos de consultas por similaridade mais utilizadas, isto é, consulta por abrangência e aos  $k$  vizinhos mais próximos. Já na seção 3.4 foram apresentados alguns métodos de acesso métricos importantes, dando ênfase para a *Slim-tree*.

## 4 SISTEMAS PARA RECUPERAÇÃO DE IMAGENS POR CONTEÚDO

### 4.1 CONSIDERAÇÕES INICIAIS

Atualmente é possível encontrar na literatura um grande número de sistemas para a tarefa de recuperar imagens por conteúdo, já que essa tarefa pode se basear em vários tipos de conteúdo (cores, texturas, medidas geométricas, anotações manuais, etc) e em diferentes abordagens. Porém, quando o levantamento bibliográfico se restringe a buscar ferramentas para *CBIR* com características consideravelmente específicas, geralmente o número de trabalhos publicados é pequeno.

O objetivo deste capítulo é apresentar os principais sistemas para *CBIR* que de alguma forma foram importantes para o desenvolvimento do sistema *MIGFIR*.

### 4.2 PRINCIPAIS SISTEMAS

#### 4.2.1 QBIC

O sistema *QBIC* (*Query By Image Content*) [NIBLACK et al., 1993] foi desenvolvido para possibilitar a execução de consultas baseadas em imagens-exemplo, em esboços de imagens (*sketches*) construídos pelos usuários, em cores e em padrões de textura. No caso de utilização de cores, o usuário pode estabelecer o percentual desejado de cada uma delas nos objetos a serem recuperados.

As representações de cores que esse sistema emprega são geralmente baseadas nas imagens inteiras nos espaços *RGB*, *YIQ*, *LAB* ou *Munsell* e em um histograma de cores *RGB* de 256 dimensões. As características de forma correspondem à área da forma, circularidade, excentricidade, dentre outras. Já com relação à textura, são empregadas

versões modificadas da espessura, da direcionalidade e do contraste originalmente propostos em [TAMURA; MORI; YAMAWAKI, 1978].

Para obter os objetos da base de dados que são similares a uma dada imagem de consulta, *QBIC* usa diferentes funções de similaridade. Se o tipo de característica considerada é cor, são empregadas a métrica Euclideana ponderada, a distância *average color* e a distância quadrática entre histogramas. Quando se considera a forma ou a textura, utiliza-se a métrica Euclideana ponderada.



Figura 4.1 – Tela do sistema *QBIC* exibindo imagens do resultado de uma consulta. Fonte: [FLICKNER et al., 1995]

Esse sistema foi um dos primeiros a aplicar indexação multidimensional para reduzir o tempo necessário para a execução de consultas. Ele emprega o método de acesso *R\*-tree* [BECKMANN et al., 1990] para indexar as representações de cor, de forma ou de textura.

Os resultados das consultas são apresentados em ordem decrescente, segundo os valores de similaridade em relação às imagens-exemplo.

Uma grande diferença entre *QBIC* e *MIGFIR* consiste no fato de que o primeiro não emprega um método de acesso métrico para indexar e recuperar imagens. Além disso, *QBIC* não utiliza medidas geométricas que poderiam ser extraídas das imagens como conteúdo, usa funções de similaridade diferentes das que são empregadas por *MIGFIR*, não permite a elaboração e execução de consultas textuais e nem a aplicação de predicados nebulosos.

Em [RUI; HUANG; CHANG, 1999], afirma-se que *QBIC* foi o primeiro sistema comercial destinado à recuperação de imagens baseada em conteúdo. Acima, a Figura 4.1 mostra uma tela desse sistema que exhibe os itens que constituem o resultado de uma consulta.

#### 4.2.2 Garlic

*Garlic* [CAREY et al., 1995] é um software capaz de integrar conteúdos de diferentes sistemas de bancos de dados e também de servidores que não são bases de dados. Assim, uma única consulta elaborada nele pode acessar dados de diversos subsistemas (por exemplo, *QBIC* pode atuar como um desses subsistemas).

A semântica associada às consultas elaboradas no sistema *Garlic* emprega conjuntos nebulosos. Por exemplo, no caso de uma consulta atômica (cuja expressão é a mais simples possível) para recuperar imagens contendo automóveis azuis, representada pela expressão *Cor\_do\_Automóvel = 'Azul'*, o resultado corresponderia a uma lista de imagens ordenada de tal maneira que seu primeiro elemento conteria o automóvel “mais azul”, isto é, aquele com grau de pertinência ao conjunto nebuloso *Azul* mais próximo de 1, seu segundo elemento conteria o segundo carro com grau de pertinência mais próximo de 1, e assim por diante.

Esse sistema permite combinar expressões atômicas através dos operadores booleanos *and* e *or* para compor consultas complexas. Quando se emprega o *or* para caracterizar uma disjunção entre expressões atômicas, *Garlic* aplica a função de agregação ou combinação *Max* (apresentada no capítulo 2, seção 2.3) para calcular os graus de pertinência ao conjunto nebuloso que representa a consulta complexa. Por outro lado, se o *and* é usado

para caracterizar uma conjunção, a função de agregação *Min* (também apresentada no capítulo 2, seção 2.3) é utilizada para determinar os graus de pertinência em relação à consulta como um todo.

Esse sistema também possibilita atribuir graus de relevância às subconsultas (expressões atômicas) que constituem uma consulta complexa. Porém, como a função de combinação para o caso de conjunções é a *Min*, a forma como esses níveis de relevância são utilizados pelo *Garlic* é diferente do modo que será apresentado para o sistema *MIGFIR*.

Sobre a indexação de imagens, *Garlic* não emprega um método de acesso para esse fim, já que ele é um sistema que gera resultados finais de consultas a partir de resultados parciais fornecidos por seus subsistemas. Mas esses subsistemas podem utilizar estruturas de indexação.

Portanto, apesar de *Garlic* e *MIGFIR* terem em comum algumas características importantes, após a leitura do capítulo 5 ficará claro que *MIGFIR* é um sistema com funcionamento e propósito bastante diferentes daqueles de *Garlic*.

### 4.2.3 MARS

*MARS (Multimedia Analysis and Retrieval System)* [ORTEGA et al., 1997] é um sistema para *CBIR* que possibilita a realização de consultas que combinem cor, forma e textura.

Histogramas *2D* sobre as coordenadas *HS* do espaço de cores *HSV* são utilizados para representar as cores. Para representar a forma são empregados descritores de Fourier, e para a textura são usados valores de espessura, contraste e direcionalidade.

Esse sistema também permite a utilização de operadores booleanos para a elaboração de consultas complexas, mas sem o uso conjunto de graus de relevância. As características empregadas durante a construção dessas consultas podem ser especificadas por meio da seleção de imagens-exemplo ou escolhendo-se, por exemplo, texturas de um conjunto de padrões disponíveis ou cores de uma paleta.

Um dos métodos para a recuperação de imagens emprega o conceito de conjunto nebuloso, já que, para uma determinada característica, as distâncias entre uma imagem de



consulta e aquelas de uma base de dados são interpretadas como graus de pertinência ao conjunto nebuloso dos objetos que satisfazem tal característica. No entanto, o outro método de recuperação, denominado *probabilistic boolean retrieval*, baseia-se na medição de probabilidades no lugar de graus de pertinência.

As funções de distância aplicadas por *MARS* são a métrica Euclideana e uma soma ponderada. Não foi encontrado na literatura um trabalho que explique o tipo de método de acesso que esse sistema utiliza para indexar as imagens.

Com relação às formas de consulta, *MARS* não executa os mesmos tipos de consultas por similaridade suportadas por *MIGFIR*. Além disso, *MARS* não emprega medidas geométricas que poderiam ser extraídas das imagens e usa funções de similaridade diferentes das que são empregadas por *MIGFIR*.

#### 4.2.4 VisualSEEK

O sistema *VisualSEEK* [SMITH; CHANG, 1997] inicialmente decompõe cada imagem da base de dados considerada em regiões de cores igualmente dominantes. Para cada região, os valores que representam as cores e as propriedades espaciais são armazenados, para serem considerados durante a execução de consultas. A técnica denominada *back-projection* é empregada para a extração dessas regiões.

Para formular uma consulta, o usuário deve esboçar as regiões desejadas, posicionando-as e dimensionando-as em uma tela de edição gráfica e também escolhendo uma cor para cada região. Assim, quando *VisualSEEK* executar essa consulta serão recuperadas as imagens mais similares ao esboço.

Para determinar a similaridade entre uma dada imagem-exemplo e qualquer imagem da base de dados considerada, uma das funções de distância empregadas é a métrica Euclideana. Para combinar valores de distância associados a diferentes regiões, calcula-se a média ponderada desses valores.

Com relação à indexação das imagens, *VisualSEEK* utiliza os métodos de acesso multidimensionais *quadtree* [SAMET, 1984] e *R-tree* [GUTTMAN, 1984]. Ele não emprega estruturas de indexação métricas. Além disso, *VisualSEEK* não utiliza medidas geométricas que poderiam ser extraídas das imagens, emprega funções de similaridade

diferentes das que *MIGFIR* usa, não permite a elaboração e execução de consultas textuais e nem a utilização de predicados nebulosos.

#### 4.2.5 Blobworld

*Blobworld* [CARSON et al., 1999] é um sistema que possibilita a utilização de textura, cor e forma nas consultas. A textura é descrita basicamente por meio de contraste, cores são representadas por histogramas no espaço *LAB* e forma é descrita por área, excentricidade e orientação.

Query image: 130065      Query blob

Querying from 35000 images (2000 returned by the filter).

blob and feature importance:				
blob (overall)	color	texture	location	shape
blob 2	very	not	not	very

1: 77045 (score = 0.97)      [New query](#)

2: 107090 (score = 0.96)      [New query](#)

3: 374088 (score = 0.96)      [New query](#)

4: 228093 (score = 0.96)      [New query](#)

5: 205042 (score = 0.96)      [New query](#)

6: 26052 (score = 0.96)      [New query](#)

Figura 4.2 – Interface de consulta do sistema *Blobworld*. Fonte: [CARSON et al., 1999]

Para elaborar uma consulta, o usuário deve escolher uma imagem inicial e, em seguida, selecionar regiões (*blobs*) dessa imagem que lhe interessam. A cada região podem ser associados alguns qualificadores nebulosos, com o propósito de indicar o grau de relevância da textura, da cor e da forma da região atual para a consulta.

A Figura 4.2 exibe a interface do sistema *Blobworld* que contém a imagem e os *blobs* de consulta, alguns qualificadores nebulosos que indicam níveis de relevância e as imagens recuperadas após a execução da consulta por similaridade.

As dissimilaridades entre pares de histogramas são calculadas através do uso da distância quadrática [CARSON et al., 1999], enquanto a métrica Euclideana é empregada para o caso de texturas. Para obter um valor que indique a dissimilaridade final entre duas imagens, realiza-se uma combinação dos valores relativos a cada tipo de característica.

Com relação à indexação, a estrutura *R\*-tree* é utilizada para indexar os histogramas. O fato de *Blobworld* não empregar métodos de acesso métricos é uma diferença em relação a *MIGFIR*. Outras diferenças relevantes são que *Blobworld* não usa medidas geométricas como conteúdo das imagens, utiliza funções de similaridade diferentes das que *MIGFIR* emprega, e não permite que consultas textuais sejam elaboradas e executadas.

#### 4.2.6 Sistema de Berretti, Del Bimbo, e Pala

Berretti, Del Bimbo e Pala apresentaram em [BERRETTI; DEL BIMBO; PALA, 2000] um protótipo de sistema que utiliza o método de acesso métrico *M-tree* para indexar e recuperar imagens representadas por características de forma. Para a tarefa de recuperação, esse protótipo possibilita a execução de consultas por similaridade.

Conforme detalhado pelos autores, cada forma é dividida em unidades chamadas de *tokens*, e cada uma dessas unidades, que corresponde a um conjunto de atributos perceptíveis, é representada por um ponto em um espaço multidimensional. Esses *tokens* são inseridos em uma árvore *M-tree*.

Uma métrica é empregada para determinar as dissimilaridades entre *tokens*, e uma função não métrica estabelecida como uma combinação de distâncias entre *tokens* é aplicada para obter as dissimilaridades entre formas.

O protótipo de sistema gerencia cada consulta baseada em exemplo, que envolve uma dada forma, como uma combinação de múltiplos *tokens*. Assim, inicialmente os *tokens* de uma forma *F* são sucessivamente considerados por um índice como objetos de consulta, e em seguida os resultados parciais são combinados para a geração do resultado final.

Ainda em [BERRETTI; DEL BIMBO; PALA, 2000] foram mostrados resultados obtidos durante os experimentos que visaram analisar o desempenho do sistema em termos da eficiência da indexação. Esses resultados evidenciaram que o emprego da *M-tree* propiciou uma economia média de 85% a 90% nas leituras de blocos da memória secundária, em comparação com a varredura seqüencial.

Em uma comparação com *MIGFIR*, é importante observar que o sistema de Berretti, Del Bimbo e Pala não utiliza medidas geométricas que poderiam ser extraídas das imagens, emprega funções de similaridade diferentes, não suporta a elaboração e a execução de consultas textuais, e não permite o uso de predicados nebulosos.

#### 4.2.7 RETIN

O sistema *RETIN* (em francês: *REcherche et Traque INteractive d'images*) [FOURNIER; CORD; PHILLIP-FOLIGUET, 2001] foi desenvolvido para permitir a indexação e a recuperação de imagens através do uso de cor e textura. No caso da cor, o usuário pode optar por um ou mais dos seguintes espaços: *RGB* (tradicional ou normalizado), *CIELAB*, *CIELUV* e *HSV*. A textura é determinada com o emprego de filtros de Gabor [GABOR, 1946]. No entanto, esse sistema não utiliza medidas geométricas que poderiam ser extraídas das imagens.

Inicialmente, *RETIN* efetua uma classificação automática empregando uma rede neural Kohonen [KOHONEN, 1990] para fazer o agrupamento dos dados de uma determinada base de dados. A fase de aprendizado dessa rede Kohonen utiliza um conjunto de treinamento formado por uma grande quantidade de *pixels* extraídos do banco de imagens.

Para cada imagem, esse sistema usa uma distribuição estatística para os agrupamentos previamente encontrados como forma de representar o conteúdo. Essa representação, chamada de assinatura, é utilizada como um índice para a imagem. Assim, cada *pixel* é classificado pela rede neural e o conjunto de frequências de cada agrupamento compõe a

assinatura. Não foram encontradas informações sobre o emprego de algum método de acesso métrico ou multidimensional em *RETIN*.

Há a possibilidade de elaborar consultas baseadas em exemplos globais ou parciais. Uma consulta com parâmetro global considera a imagem-exemplo como um todo no momento de recuperar imagens similares, e uma com parâmetro parcial permite que sejam considerados objetos ou regiões da imagem-exemplo para a tarefa de recuperação. Diferentemente de *MIGFIR*, o sistema *RETIN* não permite a elaboração e a execução de consultas textuais e nem o uso de conceitos da teoria dos conjuntos difusos nas consultas.

Com relação às funções utilizadas para calcular as dissimilaridades entre distribuições estatísticas de imagens, *RETIN* aplica duas métricas derivadas das distâncias *Minkowski* [LIMA, 1977], que inclusive permitem a atribuição de graus de relevância às imagens.

Os resultados de testes mostrados em [FOURNIER; CORD; PHILLIP-FOLIGUET, 2001] correspondem apenas a uma avaliação qualitativa do sistema, em termos de precisão na recuperação; não foram exibidos resultados comparativos correspondentes à eficiência, medida por exemplo em termos de tempo de execução de consulta.

#### 4.2.8 CbPACS

Bueno *et al.* apresentaram em [BUENO et al., 2002] um sistema denominado *cbPACS* (*Content-based Picture Archiving and Communication System*) com o intuito de viabilizar a recuperação eficiente de objetos de uma base de imagens médicas.

Esse sistema utiliza a árvore métrica *Slim-tree* para indexar vetores de características (representações numéricas concisas que denotam os detalhes mais relevantes das mídias), que são construídos a partir das distribuições de cores das imagens por meio de histogramas normalizados e histogramas métricos. As cores são extraídas por um módulo de processamento de imagens.

Para executar de forma eficiente consultas por abrangência ou aos  $k$  vizinhos mais próximos, *cbPACS* emprega os índices do tipo *Slim-tree* previamente construídos. A capacidade desse sistema está sendo ampliada, para que ele suporte a combinação de cores com texturas e formas.

Embora o sistema *MIGFIR* utilize o mesmo método de acesso métrico empregado por *cbPACS*, há algumas diferenças relevantes entre esses dois sistemas: *cbPACS* não usa medidas geométricas como conteúdo, utiliza funções de similaridade diferentes das usadas por *MIGFIR*, não permite a elaboração e a execução de consultas textuais e nem a aplicação de conceitos da teoria dos conjuntos nebulosos nas consultas.

### 4.3 CONSIDERAÇÕES FINAIS

Este capítulo teve como propósito apresentar os principais sistemas para *CBIR* que foram relevantes durante o desenvolvimento de *MIGFIR*. Alguns dos sistemas discutidos na seção 4.2 estão relacionados a *MIGFIR* por também empregarem métodos de acesso métricos ou multidimensionais para acelerarem a recuperação de imagens, enquanto outros deles são considerados trabalhos relacionados pelo fato de também permitirem a utilização de conceitos da teoria dos conjuntos nebulosos durante a formulação e/ou execução de consultas por similaridade.

Não está no escopo desta dissertação apresentar abordagens pouco relacionadas ao sistema *MIGFIR*. Mas há bons trabalhos na literatura que possibilitam ter uma noção da quantidade e da diversidade de sistemas para *CBIR*, como por exemplo [ASLANDOGAN; YU, 1999], [RUI; HUANG; CHANG, 1999], [SMEULDERS et al., 2000], [ANTANI; KASTURI; JAIN, 2002], [FEND; SIU; ZHANG, 2003].

## 5 O SISTEMA MIGFIR

### 5.1 CONSIDERAÇÕES INICIAIS

Com base nos aspectos apresentados nos capítulos anteriores como motivadores do projeto de mestrado, foi desenvolvido o sistema denominado *MIGFIR* (*Metric Indexing and Geometric Features-based Image Retrieval*). Um dos objetivos fundamentais desse sistema, cuja primeira versão foi apresentada em [GAVIOLI; BIAJIZ; MOREIRA, 2005], é possibilitar que determinados tipos de imagens (como de faces humanas, por exemplo) sejam indexados e recuperados com eficiência e precisão<sup>2</sup> satisfatórias utilizando-se medidas geométricas das imagens, duas novas funções de similaridade que serão apresentadas neste capítulo, e a estrutura de indexação *Slim-tree*. O outro objetivo fundamental é permitir que os usuários configurem importantes parâmetros durante a definição dos dois tipos de consultas que serão discutidos e também possibilitar que conceitos como conjuntos e predicados nebulosos sejam empregados para tornar as expressões de busca de um dos tipos de consultas mais próximas da linguagem que os humanos normalmente usam.

Este capítulo apresentará de modo detalhado o sistema *MIGFIR*. Durante essa apresentação, dois módulos desse sistema receberão maior destaque:

- inicialmente, o módulo de *MIGFIR* que possibilita definir expressões de busca para consultas com diversos níveis de complexidade, atuando como interface entre os usuários e os algoritmos de processamento de consultas;

---

<sup>2</sup> Nesta dissertação, a medida de desempenho *precisão* significa o quão próxima está a ordenação (por similaridade em relação a um objeto de consulta) de um conjunto de imagens realizada por *MIGFIR* da ordenação do mesmo conjunto efetuada por um ou mais humanos, tal que o grau de precisão de *MIGFIR* é diretamente proporcional ao grau de proximidade entre as duas ordenações.

- em seguida, o módulo mais importante do sistema, que é a parte responsável pela criação de objetos que representem imagens e funções de similaridade métricas, pelo gerenciamento de índices métricos e pela execução de todas as consultas que tenham sido definidas no módulo de interface.

## 5.2 ARQUITETURA DO SISTEMA

O sistema *MIGFIR* foi configurado para indexar e recuperar conjuntos de imagens de faces humanas devido a importantes características que estão associadas a esse tipo de imagem (essas características foram apresentadas no capítulo 1, seção 1.1).

Porém, é importante destacar que esse sistema pode ser ajustado, em um intervalo de tempo relativamente curto, para atuar também com outros conjuntos de imagens.

*MIGFIR* foi implementado em linguagem *C++*, utilizando-se o ambiente de desenvolvimento *Borland C++ Builder 5*. Para simplificar sua apresentação, pode-se dividi-lo em três módulos principais, denominados *MIGFIR\_PredicateGenerator*, *MIGFIR\_QueryInterfaces* e *MIGFIR\_QueryImplementation*.

A arquitetura de *MIGFIR* está ilustrada na Figura 5.1. Nela, observe inicialmente que a partir de um conjunto de imagens  $C$  pode-se efetuar a extração dos valores de suas características geométricas através do uso de uma ferramenta apropriada, com o intuito de permitir a representação de cada imagem por meio de suas medidas geométricas.

Em seguida, a partir do conjunto de medidas extraídas dos elementos de  $C$ , o módulo *MIGFIR\_PredicateGenerator* (Figura 5.1, retângulo de linhas tracejadas verdes) possibilita determinar os valores-verdade dos predicados nebulosos que também podem representar cada imagem (chamados neste trabalho de *predicados válidos* ou *verdadeiros*).

O módulo *MIGFIR\_QueryImplementation* (Figura 5.1, retângulo de linhas tracejadas azuis) pode utilizar os conjuntos  $C$ ,  $M_C$  e  $P_C$  para criar e gerenciar os índices métricos que são utilizados durante o processamento de consultas por similaridade complexas, isto é, consultas baseadas em múltiplas imagens-exemplo ou baseadas em múltiplos predicados nebulosos.



Ainda na Figura 5.1, note que o módulo *MIGFIR\_QueryInterfaces* (retângulo de linhas tracejadas vermelhas) atua como interface entre os usuários e os algoritmos de processamento de consultas.

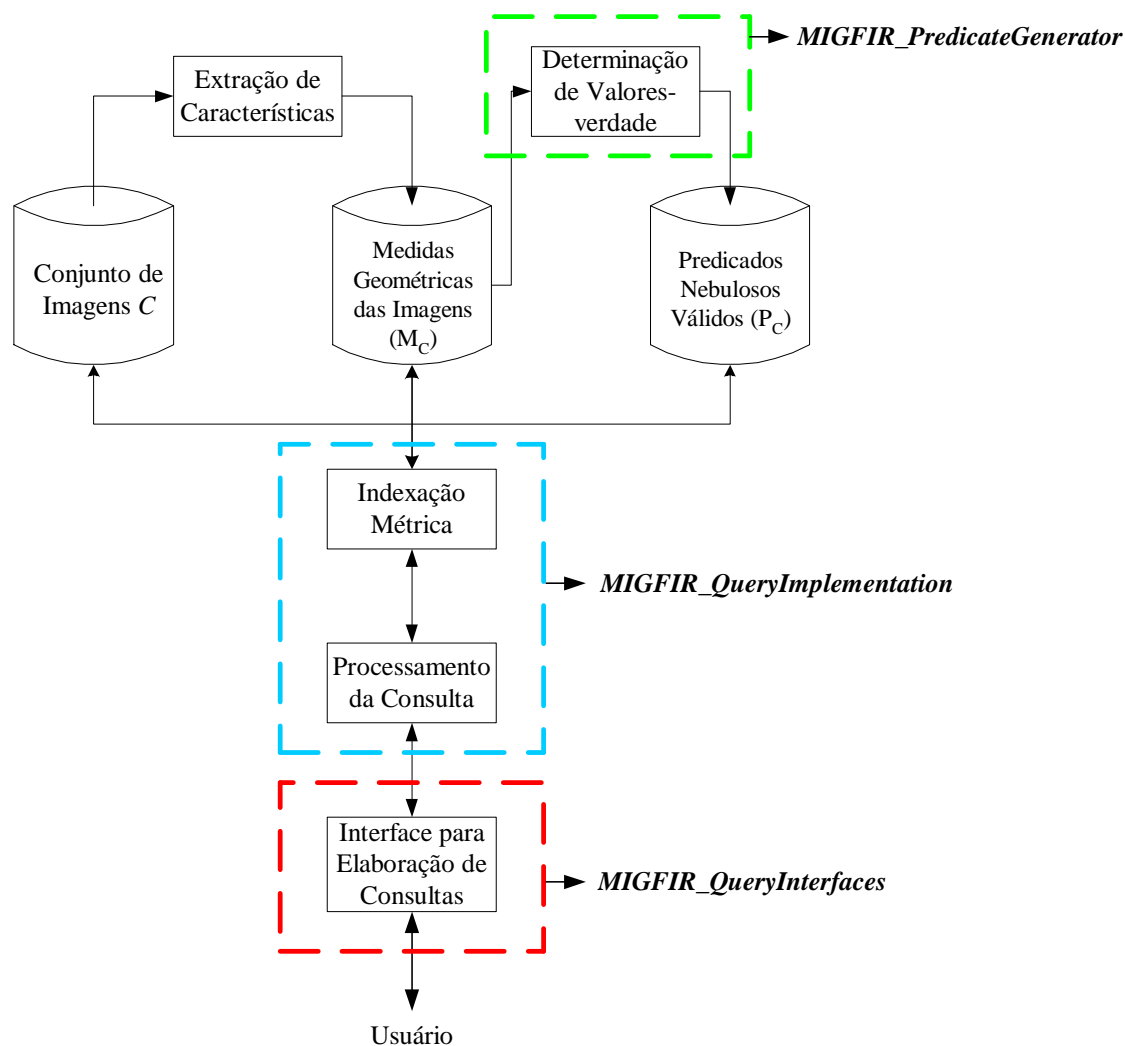


Figura 5.1 – Arquitetura básica do sistema *MIGFIR*.

Os detalhes de cada um dos três módulos serão apresentados a seguir, nas seções 5.3, 5.4 e 5.5.

### 5.3 MÓDULO *MIGFIR\_PREDICATEGENERATOR*

O módulo *MIGFIR\_PredicateGenerator* foi implementado para viabilizar a realização de uma tarefa fundamental para o uso de predicados nebulosos na composição de

consultas: a partir de funções de pertinência associadas às características geométricas extraídas das imagens, a tarefa é obter os graus de pertinência dos possíveis termos lingüísticos a fim de que, para cada medida geométrica extraída, o termo que possuir o maior valor de pertinência possa ser empregado na formação de um predicado nebuloso válido.

É importante destacar que a escolha de funções de pertinência adequadas é uma tarefa fundamental para garantir um grau de precisão satisfatório na recuperação de imagens que satisfaçam consultas baseadas em predicados nebulosos (ou, por simplicidade, *consultas nebulosas*). Prover as funções apropriadas é atribuição de um ou mais especialistas do domínio em questão.

A geração dos predicados nebulosos válidos para cada imagem de um dado conjunto pode ser efetuada clicando-se sobre o botão *Get Fuzzy Predicates* da interface inicial do sistema, mostrada na Figura 5.2. Quando se seleciona esta opção, é automaticamente criado um arquivo que armazena os predicados verdadeiros. Portanto, só há a necessidade de clicar sobre *Get Fuzzy Predicates* quando for considerado um novo conjunto de imagens.

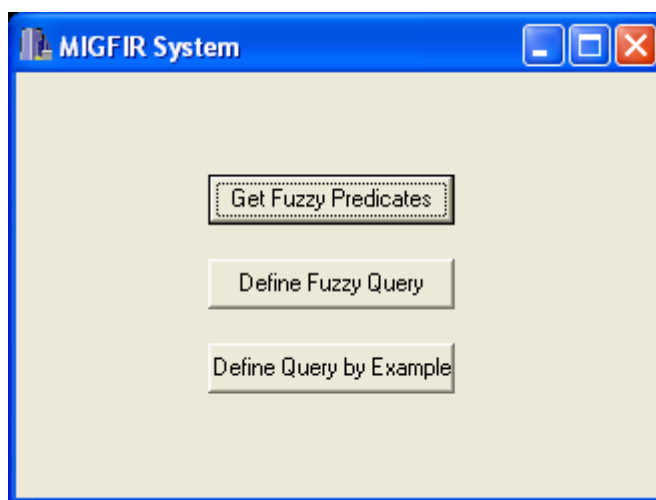


Figura 5.2 – Tela inicial do sistema *MIGFIR*.

Ainda na Figura 5.2, os botões *Define Fuzzy Query* e *Define Query by Example* conduzem o usuário, respectivamente, à interface de definição de consultas nebulosas e à interface de elaboração de consultas baseadas em imagens-exemplo.

## 5.4 MÓDULO *MIGFIR\_QUERYINTERFACES*

O módulo *MIGFIR\_QueryInterfaces* é responsável por oferecer ao usuário as interfaces para elaboração de consultas baseadas em imagens-exemplo e baseadas em predicados nebulosos, e também as interfaces para exibição dos resultados dessas consultas.

Para os dois tipos de consultas por similaridade complexas, esse módulo permite a utilização dos operadores lógicos *and* e *or* com o propósito de relacionar entre si os componentes de cada subconsulta (subexpressão da expressão de busca) e as próprias subconsultas. Além disso, conforme será detalhado a seguir, o usuário pode definir parâmetros de execução relevantes para cada componente de uma expressão de busca, isto é, para cada medida geométrica de uma imagem, predicado nebuloso, imagem-exemplo ou conjunto de predicados nebulosos.

As duas subseções a seguir apresentam detalhadamente as interfaces disponibilizadas por *MIGFIR\_QueryInterfaces*.

### 5.4.1 Interface para Consultas baseadas em Imagens-exemplo

A interface do módulo *MIGFIR\_QueryInterfaces* relativa a consultas baseadas em imagens-exemplo possibilita elaborar expressões de busca envolvendo uma ou mais imagens selecionadas pelo usuário e também visualizar apropriadamente os resultados desse tipo de consulta.

Essa interface começa a ser apresentada na Figura 5.3, que mostra a tela para definição de expressões de busca. Com o uso dessa tela, pode-se definir uma expressão composta por uma ou mais subexpressões, tal que cada subexpressão pode ser constituída por um conjunto qualquer de medidas geométricas provenientes de uma ou mais imagens (tipo de expressão que será chamado aqui de *tipo 1*) ou pode ser constituída por todas as medidas geométricas de uma dada imagem (tipo que será denominado *tipo 2*).

Note que a Figura 5.3 ilustra a construção de uma consulta cuja expressão de busca é *tipo 1*. Para elaborar uma expressão desse tipo, o usuário deve seguir nove passos:

1. Clicar sobre o botão *Open Image* para selecionar uma imagem-exemplo.
2. Clicar sobre o botão *Get Measures* para obter os valores das características geométricas da imagem escolhida.

3. Selecionar uma das medidas geométricas exibidas na parte da Figura 5.3 destacada como (1).
4. Definir o grau de relevância da medida geométrica selecionada em relação à subexpressão de busca na qual tal medida será inserida. Esse grau pode ser qualquer valor inteiro entre 1% e 100%; caso o usuário não defina um valor, o sistema estabelecerá automaticamente a relevância 100% (observe o componente destacado como (2) na Figura 5.3).
5. Escolher o tipo de consulta por similaridade (por abrangência ou aos  $k$  vizinhos mais próximos) a ser empregada na busca das imagens com determinado grau de similaridade em relação à imagem-exemplo corrente  $I_e$ , tal que esse grau será calculado entre qualquer imagem  $I_k$  do conjunto de dados e  $I_e$  apenas com base na medida geométrica de  $I_e$  selecionada e na correspondente medida de  $I_k$  (observe o componente marcado como (3) na Figura 5.3).
6. Informar um parâmetro adequado para o tipo de consulta por similaridade escolhida no passo 5, isto é, um número real  $r \geq 0$  que indique a diferença máxima permitida em relação à medida geométrica selecionada para uma consulta por abrangência (*query radius*) ou um número inteiro positivo  $k$  para uma consulta às  $k$  imagens mais próximas segundo a medida selecionada (*query neighbors*).
7. Clicar sobre o botão *Confirm* para inserir a medida geométrica corrente na expressão de busca, que é exibida pelo componente destacado como (4) na figura; note que o sistema mostra os graus de relevância entre colchetes (por exemplo, [66%]).
8. Inserir o operador lógico *and* ou o *or* na expressão de busca, clicando sobre o botão apropriado. Assim, o usuário pode inserir estes operadores para relacionar entre si subexpressões ou medidas geométricas da mesma subexpressão. Cada subexpressão deve ser delimitada através do uso de parênteses.
9. Clicar sobre o botão *Run* para que o módulo *MIGFIR\_QueryImplementation* execute a consulta complexa ou retornar ao passo 1 (ou ao passo 3, se desejar utilizar a mesma imagem-exemplo).

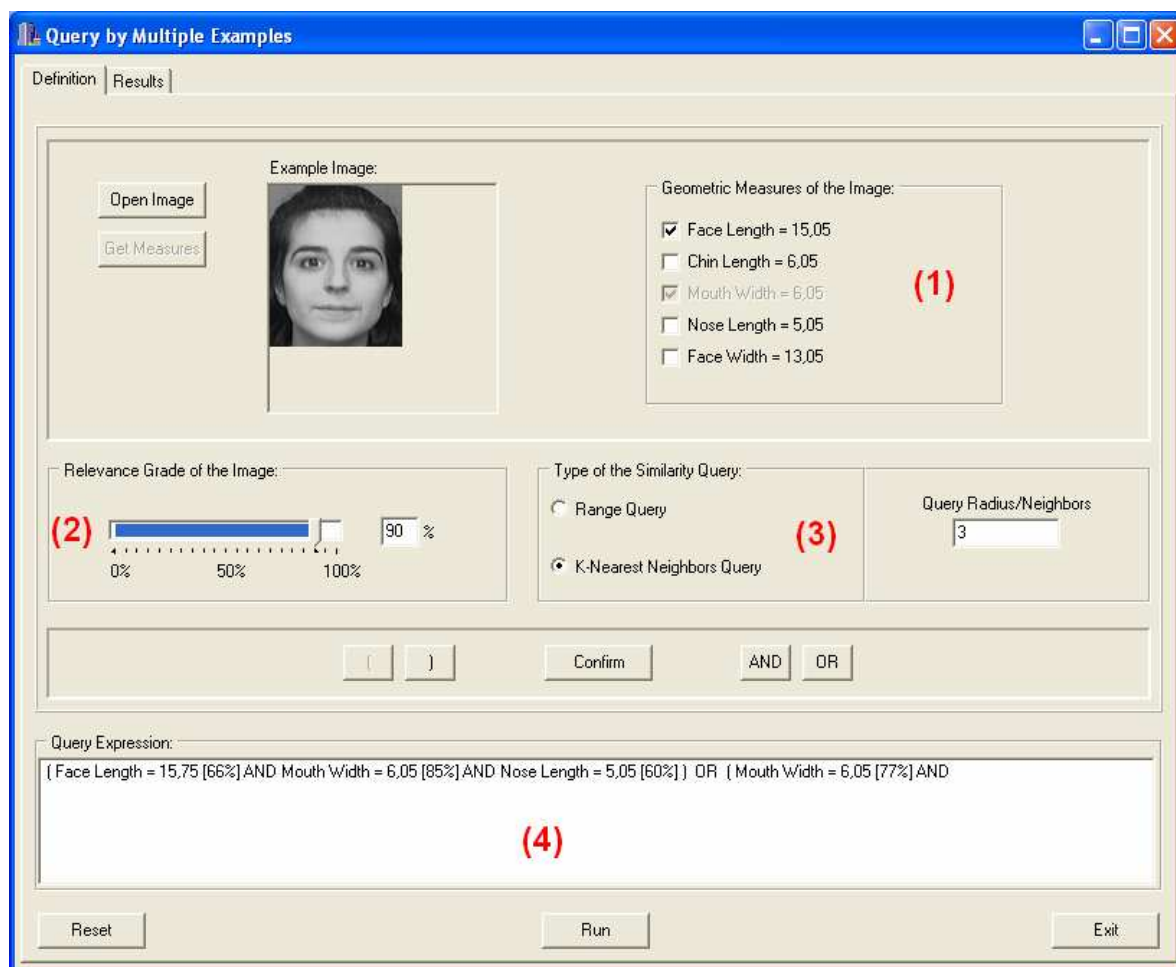


Figura 5.3 – Tela para elaboração de expressões de busca para consultas baseadas em imagens-exemplo.

Para definir uma expressão de busca *tipo 2*, deve-se seguir a mesma seqüência de passos das expressões *tipo 1*, porém com algumas diferenças.

A primeira diferença ocorre no passo 3, no qual devem ser selecionadas todas as medidas geométricas da imagem-exemplo corrente. Fazendo-se isso, o grau de relevância estabelecido no passo 4 será associado à imagem selecionada como um todo, que constitui uma subexpressão, e não apenas a uma de suas medidas. Logo, o valor definido no passo 4 significa a relevância da imagem corrente em relação à expressão de busca completa.

Outra diferença é que o tipo de consulta escolhido no passo 5 para buscar as imagens com certo grau de similaridade em relação à imagem-exemplo corrente considerará distâncias calculadas a partir de todas as medidas geométricas consideradas.

No passo 7, quando o usuário clica sobre o botão *Confirm* o sistema inclui na expressão de busca uma nova subexpressão, composta por todas as medidas da imagem-

exemplo atual relacionadas entre si por meio do operador lógico *and*. Como consequência disto o usuário poderá incluir o operador *or* somente para relacionar entre si subexpressões, ou seja, imagens como um todo.

Por fim, no passo 9 pode-se clicar sobre o botão *Run* para que o módulo *MIGFIR\_QueryImplementation* execute a consulta complexa ou pode-se retornar ao passo 1.

Diante da possibilidade de utilização dos operadores lógicos *and* e *or* e de configuração dos parâmetros discutidos acima para cada componente de uma expressão de busca, *MIGFIR* proporciona ao usuário alta flexibilidade no processo de definição de qualquer consulta. Portanto, o usuário pode elaborar desde consultas muito simples até bastante complexas.

Com relação ao número máximo de medidas geométricas que podem ser empregadas para representar cada imagem, é importante ressaltar que o sistema *MIGFIR* suporta qualquer número. No entanto, para a avaliação experimental, em que foram considerados conjuntos de faces humanas, o sistema foi ajustado para dar suporte à indexação e à recuperação de imagens utilizando até cinco medidas geométricas faciais. Foram escolhidas as mesmas medidas que já haviam sido empregadas com sucesso em [SANTINI; JAIN, 1999] na avaliação do modelo *FFC*, isto é: comprimento da face, comprimento do queixo, largura da boca, comprimento do nariz e largura da face.

A justificativa para o uso dessas cinco medidas citadas baseia-se em dois aspectos: Santini e Jain mostraram em [SANTINI; JAIN, 1999] que elas são poderosas o bastante para caracterizar o mesmo tipo de face que é considerado na avaliação de *MIGFIR*; e experimentos efetuados durante o desenvolvimento do projeto de mestrado, envolvendo quinze alunos (sendo seis alunos de pós-graduação e nove de graduação) que não tinham conhecimento do assunto tratado neste trabalho, permitiram observar que as cinco medidas escolhidas realmente são suficientes para avaliar a similaridade entre faces quando são consideradas apenas as características geométricas. Uma outra justificativa pode ser encontrada em [EDELMAN; INTRATOR, 1997].

De acordo com a literatura da área, a medição da similaridade entre faces humanas é uma questão complexa, que depende de outros fatores além das medidas geométricas, como a cor da pele, a textura do cabelo, as curvas do contorno facial, etc. Todavia, foi escolhida

uma abordagem simplificada durante o desenvolvimento deste trabalho porque o objetivo não era avaliar os tipos de características que poderiam representar as faces nem desenvolver um sistema que combinasse diversos tipos de características.

A outra tela da interface para consultas baseadas em exemplos, mostrada na Figura 5.4, tem a função de exibir de forma ordenada as imagens que satisfazem a expressão de busca mostrada na região desta tela rotulada como *Query Expression*.

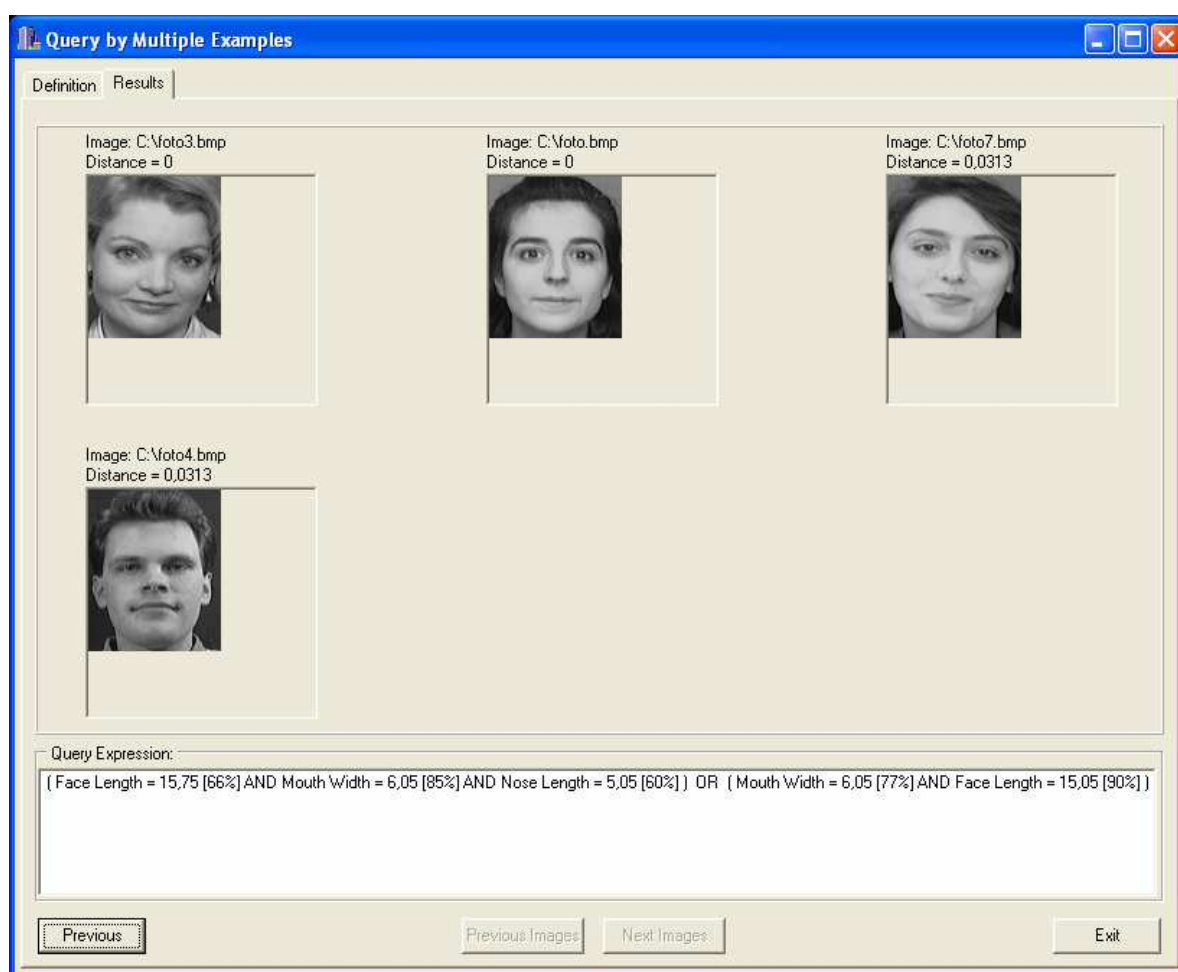


Figura 5.4 – Tela para exibição das imagens que satisfazem uma consulta baseada em imagens-exemplo.

Após o módulo *MIGFIR\_QueryImplementation* executar a consulta complexa, se o resultado contiver mais de seis imagens então o usuário pode navegar neste resultado usando os botões *Previous Images* e *Next Images*.

Observe que as imagens são apresentadas em ordem crescente da distância em relação à expressão de busca. Quando duas ou mais imagens possuem o mesmo valor para a

distância, a ordem de exibição na tela é a mesma em que foram recuperadas durante a realização da consulta.

#### 5.4.2 Interface para Consultas baseadas em Predicados Nebulosos

A interface do módulo *MIGFIR\_QueryInterfaces* referente a consultas baseadas em predicados nebulosos permite que sejam elaboradas expressões de busca contendo um ou mais predicados definidos pelos usuários e também que sejam visualizadas as imagens que satisfazem esse tipo de consulta.

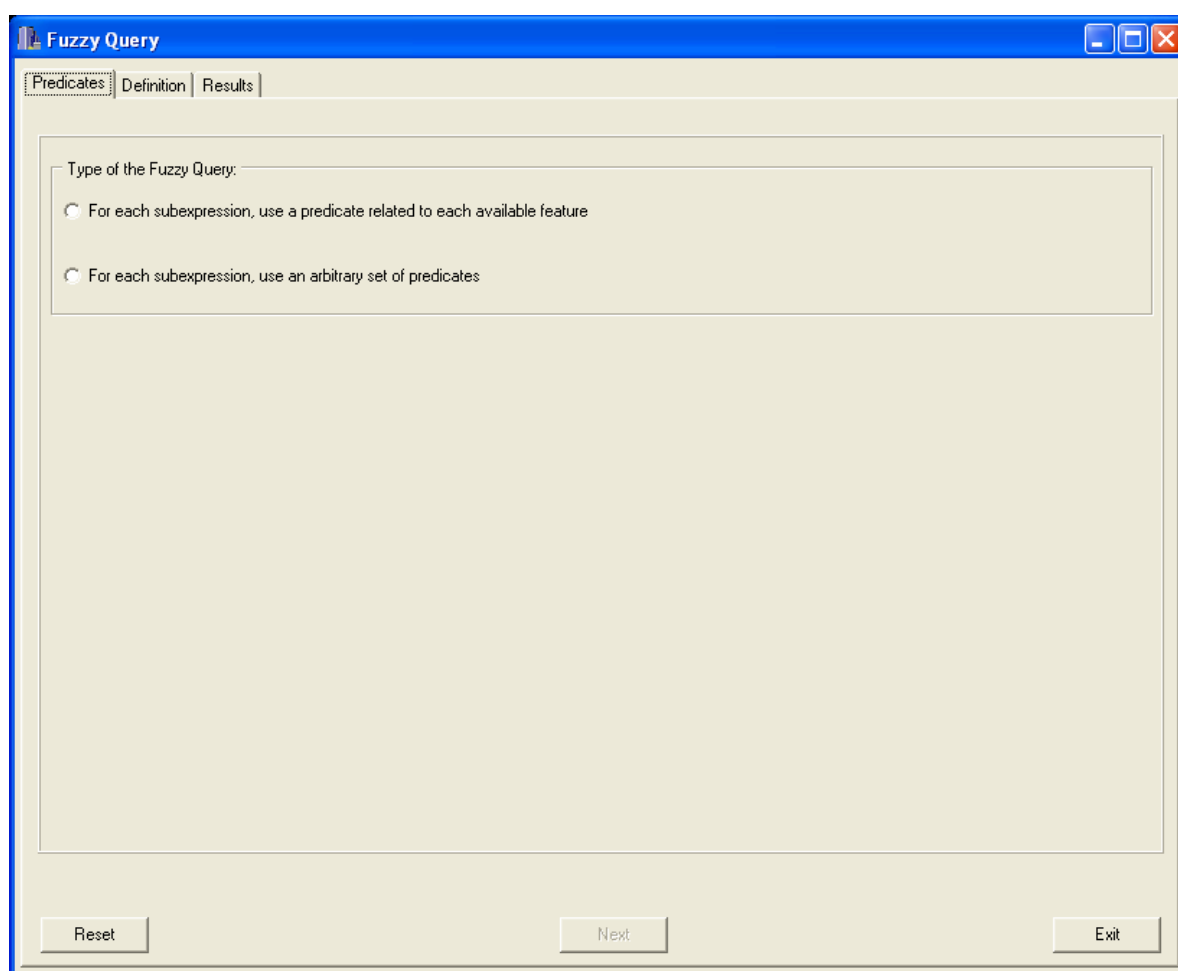


Figura 5.5 – Tela inicial da interface para consultas nebulosas.

A Figura 5.5 mostra a tela inicial da interface para consultas nebulosas. Para criar predicados nebulosos, deve-se inicialmente selecionar nesta tela o tipo de consulta que se pretende elaborar e executar (observe a região superior da Figura 5.5); ou seja, deve-se escolher se cada subexpressão da expressão de busca será composta por um conjunto

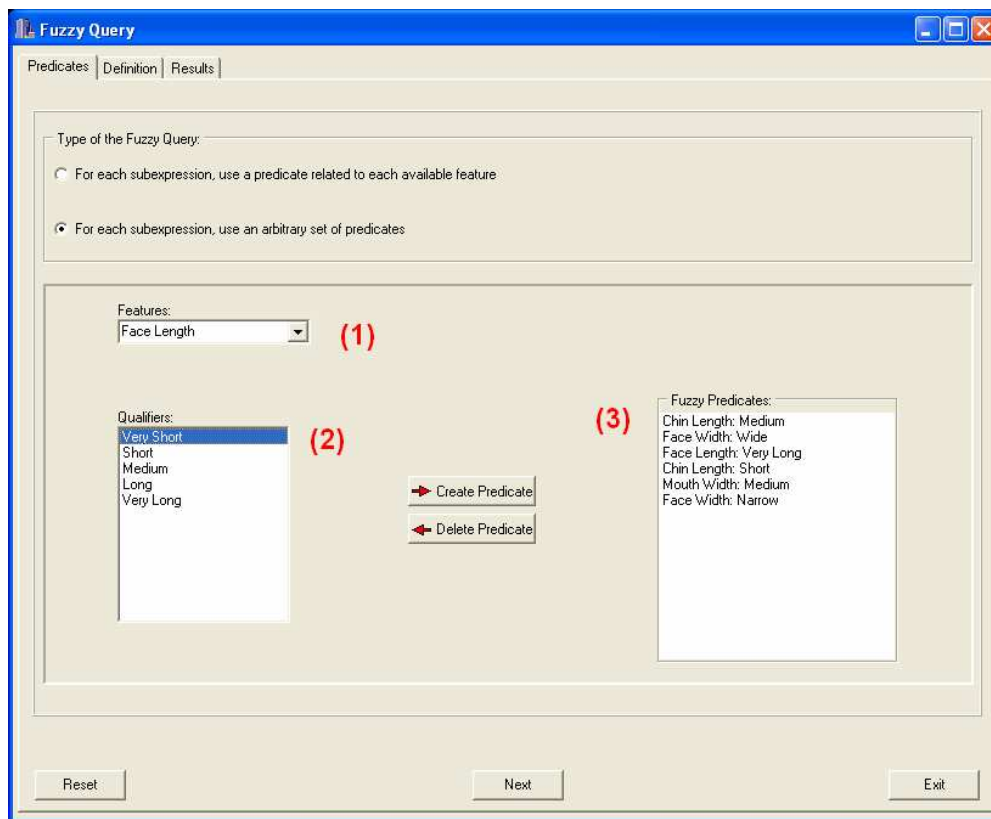


qualquer de predicados correspondentes a uma ou mais características geométricas (tipo de expressão que será denominado *tipo 3*), ou se cada subexpressão será constituída por um predicado associado a cada característica geométrica considerada (tipo que será chamado de *tipo 4*).

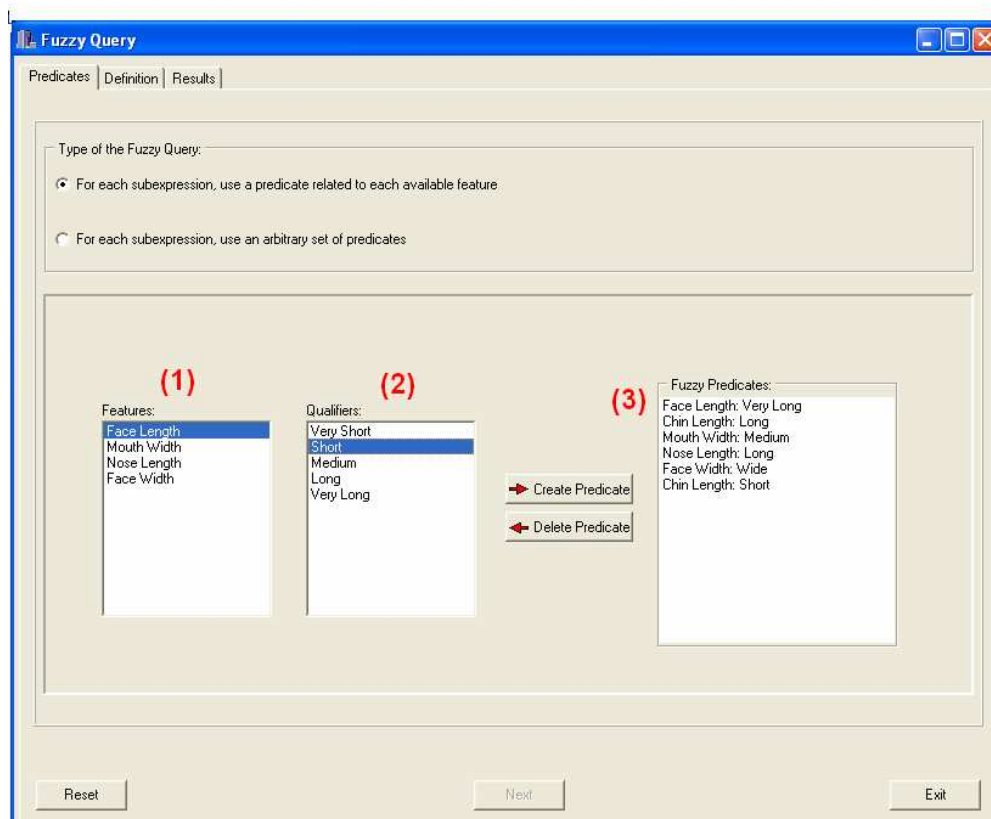
Agora, comparando-se as figuras 5.6(a) e 5.6(b), pode-se notar que a composição da tela inicial apresentada depende do tipo de consulta nebulosa que foi selecionado. Optou-se pelo desenvolvimento de uma configuração da tela inicial para cada tipo com o propósito de tornar a tarefa de criação dos predicados nebulosos suficientemente intuitiva para o usuário. Deve-se enfatizar que é permitido criar qualquer quantidade de predicados.

Na Figura 5.6(a) ilustra-se a criação de predicados nebulosos que poderiam ser empregados para compor uma expressão de busca *tipo 3*. Essa criação deve ser iniciada pela seleção de uma das possíveis características geométricas listadas pelo componente destacado como (1) na Figura 5.6(a). Feito isso, o sistema apresenta no componente destacado como (2) nesta figura os possíveis termos lingüísticos referentes à característica selecionada. Deve-se escolher um dos termos e em seguida clicar sobre o botão *Create Predicate*, para que o novo predicado seja gerado e inserido na lista de predicados nebulosos já definidos (componente marcado como (3) na Figura 5.6(a)). O botão *Delete Predicate* possibilita a remoção de qualquer predicado da lista e o botão *Next* conduz o usuário à tela de elaboração de expressões de busca.

Já na Figura 5.6(b) ilustra-se a construção de predicados que poderiam ser utilizados para compor uma expressão *tipo 4*. Esse processo de construção difere daquele relativo aos predicados para expressões *tipo 3* no passo de seleção das características que serão empregadas. Isto porque no caso em que se deseja definir uma consulta cuja expressão é *tipo 4*, o sistema automaticamente insere no componente marcado como (1) na Figura 5.6(b) todas as características consideradas das imagens, já que cada subexpressão deverá conter um predicado para cada característica.



(a)



(b)

Figura 5.6 – As possíveis composições da tela inicial: (a) para expressões *tipo 3*; (b) para expressões *tipo 4*.

Quando um item é selecionado no componente (1) da tela mostrada na Figura 5.6(b), o sistema exibe o respectivo conjunto de possíveis termos lingüísticos no componente destacado como (2). Deve-se escolher um dos termos e em seguida clicar sobre o botão *Create Predicate*, a fim de que o novo predicado seja formado e inserido na lista de predicados nebulosos já criados (componente marcado como (3) na Figura 5.6(b)). Isso faz com que a característica utilizada seja removida do componente (1). Portanto, após a criação de um predicado a partir de cada característica, o usuário tem duas opções: pode começar a definir novos predicados para uma próxima subexpressão da consulta ou pode clicar sobre o botão *Next*. As funções deste e do botão *Delete Predicate* são as mesmas já citadas para o caso em que são formados predicados para expressões *tipo 3*.

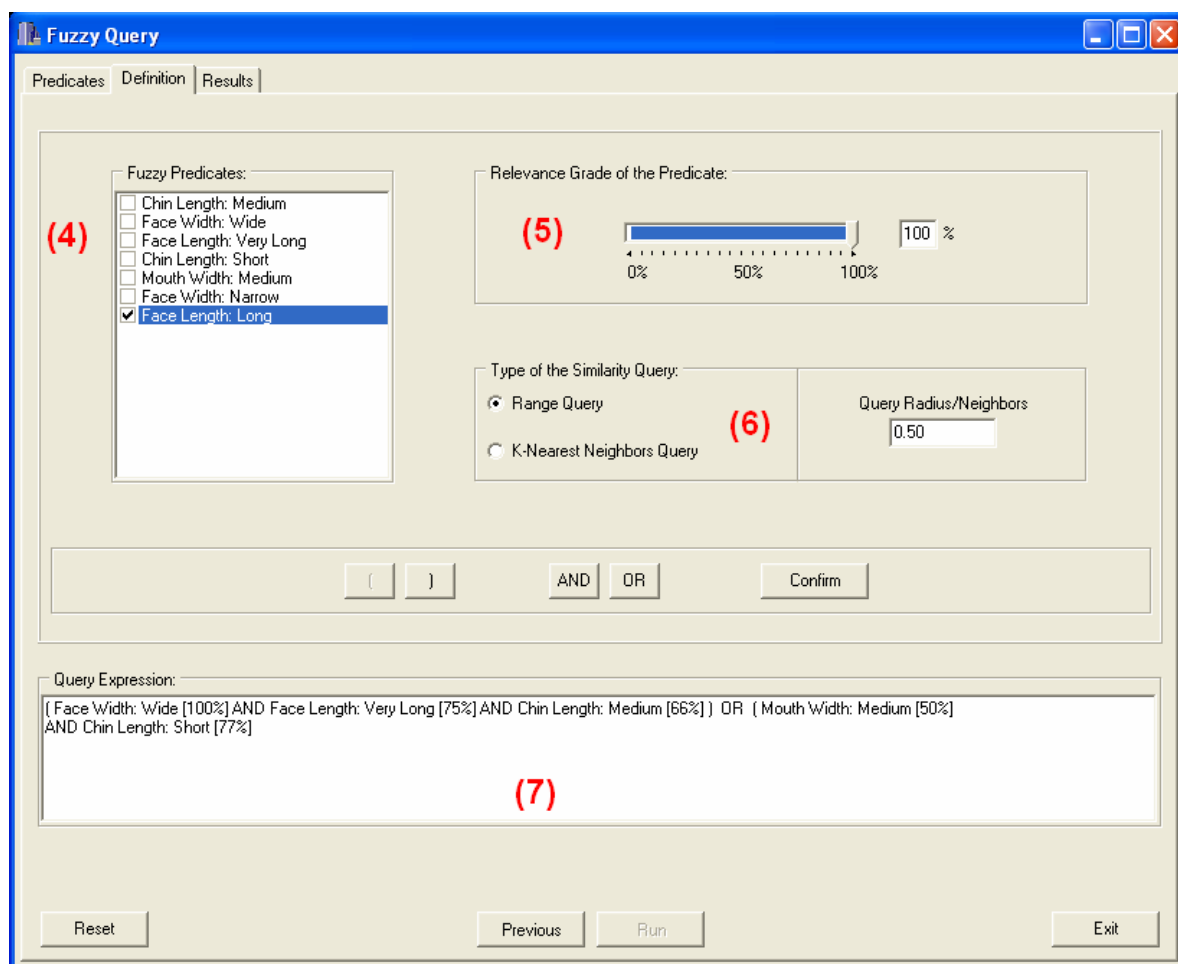


Figura 5.7 – Tela para elaboração de expressões de busca para consultas baseadas em predicados nebulosos.

A Figura 5.7 mostra a tela para elaboração de expressões de busca *tipo 3* e *tipo 4*. Essas expressões podem ser construídas com base nos predicados criados previamente, os quais são exibidos na parte desta tela destacada como (4).

Note que a Figura 5.7 ilustra a preparação de uma consulta cuja expressão de busca é *tipo 3*. A fim de construir uma expressão desse tipo, o usuário deve seguir sete passos:

1. Selecionar um dos predicados nebulosos exibidos na região da Figura 5.7 marcada como (4).
2. Definir o grau de relevância do predicado nebuloso selecionado em relação à subexpressão de busca na qual tal predicado será inserido (veja o componente destacado como (5) na Figura 5.7). Esse grau pode ser qualquer valor entre 1% e 100%; caso o usuário não defina um valor, o sistema estabelecerá automaticamente a relevância 100%.
3. Escolher o tipo de consulta por similaridade (por abrangência ou aos  $k$  vizinhos mais próximos) a ser empregada na busca das imagens que contêm, no respectivo conjunto de predicados usados para representar a imagem, o predicado selecionado; observe o componente marcado como (6) na Figura 5.7.
4. Informar um parâmetro adequado para o tipo de consulta por similaridade escolhido no passo 3. No caso de uma consulta por abrangência, deve-se inserir um número real pertencente ao intervalo  $[0, 1]$  que indique a diferença máxima permitida em relação ao número 1,0, que é o valor-verdade atribuído ao chamado predicado de consulta (predicado escolhido no passo 1). Já no caso de uma consulta aos  $k$  vizinhos mais próximos, deve-se informar um número inteiro positivo para que sejam recuperadas as  $k$  imagens que possuam os maiores valores-verdade para o predicado corrente.
5. Clicar sobre o botão *Confirm* para inserir o predicado corrente na expressão de busca, que é exibida pelo componente destacado como (7) na figura; nessa expressão, observe que o sistema mostra os graus de relevância entre colchetes (por exemplo,  $[75\%]$ ).
6. Inserir o operador lógico *and* ou o *or* na expressão de busca, clicando sobre o botão apropriado. Assim, o usuário pode inserir esses operadores para relacionar

entre si subexpressões ou predicados nebulosos da mesma subexpressão. Cada subexpressão deve ser delimitada através do uso de parênteses.

7. Clicar sobre o botão *Run* para que o módulo *MIGFIR\_QueryImplementation* execute a consulta complexa ou retornar ao passo 1.

Para definir uma expressão de busca *tipo 4*, deve-se seguir a mesma seqüência de passos das expressões *tipo 3*, porém com algumas diferenças.

A primeira diferença ocorre no passo 1, no qual deve ser selecionada uma quantidade de predicados igual ao total de características geométricas consideradas, devendo ser um predicado referente a cada característica. Fazendo-se isto, o grau de relevância estabelecido no passo 2 será associado ao conjunto de predicados selecionados, que constituem uma subexpressão. Assim, o valor definido no passo 2 significa a relevância do conjunto de predicados em relação à expressão de busca completa.

Outra diferença é que o tipo de consulta escolhido no passo 3 para buscar as imagens representadas pelo mesmo conjunto de predicados selecionados no passo 1 considerará valores de similaridade, que são números reais pertencentes ao intervalo  $[0, 1]$ , calculados a partir de todos os valores-verdade associados aos predicados dos conjuntos.

No passo 5, quando o usuário clicar sobre o botão *Confirm* o sistema incluirá na expressão de busca uma nova subexpressão, composta por todos os predicados selecionados relacionados entre si por meio do operador lógico *and*. Como consequência disto, no passo 6 o usuário poderá incluir o operador *or* apenas para relacionar entre si subexpressões.

Por fim, no passo 7 pode-se clicar sobre o botão *Run* para que o módulo *MIGFIR\_QueryImplementation* execute a consulta nebulosa ou pode-se retornar ao passo 1.

Em relação ao número máximo de predicados nebulosos que podem ser associados a cada imagem, deve-se enfatizar que o sistema suporta qualquer quantidade. No entanto, como a configuração do sistema para a avaliação experimental permite o uso de até cinco medidas geométricas relativas a imagens de faces humanas, pode-se efetuar a indexação e a recuperação utilizando predicados criados ao se combinar essas características a seus respectivos conjuntos de termos lingüísticos. A Tabela 5.1 mostra as cinco características consideradas junto aos possíveis termos.

Tabela 5.1 – Características extraídas de imagens e os possíveis termos lingüísticos.

<b>Característica da Imagem</b>	<b>Possíveis Termos Lingüísticos</b>
comprimento da face	{ muito curta, curta, média, longa, muito longa }
comprimento do queixo	{ curto, médio, longo }
largura da boca	{ estreita, média, larga }
comprimento do nariz	{ curto, médio, longo }
largura da face	{ muito estreita, estreita, média, larga, muito larga }

A justificativa para o uso dos conjuntos de termos mostrados na Tabela 5.1 baseia-se em experimentos efetuados durante o desenvolvimento do projeto, que permitiram observar que tais conjuntos são suficientes para possibilitar a representação de faces humanas através de predicados nebulosos. Esses experimentos e aqueles referentes à escolha do total de medidas geométricas (comentados na subseção 5.4.1) contaram com a participação do mesmo grupo de pessoas.

A última tela da interface para consultas nebulosas é mostrada na Figura 5.8. Ela é responsável por apresentar de forma ordenada as imagens que satisfazem a expressão de busca exibida na região desta tela rotulada como *Query Expression*.

Pode-se notar que a tela dada na Figura 5.8 é quase idêntica àquela mostrada na Figura 5.4, já que a única diferença entre elas é que na tela de resultados de consultas nebulosas as imagens são apresentadas em ordem decrescente do valor de similaridade (grau de “satisfação”) em relação à expressão de busca, e não em ordem crescente de distância. Quando duas ou mais imagens possuem o mesmo valor de similaridade, a ordem de exibição é a mesma em que foram recuperadas durante a execução da consulta.

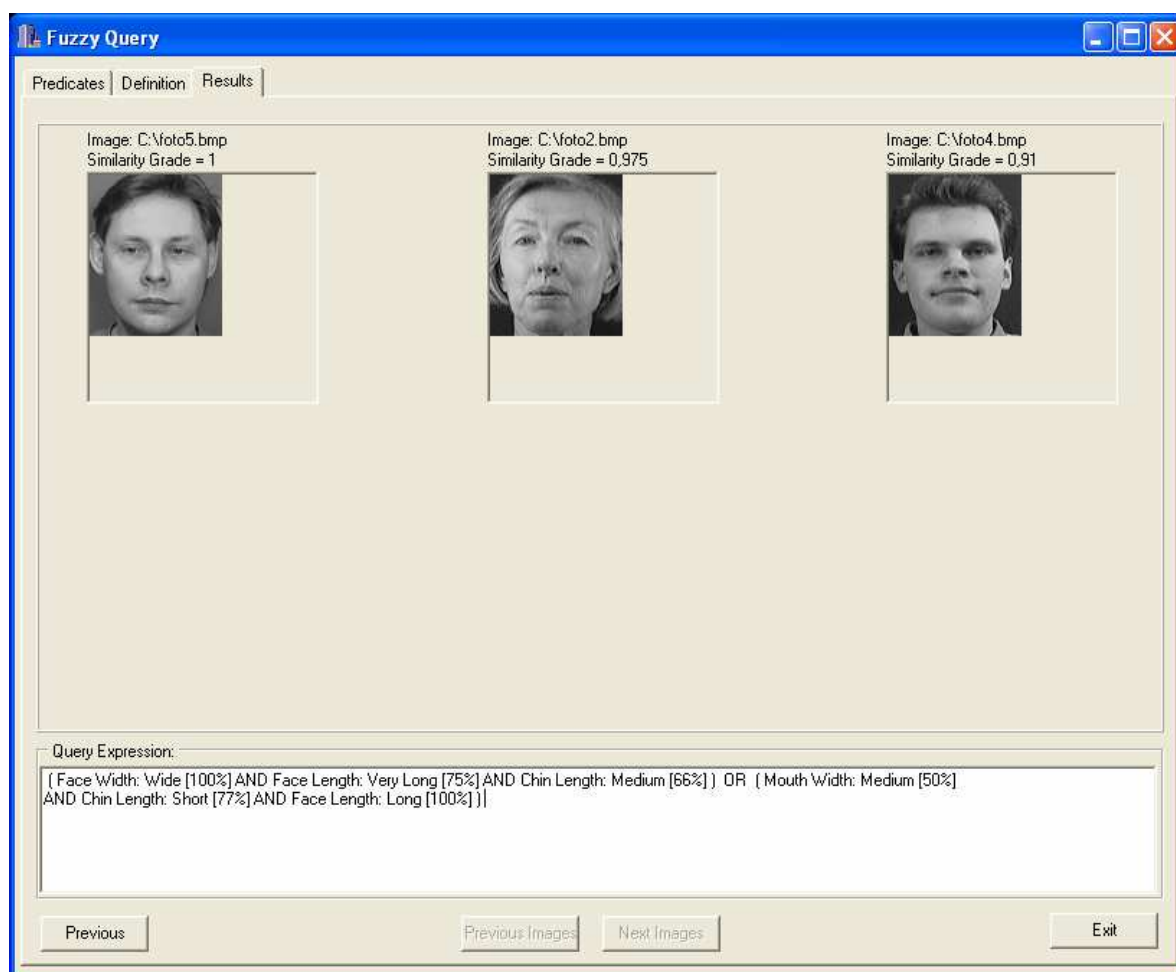


Figura 5.8 – Tela para exibição das imagens que satisfazem uma consulta baseada em predicados nebulosos.

## 5.5 MÓDULO MIGFIR\_QUERYIMPLEMENTATION

Esse módulo é a parte do sistema responsável pela criação de objetos que representem imagens e funções de similaridade métricas, pelo gerenciamento de índices métricos e pela execução de todas as consultas que tenham sido definidas no módulo *MIGFIR\_QueryInterfaces*.

Para facilitar a apresentação do módulo *MIGFIR\_QueryImplementation*, pode-se dividi-lo em duas partes, denominadas *MIGFIR\_QueryImplementation\_Data* e *MIGFIR\_QueryImplementation\_Application*. Elas serão discutidas nas subseções 5.5.1 e 5.5.2, respectivamente.

### 5.5.1 MIGFIR\_QueryImplementation\_Data

Antes de apresentar os tipos de dados (as classes) que foram definidos na parte chamada de *MIGFIR\_QueryImplementation\_Data*, é importante abordar alguns aspectos relativos à implementação da estrutura de indexação métrica *Slim-tree*.

O código-fonte da *Slim-tree* foi incluído em uma biblioteca de métodos de acesso métricos desenvolvida pelo Grupo de Bases de Dados e Imagens do Instituto de Ciências Matemáticas e de Computação, da Universidade de São Paulo. Essa biblioteca, denominada *Arboretum*<sup>3</sup>, foi implementada em linguagem C++ e contém também as estruturas de indexação *DBM-tree*, *DF-tree*, *GH-tree*, *M-tree*, *VP-tree*, e a varredura seqüencial.

A *Arboretum* foi desenvolvida com base em uma arquitetura de três camadas, como pode ser visto na Figura 5.9.

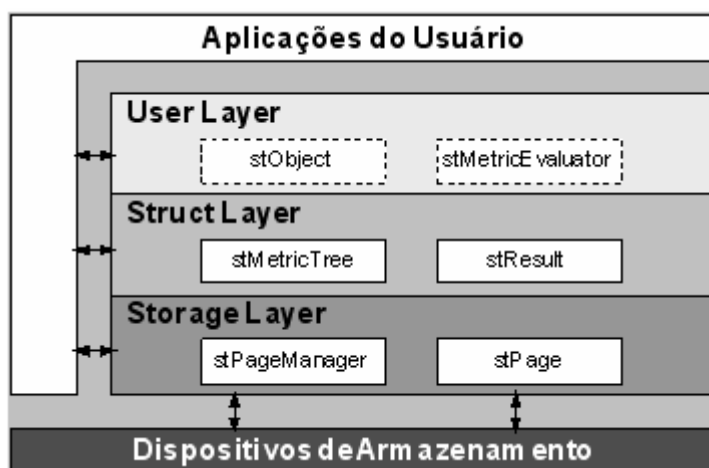


Figura 5.9 – Arquitetura da biblioteca de estruturas de indexação *Arboretum*.

Nesta arquitetura, a camada denominada *Storage Layer* provê serviços de gerenciamento de páginas do disco rígido ou da memória principal para a camada *Struct Layer*. A camada *Struct Layer* implementa os métodos de acesso métricos utilizando classes do tipo *template*, com o objetivo de dar suporte à indexação e à recuperação de tipos de objetos previamente desconhecidos. Por fim, a camada *User Layer* deve conter uma ou

<sup>3</sup> A biblioteca de métodos de acesso métricos *Arboretum* pode ser obtida em <http://gbdi.icmc.usp.br/arboretum/>



mais classes que implementem a interface *stObject* para instanciar objetos a serem indexados e recuperados, e uma ou mais classes que implementem a interface *stMetricEvaluator* para instanciar métricas de similaridade.

Portanto, a parte *MIGFIR\_QueryImplementation\_Data* foi implementada como a camada *User Layer* da biblioteca *Arboretum*. Para isso, foram definidas quatro classes para instanciar objetos que representem apropriadamente as imagens que devem ser indexadas e recuperadas (veja a Figura 5.10), e duas classes para instanciar as funções métricas utilizadas no cálculo da similaridade entre as imagens (veja a Figura 5.11).

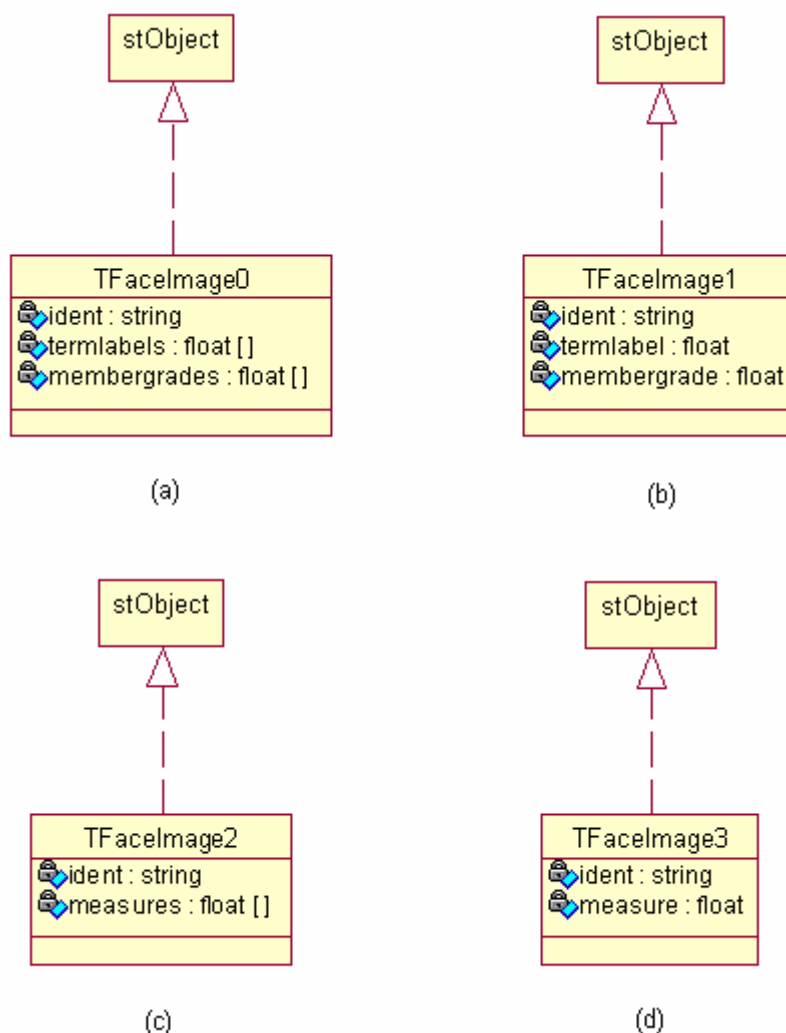


Figura 5.10 – Classes utilizadas para representar imagens de faces humanas: (a) Classe *TFaceImage0*; (b) Classe *TFaceImage1*; (c) Classe *TFaceImage2*; (d) Classe *TFaceImage3*.

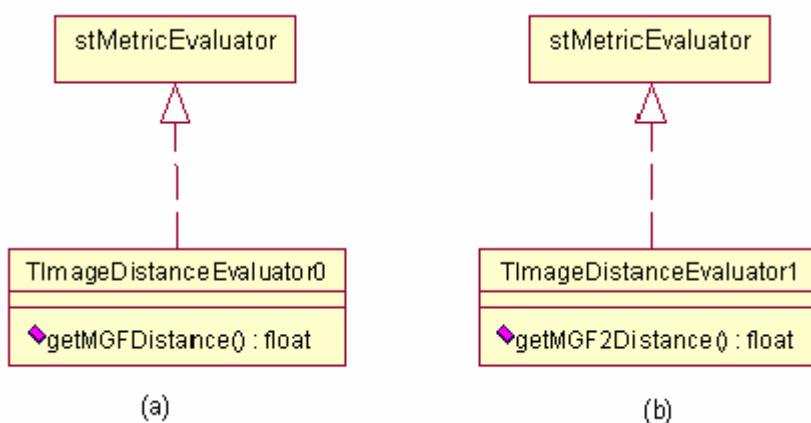


Figura 5.11 – Classes utilizadas para instanciar as duas funções de similaridade consideradas: (a) Classe correspondente à métrica *MGF*; (b) Classe correspondente à métrica *MGF2*.

A classe *TFaceImage0*, mostrada na Figura 5.10(a), possibilita representar cada imagem por meio de cinco predicados nebulosos válidos gerados a partir das medidas geométricas consideradas; assim, o atributo *ident* armazena o identificador exclusivo da imagem corrente, *termlabels* é um vetor que contém cinco números inteiros gerados automaticamente para representar de forma numérica, nos cálculos de distância, os cinco termos lingüísticos dos predicados, e o vetor *membergrades* armazena cinco valores-verdade, sendo um para cada predicado. A classe *TFaceImage1* (Figura 5.10(b)) difere da classe *TFaceImage0* apenas em um aspecto: ela permite representar cada imagem através de apenas um predicado nebuloso válido, de modo que o termo e o valor-verdade são acessados por meio dos atributos *termlabel* e *membergrade*, respectivamente.

Já a classe *TFaceImage2*, mostrada na Figura 5.10(c), permite representar cada imagem de face por meio de suas cinco medidas geométricas; logo, o atributo *ident* armazena o identificador exclusivo da imagem corrente e *measures* é um vetor que guarda os valores referentes às cinco características geométricas. A classe *TFaceImage3* (Figura 5.10(d)) difere da classe *TFaceImage2* somente em um aspecto: ela possibilita representar cada imagem por meio de apenas uma característica geométrica, cujo valor é armazenado no atributo *measure*.

Note-se que na Figura 5.10 são especificados somente os principais atributos de cada classe, já que discutir aspectos bastante específicos da implementação dos métodos das classes não está no escopo desta dissertação.

Com relação à Figura 5.11, são ilustradas as classes que permitem utilizar as funções de similaridade *MGF* (*Metric for Geometric Features*) e *MGF2*, que foram desenvolvidas durante a realização do projeto de mestrado e que serão discutidas detalhadamente na subseção 5.5.2.4. Na Figura 5.11(a), correspondente à métrica *MGF*, o método *getMGFDistance()* usa essa função para determinar a distância entre dois objetos quaisquer da classe *TFaceImage0* ou da classe *TFaceImage1*. Já o método *getMGF2Distance()* (Figura 5.11(b)) emprega a métrica *MGF2* para calcular a distância entre dois objetos quaisquer de *TFaceImage2* ou de *TFaceImage3*.

### 5.5.2 MIGFIR\_QueryImplementation\_Application

*MIGFIR\_QueryImplementation\_Application* é a parte do módulo *MIGFIR\_QueryImplementation* na qual foram implementados os algoritmos responsáveis pela execução das consultas baseadas em imagens-exemplo e baseadas em predicados nebulosos. O processo de execução das consultas inclui as tarefas de criação e gerenciamento de instâncias da *Slim-tree*.

Os algoritmos implementados nessa parte do módulo *MIGFIR\_QueryImplementation* realizam até sete tarefas<sup>4</sup> principais que compõem o processo de execução de qualquer consulta complexa suportada pelo sistema *MIGFIR*. As subseções 5.5.2.1 a 5.5.2.7 detalham essas tarefas e os algoritmos. Para estas subseções considere, como exemplos, as seguintes consultas:

- $C_1$ : consulta baseada em exemplos com expressão tipo 1: ( ( *Chin Length* = 4,75 [90%] OR *Mouth Width* = 6,05 [77%] ) OR ( *Face Length* = 14,95 [83%] AND *Chin Length* = 4,90 [100%] AND *Face Width* = 13,85 [100%] ) ).
- $C_2$ : consulta baseada em exemplos com expressão tipo 2: ( ( *Face Length* = 15,45 AND *Chin Length* = 6,05 AND *Mouth Width* = 6,05 AND *Nose Length* = 5,05 AND *Face Width* = 13,05 [66%] ) OR ( *Face Length* = 15,75 AND *Chin Length* = 5,05 AND *Mouth Width* = 5,65 AND *Nose Length* = 4,95 AND *Face Width* = 12,75 [100%] ) ).

---

<sup>4</sup> O total de tarefas depende da complexidade da expressão de busca.

- $C_3$ : consulta nebulosa com expressão *tipo 3*: ( ( *Face Width: Wide [90%] AND Face Length: Very Long [85%] AND Mouth Width: Medium [73%]* ) OR ( *Face Length: Long [97%] AND Nose Length: Medium [80%]* ) ).
- $C_4$ : consulta nebulosa com expressão *tipo 4*: ( ( *Face Length: Long AND Chin Length: Medium AND Mouth Width: Wide AND Nose Length: Short AND Face Width: Narrow [75%]* ) OR ( *Face Length: Very Long AND Chin Length: Long AND Mouth Width: Medium AND Nose Length: Medium AND Face Width: Wide [100%]* ) ).

### 5.5.2.1 Leitura dos dados das imagens

A leitura dos dados das imagens pode ser a primeira tarefa realizada durante a execução de uma consulta complexa. Essa tarefa consiste na leitura de um arquivo que contém os dados extraídos das imagens a serem indexadas.

Como a configuração atual do sistema *MIGFIR* faz com que todos os índices criados sejam armazenados no disco rígido, é verificado automaticamente, antes da realização da leitura, se já existe um índice correspondente aos dados que deveriam ser lidos do arquivo de dados de imagens. Logo, essa primeira tarefa é efetuada somente se não existir um índice correspondente aos dados em questão.

Para o caso considerado neste trabalho, existem dois tipos de arquivos que armazenam dados de imagens de faces humanas: um tipo contém as cinco medidas faciais consideradas e um identificador exclusivo para cada imagem; e o outro contém os cinco predicados nebulosos válidos, os valores-verdade desses predicados e um identificador exclusivo para cada imagem.

Arquivos que armazenam medidas geométricas faciais podem ser lidos durante a execução de consultas baseadas em imagens-exemplo. Já no caso de consultas nebulosas, os arquivos que podem ser lidos são aqueles que contêm predicados nebulosos válidos.

### 5.5.2.2 Criação de objetos para representar imagens

Se houver a necessidade de ler algum arquivo *A* de dados de imagens durante a execução de uma consulta complexa, tal leitura ocorrerá apenas uma vez. Para tornar isso possível, foram definidas as quatro classes apresentadas na Figura 5.10, que permitem

instanciar objetos apropriados para armazenar dados lidos de  $A$ . Depois de criados, esses objetos, que representam as imagens propriamente ditas, serão indexados em árvores do tipo *Slim-tree* e poderão ser recuperados para compor resultados de consultas.

Assim, se a expressão de busca da consulta for *tipo 1*, como ocorre com  $C_1$  (definida acima), o algoritmo desenvolvido para a instanciação de objetos cria instâncias da classe *TFaceImage3* para armazenar as medidas geométricas lidas de um arquivo que são correspondentes a cada característica geométrica utilizada na expressão de busca. No caso de  $C_1$ , por exemplo, se for necessário ler um arquivo de dados de imagens, o algoritmo de instanciação criará objetos para armazenar os valores lidos referentes ao comprimento do queixo, à largura da boca, ao comprimento da face e à largura da face.

No caso da expressão ser *tipo 2*, como ocorre com  $C_2$  (especificada acima), o algoritmo cria objetos da classe *TFaceImage2* para guardar os valores das cinco características geométricas faciais que podem ser lidas de um arquivo de dados.

Quando a expressão é *tipo 3*, o algoritmo de instanciação cria objetos do tipo *TFaceImage1* para armazenar os predicados nebulosos e os respectivos valores-verdade que são correspondentes a cada predicado utilizado na expressão de busca. No caso de  $C_3$  (definida acima), por exemplo, se for necessário ler um arquivo de dados de imagens o algoritmo criará objetos para armazenar os predicados e os valores-verdade lidos referentes à largura da face, ao comprimento da face, à largura da boca e ao comprimento do nariz.

Já no caso da expressão ser *tipo 4*, como ocorre com  $C_4$  (especificada acima), o algoritmo de instanciação cria objetos do tipo *TFaceImage0* para guardar os cinco predicados nebulosos válidos e os respectivos valores-verdade que podem ser lidos de um arquivo de dados.

### 5.5.2.3 Criação de instâncias da *Slim-tree*

Essa tarefa consiste em criar, no disco rígido ou na memória principal, instâncias da *Slim-tree* com configurações apropriadas para o tipo de objeto a ser indexado.

A criação de uma instância da *Slim-tree* ocorre somente quando é realizada a leitura de dados das imagens (primeira tarefa), pois a execução dessa leitura indica que ainda não foi construído um índice referente aos dados lidos.

No algoritmo desenvolvido para a criação de árvores *Slim-tree*, além de ser possível configurar se elas devem ser armazenadas no disco ou na memória, também é possível definir diferentes tamanhos para os nós. Assim, durante a execução de uma consulta, esse algoritmo seleciona um dos possíveis valores de tamanho para os nós de uma árvore de acordo com o tipo de objeto que será indexado. Com relação aos quatro tipos de objetos considerados neste trabalho, pôde-se concluir através de experimentos que dois possíveis valores de tamanho são adequados, sendo eles 1024 bytes (1 KB) e 4096 bytes (4 KB).

O algoritmo de instanciação cria árvores em que cada nó corresponde a uma página de 1024 bytes de tamanho se os objetos que serão indexados forem da classe *TFaceImage1* ou da classe *TFaceImage3*. Portanto, índices com páginas de 1 KB são utilizados durante a execução de consultas cujas expressões de busca são *tipo 1* ou *tipo 3*. Já para as situações em que os objetos a serem indexados são da classe *TFaceImage0* ou da classe *TFaceImage2*, o algoritmo de instanciação gera árvores em que cada nó corresponde a uma página de 4096 bytes de tamanho. Logo, índices com páginas de 4 KB são usados durante a realização de consultas cujas expressões são *tipo 2* ou *tipo 4*.

#### 5.5.2.4 Inserção dos objetos nas instâncias da *Slim-tree*

Após a criação dos objetos que armazenam os dados das imagens consideradas e a criação de árvores *Slim-tree*, o sistema deve realizar a inserção de cada objeto na árvore métrica adequada. Tanto para efetuar esse tipo de inserção como para executar qualquer consulta por abrangência ou por vizinhança, é necessário utilizar uma função métrica apropriada.

Conforme foi informado na subseção 5.5.1, para possibilitar o uso dos quatro tipos de objetos que podem ser instanciados para representar imagens de faces humanas foram desenvolvidas e implementadas duas funções de similaridade métricas denominadas *MGF* e *MGF2*. Essas funções foram desenvolvidas com o propósito de que os cálculos de similaridade no sistema *MIGFIR* sejam realizados utilizando-se métricas cujas representações sejam intuitivas, considerando-se as peculiaridades dos conceitos empregados para representar as imagens (conjuntos de valores-verdade de predicados e conjuntos de medidas geométricas).

Para determinar o grau de similaridade entre qualquer par de imagens representadas por dois objetos de *TFaceImage0* ou de *TFaceImage1*, *MIGFIR* aplica a métrica *MGF*.

Como qualquer objeto  $C$  de qualquer uma dessas duas classes representa uma imagem  $I_C$  por meio de um conjunto nebuloso  $\mu(C) = \{\mu_1, \dots, \mu_n\}$  constituído por  $n \geq 1$  valores-verdade correspondentes a  $n$  predicados nebulosos válidos para  $I_C$ , decidiu-se que a similaridade entre  $C$  e qualquer objeto  $F$  da mesma classe seria calculada com base na diferença entre pares de valores-verdade correspondentes dos conjuntos nebulosos  $\mu(C)$  e  $\mu(F)$ .

Assim, seguindo um princípio do modelo *FFC*, a função *MGF* começou a ser definida a partir da seguinte equação:

$$D(C, F) = \frac{1}{n} [f(\mu(C) - \mu(F)) + f(\mu(F) - \mu(C))], \quad (5.1)$$

onde  $D(C, F)$  corresponde à distância entre as imagens representadas pelos objetos  $C$  e  $F$ ,  $\mu(C)$  e  $\mu(F)$  são os conjuntos nebulosos correspondentes a  $C$  e  $F$  (respectivamente),  $n \geq 1$  é a quantidade de valores-verdade em cada conjunto, e  $f(\cdot)$ , que é uma função cujo argumento deve ser um conjunto nebuloso  $\mu = \{\mu_1, \dots, \mu_n\}$ , é definida pela equação 5.2:

$$f(\mu) = \sum_{i=1}^n \mu_i. \quad (5.2)$$

Utilizando a equação (5.2) e a fórmula para obtenção da diferença entre dois conjuntos nebulosos (dada no capítulo 2 pela equação (2.7)), é possível reescrever (5.1), obtendo-se:

$$D(C, F) = \frac{1}{n} \left[ \sum_{i=1}^n \text{Max}[\mu_i(C) - \mu_i(F), 0] + \sum_{i=1}^n \text{Max}[\mu_i(F) - \mu_i(C), 0] \right], \quad (5.3)$$

onde se utiliza a função *Max*, que foi apresentada no capítulo 2, seção 2.3. A equação (5.3) pode ser simplificada, o que resulta em:

$$D(C, F) = \frac{1}{n} \sum_{i=1}^n [\text{Max}[\mu_i(C) - \mu_i(F), 0] + \text{Max}[\mu_i(F) - \mu_i(C), 0]]. \quad (5.4)$$

Como a equação (5.4) determina a *distância*  $D(C, F)$  entre as imagens  $I_C$  e  $I_F$ , e como tal distância sempre é um número real pertencente a  $[0, 1]$ , para se obter o valor da *similaridade*  $S(C, F)$  entre  $I_C$  e  $I_F$  basta utilizar a equação (5.5):

$$S(C, F) = 1 - D(C, F) . \quad (5.5)$$

Assim, as equações (5.4) e (5.5) constituem a métrica *MGF*. A equação (5.5) deixa evidente que a similaridade  $S(C, F)$  também é um número real pertencente a  $[0, 1]$ .

A partir da equação (5.4) pode-se obter a métrica *MGF2*. Para isso, bastam ser efetuadas duas modificações em (5.4): considerar o uso de conjuntos de medidas geométricas nessa equação, no lugar dos conjuntos nebulosos; e remover o termo de normalização  $1/n$ , já que as medidas geométricas não são normalizadas para o intervalo  $[0, 1]$ . Após essas modificações, a fórmula da função *MGF2*, para calcular a distância entre qualquer par de imagens representadas por dois objetos  $A$  e  $B$  de *TFaceImage2* ou de *TFaceImage3*, corresponde à seguinte equação:

$$D(A, B) = \sum_{i=1}^n [\text{Max}[\gamma_i(A) - \gamma_i(B), 0] + \text{Max}[\gamma_i(B) - \gamma_i(A), 0]] , \quad (5.6)$$

onde  $D(A, B)$  é o valor da distância entre os objetos  $A$  e  $B$ ,  $\gamma(A) = \{\gamma_1, \dots, \gamma_n\}$  é o conjunto de  $n \geq 1$  medidas geométricas que o objeto  $A$  utiliza para representar a imagem  $I_A$  (o mesmo vale para  $\gamma(B)$ ),  $\gamma_i(A)$  é a  $i$ -ésima medida geométrica inserida em  $\gamma(A)$ , e  $\gamma_i(B)$  é a  $i$ -ésima medida inserida em  $\gamma(B)$ .

É relevante enfatizar que como a similaridade entre as imagens representadas por  $A$  e  $B$  é inversamente proporcional à distância entre elas, tem-se que quanto maior é o valor  $D(A, B)$  dado pela equação (5.6), menor é a similaridade entre essas duas imagens.

Considerando-se novamente a função *MGF*, para ilustrar o seu uso será apresentado um exemplo simples a partir da Figura 5.12, que mostra três imagens de esboços de faces humanas.

Considere que apenas as cinco medidas destacadas na Figura 5.13, que são exatamente as mesmas características geométricas consideradas neste trabalho, são utilizadas para representar cada esboço de face mostrado na Figura 5.12. Para esses



esboços, suponha que os valores das cinco características geométricas são os mostrados na Tabela 5.2.

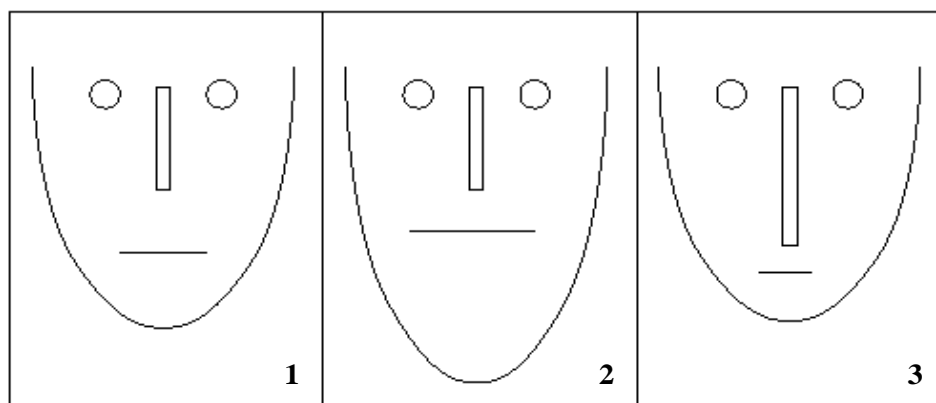


Figura 5.12 – Três esboços simples de faces humanas.

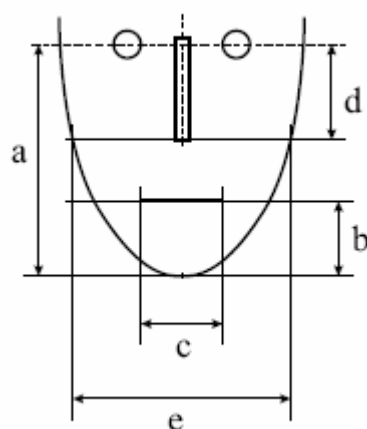


Figura 5.13 – Esboço de uma face humana com 5 medidas geométricas em destaque.

Tabela 5.2 – Valores de características geométricas referentes aos três esboços de faces considerados.

<b>Característica da Imagem</b>	<b>Face 1</b>	<b>Face 2</b>	<b>Face 3</b>
comprimento da face (medida <i>a</i> )	13	17	12
comprimento do queixo (medida <i>b</i> )	4	7	2
largura da boca (medida <i>c</i> )	4	6	2
comprimento do nariz (medida <i>d</i> )	5	5	7
largura da face (medida <i>e</i> )	10	9	10

A partir dos valores de características exibidos na Tabela 5.2 e considerando o uso de funções de pertinência nebulosas apropriadas (fornecidas por especialistas do domínio), suponha que tenham sido determinados os predicados nebulosos e os respectivos valores-verdade mostrados na Tabela 5.3.

Tabela 5.3 – Predicados nebulosos válidos e respectivos valores-verdade referentes aos três esboços de faces considerados.

Predicados Nebulosos	Face 1	Face 2	Face 3
Face longa	0,7	0,9	0,7
Queixo longo	0,4	0,9	0,2
Boca larga	0,4	0,7	0,3
Nariz curto	0,7	0,7	0,1
Face larga	0,6	0,5	0,6

Os valores-verdade mostrados na Tabela 5.3 podem ser agrupados em conjuntos nebulosos  $\mu$  para representar as três imagens consideradas nesse exemplo. Fazendo-se isso, tem-se que  $\mu_{Face1} = \{0,7, 0,4, 0,4, 0,7, 0,6\}$ ,  $\mu_{Face2} = \{0,9, 0,9, 0,7, 0,7, 0,5\}$  e  $\mu_{Face3} = \{0,7, 0,2, 0,3, 0,1, 0,6\}$ .

Por fim, para calcular o grau de similaridade entre cada par de esboços de faces, basta utilizar os valores-verdade dos respectivos conjuntos nebulosos e as equações (5.4) e (5.5), correspondentes à métrica *MGF*. Fazendo-se isso, obtém-se o seguinte:

- Similaridade entre *Face 1* e *Face 2*:  $S(Face1, Face2) = 0,78$ .
- Similaridade entre *Face 1* e *Face 3*:  $S(Face1, Face3) = 0,82$ .
- Similaridade entre *Face 2* e *Face 3*:  $S(Face2, Face3) = 0,62$ .

Portanto, de acordo com a função *MGF*, as imagens mais similares são *Face 1* e *Face 3*, e as menos similares são *Face 2* e *Face 3*.

Com relação aos conjuntos de dados, caso sejam incluídas novas imagens em um conjunto que já foi indexado, o sistema insere os dados extraídos das novas imagens nos

respectivos índices preexistentes. Desse modo, esses índices não precisam ser reconstruídos. Entretanto, para a versão do sistema desenvolvida até a elaboração desta dissertação, se forem excluídas imagens já indexadas é necessário reconstruir os índices que as contêm.

#### **5.5.2.5 Recuperação das imagens similares a cada componente da expressão**

Neste ponto do processo de execução de uma consulta complexa tem-se a primeira tarefa relacionada diretamente à recuperação de imagens que podem satisfazer tal consulta. Essa tarefa consiste em buscar e recuperar com eficiência e precisão satisfatórias as imagens similares a cada componente da expressão de busca corrente.

Para realizar essa tarefa e as das próximas subseções, foi desenvolvido o algoritmo *Search\_Result\_Images*. Esse algoritmo, apresentado na Figura 5.14, inicialmente executa a tarefa relatada no parágrafo anterior; em seguida, emprega funções de combinação de valores para determinar as listas de imagens que satisfazem as subexpressões da consulta e também para determinar a lista de imagens que satisfazem a expressão de busca completa.

É importante destacar que não foram incluídos na Figura 5.14 diversos detalhes da implementação, pois o objetivo é apresentar apenas a idéia fundamental do algoritmo.

A Figura 5.14 permite notar que a tarefa à qual esta subseção se refere é realizada através da execução dos passos 5 e 6. É importante ressaltar que, no passo 5, a ordem de execução das consultas correspondentes aos componentes é sempre da esquerda para a direita.

Para exemplificar a realização dessa tarefa, considere que o sistema está executando a consulta  $C_3$  (definida no início da seção 5.5.2), cuja expressão de busca contém cinco componentes: *Face Width: Wide*, *Face Length: Very Long*, *Mouth Width: Medium*, *Face Length: Long* e *Nose Length: Medium*. Neste caso, quando são realizados os passos 5 e 6 de *Search\_Result\_Images*, são construídas cinco listas encadeadas de objetos que representam as imagens com determinados graus de similaridade em relação a cada um desses cinco componentes da expressão.

### Algoritmo *Search\_Result\_Images*

Início

1.  $i = 1$ ;
2. Enquanto  $i$  for menor ou igual ao total de subexpressões que constituem a expressão de busca, faça:
  3.  $j = 1$ ;
  4. Enquanto  $j$  for menor ou igual ao total de componentes que constituem a subexpressão corrente, faça:
    5. Para o  $j$ -ésimo componente da subexpressão, execute sua respectiva consulta por similaridade sobre o índice métrico apropriado, para gerar uma lista encadeada ordenada de objetos que representam as imagens que satisfazem tal consulta (imagens com determinados graus de similaridade em relação ao  $j$ -ésimo componente);
    6. Insira o endereço do início da lista de imagens gerada no passo 5 no vetor *components\_vector*, que deve armazenar os endereços do início de todas as listas geradas para os componentes da subexpressão corrente;
    7.  $j = j + 1$ ;
  8. Fim do Enquanto;
  9. *subresult\_pointer* = *Combine\_Subresults(components\_vector, subexpression\_op, relevance\_grades)*;
  10. Insira *subresult\_pointer*, que é o endereço do início da lista de imagens gerada no passo 9, no vetor *subexpressions\_vector*, que deve armazenar os endereços de todas as listas criadas através da execução das subexpressões;
11.  $i = i + 1$ ;
12. Fim do Enquanto;
13. *result\_pointer* = *Combine\_Subresults(subexpressions\_vector, expression\_op, relevance\_grades)*;
14. Enviar o resultado final da consulta, isto é, a lista encadeada ordenada de imagens que inicia no endereço *result\_pointer*, para o módulo *MIGFIR\_QueryInterfaces* exibir as imagens recuperadas apropriadamente.

Fim.

Figura 5.14 – Algoritmo *Search\_Result\_Images*.

### 5.5.2.6 Execução de operações lógicas entre componentes

Após a geração de listas de imagens correspondentes aos componentes da expressão de busca, é necessário, para cada subexpressão da consulta, executar a operação lógica especificada pelo uso do operador *and* ou do operador *or* entre os componentes da subexpressão. Essa tarefa corresponde à realização dos passos 9 e 10 do algoritmo *Search\_Result\_Images* (Figura 5.14).

A execução da operação booleana especificada para uma dada subexpressão  $D_I$  que contém  $n$  componentes visa combinar as listas de imagens associadas a esses componentes em uma lista encadeada  $L_{D_I}$  de objetos que representem as imagens que satisfazem  $D_I$ . Portanto, se o operador lógico *and* for empregado para relacionar os  $n$  componentes de  $D_I$  entre si, a lista  $L_{D_I}$  será composta apenas pelas imagens que estiverem presentes nas  $n$  listas. Por outro lado, se o operador usado para relacionar tais componentes entre si for o *or*, a lista  $L_{D_I}$  será formada por todas as imagens das  $n$  listas, sem objetos repetidos.

A criação de uma lista encadeada  $L_{D_I}$  ocorre a partir da chamada da função *Combine\_Subresults(components\_vector, subexpression\_op, relevance\_grades)* no passo 9 de *Search\_Result\_Images*. Para uma subexpressão  $D_I$ , o parâmetro *components\_vector* é o vetor que guarda os endereços do início de todas as listas de imagens correspondentes aos componentes de  $D_I$ , *subexpression\_op* é o operador lógico usado em  $D_I$  e *relevance\_grades* é o vetor que contém os graus de relevância atribuídos aos componentes.

Para a inserção de uma imagem  $I_0$  na lista encadeada  $L_{D_I}$ , é necessário calcular o grau de similaridade de  $I_0$  em relação à subexpressão  $D_I$ , já que os valores de similaridade são usados para ordenar  $L_{D_I}$ . Para calcular esses valores, são utilizadas funções chamadas neste trabalho de funções de combinação.

No caso de uma subexpressão  $D_k$  conter o operador booleano *and* entre seus  $n$  componentes, a função de combinação a ser empregada para determinar o grau de similaridade  $S_X(I_0, D_k)$  de uma imagem  $I_0$  em relação a  $D_k$  corresponde à equação (5.7):

$$S_X(I_0, D_k) = \frac{\sum_{i=1}^n (rlv_{C_i} \times S_C(I_0, C_i))}{\sum_{i=1}^n rlv_{C_i}}, \quad (5.7)$$

onde  $C_i$  é o  $i$ -ésimo componente de  $D_k$ ,  $rlv_{C_i}$  é o grau de relevância estabelecido pelo usuário para  $C_i$  e  $S_C(I_0, C_i)$  é o grau de similaridade de  $I_0$  em relação a  $C_i$ .

A equação (5.7) determina o valor de similaridade  $S_X(I_0, D_k)$  através do cálculo da média ponderada dos valores de similaridade referentes aos componentes de  $D_k$ , tal que os graus de relevância são usados como pesos. Assim, quanto maior a relevância atribuída a um determinado componente, maior será a sua influência no valor  $S_X(I_0, D_k)$ .

Já no caso em que uma subexpressão  $D_j$  contém o operador booleano *or* entre seus  $n$  componentes, há duas funções de combinação utilizadas para determinar o grau de similaridade  $S_X(I_0, D_j)$  de uma imagem  $I_0$  em relação a  $D_j$ : se a subexpressão contém predicados nebulosos, emprega-se a equação (5.8); mas se a subexpressão contém medidas geométricas, utiliza-se a equação (5.9).

$$S_X(I_0, D_j) = \text{Max}\{(rlv_{C_i} \times S_C(I_0, C_i)) \mid i = 1..n\} \quad (5.8)$$

$$S_X(I_0, D_j) = \text{Min}\left\{\left(\frac{1}{rlv_{C_i}} \times S_C(I_0, C_i)\right) \mid i = 1..n\right\} \quad (5.9)$$

Nas equações (5.8) e (5.9),  $C_i$  é o  $i$ -ésimo componente de  $D_j$ , e  $rlv_{C_i}$  é o grau de relevância definido pelo usuário para  $C_i$ . Enquanto a função representada pela equação (5.8) determina o valor de similaridade  $S_X(I_0, D_j)$  usando a função *Max*, a função dada pela equação (5.9) calcula  $S_X(I_0, D_j)$  empregando *Min*. Esta diferença se deve ao fato de que na equação (5.8) o parâmetro  $S_C(I_0, C_i)$  significa o valor de similaridade de  $I_0$  com relação a  $C_i$ , ao passo que na equação (5.9) esse mesmo parâmetro significa o valor da distância de  $I_0$  a  $C_i$ .

### 5.5.2.7 Geração do resultado final da consulta

Após a criação das listas de imagens correspondentes às subexpressões, é necessário executar a operação lógica especificada pelo emprego de *and* ou de *or* entre essas subexpressões a fim de gerar o resultado final da consulta complexa. Essa tarefa corresponde à realização dos passos 13 e 14 do algoritmo *Search\_Result\_Images* (Figura 5.14).

A execução da operação lógica especificada para uma expressão de busca  $E$  composta por  $m$  subexpressões tem como objetivo combinar as listas de imagens associadas às

subexpressões em uma lista encadeada  $L_E$  de objetos que representem as imagens que satisfazem  $E$ . Logo, se o operador *and* for empregado para relacionar as  $m$  subexpressões entre si, a lista  $L_E$  será composta somente pelas imagens que estiverem presentes nas outras  $m$  listas. Entretanto, se o operador usado para relacionar tais subexpressões entre si for o *or*,  $L_E$  será formada por todas as imagens das outras  $m$  listas, sem objetos repetidos.

A criação da lista encadeada final  $L_E$  ocorre a partir da chamada da função *Combine\_Subresults(subexpressions\_vector, expression\_op, relevance\_grades)* no passo 13 de *Search\_Result\_Images*. O parâmetro *subexpressions\_vector* é o vetor que armazena os endereços do início de todas as listas de imagens correspondentes às subexpressões, *expression\_op* é o operador lógico empregado na expressão de busca e *relevance\_grades* é o vetor que contém os graus de relevância atribuídos às subexpressões.

Para incluir uma imagem  $I_I$  no resultado final  $L_E$ , é necessário determinar o grau de similaridade de  $I_I$  em relação à expressão  $E$ , pois os valores de similaridade são utilizados para ordenar  $L_E$ . Para determinar esses valores, são empregadas funções de combinação.

Se uma expressão de busca  $E_k$  contiver o operador lógico *and* entre suas  $m$  subexpressões, a função de combinação a ser utilizada para calcular o grau de similaridade  $S_F(I_I, E_k)$  de uma imagem  $I_I$  em relação a  $E_k$  corresponde à equação (5.10):

$$S_F(I_I, E_k) = \frac{\sum_{i=1}^m (rlv_{D_i} \times S_X(I_I, D_i))}{\sum_{i=1}^m rlv_{D_i}}, \quad (5.10)$$

onde  $D_i$  é a  $i$ -ésima subexpressão de  $E_k$ ,  $rlv_{D_i}$  é o grau de relevância estabelecido pelo usuário para  $D_i$  e  $S_X(I_I, D_i)$  é o valor de similaridade de  $I_I$  em relação a  $D_i$ .

A equação (5.10) determina o grau de similaridade  $S_F(I_I, E_k)$  por meio do cálculo da média ponderada dos valores de similaridade referentes às subexpressões de  $E_k$ , de modo que os graus de relevância são empregados como pesos. Portanto, quanto maior a relevância atribuída a uma determinada subexpressão, maior será a sua influência no valor  $S_F(I_I, E_k)$ .

Já na situação em que uma expressão de busca  $E_j$  contém o operador *or* entre suas  $m$  subexpressões, duas funções de combinação são empregadas para determinar o valor de

similaridade  $S_F(I_I, E_j)$  de uma imagem  $I_I$  em relação a  $E_j$ : se cada subexpressão de  $E_j$  é constituída por predicados nebulosos, utiliza-se a equação (5.11); mas se cada subexpressão contém medidas geométricas, emprega-se a equação (5.12).

$$S_F(I_I, E_j) = \text{Max}\{rlv_{D_i} \times S_X(I_I, D_i) \mid i = 1..m\} \quad (5.11)$$

$$S_F(I_I, E_j) = \text{Min}\left\{\left(\frac{1}{rlv_{D_i}} \times S_X(I_I, D_i)\right) \mid i = 1..m\right\} \quad (5.12)$$

Nas equações (5.11) e (5.12),  $D_i$  é a  $i$ -ésima subexpressão de  $E_j$ , e  $rlv_{D_i}$  é o grau de relevância definido pelo usuário para  $D_i$ . Enquanto a função representada pela equação (5.11) determina o valor de similaridade  $S_F(I_I, E_j)$  usando a função *Max*, a equação (5.12) mostra uma função que calcula  $S_F(I_I, E_j)$  com a utilização da função *Min*. Esta diferença se deve ao fato de que na equação (5.11) o parâmetro  $S_X(I_I, D_i)$  significa o valor de similaridade de  $I_I$  em relação à subexpressão  $D_i$ , ao passo que em (5.12) esse mesmo parâmetro significa o valor da distância de  $I_I$  a  $D_i$ .

As funções de combinação representadas pelas equações (5.7) a (5.12) foram definidas para a execução de operações *and* e *or* porque propiciaram resultados satisfatórios em termos de precisão.

Por fim, após a realização das sete tarefas discutidas desde a subseção 5.5.2.1, o endereço do início da lista de imagens final é enviado ao módulo *MIGFIR\_QueryInterfaces* para que as imagens que constituem o resultado final da consulta complexa sejam exibidas ordenadamente.

## 5.6 CONSIDERAÇÕES FINAIS

O objetivo deste capítulo foi apresentar detalhadamente o sistema *MIGFIR*, destacando a alta flexibilidade na elaboração de consultas, o desenvolvimento de duas métricas para o cálculo de graus de similaridade entre imagens, e os principais algoritmos para a execução das consultas.

Inicialmente, foram abordados aspectos relativos à arquitetura do sistema na seção 5.2, com destaque para sua divisão em três módulos: *MIGFIR\_PredicateGenerator*, *MIGFIR\_QueryInterfaces* e *MIGFIR\_QueryImplementation*.



Na seção 5.3 abordou-se o módulo *MIGFIR\_PredicateGenerator*, o qual é fundamental para que predicados nebulosos possam ser utilizados por *MIGFIR* na indexação e na recuperação de imagens.

Na seção 5.4 foi apresentada a parte do sistema denominada *MIGFIR\_QueryInterfaces*, cuja função básica é oferecer ao usuário as interfaces para a elaboração de consultas e para a exibição dos resultados dessas consultas.

Por fim, na seção 5.5 foi apresentado o módulo *MIGFIR\_QueryImplementation*, que é responsável pela representação apropriada das imagens indexadas e funções de similaridade, pelo gerenciamento de índices métricos, e pela execução de qualquer consulta baseada em imagens-exemplo ou baseada em predicados nebulosos.

O próximo capítulo apresentará a avaliação experimental do sistema *MIGFIR*.

## 6 AVALIAÇÃO EXPERIMENTAL

### 6.1 CONSIDERAÇÕES INICIAIS

Os principais objetivos deste capítulo são:

- avaliar a precisão do sistema *MIGFIR* na execução de consultas baseadas em imagens-exemplo;
- comparar a eficiência (analisada em termos de tempo de execução de consulta) do sistema *MIGFIR* à de um sistema relativamente similar durante a realização de consultas baseadas em predicados nebulosos e em imagens-exemplo.

Para a realização dos experimentos, foram utilizados conjuntos de imagens de faces humanas reais e conjuntos de medidas faciais sintéticas.

Os principais conjuntos reais considerados foram um constituído por 30 fotografias de alunos da Universidade Federal de São Carlos, um fornecido por *AT&T Laboratories* (formalmente denominado *The ORL Database of Faces*<sup>5</sup>, com 10 imagens diferentes de cada indivíduo, para um total de 40 indivíduos) e um denominado *Yale Database*<sup>6</sup> (constituído por 10 imagens diferentes de cada um dos 16 indivíduos fotografados). Para cada imagem real, foram extraídos os cinco valores das características faciais citadas no capítulo 5, subsecção 5.4.1, que possibilitam a definição dos termos lingüísticos discutidos na subsecção 5.4.2.

---

<sup>5</sup> Banco de imagens de faces humanas disponibilizado por *AT&T Laboratories Cambridge* em <http://www.cam-orl.co.uk/facedatabase.html>.

<sup>6</sup> Base de imagens de faces humanas disponibilizada em <http://cvc.yale.edu>.

A obtenção das cinco medidas faciais pôde ser realizada por meio do uso de um sistema para extração de medidas geométricas de faces humanas que foi desenvolvido por Oriani e Moreira<sup>7</sup>, pesquisadores do Departamento de Computação da Universidade Federal de São Carlos.

Já os quatro principais conjuntos sintéticos empregados contêm as medidas de 1000, 3000, 10000 e 30000 faces humanas. Esses conjuntos foram gerados de forma semi-automática por um algoritmo desenvolvido especificamente para esse fim e foram utilizados pela primeira vez durante a avaliação experimental da primeira versão de *MIGFIR*, apresentada em [GAVIOLI; BIAJIZ; MOREIRA, 2005].

Os experimentos foram executados em um computador pessoal com 1 processador *AMD Athlon XP 2400+*, sistema operacional *Windows XP Professional*, 256 megabytes de memória principal e 1 disco rígido de 80 gigabytes.

Na seção 6.2 apresentam-se os resultados experimentais referentes à avaliação da precisão do sistema *MIGFIR* e na seção 6.3 apresentam-se os resultados obtidos durante a avaliação da eficiência desse sistema.

## 6.2 AVALIAÇÃO DA PRECISÃO NA EXECUÇÃO DE CONSULTAS

Conforme foi informado e justificado no capítulo 5, subseção 5.4.1, durante o desenvolvimento do projeto de mestrado decidiu-se avaliar a similaridade entre faces humanas com base em medidas geométricas. Portanto, para evitar a interferência de outros tipos de características (tais como cor, textura e formas) nos resultados dos julgamentos humanos de similaridade, considerou-se, para a realização dos experimentos de avaliação da precisão do sistema *MIGFIR*, um conjunto constituído por dez esboços (*sketches*) de faces humanas similares àqueles exibidos na Figura 5.12.

A precisão de *MIGFIR* na realização de consultas nebulosas é completamente dependente das funções de pertinência nebulosas utilizadas, as quais podem ser definidas por especialistas do domínio com base no estudo do julgamento de similaridade empregado por prováveis usuários do sistema.

---

<sup>7</sup> Um texto que apresenta o sistema de extração de medidas desenvolvido por André Oriani e pelo professor Jander Moreira foi disponibilizado em <http://www.comp.ufscar.br/~aoriani/face/extrator.pdf>.

Como esse estudo que visa definir funções de pertinência apropriadas está fora do escopo deste trabalho, nesta seção são apresentados resultados da avaliação da precisão na execução de consultas baseadas em imagens-exemplo. É importante lembrar que para realizar este tipo de consulta, *MIGFIR* aplica a métrica *MGF2* como função de similaridade.

Para efetuar essa avaliação, decidiu-se seguir uma metodologia similar à que foi usada em [SANTINI; JAIN, 1999], pois aquela foi considerada bastante apropriada e também porque foi possível notar que ainda não há na literatura uma metodologia para esse tipo de avaliação que seja um padrão aceito universalmente.

Assim, inicialmente foram escolhidos cinco alunos de graduação que não tinham conhecimento dos assuntos discutidos nesta dissertação. Solicitou-se que eles ordenassem de modo decrescente, de acordo com o julgamento de similaridade de cada um, nove dos dez esboços considerados segundo a similaridade com o décimo esboço, que foi selecionado aleatoriamente e considerado a imagem-exemplo. Cada indivíduo elaborou três ordenações, correspondentes a três imagens-exemplo diferentes, e em seguida dividiu cada ordenação em três grupos: o primeiro grupo contendo os esboços de faces considerados muito similares ao esboço de consulta, o segundo contendo as faces consideradas não muito similares à imagem-exemplo, e o terceiro armazenando as faces completamente diferentes do exemplo. Para verificar a estabilidade das ordenações geradas, após duas semanas quatro dos cinco indivíduos repetiram o experimento descrito acima, com as mesmas imagens-exemplo.

Cada ordenação de cada indivíduo foi comparada à ordenação gerada pelo sistema *MIGFIR*, considerando-se a mesma imagem-exemplo. A medida denominada *deslocamento ponderado* (*weighted displacement* [NARASIMHALU; KANKANHALLI; WU, 1997]) foi aplicada para comparar as ordenações.

Para utilizar a medida *weighted displacement*, considere uma consulta  $q$  que é executada sobre um conjunto de  $n$  imagens. Considere também a ordenação  $O_H = \{I_1, \dots, I_n\}$  fornecida por um indivíduo como a ordenação ideal e uma medida de relevância  $R(I, q)$ , cujo valor é um número real em  $[0, 1]$  e tal que para a ordem ideal:

$$\forall i \quad R(I_i, q) \geq R(I_{i+1}, q) . \quad (6.1)$$

Assim como foi efetuado em [SANTINI; JAIN, 1999], optou-se por considerar  $R(I_i, q) = 0,8$  para imagens de faces muito similares ao esboço de consulta,  $R(I_i, q) = 0,5$  para imagens não muito similares e  $R(I_i, q) = 0,05$  para as faces completamente diferentes do esboço-exemplo.

A ordenação do conjunto de  $n$  imagens obtida através do uso da função de similaridade *MGF2* será representada por  $O_M = \{I_{k_1}, \dots, I_{k_n}\}$ , tal que  $k_1, \dots, k_n$  é uma permutação de  $1, \dots, n$ . Portanto, o *deslocamento* de  $I_i$  em  $O_M$  é estabelecido como:

$$d_q(I_i) = |i - k_i|, \quad (6.2)$$

e o *deslocamento ponderado relativo* de  $O_M$  é estabelecido como:

$$W_q = \frac{\sum_{i=1}^n R(I_i, q) d_q(I_i)}{F}, \quad (6.3)$$

tal que  $F$  é um fator de normalização definido como  $\lfloor n^2/2 \rfloor$  (a justificativa para esta definição de  $F$  pode ser encontrada em [SANTINI; JAIN, 1999]). Devido ao uso deste fator de normalização, tem-se  $W_q = 0$  quando  $O_M = O_H$ , e  $W_q \leq 1$  para qualquer  $O_M$  e  $R$ .

Com a utilização de (6.2) e (6.3), calculou-se o deslocamento ponderado de cada ordenação gerada por *MIGFIR* para corresponder a cada ordenação fornecida por cada um dos indivíduos. Em seguida, computou-se a média aritmética  $m$  dos valores de deslocamento ponderado, obtendo-se  $m = 0,088$ .

Comparando o valor de  $m$  produzido por *MIGFIR* aos valores obtidos nos experimentos apresentados em [SANTINI; JAIN, 1999], pôde-se notar que as diferenças entre o valor de  $m$  e as médias correspondentes à métrica Euclideana e à função *FFC* são muito pequenas sob o ponto de vista prático, além do fato de  $m$  ser um número bastante próximo de 0.

Pode-se concluir que o sistema *MIGFIR* exibe um grau de precisão satisfatório na recuperação de imagens de faces humanas similares a uma imagem-exemplo quando são empregadas as cinco medidas geométricas consideradas neste trabalho, pois a avaliação de similaridade entre esboços de faces efetuada por *MIGFIR* se aproxima bastante do julgamento humano de similaridade (considerado gerador de ordenações ideais).

### 6.3 AVALIAÇÃO DA EFICIÊNCIA NA EXECUÇÃO DE CONSULTAS

Para a realização de uma comparação justa, também foi implementado um sistema bastante similar ao *MIGFIR*, mas que usa a varredura seqüencial ao invés da *Slim-tree* para acessar os dados das imagens que satisfazem cada consulta. Esse sistema será chamado a partir daqui de *SeqScan*.

Com o propósito de obter resultados consistentes, foram realizadas muitas consultas baseadas em exemplos e em predicados com diferentes níveis de complexidade. Nesta seção serão apresentadas apenas sete consultas, já que as conclusões proporcionadas pelas demais em termos de economia de tempo de execução são similares às conclusões que as escolhidas proporcionam. Das sete, que foram executadas tanto pelo sistema *MIGFIR* como por *SeqScan*, quatro baseiam-se em predicados e três baseiam-se em exemplos. Além disso, serão exibidos resultados referentes apenas aos conjuntos de dados sintéticos, devido ao fato dos conjuntos reais possuírem quantidades de imagens consideradas muito pequenas.

Para simplificar os experimentos, foi considerado o grau de relevância 100% para todos os predicados que compõem cada consulta nebulosa e para todas as medidas faciais que constituem cada consulta baseada em exemplos. Por outro lado, foram considerados vários valores diferentes para os raios de abrangência e para as quantidades de vizinhos mais próximos das consultas por similaridade utilizadas, que propiciaram a obtenção de resultados conclusivos em termos da economia de tempo alcançada com o uso de *MIGFIR*.

As quatro consultas nebulosas consideradas nesta seção são:

- Consulta  $F_1$ : encontrar as imagens que satisfazem a expressão de busca (*Face Length: Very Long and Mouth Width: Wide and Chin Length: Long*).
- Consulta  $F_2$ : encontrar as imagens que satisfazem a expressão de busca (*Face Length: Long and Chin Length: Medium and Mouth Width: Wide and Nose Length: Short and Face Width: Wide*).
- Consulta  $F_3$ : encontrar as imagens que satisfazem a expressão de busca ( (*Face Length: Long or Face Length: Medium*) and (*Mouth Width: Medium and Nose Length: Short and Chin Length: Short*) ).

- Consulta  $F_4$ : encontrar as imagens que satisfazem a expressão de busca ( (*Face Length: Long and Chin Length: Medium and Mouth Width: Narrow*) or (*Face Width: Wide and Nose Length: Long and Face Length: Very Long*) or (*Mouth Width: Wide and Face Width: Medium and Face Length: Short*) ).

A consulta  $F_1$ , cuja expressão de busca é *tipo 3*, é constituída por três predicados nebulosos relacionados entre si através do operador booleano *and*. A Tabela 6.1 exhibe os parâmetros escolhidos para as consultas por similaridade que foram associadas aos predicados de  $F_1$ .

Note-se que na primeira coluna da Tabela 6.1 há um rótulo para identificar cada configuração dos parâmetros das consultas por similaridade.

Tabela 6.1 – Quatro configurações de parâmetros para a execução da consulta nebulosa  $F_1$ .

<b>Identificador da configuração das consultas por similaridade</b>	<b>1º predicado</b>	<b>2º predicado</b>	<b>3º predicado</b>
<i>Config1</i>	<i>RQ</i> com raio = 0,1	<i>K-NNQ</i> com k = 3	<i>RQ</i> com raio = 0,1
<i>Config2</i>	<i>K-NNQ</i> com k = 3	<i>RQ</i> com raio = 0,1	<i>K-NNQ</i> com k = 3
<i>Config3</i>	<i>RQ</i> com raio = 0,3	<i>K-NNQ</i> com k = 15	<i>RQ</i> com raio = 0,3
<i>Config4</i>	<i>K-NNQ</i> com k = 15	<i>RQ</i> com raio = 0,3	<i>K-NNQ</i> com k = 15

Utilizando os rótulos definidos na Tabela 6.1, os gráficos ilustrados na Figura 6.1 apresentam o tempo consumido durante a execução da consulta  $F_1$  para cada configuração das consultas por similaridade e cada conjunto de medidas faciais considerado. Nesta e nas demais figuras semelhantes desta seção, o eixo vertical representa o tempo consumido (medido em *segundos*) e o eixo horizontal indica o tamanho de cada conjunto de imagens considerado.

A Figura 6.1 permite observar que em todas as situações avaliadas envolvendo a consulta  $F_1$  o sistema *MIGFIR* foi mais eficiente que *SeqScan*. A economia de tempo para a execução de  $F_1$  proporcionada por *MIGFIR* variou de 19,2% (na situação em que o tempo gasto por ele foi 1,005s e o tempo de *SeqScan* foi 1,198s) até 358,3% (na situação em que o tempo consumido por *MIGFIR* foi 0,012s e o de *SeqScan* foi 0,055s).

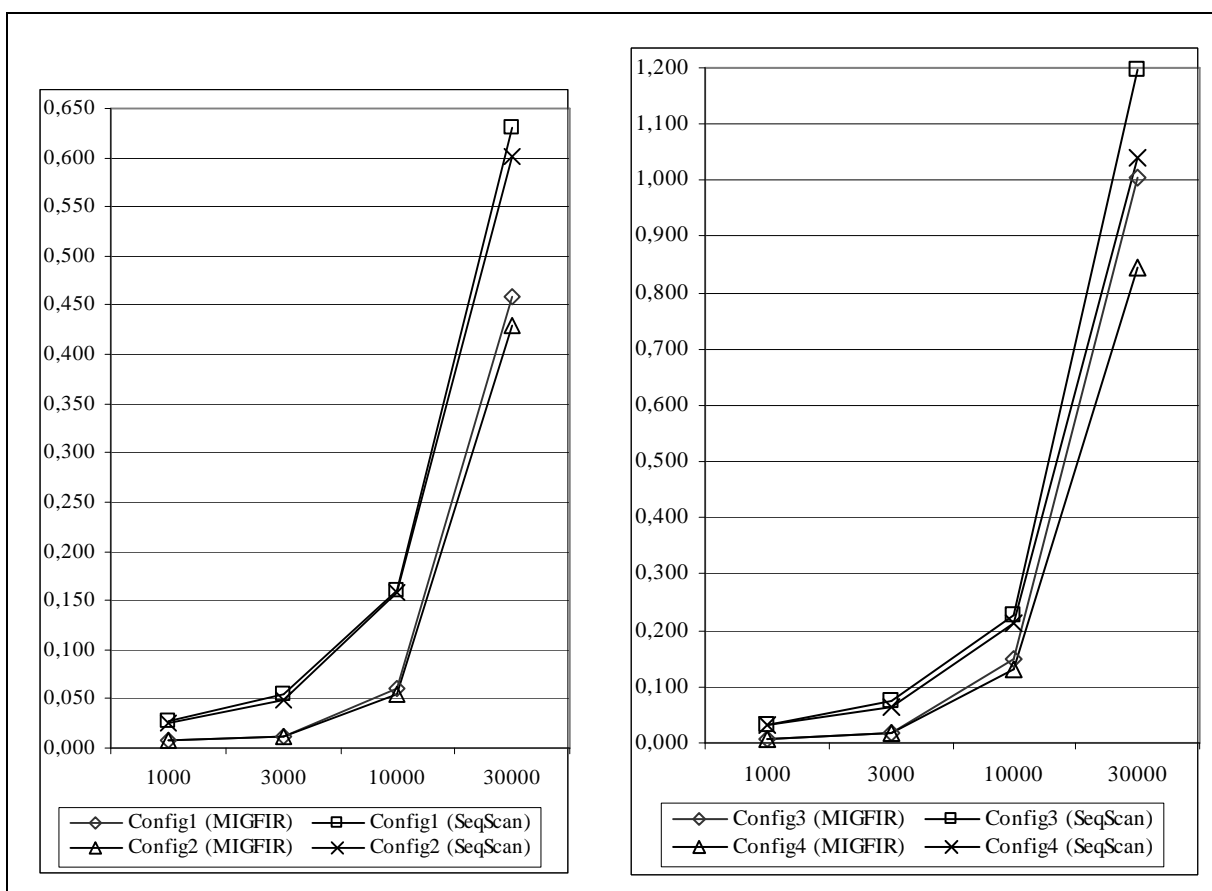


Figura 6.1 – Gráficos apresentando o tempo consumido durante a execução de  $F_1$  para cada configuração de consulta por similaridade e cada conjunto de dados, considerando-se a execução em *MIGFIR* e em *SeqScan*.

A consulta  $F_2$  possui expressão de busca *tipo 4*, ou seja, ela é constituída por uma subexpressão de busca que contém um predicado nebuloso para cada medida facial considerada neste trabalho. A Tabela 6.2 mostra os parâmetros selecionados para as consultas por similaridade que foram associadas à subexpressão de  $F_2$ .

Tabela 6.2 – Quatro configurações de parâmetros para a execução da consulta nebulosa  $F_2$ .

Identificador da configuração da consulta por similaridade	Subexpressão (5 predicados)
<i>Config21</i>	<i>K-NNQ</i> com $k = 3$
<i>Config22</i>	<i>K-NNQ</i> com $k = 15$
<i>Config23</i>	<i>RQ</i> com raio = 0,1
<i>Config24</i>	<i>RQ</i> com raio = 0,3



Empregando os rótulos exibidos na primeira coluna da Tabela 6.2, os gráficos que compõem a Figura 6.2 apresentam o tempo gasto durante a execução de  $F_2$  para cada configuração de consulta por similaridade e cada conjunto de dados considerado.

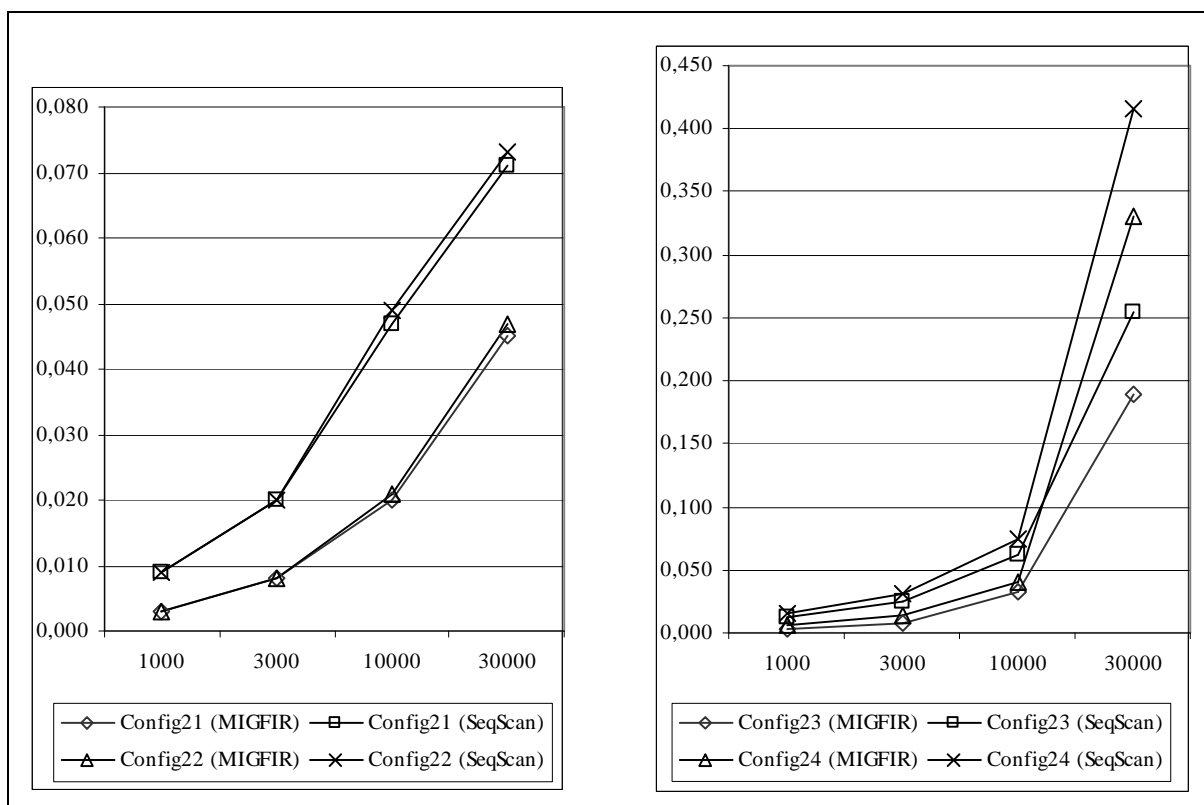


Figura 6.2 – Gráficos apresentando o tempo consumido durante a execução de  $F_2$  para cada configuração de consulta por similaridade e cada conjunto de dados, considerando-se a execução em *MIGFIR* e em *SeqScan*.

A Figura 6.2 possibilita notar que em todos os casos analisados envolvendo  $F_2$  o sistema *MIGFIR* foi mais eficiente que *SeqScan*. A economia de tempo de execução variou de 26,2% (na situação em que *MIGFIR* gastou 0,330s e *SeqScan* gastou 0,416s) até 300% (na situação em que o tempo consumido por *MIGFIR* foi 0,003s e o de *SeqScan* foi 0,012s).

A consulta  $F_3$ , cuja expressão de busca é *tipo 3*, é constituída por duas subexpressões relacionadas entre si através do operador booleano *and*, tal que a primeira subexpressão contém dois predicados relacionados entre si pelo operador *or* e a segunda contém três predicados relacionados por *and*. A Tabela 6.3 exhibe os parâmetros escolhidos para as consultas por similaridade que foram associadas aos predicados de  $F_3$ .

Tabela 6.3 – Quatro configurações de parâmetros para a execução da consulta nebulosa  $F_3$ .

Identificador da configuração das consultas por similaridade	1° predicado	2° predicado	3° predicado	4° predicado	5° predicado
<i>Config31</i>	<i>K-NNQ</i> com $k = 3$	<i>RQ</i> com raio = 0,1	<i>K-NNQ</i> com $k = 3$	<i>RQ</i> com raio = 0,1	<i>K-NNQ</i> com $k = 3$
<i>Config32</i>	<i>RQ</i> com raio = 0,1	<i>K-NNQ</i> com $k = 3$	<i>RQ</i> com raio = 0,1	<i>K-NNQ</i> com $k = 3$	<i>RQ</i> com raio = 0,1
<i>Config33</i>	<i>K-NNQ</i> com $k = 15$	<i>RQ</i> com raio = 0,3	<i>K-NNQ</i> com $k = 15$	<i>RQ</i> com raio = 0,3	<i>K-NNQ</i> com $k = 15$
<i>Config34</i>	<i>RQ</i> com raio = 0,3	<i>K-NNQ</i> com $k = 15$	<i>RQ</i> com raio = 0,3	<i>K-NNQ</i> com $k = 15$	<i>RQ</i> com raio = 0,3

Os dois gráficos que constituem a Figura 6.3 mostram o tempo consumido durante a execução de  $F_3$  para cada configuração apresentada na Tabela 6.3 e cada conjunto de medidas considerado.

Observando-se a Figura 6.3, pode-se notar que em todos os casos analisados o sistema *MIGFIR* superou *SeqScan* em termos de eficiência. A redução de tempo de execução propiciada por *MIGFIR* variou de 17,3% (na situação em que ele gastou 2,915s e *SeqScan* gastou 3,419s) até 213,3% (quando *MIGFIR* consumiu 0,015s e *SeqScan* consumiu 0,047s).

A consulta  $F_4$ , que tem expressão de busca *tipo 3*, é composta por três subexpressões relacionadas entre si por meio do operador booleano *or*, tal que cada subexpressão contém três predicados nebulosos relacionados entre si por *and*. A Tabela 6.4 mostra os parâmetros selecionados para as consultas por similaridade associadas aos predicados de  $F_4$ .

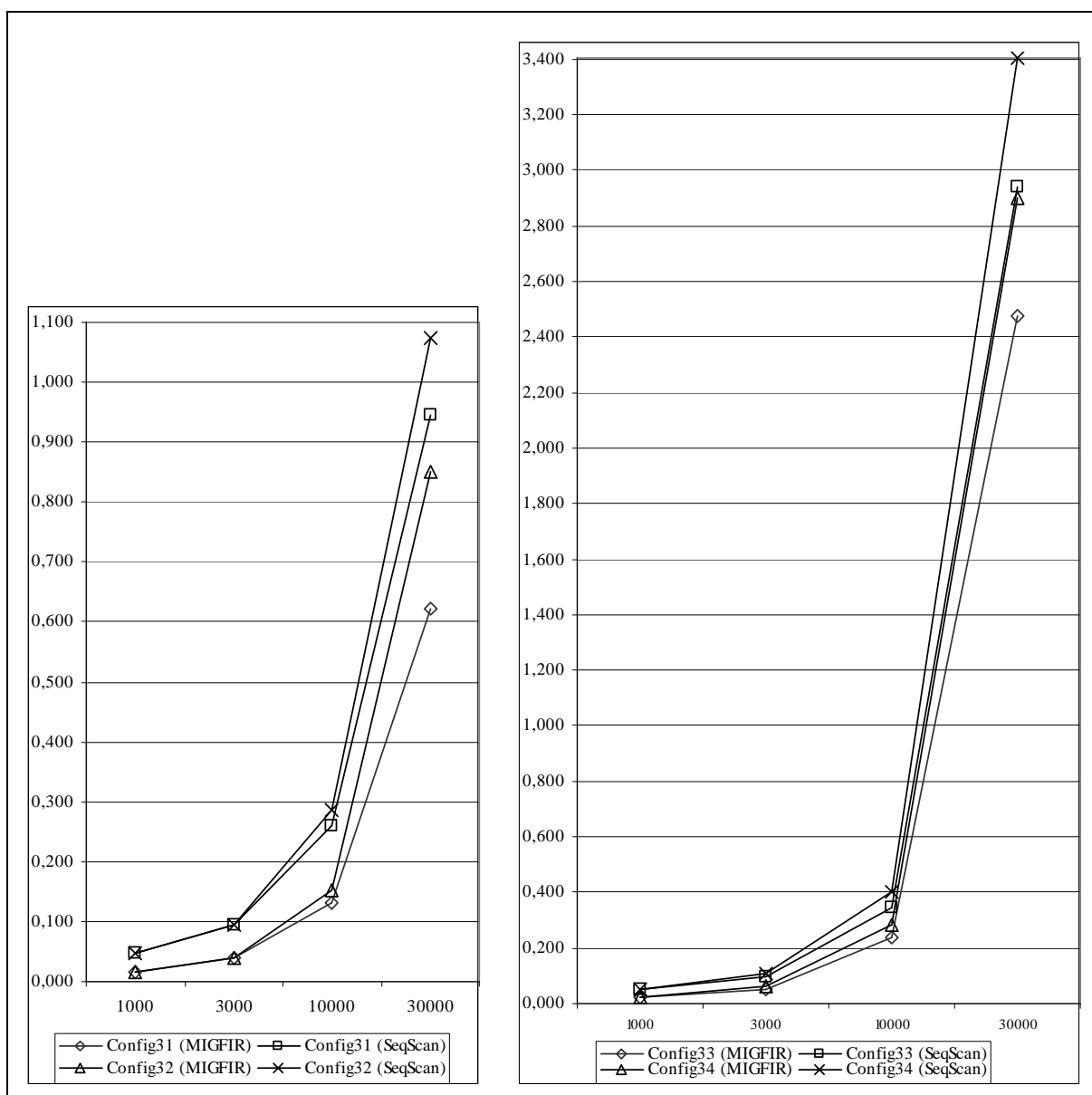


Figura 6.3 – Gráficos apresentando o tempo consumido durante a execução de  $F_3$  para cada configuração de consulta por similaridade e cada conjunto de dados, considerando-se a execução em *MIGFIR* e em *SeqScan*.

Os quatro gráficos que compõem a Figura 6.4 exibem o tempo consumido durante a realização de  $F_4$  para cada configuração mostrada na Tabela 6.4 e cada conjunto de dados sintéticos.

Tabela 6.4 – Quatro configurações de parâmetros para a execução da consulta nebulosa  $F_4$ .

Identificador da configuração das consultas por similaridade	1º, 2º e 3º predicados	4º, 5º e 6º predicados	7º, 8º e 9º predicados
<i>Config41</i>	<i>K-NNQ</i> com $k = 3$	<i>RQ</i> com raio = 0,1	<i>K-NNQ</i> com $k = 3$
<i>Config42</i>	<i>RQ</i> com raio = 0,1	<i>K-NNQ</i> com $k = 3$	<i>RQ</i> com raio = 0,1
<i>Config43</i>	<i>K-NNQ</i> com $k = 15$	<i>RQ</i> com raio = 0,3	<i>K-NNQ</i> com $k = 15$
<i>Config44</i>	<i>RQ</i> com raio = 0,3	<i>K-NNQ</i> com $k = 15$	<i>RQ</i> com raio = 0,3

Observando-se a Figura 6.4, é possível perceber que em todas as situações envolvendo  $F_4$  o sistema *MIGFIR* foi mais eficiente que *SeqScan*. A redução de tempo de execução propiciada por *MIGFIR* variou de 17,1% (quando ele gastou 20,250s e *SeqScan* gastou 23,719s) até 220% (quando *MIGFIR* consumiu 0,025s e *SeqScan* consumiu 0,080s).

Já as três consultas baseadas em imagens-exemplo consideradas nesta seção são:

- Consulta  $E_1$ : encontrar as imagens que satisfazem a expressão de busca (*Face Length = 1,9235 and Chin Length = 0,6176 and Mouth Width = 0,6471 and Nose Length = 0,6765 and Face Width = 2,1059*), onde as cinco medidas faciais em  $E_1$  foram extraídas da imagem mostrada na Figura 6.5(a).
- Consulta  $E_2$ : encontrar as imagens que satisfazem a expressão de busca ( (*Face Length = 1,9235 and Chin Length = 0,6176 and Mouth Width = 0,6471 and Nose Length = 0,6765 and Face Width = 2,1059*) and (*Face Length = 1,8526 and Chin Length = 0,5795 and Mouth Width = 0,5937 and Nose Length = 0,6911 and Face Width = 2,1211*) ), onde as cinco primeiras medidas faciais em  $E_2$  são da imagem mostrada na Figura 6.5(a) e as outras cinco medidas foram extraídas da imagem exibida na Figura 6.5(b).
- Consulta  $E_3$ : encontrar as imagens que satisfazem a expressão de busca ( (*Face Length = 1,9893 or Face Length = 1,7222*) and (*Mouth Width = 0,5003 and*

*Face Width = 2,0647 and Nose Length = 0,6263* ), onde a primeira medida nesta expressão é da imagem mostrada na Figura 6.5(c), a segunda medida é da face na Figura 6.5(d) e os outros três valores são da face na Figura 6.5(e).

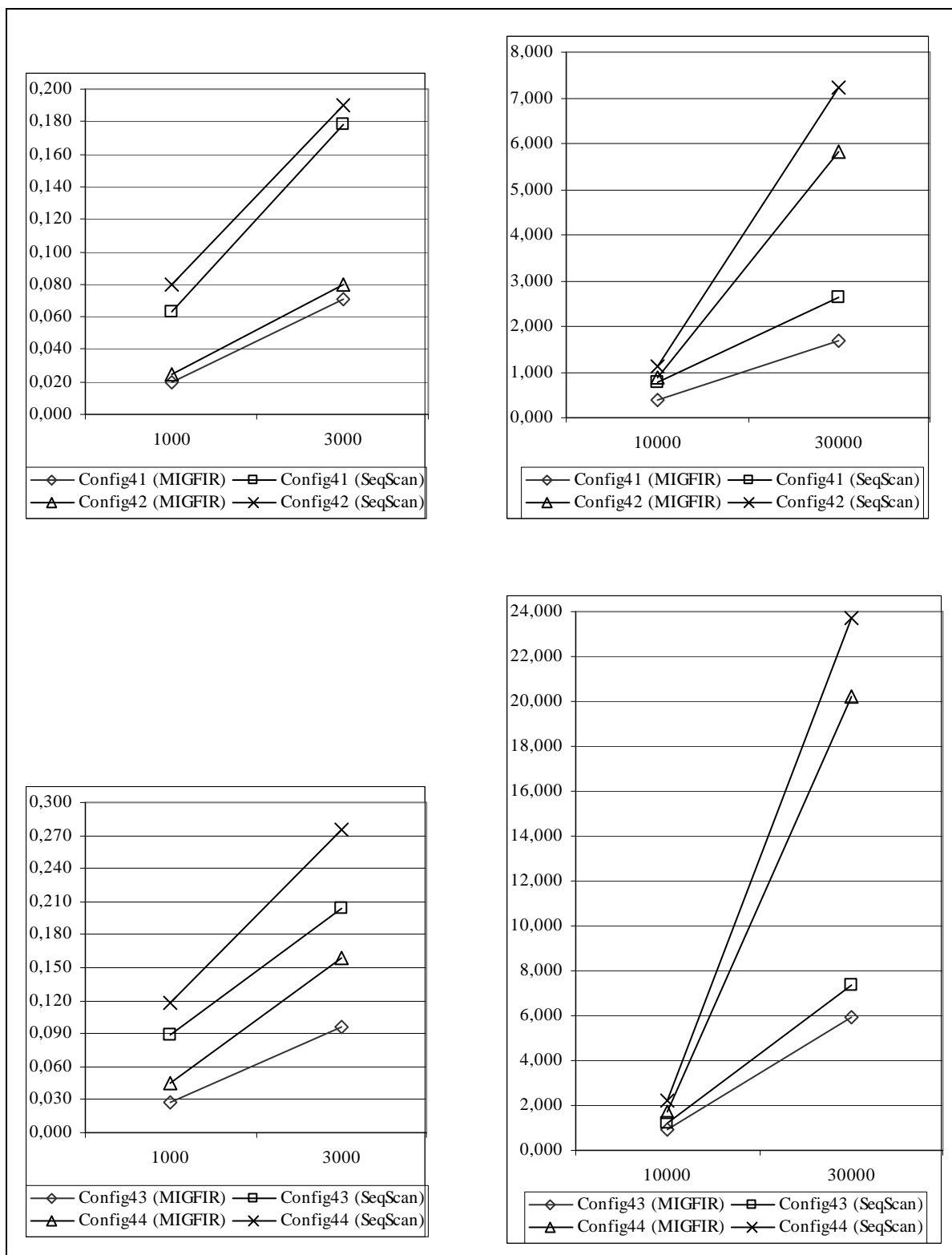


Figura 6.4 – Gráficos apresentando o tempo consumido durante a execução de  $F_4$  para cada configuração de consulta por similaridade e cada conjunto de dados, considerando-se a execução em *MIGFIR* e em *SeqScan*.

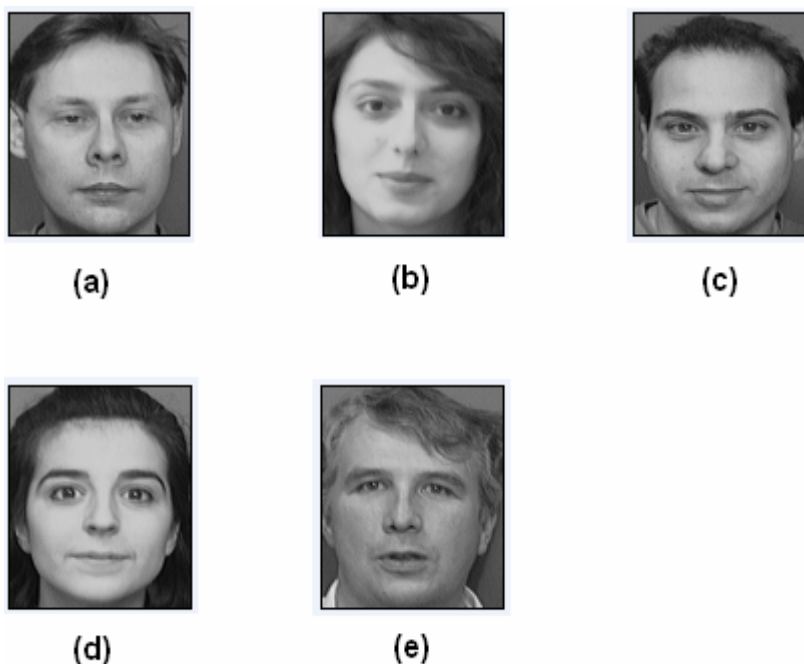


Figura 6.5 – Cinco imagens usadas nas consultas baseadas em exemplos que foram analisadas.

As imagens exibidas na Figura 6.5 são provenientes do conjunto de faces humanas denominado *The ORL Database of Faces*, que foi fornecido por *AT&T Laboratories*.

É importante destacar que as cinco medidas faciais de todas as imagens que foram empregadas durante os testes envolvendo *MIGFIR* foram normalizadas pela distância entre os olhos para evitar erros durante a execução de consultas por causa de variações de escala. Além disso, todas as faces humanas estavam em poses frontais.

A consulta  $E_1$ , cuja expressão de busca é *tipo 2*, é constituída por apenas uma subexpressão que contém as cinco medidas faciais extraídas de uma imagem-exemplo relacionadas entre si pelo operador booleano *and*. A Tabela 6.5 exibe os parâmetros escolhidos para as consultas por similaridade que foram associadas a essa subexpressão.

Tabela 6.5 – Três configurações de parâmetros para a execução da consulta baseada em exemplo  $E_1$ .

Identificador da configuração da consulta por similaridade	Subexpressão (5 medidas faciais)
<i>Config51</i>	<i>K-NNQ</i> com $k = 3$
<i>Config52</i>	<i>K-NNQ</i> com $k = 15$
<i>Config53</i>	<i>RQ</i> com raio = 0,4

É relevante salientar que durante a elaboração de uma consulta baseada em imagens-exemplo é permitido definir raios maiores que 1 para consultas por abrangência.

Empregando os rótulos estabelecidos na primeira coluna da Tabela 6.5, os gráficos exibidos na Figura 6.6 apresentam o tempo consumido durante a execução de  $E_1$  para cada configuração de consulta por similaridade e cada conjunto de dados considerado.

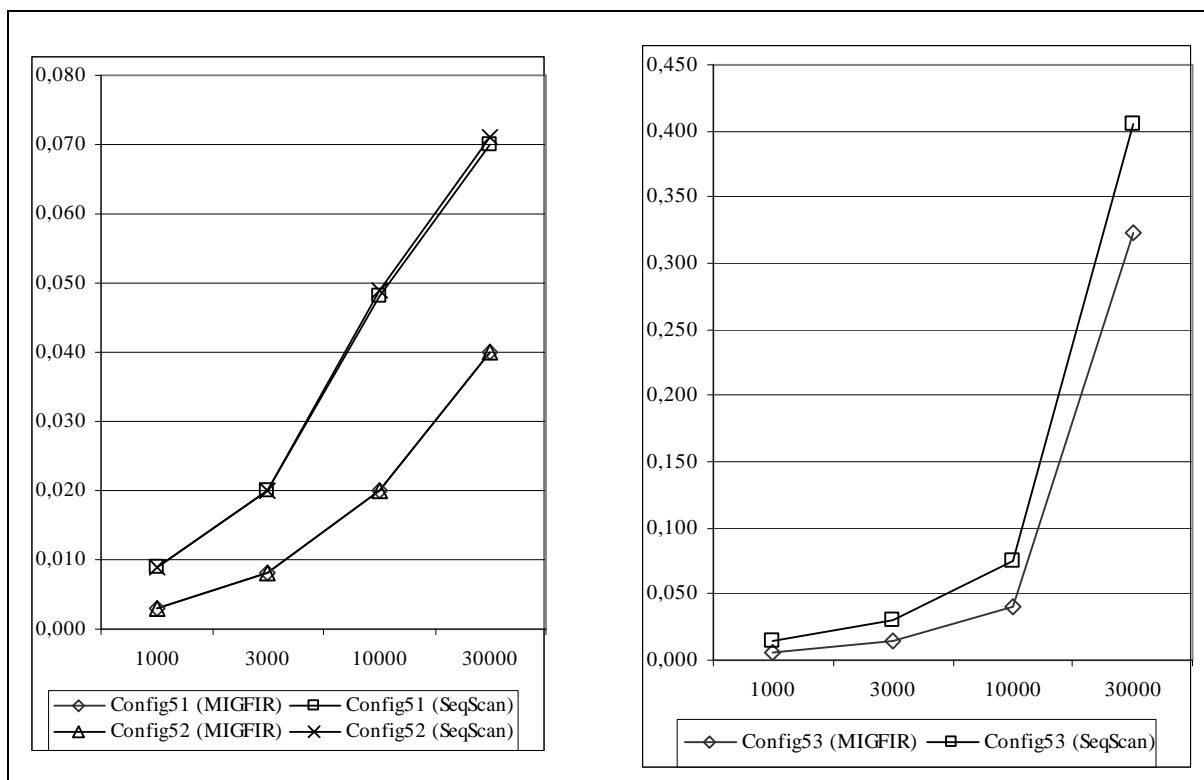


Figura 6.6 – Gráficos apresentando o tempo consumido durante a execução de  $E_1$  para cada configuração de consulta por similaridade e cada conjunto de dados, considerando-se a execução em *MIGFIR* e em *SeqScan*.

A Figura 6.6 permite notar que em todos os casos analisados envolvendo  $E_1$  o sistema *MIGFIR* superou *SeqScan* em termos de eficiência. A redução de tempo de execução variou de 25,8% (na situação em que o tempo gasto por *MIGFIR* foi 0,323s e o tempo de *SeqScan* foi 0,406s) até 200% (quando *MIGFIR* consumiu 0,003s e *SeqScan* consumiu 0,009s).

A consulta  $E_2$ , que tem expressão *tipo 2*, é composta por duas subexpressões relacionadas entre si por meio do operador *and*, tal que cada subexpressão contém as cinco medidas faciais de uma imagem-exemplo relacionadas entre si também por *and*.

A Tabela 6.6 mostra os parâmetros selecionados para as consultas por similaridade que foram associadas a cada subexpressão de  $E_2$ .

Tabela 6.6 – Três configurações de parâmetros para a execução da consulta baseada em exemplos  $E_2$ .

Identificador da configuração das consultas por similaridade	1ª subexpressão (5 medidas faciais)	2ª subexpressão (5 medidas faciais)
<i>Config61</i>	<i>K-NNQ</i> com $k = 5$	<i>K-NNQ</i> com $k = 5$
<i>Config62</i>	<i>K-NNQ</i> com $k = 15$	<i>K-NNQ</i> com $k = 15$
<i>Config63</i>	<i>RQ</i> com raio = 0,4	<i>RQ</i> com raio = 0,4

Os gráficos que constituem a Figura 6.7 apresentam o tempo gasto durante a execução de  $E_2$  para cada configuração dada na Tabela 6.6 e cada conjunto de medidas sintéticas.

Observando-se a Figura 6.7, pode-se perceber que em todas as situações em que  $E_2$  foi executada o sistema *MIGFIR* foi mais eficiente que *SeqScan*. A economia de tempo de execução proporcionada por *MIGFIR* variou de 18,9% (no caso em que ele gastou 6,991s e *SeqScan* gastou 8,312s) até 275% (quando *MIGFIR* consumiu 0,012s e *SeqScan* consumiu 0,045s).

A consulta  $E_3$ , cuja expressão é *tipo 1*, é constituída por duas subexpressões relacionadas entre si através do operador *and*, tal que a primeira subexpressão contém dois valores de características faciais relacionados entre si pelo operador *or* e a segunda contém três medidas relacionadas por *and*. A Tabela 6.7 exhibe os parâmetros estabelecidos para as consultas por similaridade associadas às medidas faciais de  $E_3$ .



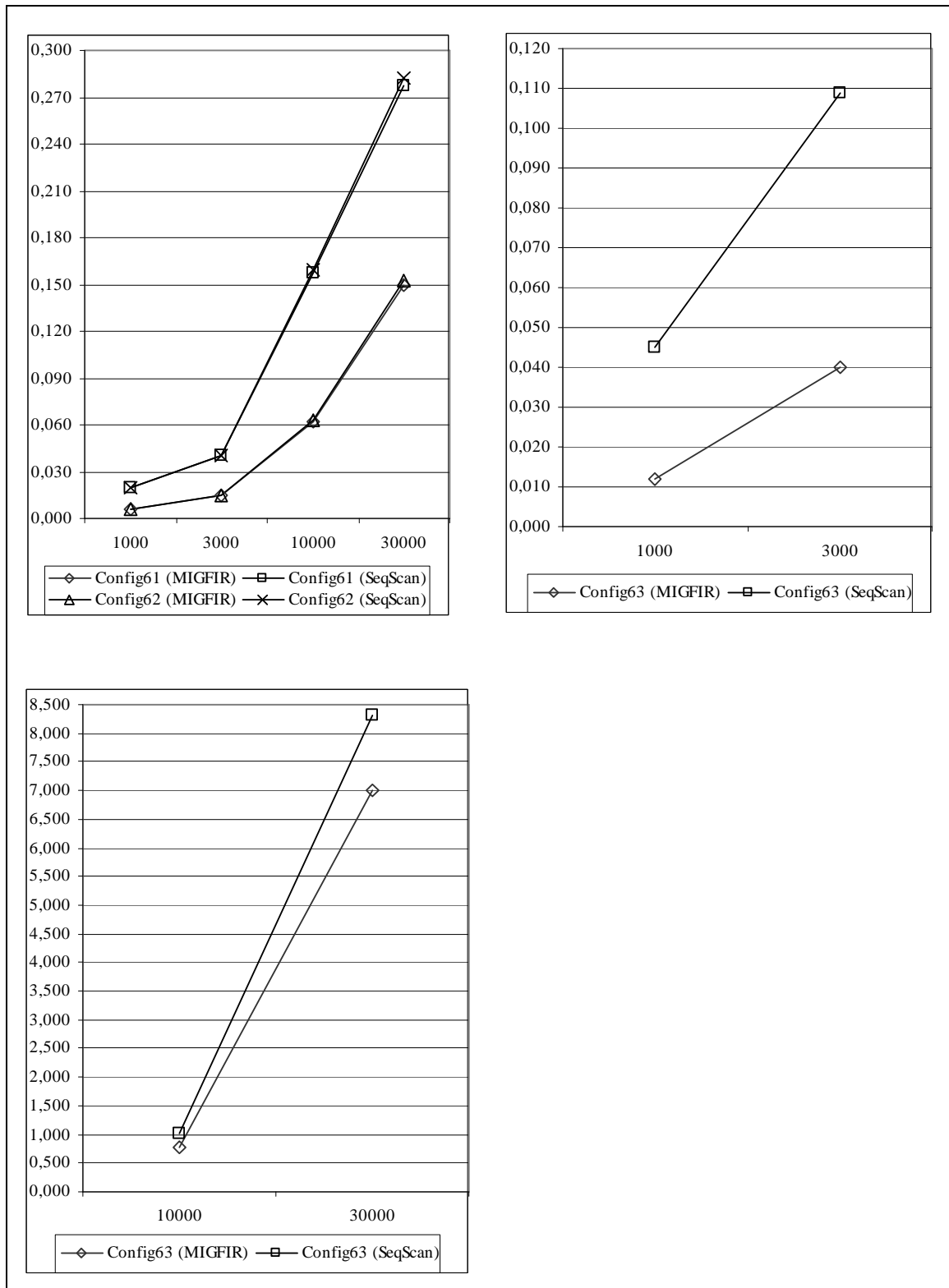


Figura 6.7 – Gráficos apresentando o tempo consumido durante a execução de  $E_2$  para cada configuração de consulta por similaridade e cada conjunto de dados, considerando-se a execução em *MIGFIR* e em *SeqScan*.

Tabela 6.7 – Duas configurações de parâmetros para a execução da consulta baseada em exemplos  $E_3$ .

Identificador da configuração das consultas por similaridade	1ª medida facial	2ª medida facial	3ª medida facial	4ª medida facial	5ª medida facial
<i>Config71</i>	<i>K-NNQ</i> com k = 15	<i>K-NNQ</i> com k = 3	<i>K-NNQ</i> com k = 15	<i>K-NNQ</i> com k = 3	<i>K-NNQ</i> com k = 15
<i>Config72</i>	<i>K-NNQ</i> com k = 15	<i>RQ</i> com raio = 0,4	<i>K-NNQ</i> com k = 15	<i>RQ</i> com raio = 0,4	<i>K-NNQ</i> com k = 15

Os gráficos apresentados na Figura 6.8 mostram o tempo consumido para a realização de  $E_3$  para cada configuração dada na Tabela 6.7 e cada conjunto de dados considerado.

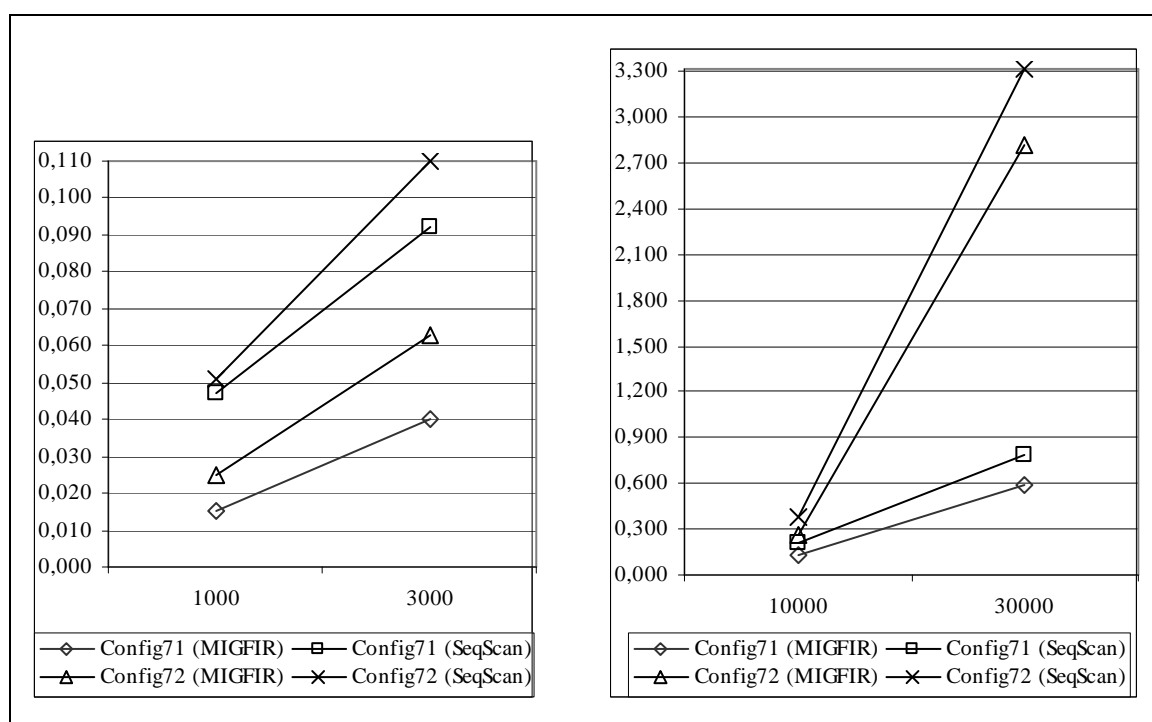


Figura 6.8 – Gráficos apresentando o tempo consumido durante a execução de  $E_3$  para cada configuração de consulta por similaridade e cada conjunto de dados, considerando-se a execução em *MIGFIR* e em *SeqScan*.

Observando-se a Figura 6.8, é possível notar que em todos os casos em que  $E_3$  foi executada o sistema *MIGFIR* superou *SeqScan* em termos de eficiência. A redução de tempo de execução propiciada por *MIGFIR* variou de 17,9% (quando ele gastou 2,811s e *SeqScan* gastou 3,315s) até 213,3% (na situação em que ele consumiu 0,015s e *SeqScan* consumiu 0,047s).

Diante dos resultados apresentados nesta seção, pôde-se concluir que para várias configurações distintas de consultas por similaridade e conjuntos de imagens com até 30000 elementos (este foi o tamanho máximo considerado durante os experimentos) a utilização da árvore métrica *Slim-tree* em *MIGFIR* possibilita uma redução bastante acentuada dos tempos de execução de consultas baseadas em predicados nebulosos e em exemplos, em comparação com um sistema similar que emprega varredura seqüencial.

#### **6.4 CONSIDERAÇÕES FINAIS**

Este capítulo apresentou a avaliação experimental do sistema *MIGFIR*, que teve o propósito de verificar a precisão e a eficiência desse sistema na execução de consultas com vários níveis de complexidade sobre diversos conjuntos de imagens.

Na seção 6.2, inicialmente foi explicada a metodologia que seria usada para a avaliação da precisão de *MIGFIR* na execução de consultas baseadas em imagens-exemplo e em seguida foram discutidos os resultados gerados por essa avaliação, que permitiram concluir que o sistema desenvolvido apresenta um grau de precisão satisfatório na recuperação ordenada de faces humanas similares a uma face-exemplo quando todas as imagens são representadas por suas medidas geométricas.

Já na seção 6.3 foram apresentados os resultados dos experimentos referentes à análise da eficiência de *MIGFIR*, que mostraram que esse sistema reduziu consideravelmente os valores de tempo de execução de consultas bastante representativas em comparação com um sistema similar que empregou a varredura seqüencial como método de acesso aos dados.

## 7 CONCLUSÕES

### 7.1 CONSIDERAÇÕES FINAIS E RESULTADOS

Durante o desenvolvimento do projeto de mestrado não foi encontrado um sistema para a recuperação de imagens por conteúdo que levasse em consideração três aspectos que alguns trabalhos de destaque na literatura classificam como muito importantes para a linha de pesquisa *CBIR*:

- A relevância da indexação e da recuperação adequadas de faces humanas para determinadas aplicações.
- A importância do uso de conceitos da teoria dos conjuntos nebulosos na linguagem empregada para formular consultas, com o intuito de aproximar mais essas formulações da linguagem geralmente utilizada pelos seres humanos.
- A relevância de medidas geométricas que podem ser extraídas de imagens, como por exemplo das de faces humanas, para a avaliação de similaridade realizada por humanos.

Assim, o sistema *MIGFIR*, que foi apresentado detalhadamente nesta dissertação, foi idealizado e desenvolvido sob forte influência desses três aspectos. Chegou-se a um sistema capaz de indexar e recuperar com eficiência e precisão satisfatórias certos tipos de imagens, como de faces humanas, usando suas medidas geométricas, as funções de similaridade *MGF* e *MGF2*, e o método de acesso métrico *Slim-tree*. Além disso, *MIGFIR* permite que os usuários configurem parâmetros importantes durante a definição de consultas por similaridade complexas e também possibilita a aplicação de conjuntos nebulosos e conceitos relacionados para tornar as expressões de busca das consultas nebulosas mais próximas da linguagem geralmente empregada por humanos.

A precisão e a eficiência de *MIGFIR* na execução de consultas com diferentes graus de complexidade sobre conjuntos de dados reais e sintéticos de vários tamanhos (entre 1000 e 30000 representações de faces humanas) foram verificadas durante sua avaliação experimental. Em termos de precisão os resultados alcançados possibilitaram concluir que, considerando-se a abordagem de avaliação simplificada que foi apresentada no capítulo 6, seção 6.2, o sistema *MIGFIR* mostra precisão satisfatória na recuperação de faces humanas similares a imagens-exemplo, já que a análise de similaridade entre esboços de faces efetuada por *MIGFIR* se aproxima bastante do julgamento humano de similaridade. Já em termos de eficiência, os resultados obtidos, correspondentes a um grande número de consultas executadas, mostram que a utilização de *MIGFIR* pode promover uma redução do tempo de execução que varia entre 17,1% (para conjuntos de dados maiores, e para expressões de busca mais complexas e com predomínio de consultas do tipo *RQ*) e 358,3% (para conjuntos de dados menores, e para expressões mais simples e com predomínio de consultas do tipo *K-NNQ*).

## 7.2 PRINCIPAIS CONTRIBUIÇÕES

Além das contribuições que já foram citadas na seção 7.1, outras contribuições muito importantes do projeto de mestrado desenvolvido são:

- Desenvolvimento e implementação das duas funções de similaridade métricas denominadas *MGF* e *MGF2*, que permitem determinar os graus de similaridade entre imagens representadas por conjuntos de medidas geométricas ou por conjuntos de valores-verdade de predicados nebulosos de uma forma intuitiva.
- Implementação do suporte para consultas baseadas em imagens-exemplo, que permite que qualquer expressão de busca seja definida por uma subexpressão de busca ou como uma combinação de duas ou mais subexpressões, tal que cada subexpressão pode ser constituída por um conjunto qualquer de medidas geométricas provenientes de uma ou mais imagens ou pode ser constituída por todas as medidas geométricas de uma dada imagem.
- Implementação do suporte para consultas baseadas em predicados nebulosos, que possibilita que qualquer expressão de busca seja formada por uma subexpressão de busca ou como uma combinação de duas ou mais subexpressões, tal que cada

subexpressão pode ser composta por um conjunto aleatório de predicados relacionados a uma ou mais medidas geométricas de imagens ou pode ser composta por um predicado associado a cada medida geométrica considerada.

- Interface que oferece ao usuário alta flexibilidade durante a definição de uma consulta baseada em exemplos ou baseada em predicados, já que com tal interface o usuário tem a possibilidade de empregar os operadores lógicos *and* e *or* para combinar predicados, medidas geométricas ou subexpressões de busca, e pode ainda configurar parâmetros muito importantes para cada componente da expressão que representa a consulta.
- Implementação que possibilita que *MIGFIR* seja ajustado, em um intervalo de tempo relativamente curto, para atuar com outros tipos de imagens além das de faces humanas ou até mesmo para utilizar outros métodos de acesso métricos.

### 7.3 PROPOSTAS PARA TRABALHOS FUTUROS

O desenvolvimento do sistema *MIGFIR* e os levantamentos bibliográficos realizados permitem apontar algumas propostas para trabalhos futuros:

- Estudar a possibilidade de uso no sistema *MIGFIR* de outros tipos de características de imagens de faces humanas (por exemplo: cor e forma) em conjunção com as medidas geométricas faciais, com o intuito de tentar aumentar o grau de precisão do sistema na recuperação de imagens por similaridade, em comparação com o julgamento humano de similaridade.
- Avaliar a possível utilização das estruturas de indexação métricas *HCS-tree* e *DBM-tree* no sistema *MIGFIR*, com o propósito de procurar elevar ainda mais a eficiência em relação aos valores de tempo de execução das consultas.
- Estudar a possibilidade de se aplicarem conceitos de otimização de processamento de consultas para tentar reduzir o tempo consumido pelo sistema desenvolvido durante a realização de consultas complexas, mais especificamente durante a execução de operações *and* e *or* entre componentes de expressões de busca.

- Integrar a ferramenta para extração de medidas geométricas de imagens de faces humanas desenvolvida por pesquisadores do Departamento de Computação da Universidade Federal de São Carlos ao sistema *MIGFIR*.
- Utilizar as medidas de desempenho que na literatura são denominadas *recall* e *precision* [SMEULDERS et al., 2000] (*precision* é completamente diferente da medida que foi chamada neste trabalho de *precisão*) na avaliação da qualidade da recuperação de imagens realizada por *MIGFIR*, visando ampliar as conclusões proporcionadas pela análise do grau de precisão.
- Investigar a possibilidade de melhorar ainda mais a qualidade dos resultados das consultas através de uma ampliação de *MIGFIR* que permita ao usuário realizar refinamentos sucessivos dos resultados parciais de qualquer consulta, para que o resultado final contenha somente as imagens que o usuário considerar relevantes (conceito de *relevance feedback* [WOOD; CAMPBELL; THOMAS, 1998]).
- Com o auxílio de especialistas ou de trabalhos dedicados ao estudo do julgamento de similaridade do ser humano, pesquisar funções de pertinência nebulosas adequadas para possibilitar a realização de uma avaliação consistente do grau de precisão do sistema *MIGFIR* na execução de consultas nebulosas.

## 8 REFERÊNCIAS BIBLIOGRÁFICAS

ANTANI, S.; KASTURI, R.; JAIN, R. **A survey on the use of pattern recognition methods for abstraction, indexing and retrieval of images and video.** Pattern Recognition, n. 35, p. 945-965, 2002.

ASLANDOGAN, Y. A.; YU, C. T. **Techniques and Systems for Image and Video Retrieval.** IEEE Transactions on Knowledge and Data Engineering, v. 11, n. 1, 1999.

BAEZA-YATES, R. et al. **Proximity matching using fixed-queries trees.** In: Anual Symposium on Combinatorial Pattern Recognition (CPM), 5., 1994, Asilomar, USA. **Anais...** Asilomar, USA: 1994. p. 198-212.

BECKMANN, N. et al. **The R\*-tree: an efficient and robust access method for points and rectangles.** In: ACM SIGMOD International Conference on Management of Data, 1990, Atlantic City, USA. **Anais...** Atlantic City, USA: 1990. p. 322-331.

BERRETTI, S.; DEL BIMBO, A.; PALA, P. **Indexed retrieval by shape appearance.** IEE Proceedings on Vision Image and Signal Processing, v. 147, n. 4, p. 356-362, august 2000.

BOJADZIEV, G.; BOJADZIEV, M. **Fuzzy sets, fuzzy logic, applications.** Singapore: World Scientific, 1995. 283p.

BORGES JR., S. R. **Consultas Nebulosas baseadas em Informações Semânticas em um Banco de Dados Multimídia.** 2001. 89p. Dissertação (Mestrado em Ciência da Computação) – Departamento de Computação, Universidade Federal de São Carlos, São Carlos.

BOZKAYA, T.; OZSOYOGLU, M. **Distance-based indexing for high-dimensional metric spaces.** In: ACM SIGMOD International Conference on Management of Data, 1997, Tucson, USA. **Anais...** Tucson, USA: 1997. p. 357-368.

BUENO, J. M. et al. **How to Add Content-based Image Retrieval Capability in a PACS.** In: 15th IEEE Symposium on Computer-Based Medical Systems, 2002, Maribor, Slovenia. **Anais...** Maribor, Slovenia: 2002. p. 321-326.



BURKHARD, W. A.; KELLER, R. M. **Some approaches to best-match file searching.** Communications of the ACM, v. 16, n. 4, p. 230-236, april 1973.

CAREY, M. J. et al. **Towards Heterogeneous Multimedia Information Systems: the Garlic Approach.** In: RIDE-DOM '95, 5th International Workshop on Research Issues in Data Engineering: Distributed Object Management, 1995, **Anais...** 1995. p. 124-131.

CARSON, C. et al. **Blobworld: a system for region-based image indexing and retrieval.** In: The Third International Conference VISUAL '99, 1999, Amsterdam, The Netherlands. **Anais...** Amsterdam, The Netherlands: Springer 1999.

CHANG, N. S.; FU, K. S. **Query-by-pictorial-example.** IEEE Transactions on Software Engineering, v. 6, n. 6, p. 519-524, 1980.

CHUA, T. S.; ZHAO, Y. L.; KANKANHALLI, M. S. **Detection of Human Faces in Compressed Domain for Video Stratification.** The Visual Computer, v. 18, n. 2, p. 121-133, april 2002.

CIACCIA, P.; PATELLA, M. **The M<sup>2</sup>-tree: processing complex multi-feature queries with just one index.** In: DELOS Workshop: Information Seeking, Searching and Querying in Digital Libraries, 1., 2000, Zurich, Switzerland. **Anais...** Zurich, Switzerland: 2000.

CIACCIA, P.; PATELLA, M.; ZEZULA, P. **M-Tree: an efficient access method for similarity search in metric spaces.** In: International Conference on Very Large Data Bases (VLDB), 23., 1997, Athens, Greece. **Anais...** Athens, Greece: 1997. p. 426-435.

DOMINGUES, H. H. **Espaços métricos e introdução à topologia.** São Paulo: Atual Editora, 1982. 184p.

EDELMAN, S.; INTRATOR, N. **Learning as extraction of low-dimensional representations.** Academic Press, 1997. p. 353-380.

ELMASRI, R.; NAVATHE, S. B. **Sistemas de Banco de Dados.** 3. ed. Rio de Janeiro: LTC, 2001. 837p.

FEND, D.; SIU, W.; ZHANG, H. **Multimedia Information Retrieval and Management - Technological Fundamentals and Applications.** Berlin: Springer Verlag, 2003. 476p.

FLICKNER, M. et al. **Query by Image and Video Content: The QBIC System.** IEEE Computer, 1995.

FOURNIER, J.; CORD, M.; PHILLIP-FOLIGUET, S. **RETIN: A content-based image indexing and retrieval system.** Pattern Analysis and Applications Journal, v. 4, n. 2/3, p. 153-173, 2001.

GABOR, D. **Theory of communication**. The Journal of the Institute of Electrical Engineers, v. 93, n. 21, p. 429-457, january 1946.

GAEDE, V.; GÜNTHER, O. **Multidimensional access methods**. ACM Computing Surveys, v. 30, n. 2, p. 170-231, 1998.

GAVIOLI, A.; BIAJIZ, M.; MOREIRA, J. **MIFLIR: A Metric Indexing and Fuzzy Logic-based Image Retrieval System**. In: The 21th IEEE International Conference on Data Engineering (ICDE 2005), 2005. **Anais...** Tokio, Japan: 2005. p. 1178.

GAVIOLI, A. et al. **Component Retrieval using Metric Indexing**. In: The 2004 IEEE International Conference on Information Reuse and Integration (IEEE IRI-2004), 2004, Las Vegas, EUA. **Anais...** Las Vegas, EUA: IEEE Systems, Man, and Cybernetics Society (SMC), 2004. p. 79-84.

GUTTMAN, A. **R-trees: a dynamic index structure for spatial searching**. In: ACM SIGMOD International Conference on Management of Data, 1984, Boston, USA. **Anais...** Boston, USA: 1984. p. 45-57.

KLIR, G. J.; YUAN, B. **Fuzzy Sets and Fuzzy Logic: Theory and Applications**. Upper Saddle River: Prentice Hall, 1995. 574p.

KOHONEN, T. **The self-organizing map**. Proceedings of the IEEE, v. 78, n. 9, 1990.

KRUSKAL, J. B. **On the shortest spanning subtree of a graph and the traveling salesman problem**. In: American Mathematical Society, 1956, **Anais...** 1956. p. 48-50.

LIMA, E. L. **Espaços métricos**. São Paulo: Edgard Blücher, 1977. 299p.

LOSADA, D. E. et al. **Experiments on using fuzzy quantified sentences in adhoc retrieval**. In: The 2004 ACM Symposium on Applied Computing, 2004, Nicosia, Cyprus. **Anais...** Nicosia, Cyprus: 2004. p. 1059-1064.

NADLER, M.; SMITH, E. P. **Pattern recognition engineering**. New York, USA: Wiley, 1993.

NARASIMHALU, A. D.; KANKANHALLI, M. S.; WU, J. **Benchmarking multimedia databases**. Multimedia Tools and Applications, v. 4, n. 3, p. 333-355, 1997.

NIBLACK, W. et al. **The QBIC Project: Querying Images by Content using Color, Texture and Shape**. In: SPIE Conference on Storage and Retrieval for Image and Video Databases, 1993, San Jose, USA. **Anais...** San Jose, USA: 1993. p. 173-187.

ORTEGA, M. et al. **Supporting Similarity Queries in MARS**. In: The 5th ACM International Multimedia Conference, 1997, Seattle, USA. **Anais...** Seattle, USA: 1997. p. 403-413.

RUI, Y.; HUANG, T. S.; CHANG, S.-F. **Image Retrieval: Past, Present, and Future**. Journal of Visual Communication and Image Representation, v. 10, p. 1-23, 1999.

SAMET, H. **The quadtree and related hierarquical data structure**. ACM Computing Surveys, v. 16, n. 2, p. 187-260, 1984.

SANTINI, S.; JAIN, R. **Similarity Measures**. IEEE Transactions on Pattern Analysis and Machine Intelligence, v. 21, n. 9, p. 871-883, september 1999.

SANTOS, R. F. **Métodos de acesso métricos para suporte a consultas por similaridade: apresentação da técnica Omni**. 2003. 145p. Tese (Doutorado em Ciência da Computação) – ICMC, Universidade de São Paulo, São Carlos.

SANTOS, R. F. et al. **Similarity search without tears: the OMNI family of all-purpose access methods**. In: IEEE International Conference on Data Engineering, 17., 2001, Heidelberg, Germany. **Anais...** Heidelberg, Germany: 2001. p. 623-630.

SHEPARD, R. N. **Toward a universal law of generalization for physical science**. Science, n. 237, p. 1317-1323, 1987.

SMEULDERS, A. W. et al. **Content-based image retrieval at the end of the early years**. IEEE Transactions on Pattern Analysis and Machine Intelligence, v. 22, n. 12, p. 1349-1380, 2000.

SMITH, J. R.; CHANG, S. F. **Querying by color regions using the VisualSEEK content-based visual query system**. Intelligent Multimedia Information Retrieval, 1997.

TAMURA, H.; MORI, S.; YAMAWAKI, T. **Texture Features corresponding to Visual Perception**. IEEE Transactions on Systems, Man and Cybernetics, v. 8, n. 6, p. 460-473, 1978.

TERANO, T.; ASAI, K.; SUGENO, M. **Fuzzy Systems Theory and its Applications**. San Diego: Academic Press Inc, 1991. 268p.

TRAINA JR., C. et al. **Fast indexing and visualization of metric data sets using Slim-trees**. IEEE Transactions on Knowledge and Data Engineering, v. 30, n. 20, p. 244-260, 2001.

TRAINA JR., C. et al. **How to improve the pruning ability of dynamic metric access methods**. In: International Conference on Information and Knowledge Management, 2002, MacLean, USA. **Anais...** MacLean, USA: 2002. p. 219-226.

TRAINA JR., C. et al. **Slim-trees: high performance metric trees minimizing overlap between nodes**. In: International Conference on Extending Database Technology (EDBT), 8., 2000, Konstanz, Germany. **Anais...** Konstanz, Germany: 2000. p. 41-65.

TVERSKY, A. **Features of similarity**. Psychological Review, v. 84, n. 4, p. 327-352, July 1977.

UHLMANN, J. K. **Satisfying general proximity/similarity queries with metric trees**. Information Processing Letters (IPL), v. 40, n. 4, p. 175-179, November 1991.

VELTKAMP, R. C. **Shape matching: similarity measures and algorithms**. In: International Conference on Shape Modeling and Applications, 2001, Genova, Italy. **Anais...** Genova, Italy: 2001. p. 188-197.

VIEIRA, M. R. et al. **DBM-Tree: A Dynamic Metric Access Method Sensitive to Local Density Data**. In: 19º Simpósio Brasileiro de Bancos de Dados, 2004, Brasília. **Anais...** Brasília: 2004. p. 163-177.

WOOD, M.; CAMPBELL, N.; THOMAS, B. **Iterative Refinement by Relevance Feedback in Content-based Digital Image Retrieval**. In: ACM Multimedia, 1998, Bristol, United Kingdom. **Anais...** Bristol, United Kingdom: 1998. p. 13-20.

WORRING, M. et al. **Interactive Indexing and Retrieval of Multimedia Content**. In: 29th Annual Conference on Current Trends in Theory and Practice of Informatics, 2002, Milovy, Czech Republic. **Anais...** Milovy, Czech Republic: 2002.

YAMAMOTO, C. H. **Definição e construção de uma estrutura de indexação métrica com uso de conceitos da lógica nebulosa**. 2003. 89p. Dissertação (Mestrado em Ciência da Computação) – Departamento de Computação, Universidade Federal de São Carlos, São Carlos.

YAMAMOTO, C. H.; BIAJIZ, M.; TRAINA JR., C. **Aumento da Eficiência das Estruturas de Indexação Métricas com Uso de Conceitos da Lógica Nebulosa**. In: 18º Simpósio Brasileiro de Bancos de Dados, 2003, Manaus. **Anais...** Manaus: 2003. p. 185-199.

YIANILOS, P. N. **Data structures and algorithms for nearest neighbor search in general metric spaces**. In: ACM SIGACT-SIAM Symposium on Discrete Algorithms, 4., 1993, Austin, USA. **Anais...** Austin, USA: 1993. p. 311-321.

ZADEH, L. A. **Fuzzy sets**. Information and Control, v. 8, n. 3, p. 338-353, 1965.