Processamento de consultas analíticas com
predicados de similaridade entre imagens em
ambientes de data warehousing

Jefferson William Teixeira

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: _____

Jefferson William Teixeira

Processamento de consultas analíticas com predicados de similaridade entre imagens em ambientes de data warehousing

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

Área de Concentração: Ciências de Computação e Matemática Computacional

Orientadora: Profa. Dra. Cristina Dutra de Aguiar Ciferri

USP – São Carlos
Julho de 2015

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados fornecidos pelo(a) autor(a)

T266p	Teixeira, Jefferson William Processamento de consultas analíticas com predicados de similaridade entre imagens em ambientes de data warehousing / Jefferson William Teixeira; orientadora Cristina Dutra de Aguiar Ciferri. -- São Carlos, 2015. 108 p. Dissertação (Mestrado - Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional) -- Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2015. 1. Data warehouse de imagens. 2. Índice bitmap. 3. Consultas por similaridade. 4. Consulta OLAP. 5. Método de acesso métrico. I. Dutra de Aguiar Ciferri, Cristina , orient. II. Título.

Jefferson William Teixeira

**Processing of analytical queries with similarity search
predicates over images in data warehousing environments**

Master dissertation submitted to the Instituto de
Ciências Matemáticas e de Computação - ICMC-
USP, in partial fulfillment of the requirements for the
degree of the Master Program in Computer Science
and Computational Mathematics. *FINAL VERSION*

Concentration Area: Computer Science and
Computational Mathematics

Advisor: Profa. Dra. Cristina Dutra de Aguiar Ciferri

**USP – São Carlos
July 2015**

“Porque o Senhor dá a sabedoria; da sua boca é que vem o conhecimento e o entendimento”

Provérbios 2.6

Agradecimentos

Àquele que é mestre por excelência, Senhor Jesus Cristo.

À minha orientadora Cristina D. A. Ciferri pelo ótimo trabalho, por sua paciência e compreensão nos momentos difíceis.

Ao professor Ricardo R. Ciferri, por suas importantes contribuições para esta dissertação de mestrado.

Aos meus pais, pelo apoio incondicional.

Aos meus colegas e amigos do GBDI, os quais tiveram um papel essencial no desenvolvimento desta dissertação de mestrado, durante nossa convivência no laboratório e em reuniões de grupo.

Ao ICMC-USP, pelo apoio institucional e por ter sido minha segunda casa desde os tempos de graduação.

À FAPESP e CNPQ, pelo apoio financeiro.

Resumo

Um ambiente de *data warehousing* oferece suporte ao processo de tomada de decisão. Ele consolida dados de fontes de informação distribuídas, autônomas e heterogêneas em um único componente, o *data warehouse*, e realiza o processamento eficiente de consultas analíticas, denominadas OLAP (*on-line analytical processing*). Um *data warehouse* convencional armazena apenas dados alfanuméricos. Por outro lado, um *data warehouse* de imagens armazena, além desses dados convencionais, características intrínsecas de imagens, permitindo a realização de consultas analíticas estendidas com predicados de similaridade entre imagens. Esses ambientes demandam, portanto, a criação de estratégias que possibilitem o processamento eficiente dessas consultas complexas e custosas. Apesar de haver na literatura trabalhos voltados a índices bitmap para ambientes de *data warehousing* e métodos de acesso métricos para melhorar o desempenho de consultas por similaridade entre imagens, no melhor do nosso conhecimento, não há uma técnica que investigue essas duas questões em um mesmo contexto. Esta dissertação visa preencher essa lacuna na literatura por meio das seguintes contribuições: (i) proposta do ImageDW-index, um mecanismo para a otimização de consultas analíticas estendidas com predicados de similaridade entre imagens; e (ii) definição de diferentes estratégias de processamento de consultas sobre *data warehouses* de imagens usando o ImageDW-index. Para validar as soluções propostas, foram desenvolvidas duas outras contribuições secundárias, que são: (iii) o ImageDW-Gen, um gerador de dados com o objetivo de povoar o *data warehouse* de imagens; e (iv) a proposta de quatro classes de consulta, as quais enfocam em diferentes custos de processamento dos predicados de similaridade entre imagens. Utilizando o ImageDW-Gen, foram realizados testes de desempenho para investigar as vantagens introduzidas pelas estratégias propostas, de acordo com as classes de consultas definidas. Comparado com o trabalho mais correlato existente na literatura, o uso do ImageDW-index proveu uma melhora no desempenho do processamento de consultas IOLAP que variou em média de 55,57% até 82,16%, considerando uma das estratégias propostas.

Palavras-chave: data warehouse de imagens, consulta por similaridade, consulta OLAP, índice bitmap, método de acesso métrico

Abstract

A data warehousing environment offers support to the decision-making process. It consolidates data from distributed, autonomous and heterogeneous information sources into one of its main components, the data warehouse. Furthermore, it provides efficient processing of analytical queries (i.e. OLAP queries). A conventional data warehouse stores only alphanumeric data. On the other hand, an image data warehouse stores not only alphanumeric data but also intrinsic features of images, thus allowing data warehousing environments to perform analytical similarity queries over images. This requires the development of strategies to provide efficient processing of these complex and costly queries. Although there are a number of approaches in the literature aimed at the development of bitmap index for data warehouses and metric access methods for the efficient processing of similarity queries over images, to the best of our knowledge, there is not an approach that investigate these two issues in the same setting. In this research, we fill this gap in the literature by introducing the following main contributions: (i) the proposal of the ImageDW-index, an optimization mechanism aimed at the efficient processing of analytical queries extended with similarity predicates over images; and (ii) definition of different processing strategies for image data warehouses using the ImageDW-index. In order to validate these main proposals, we also introduce two secondary contributions, as follows: (iii) the ImageDW-Gen, a data generator to populate image data warehouses; and (iv) the proposal of four query classes, each one enforcing different query processing costs associated to the similarity predicates in image data warehousing environments. Using the ImageDW-Gen, performance tests were carried out in order to investigate the advantages introduced by the proposed strategies, according to the query classes. Compared to the most related work available in the literature, the ImageDW-index provided a performance gain that varied from 55.57% to 82.16%, considering one of the proposed strategies.

Keywords: image data warehouse, similarity query, OLAP query, bitmap index, metric access method

Sumário

Lista de Figuras	xi
Lista de Tabelas	xiii
Lista de Algoritmos	xv
1 Introdução	1
1.1 Motivação e Contribuições	2
1.2 Estrutura da Dissertação	4
2 Data Warehouse Convencional	7
2.1 Introdução	8
2.2 Características dos Dados	8
2.3 Modelagem dos Dados	9
2.4 Estrutura Lógica	11
2.5 Consultas OLAP	12
2.6 Índices Bitmap	14
2.6.1 Índice Bitmap básico	14
2.6.2 Técnica de <i>binning</i>	15
2.6.3 Técnicas de compressão	16
2.6.4 Técnicas de codificação	18
2.6.5 Índice bitmap de junção	20
2.6.6 Biblioteca FastBit	21
2.7 Considerações Finais	22
3 Data Warehouse de Imagens	25
3.1 Introdução	26
3.2 Espaços Métricos e Consultas por Similaridade	26

3.2.1	Espaços Métricos e Vetoriais	27
3.2.2	Funções de Distância	28
3.2.3	Extração de Características	30
3.2.4	Técnica Omni	31
3.3	Camadas Perceptuais	34
3.4	Consulta IOLAP	35
3.5	Arquitetura	36
3.6	Considerações Finais	40
4	Trabalhos Correlatos	43
4.1	iCube: A Similarity-Based Data Cube for Medical Images	43
4.2	Índice Bitmap Baseado em Dimensões Representativas	45
4.3	Hierarchical Bitmap Index (HBI)	47
4.4	Grid Bitmap Index (GBI)	51
4.5	Considerações Finais	53
5	Processamento de Consultas IOLAP	55
5.1	Proposta do ImageDW-index	56
5.1.1	Diretrizes para o Ambiente de DWing de Imagens	56
5.1.2	Detalhamento do ImageDW-index	57
5.2	Estratégias de Processamento de Consultas	61
5.2.1	Abordagem Especialista	63
5.2.2	Abordagem Unificada	66
5.2.3	Algoritmos Executados pelos Módulos de Processamento de Consultas IOLAP	68
5.3	Gerador de Dados ImageDW-Gen	72
5.3.1	Origem dos Dados e Ferramentas Utilizadas para o ETL de Imagens	72
5.3.2	Esquema Estrela Adotado	73
5.3.3	Estrutura do ImageDW-Gen	75
5.4	Classes de Consultas IOLAP	77
5.5	Considerações Finais	78
6	Testes de Desempenho	79
6.1	Testes Baseados na Seletividade dos Predicados de Similaridade entre Imagens	80
6.2	Testes Baseados na Seletividade dos Predicados Convencionais e dos Predicados de Similaridade entre Imagens	84
6.3	Considerações Finais	91

7	Conclusões	93
7.1	Contribuições	95
7.2	Trabalhos Futuros	95
	Referências Bibliográficas	97
	Apêndice A Análise das Principais Técnicas de Índice Bitmap	103
A.1	Descrição dos Testes	103
A.2	Resultados	104
A.2.1	Comparação das Técnicas de Codificação sem <i>Binning</i>	105
A.2.2	Comparação das Técnicas de Codificação com <i>Binning</i>	106
A.3	Considerações Finais	108

Lista de Figuras

2.1	Estrutura do DW em níveis de agregação.	9
2.2	Exemplo de um cubo de dados.	10
2.3	Exemplo de um esquema estrela sobre exames médicos.	12
2.4	Exemplo das operações <i>drill-down</i> e <i>roll-up</i>	13
2.5	Exemplo de uma operação <i>slice and dice</i>	13
2.6	Exemplo de uma operação <i>pivot</i>	14
2.7	Exemplo de consulta utilizando a técnica <i>binning</i> . Adaptado de (Stockinger and Wu, 2007).	16
2.8	Passos do método de compressão WAH. Adaptado de (Stabno and Wrembel, 2007). 17	
2.9	Aplicação de formas de codificação simples sobre um atributo X. O RID (<i>Row ID</i>) representa a tupla na base de dados. Adaptado de (Wu et al., 2010a). . . .	18
2.10	Ilustração de um índice bitmap de dois níveis do tipo <i>equality-equality</i> . Adaptado de (Wu et al., 2010a).	19
2.11	Exemplo de um índice bitmap de junção.	21
3.1	Exemplos de consultas em espaços métricos. (a) Consulta por abrangência e (b) consulta pelos 4 vizinhos mais próximos. Adaptado de (Traina-Jr et al., 2007). .	28
3.2	Exemplos das formas geométricas geradas pelas distâncias L_p , considerando um espaço bidimensional. (a) Distância <i>Manhattan</i> (L_1), (b) Distância Euclidiana (L_2) e (c) Distância <i>Chebychev</i> (L_∞).	29
3.3	Definição de uma <i>mbOr</i> (área hachurada) para uma consulta ao elemento s_q , com raio r_q e dois elementos representativos, f_1 e f_2 . Adaptado de (Traina-Jr et al., 2007).	33

3.4	Influência do posicionamento dos representantes globais na determinação de uma <i>mbOr</i> ótima. Na Figura 3.4a, os elementos estão dispostos nas bordas do conjunto de dados. Já na Figura 3.4b, os representantes estão muito próximos entre si. Adaptado de (Traina-Jr et al., 2007)	34
3.5	Arquitetura de um ambiente de DWing de imagens.	37
3.6	Processo de armazenamento das imagens no DW de imagens. Adaptado de (Annibal et al., 2010).	38
3.7	Esquema estrela estendido com a dimensão <i>Image</i> . Adaptado de (Annibal et al., 2010).	39
4.1	Etapas do processamento de consultas no iCube. Adaptado de (Annibal et al., 2010).	45
4.2	Índice bitmap para uma base de dados multimídia, com $T = 3$. Adaptado de (Jeong and Nang, 2004).	47
4.3	Particionamento do espaço de dados. Adaptado de (Nang et al., 2010).	48
4.4	Árvore de particionamento em intervalos. Os intervalos com os mesmos limitantes estão representados com uma mesma textura. Adaptado de (Nang et al., 2010).	50
4.5	Exemplo de uma consulta <i>Knn</i> . A região hachurada representa o conjunto final de elementos candidatos após a intersecção dos candidatos em x e em y . Adaptado de (Cha, 2004).	52
5.1	Estrutura genérica do ImageDW-index, a qual representa o índice bitmap construído sobre os atributos de distância entre as imagens armazenadas e os elementos representativos, segundo a técnica Omni, para cada camada perceptual.	57
5.2	Ilustração de duas camadas perceptuais de um DW de imagens em seus espaços métricos e com seus respectivos elementos representativos, os quais são definidos de acordo com a técnica Omni. Em (a) é representado <i>PL0</i> , com três elementos representativos. Em (b) é ilustrado <i>PL1</i> com dois elementos representativos.	58
5.3	ImageDW-index para uma amostra dos dados do DW de imagens, considerando a camada perceptual <i>PL0</i> . Para cada imagem identificada por um <i>ID</i> , o índice bitmap é ilustrado.	59
5.4	ImageDW-index para uma amostra dos dados do DW de imagens, considerando a camada perceptual <i>PL1</i> . Para cada imagem identificada por um <i>ID</i> , o índice bitmap é ilustrado.	59
5.5	Processamento de consultas IOLAP de acordo com a Estratégia EspecialistaAccessaDW.	64

5.6	Processamento de consultas IOLAP de acordo com a Estratégia EspecialistaU- saBitmap.	65
5.7	Processamento de consultas IOLAP de acordo com a Estratégia UnificadaAces- saDW.	67
5.8	Processamento de consultas IOLAP de acordo com a Estratégia UnificadaUsa- Bitmap.	67
5.9	Esquema Estrela do DW de imagens.	74
5.10	Principais componentes da estrutura do ImageDW-Gen.	76
6.1	Resultados da execução de Q1, Q2, Q3 e Q4 utilizando as estratégias Especialis- taAcessaDW, EspecialistaUsaBitmap e o trabalho correlato, para as seletividades consideradas.	82
6.2	Gráfico com a média dos resultados obtidos pela execução das consultas Q1, Q2, Q3 e Q4.	84
6.3	Desempenho das estratégias propostas, considerando as consultas em que a se- letividade dos predicados convencionais é variada.	87
6.4	Desempenho da estratégia de Annibal et. al. (2010), considerando a execução das consultas variando a seletividade dos predicados convencionais.	89
A.1	Gráfico com o tempo médio obtido da execução das consultas Q1, Q2, Q3 e Q4 em diferentes seletividades, utilizando as técnicas de codificação InEnc, BiEnc e IEEnc isoladamente.	106
A.2	Gráficos com os resultados obtidos pela execução de Q1, Q2, Q3 e Q4 sobre índices bitmap configurados de acordo com as técnicas de codificação IEnc, IEEnc e BiEnc com <i>binning</i>	107

Lista de Tabelas

3.1	Principais descritores de Haralick (Haralick et al., 1973).	31
4.1	Casos possíveis em que os valores σ_p^j e σ_q^j podem se enquadrar no intervalo I^1 . .	50
4.2	Resumo das principais características das técnicas de indexação baseadas em bitmap para consultas por similaridade.	53
5.1	Distâncias entre a imagem de consulta img_q e os elementos representativos, bem como os intervalos obtidos para a camada perceptual $PL0$, considerando um raio $r_q = 2$	60
5.2	Distâncias entre a imagem de consulta img_q e os elementos representativos, bem como os intervalos obtidos para a camada perceptual $PL1$, considerando um raio $r_q = 2$	60
5.3	Características das camadas perceptuais.	74
6.1	Características dos atributos convencionais utilizados nessa bateria de testes. . .	85
6.2	Seletividade das consultas Q1, Q2, Q3 e Q4.	86
6.3	Tempo médio (em segundos) do desempenho das estratégias para as cinco consultas realizadas.	90

Lista de Algoritmos

1	ConveFi (<i>convPred</i> , <i>inSet</i> , <i>selectFV</i>)	68
2	ConvAndNonConveFi (<i>convPred</i> , <i>percL</i> [1.. <i>n</i>], <i>Img_q</i> , <i>r_q</i>)	70
3	OmRef (<i>candidateSet</i> , <i>percL</i> [1.. <i>n</i>], <i>featVec_q</i> [1.. <i>n</i>], <i>r_q</i>)	71

Introdução

Um ambiente de *data warehousing* (DWing) consolida grandes volumes de dados de interesse, oriundos de provedores de informação autônomos, distribuídos e heterogêneos em um único banco de dados, o *data warehouse* (DW). Esse ambiente garante eficiência e flexibilidade na recuperação de informações estratégicas voltadas aos processos de gerência e de tomada de decisão (Chaudhuri and Dayal, 1997; Mohania et al., 1999). Em um ambiente de DWing, o acesso à informação integrada é realizado, geralmente, em duas etapas:

- etapa de ETL (*extract, transform, load*), na qual a informação de cada provedor é extraída previamente, devendo ser traduzida, filtrada, integrada à informação relevante de outros provedores e, finalmente, armazenada no DW; e
- etapa de análise e consulta, na qual operações analíticas, denominadas OLAP (*on-line analytical processing*) (Codd et al., 1993), são executadas diretamente no DW, sem acessar os provedores originais.

Dessa forma, a informação integrada torna-se disponível para análise imediata dos usuários de sistemas de suporte à decisão (SSD) (Ariyachandra and Watson, 2010; Chaudhuri et al., 2011; Golfarelli and Rizzi, 2009; Gonzales et al., 2011). Por exemplo, um ambiente de DWing para a área médica pode integrar dados relativos à incidência de câncer de mama segundo diferentes hospitais e faixas etárias ao longo dos anos. Usuários de SSD dessa aplicação, tais como um superintendente de hospital ou líderes de equipes médicas, podem realizar, utilizando as informações integradas, análises de tendências

simples (ex.: “Qual a incidência de câncer de mama em julho de 2005?”), análises comparativas (ex.: “Qual a incidência de câncer de mama nos últimos três anos?”) e análises de tendências múltiplas (ex.: “Qual a incidência de câncer de mama nos últimos três anos reportadas por diferentes hospitais, considerando faixas etárias distintas?”).

As consultas OLAP exemplificadas anteriormente, bem como diversas outras similares, podem ser submetidas a DWs convencionais, por que envolvem tipos de dados alfanuméricos. Esse tipo de dado possui relação de ordem total e, portanto, pode ser ordenado e pesquisado usando operadores relacionais (ex.: menor, maior, igual e diferente). Entretanto, existe uma nova gama de consultas OLAP que não pode ser respondida por essas aplicações, desde que consideram também dados complexos em seu preâmbulo, tais como imagens, os quais não possuem relação de ordem total. Por exemplo, certa equipe médica pode estar interessada em analisar a quantidade de imagens que são similares a uma imagem de câncer de mama para avaliar a evolução de determinadas formas da patologia segundo critérios de interesse, tais como a distribuição geográfica ou fatores médicos. Nesse contexto, a equipe médica poderia estar interessada na seguinte consulta: “Qual a quantidade de imagens, similares a uma determinada imagem de câncer de mama, que ocorreu nos últimos 3 anos em hospitais da região sul do Brasil?”.

1.1 Motivação e Contribuições

A nova gama de consultas OLAP enfocada nesta dissertação envolve tanto predicados convencionais, os quais referem-se a critérios de seleção sobre os dados convencionais, quanto predicados de similaridade entre imagens, os quais referem-se a critérios que requerem a comparação de imagens. Ambientes de DWing que visam oferecer suporte a essa nova gama de análises demandam, portanto, o desenvolvimento de estratégias para o processamento eficiente dessas consultas complexas e custosas.

Na literatura, embora existam abordagens voltadas à indexação em ambientes de DWing convencionais e à indexação de imagens em bancos de dados multimídia, elas apresentam como limitação o fato de não considerarem essas duas áreas de pesquisa em um mesmo contexto. Ou seja, desde que as consultas OLAP enfocadas nesta dissertação requerem a comparação entre imagens, faz-se necessário o uso de operações baseadas em similaridade.

Portanto, esta dissertação de mestrado visa suprir a lacuna existente na literatura no que se refere ao processamento eficiente de consultas analíticas em ambientes de DWing de imagens. As propostas apresentadas nesta dissertação consideram o uso de DWs relacionais. Segundo essa abordagem, os dados são armazenados em um banco de dados

relacional de acordo com um esquema estrela (Kimball and Ross, 2002). Adicionalmente, nesta dissertação é usada a definição de DW de imagens introduzida em (Annibal, 2011), a qual considera que um DW de imagens é projetado segundo um esquema estrela diferenciado pois, além de possuir tabelas de dimensão com dados convencionais, esse esquema também possui uma ou mais tabelas de dimensão especificamente voltadas à manipulação de imagens, as quais armazenam a representação do conteúdo intrínseco das imagens por meio de vetores de características.

Esta dissertação de mestrado introduz as seguintes contribuições principais: (i) proposta do ImageDW-index, um mecanismo para a otimização de consultas analíticas estendidas com predicados de similaridade entre imagens; e (ii) definição de diferentes estratégias para o processamento dessas consultas usando o ImageDW-index. Para validar as soluções propostas, foram introduzidas as seguintes contribuições secundárias: (iii) desenvolvimento do ImageDW-Gen, um gerador de dados para povoamento do DW de imagens; e (iv) especificação de classes de consultas, cada qual enfocando em diferentes custos do processamento de consultas em ambientes de DWing de imagens.

O ImageDW-index utiliza as vantagens de duas técnicas já existentes na literatura: a técnica Omni (Traina-Jr et al., 2007) e o índice bitmap (Wu et al., 2010b). A técnica Omni introduz mecanismos de filtragem e refinamento do espaço métrico, formado pelos vetores de características das imagens, em consultas por similaridade. Para isso, são identificados pontos conhecidos do espaço métrico, denominados elementos representativos, os quais são utilizados para uma grande poda no conjunto de dados. Índices bitmap são amplamente usados em aplicações de DWing convencionais, por apresentarem melhor desempenho em grandes volumes de dados, visto que as consultas são respondidas por meio de operações lógicas bit a bit.

Utilizando o ImageDW-index, foram propostas as seguintes estratégias para o processamento de consultas analíticas sobre o DW de imagens: EspecialistaAcessaDW, EspecialistaUsaBitmap, UnificadaAcessaDW, UnificadaUsaBitmap. Essas estratégias foram elaboradas com o objetivo de se investigar como deve ser realizado o processamento de consultas que envolvem tanto predicados convencionais quanto predicados de similaridade entre imagens.

Com relação ao ImageDW-Gen, seu desenvolvimento foi motivado pelo fato de não haver na literatura um DW de imagens volumoso para ser usado em testes de desempenho. O ImageDW-Gen constrói um DW de imagens sobre exames médicos, o qual envolve hospitais, pacientes e diferentes tipos de exames realizados ao longo dos anos. Para isso, obteve-se acesso a um banco de imagens médicas reais do Hospital das Clínicas de Ribeirão Preto (HCRP), o qual armazena exames de diferentes modalidades,

tais como cérebro, mama, tórax, entre outros. Com o auxílio do ImageDW-Gen, dados convencionais sintéticos e reais também foram incorporados ao DW de imagens. Adicionalmente, foram definidas quatro classes de consultas, as quais enfocam em diferentes custos de processamento dos predicados de similaridade entre imagens, com o objetivo de avaliar o desempenho das estratégias de consultas utilizando o ImageDW-index em cenários distintos.

Utilizando o DW de imagens gerado pelo ImageDW-Gen, foram realizados testes de desempenho para investigar as vantagens introduzidas pelas estratégias de processamento de consultas propostas. Para isso, as consultas utilizadas foram definidas em termos das classes propostas considerando cenários que envolvem apenas predicados de similaridade entre imagens e também cenários envolvendo predicados convencionais. De acordo com os resultados obtidos, quando comparada ao trabalho correlato descrito em (Annibal et al., 2010), a estratégia EspecialistaAcessaDW proveu uma melhora no desempenho do processamento de consultas que envolvem apenas predicados de similaridade entre imagens, com um ganho médio que variou de 55.57% até aproximadamente 82.16%. Já a estratégia UnificadaAcessaDW proveu um ganho de aproximadamente 76,76% em consultas que envolvem tanto predicados de similaridade entre imagens quanto convencionais.

1.2 Estrutura da Dissertação

Esta dissertação de mestrado está organizada da seguinte maneira:

- No capítulo 2 são discutidos os principais tópicos relacionados a DWs convencionais e índices bitmap.
- No capítulo 3 são abordados os conceitos de DWs de imagens, consultas por similaridade e da técnica Omni, os quais são utilizados nesta dissertação de mestrado.
- No capítulo 4 são investigados os trabalhos correlatos a essa dissertação de mestrado.
- No capítulo 5 é apresentada a proposta do ImageDW-index e das estratégias de processamento de consultas. Nesse capítulo, também são descritos o ImageDW-Gen e as classes de consulta propostas.
- No capítulo 6 são detalhados os testes de desempenho realizados para validação do ImageDW-index.
- No capítulo 7 são apresentadas as conclusões finais dessa dissertação, bem como os trabalhos futuros.

-
- No Apêndice A são discutidos os resultados dos testes de desempenho realizados para determinar a estrutura do índice bitmap adotada para o ImageDW-index.

Data Warehouse Convencional

O crescente poder de processamento dos computadores e a sofisticação das técnicas e ferramentas para análise de grandes bases de dados resultaram no desenvolvimento do que é conhecido hoje como DW (Elmasri and Navathe, 2011). Um ambiente de DWing é uma coleção de tecnologias criada para auxiliar especialistas (ex.: executivos, gerentes, analistas) a tomarem decisões melhores e mais rápidas em seus negócios (Chaudhuri and Dayal, 1997). Para isso, esse ambiente consolida arquiteturas, algoritmos e ferramentas que permitem que dados selecionados de diferentes provedores de informação (fonte dos dados) sejam integrados em uma única base chamada DW.

A obtenção de informações estratégicas para o processo de tomada de decisão é de suma importância para que as empresas tenham vantagem competitiva em seus negócios. Dessa forma, tecnologias relacionadas a DW, como aplicações OLAP, têm tido um papel fundamental para o sucesso dessas organizações. Ademais, as empresas não são as únicas a se beneficiarem dessa tecnologia, visto que na área médica, por exemplo, aplicações dessa natureza têm sido amplamente exploradas para auxiliar os médicos a elaborarem diagnósticos mais precisos, determinando tratamentos mais eficazes. Como resultado, as pesquisas nessa área têm estudado a incorporação de dados multimídia ao DW, visando proporcionar uma nova gama de consultas OLAP (Annibal et al., 2010; Arigon et al., 2007; Chen et al., 2008; Jin et al., 2010; Wong et al., 2004).

Esse capítulo é organizado da seguinte maneira: Nas seções 2.1, 2.2, 2.3, 2.4 e 2.5 são descritos os principais conceitos relacionados aos DWs convencionais. Na seção 2.6 são detalhados o índice bitmap e o índice bitmap de junção, os quais são amplamente usados em aplicações de DWing. O capítulo é finalizado na seção 2.7 com as considerações finais.

2.1 Introdução

Um DW é um repositório projetado para armazenar um grande volume de dados sobre o qual operações analíticas (e.g. consultas OLAP) podem ser realizadas. DWs convencionais armazenam apenas dados alfanuméricos, os quais são considerados dados convencionais. Devido à sua finalidade analítica, a implementação de um DW exige uma forma especial de organização dos dados, além de métodos de acesso apropriados, os quais, em geral, não são implementados em SGBDs (Sistemas de Gerenciamento de Banco de Dados) destinados ao ambiente operacional de uma organização.

O ambiente operacional é aquele constituído por aplicações responsáveis pelas operações cotidianas do negócio. Nesses ambientes são realizadas, em sua maioria, transações de inserção, atualização e remoção de dados. Esse tipo de operação é conhecido como OLTP (*on-line transaction processing*). Por outro lado, um ambiente informacional refere-se às ferramentas e algoritmos que dão suporte à análise do negócio. Nesse contexto, o DW emerge como um dos principais componentes do ambiente informacional de uma organização.

Construir um DW é uma tarefa complexa que pode levar anos para ser concluída. Para diminuir essa complexidade, pode-se implementar *data marts*. *Data marts* são DWs de menor porte que armazenam subconjuntos de dados focados em assuntos específicos ou departamentais (ex.: um *data mart* para o departamento de recursos humanos de um hospital) (Chaudhuri and Dayal, 1997).

2.2 Características dos Dados

Uma das principais diferenças do DW com relação aos bancos de dados tradicionais diz respeito à origem dos dados. Enquanto no segundo eles são carregados a partir de transações cotidianas, como inserção de novos pacientes de um hospital, agendamento de um exame, etc, no primeiro, os dados, em sua grande maioria, são oriundos do próprio ambiente operacional da organização, partindo de diferentes provedores de informação.

Cada provedor de informação pode armazenar seus dados utilizando diferentes padrões, formatos e tecnologias. Por essa razão, um ambiente de DWing possui técnicas para integrar esses dados de maneira consistente, removendo incoerências e realizando as devidas padronizações nos dados.

Pode-se dizer que um DW é uma coleção de dados integrados, orientados a assunto, históricos e não voláteis (Inmon, 2005). Integrados, porque os dados são obtidos a partir

de diversos sistemas relacionados aos negócios da organização, sendo que esses sistemas podem utilizar diferentes tecnologias. Orientados a assunto, visto que o DW é construído para oferecer suporte às análises relacionadas aos temas de negócio previamente especificados. Históricos, pois são relevantes por determinado período de tempo. E, finalmente, não voláteis, pois raramente os dados são atualizados em um DW, permanecendo estáveis por longos períodos de tempo.

Outro aspecto importante é que, em um DW, os dados são armazenados segundo diferentes níveis de agregação, como ilustra a Figura 2.1. O nível inferior possui os dados coletados diretamente dos provedores de informação, servindo como base para futuras investigações (dados mais detalhados, de menor granularidade). A escolha da granularidade nesse nível impacta no tamanho do DW, bem como nos tipos de consultas que podem ser respondidas pelo ambiente. Quanto menor a granularidade do nível inferior, maior a variedade de consultas que pode ser submetida ao DW, porém o espaço em disco consumido é maior. Por outro lado, o nível superior possui dados altamente resumidos (dados menos detalhados, de maior granularidade). Por fim, os níveis intermediários armazenam os dados em graus de agregação crescente. Dessa forma, usuários de SSD podem iniciar suas análises a partir do nível superior para ter uma visão geral do negócio, e então percorrer a hierarquia para obter as informações mais detalhadas.

Figura 2.1: Estrutura do DW em níveis de agregação.

2.3 Modelagem dos Dados

Para oferecer suporte às análises complexas e diversificadas (*ad-hoc*) realizadas por usuários SSD, facilitando a organização, visualização e o resumo dos dados sob diferentes

perspectivas, o DW é modelado de acordo com o paradigma multidimensional. Uma visão multidimensional inclui um conjunto de medidas numéricas de interesse, que são os objetos de análise relevantes ao negócio, e um conjunto de dimensões, as quais contêm atributos que determinam o contexto para as medidas numéricas (Chaudhuri and Dayal, 1997). Nesse modelo, o cubo de dados consiste de uma representação gráfica de grande utilidade para exibir as visões dos usuários no espaço multidimensional (Shoshani, 1997), bem como auxiliar na identificação das várias agregações que podem ser originadas.

Para ilustrar a modelagem dos dados em um DW, considere uma aplicação para a área médica projetada para que análises sobre exames médicos de câncer de mama possam ser realizadas sob diferentes perspectivas. Para isso, as seguintes dimensões são adotadas: *Hospital*, *DataExame*, *Paciente* e *Idade*. A primeira tabela de dimensão armazena dados referentes ao hospital em que o exame foi realizado, a segunda refere-se à tabela de dimensão temporal dessa aplicação, a terceira possui dados sobre os pacientes e a quarta tabela de dimensão armazena a idade dos pacientes. Na Figura 2.2, é ilustrado um cubo de dados desse exemplo, sendo que cada célula (ponto no espaço) representa o valor da medida numérica de interesse agregada. Nesse exemplo, a incidência de exames está agregada por mês.

Figura 2.2: Exemplo de um cubo de dados.

Frequentemente, os atributos de uma dimensão podem estabelecer relações hierárquicas entre si (Harinarayan et al., 1996), as quais definem diferentes níveis de agregação e, por conseguinte, o nível máximo de detalhe do DW (granularidade). Do exemplo anterior, os atributos da tabela de dimensão temporal *DataExame* podem fazer parte de uma hierarquia de cinco níveis da seguinte maneira, por exemplo: dia → mês → bimestre → semestre → ano. Dessa forma, os dias podem ser agregados em meses, os quais podem ser

agregados em bimestres, que podem ser agrupados em semestres, que, finalmente, podem ser agregados em anos.

2.4 Estrutura Lógica

O cubo de dados é utilizado para representar de forma conceitual os dados e as agregações do DW, principalmente devido à fácil visualização das operações analíticas que ele possibilita. Para representar essa estrutura de forma lógica, existem duas abordagens comumente utilizadas: estruturas relacionais ou estruturas especializadas. De acordo com a primeira, os dados do DW são armazenados em um SGBD relacional que, geralmente, é estendido para oferecer suporte eficiente às consultas de tomada de decisão, formando assim um sistema ROLAP (OLAP relacional). Pela segunda abordagem, os dados são guardados em estruturas especializadas (geralmente matrizes), sendo que as operações analíticas são implementadas diretamente sobre elas, originando um sistema MOLAP (OLAP multidimensional). Existem ainda os sistemas HOLAP (OLAP híbrido), os quais possuem características dos sistemas ROLAP e MOLAP. Nesta dissertação o foco é em sistemas ROLAP.

Em sistemas ROLAP, os dados são armazenados em relações (tabelas), as quais refletem o modelo multidimensional por meio de dois tipos de esquemas: esquema estrela e esquema floco de neve (Kimball and Ross, 2002). O esquema estrela possui dois tipos de tabelas, a tabela de fatos, localizada visivelmente no centro, e as tabelas de dimensão, nas extremidades. A tabela de fatos armazena as medidas numéricas relevantes ao negócio e as referências (chaves estrangeiras) às tabelas de dimensão. Cada dimensão do modelo multidimensional possui sua respectiva tabela no esquema estrela, a qual possui os dados que contextualizam cada instância (tupla) da tabela de fatos. Uma estrutura mais complexa, na qual várias tabelas de fatos compartilham tabelas de dimensão em comum, é chamada de constelação de fatos.

Na Figura 2.3 é ilustrado um esquema estrela de acordo o exemplo enunciado na seção 2.3. Esse esquema é composto por uma tabela de fatos *Exame*, cuja medida numérica é a quantidade, e pelas tabelas de dimensão *Paciente*, *Idade*, *DataExame* e *Hospital*.

Figura 2.3: Exemplo de um esquema estrela sobre exames médicos.

O esquema floco de neve é gerado a partir da normalização das tabelas de dimensão do esquema estrela, de acordo com as hierarquias de relacionamento de atributos. A manutenção de tabelas desnormalizadas no esquema estrela e a consequente redundância dos dados armazenados é justificada pela necessidade de se obter um maior desempenho nas consultas analíticas. Em contrapartida, no esquema floco de neve, a normalização poupa espaço de armazenamento, porém implica em um maior número de junções durante o processamento das consultas. De fato, a economia de espaço proporcionada pelo esquema floco de neve é pequena, visto que a maior parte do custo de armazenamento de um DW deve-se à tabela de fatos. Por essa razão, o esquema estrela é mais comumente adotado em aplicações de DW. Nesta dissertação o foco é no esquema estrela.

2.5 Consultas OLAP

Os principais tipos de consultas OLAP são *drill-down*, *roll-up*, *slice and dice* e *pivot* (Kimball and Ross, 2002), os quais são ilustrados nesta seção. As consultas *drill-down* analisam os dados em níveis de agregação progressivamente mais detalhados. Por exemplo, um usuário de SSD pode pesquisar a incidência de exames de mama por ano (alto nível de agregação) e sucessivamente detalhar suas consultas, por meio das hierarquias de atributos das dimensões, finalizando esse processo no nível mais detalhado possível, como a quantidade de exames de mama por dia. Já as consultas *roll-up* fazem o inverso, ou seja, iniciam a investigação em níveis de agregação progressivamente menos detalhados,

agrupando os dados em unidades maiores ao longo de uma dimensão. Estes dois tipos de operações são ilustrados na Figura 2.4.

Figura 2.4: Exemplo das operações *drill-down* e *roll-up*.

A operação *slice and dice* permite que sejam feitas análises sobre um subconjunto dos dados, restringindo o escopo das consultas. Desse modo, os usuários podem realizar um corte (*slice*) ao fixar um valor para uma ou mais dimensões, além de selecionar uma faixa de valores para as demais dimensões (*dice*), como ilustrado na Figura 2.5.

Figura 2.5: Exemplo de uma operação *slice and dice*.

Diferentes perspectivas dos mesmos dados podem ser obtidas pela operação *pivot*, a qual reorienta a visão multidimensional dos dados de acordo com a necessidade do usuário. Essa operação altera a ordem das dimensões, isto é, modifica a orientação dos eixos sobre os quais os dados são visualizados. Dessa forma, várias combinações das dimensões podem ser analisadas de modo a investigar diferentes relacionamentos que podem existir entre os dados. Na Figura 2.6 é ilustrada a mudança de orientação das dimensões *Tempo* e *Faixa etária*.

Figura 2.6: Exemplo de uma operação *pivot*.

2.6 Índices Bitmap

No contexto de DW, existem diversos trabalhos na literatura que apresentam propostas de índices para melhorar o desempenho de consultas OLAP, sendo que as mais promissoras são aquelas relacionadas aos índices bitmap. Esse tipo de índice é adotado devido à sua eficiência na indexação de objetos multidimensionais, maior ênfase no desempenho de consultas e por apresentar melhores resultados em consultas do tipo *ad-hoc* (Stockinger and Wu, 2007).

Essa seção está organizado da seguinte forma: Na seção 2.6.1 é descrito o funcionamento de um índice bitmap básico. Nas seções 2.6.2, 2.6.3, 2.6.4 são resumidas as principais técnicas para melhorar o desempenho de índices bitmap, que são: *binning*, compressão e codificação; na seção 2.6.5 é introduzido o conceito de índice bitmap de junção, uma técnica eficiente para uso em DWs; e na seção 2.6.6 é descrito o software FastBit, o qual provê implementações eficientes de índices bitmap.

2.6.1 Índice Bitmap básico

Um índice bitmap básico consiste de vetores de bits construídos para cada valor do domínio de um atributo, sendo que cada elemento desses vetores faz referência a uma tupla da tabela original. Se há um bit “1” na posição j de um bitmap, então a tupla j da tabela original conterá o valor representado por esse vetor. Caso contrário, haverá um bit “0”.

Com relação ao desempenho das consultas, índices bitmap proporcionam ótimos resultados em termos de comparações, visto que operações lógicas bit a bit são realizadas de maneira muito eficiente pelos processadores. Já em termos de espaço de armazenamento,

índices bitmap são mais vantajosos quando aplicados a atributos de baixa cardinalidade. Outra característica desse índice é o alto custo de atualização e, por essa razão, ele é mais indicado para bancos de dados focados em operações de leitura (Stockinger and Wu, 2007), motivando sua utilização em ambientes de DWing.

Em sua forma tradicional, índices bitmap podem gerar vetores esparsos para uma grande base de dados, isto é, poucos bits com valor “1” e muito espaço sendo ocupado para armazenar zeros (O’Neil and Graefe, 1995). Entretanto, existem técnicas que podem ser aplicadas para reduzir o espaço de armazenamento do índice, resolvendo o problema de vetores esparsos, bem como para torná-lo eficiente em domínios de alta cardinalidade. As principais técnicas de otimização para índice bitmap são agrupadas e discutidas nas seções de 2.6.2 a 2.6.4.

2.6.2 Técnica de *binning*

Por meio da técnica de *binning*, são criados grupos (bins) ou identificadores, pelos quais os valores do atributo indexado são organizados. Dessa forma, os vetores de bits são criados para cada *bin*, desassociando o número de bitmaps da cardinalidade do atributo e assim, reduzindo o espaço de armazenamento necessário para o índice (Stockinger and Wu, 2007). Essa implementação é útil, por exemplo, para indexação de atributos numéricos que podem assumir vários valores.

Uma desvantagem é que essa estratégia cria um índice impreciso, isto é, apenas a busca por um valor no índice não é suficiente para responder com exatidão a algumas consultas, sendo necessário um processo de checagem dos valores candidatos (*candidate-check*). O *candidate-check* consiste do acesso aos valores reais do atributo indexado para compará-los ao valor da consulta, eliminando os falsos positivos. Em grandes bases de dados esse processo de checagem pode ser muito custoso, sendo necessário vários acessos a disco para processar uma consulta. Dessa forma, a técnica de *binning* apresenta um desempenho irregular, dependendo da consulta e dos bins escolhidos.

Na Figura 2.7 é ilustrada uma consulta utilizando um índice bitmap com a técnica de *binning*. Nesse exemplo, considera-se que um atributo numérico A possui valores no intervalo de 0 a 100 que foi particionado em cinco bins. O bit “1” significa que o valor de A, em determinada tupla, pertence ao *bin* correspondente. Caso contrário, o bit é “0”.

		0	1	2	3	4	--> Bins
RID	A	[0,20)	[20,40)	[40,60)	[60,80)	[80,100]	--> Intervalos
1	34.7	0	1	0	0	0	
2	94	0	0	0	0	1	
3	24.9	0	1	0	0	0	
4	15.5	1	0	0	0	0	
5	61.7	0	0	0	1	0	
6	67.2	0	0	0	1	0	
7	58.6	0	0	1	0	0	

 $37 \leq A < 63$

Figura 2.7: Exemplo de consulta utilizando a técnica *binning*. Adaptado de (Stockinger and Wu, 2007).

Utilizando o índice da Figura 2.7, suponha a seguinte consulta: “Qual a quantidade de tuplas em que $37 \leq A < 63$?”. Analisando o intervalo da consulta, a resposta está contida nos bins 1, 2 e 3, dos quais pode-se afirmar com certeza que apenas as entradas com bit “1” do vetor 2 fazem parte da resposta, isto é, RID 7. Assim, para confirmar o restante da resposta, é preciso analisar cada entrada com bit “1” dos vetores 1 e 3 (elementos candidatos). Portanto, essa consulta retornará o RID 7 diretamente e, após a checagem dos elementos candidatos do bin 1 (RIDs 1 e 3) e do bin 3 (RIDs 5 e 6), o RID 5 também fará parte do resultado.

2.6.3 Técnicas de compressão

Uma técnica importante para melhorar o desempenho de um índice bitmap consiste na compressão dos vetores de bits gerados. Com essa técnica, é possível reduzir o espaço necessário para armazenamento do índice usando como base algum método de compressão sem perdas. Entretanto, a compressão dos dados afeta diretamente no processamento das consultas, visto que a descompressão do índice é necessária para sua utilização (Wu et al., 2010b).

Na literatura, são encontrados métodos de compressão especializados que visam a redução do tempo no processamento de consultas que utilizam índice bitmap, tais como o *Byte-Aligned Bitmap Code* (BBC) e seus variantes (Antoshenkov, 1995; Johnson, 1999), *Word-Aligned Hybrid* (WAH) (Wu et al., 2006) e um método chamado *Run-Length Huffman* (RLH) (Stabno and Wrembel, 2007). Dentre os métodos citados, o que é utilizado nesta dissertação de mestrado é o WAH, por ser uma opção que proporciona ótimos resultados em termos de tempo de processamento (Wu, 2005).

Figura 2.8: Passos do método de compressão WAH. Adaptado de (Stabno and Wrembel, 2007).

2.6.4 Técnicas de codificação

A codificação define a forma como os valores de um atributo são traduzidos em bitmaps. Técnicas de codificação simples são: *equality encoding*, *range encoding* e *interval encoding*. Além dessas técnicas, o *binary encoding* também diminui o tamanho do índice (Wong et al., 1985). Essas técnicas são detalhadas a seguir, e ilustradas na Figura 2.9.

Em sua maneira mais simples, o índice bitmap é codificado por igualdade (*equality encoding*), isto é, para cada valor de um atributo com cardinalidade C são criados C bitmaps. Essa forma de codificação é mais adequada em consultas com critérios de igualdade, tais como “ $X = 2$ ”.

		<i>equality</i>				<i>range</i>			<i>interval</i>			<i>binary</i>	
		= 0	= 1	= 2	= 3	<=0	<=1	<=2	[0,1]	[1,2]	[2,3]	&2	&1
RID	X	e_0	e_1	e_2	e_3	r_0	r_1	r_2	i_0	i_1	i_2	b_1	b_2
1	1	0	1	0	0	0	1	1	1	1	0	0	1
2	3	0	0	0	1	0	0	0	0	0	1	1	1
3	2	0	0	1	0	0	0	1	0	1	1	1	0
4	0	1	0	0	0	1	1	1	1	0	0	0	0
...	...												

Figura 2.9: Aplicação de formas de codificação simples sobre um atributo X. O RID (*Row ID*) representa a tupla na base de dados. Adaptado de (Wu et al., 2010a).

O *range encoding* utiliza valores limitantes do atributo indexado para definir condições de pertinência das tuplas aos bitmaps. Dessa forma, um bit correspondente a uma tupla recebe “1” se o atributo indexado for menor do que aquele associado ao bitmap. Esse tipo de codificação é otimizado para responder eficientemente a consultas da forma “ $X < 2$ ” (*one-sided range query*).

O *interval encoding* define intervalos para os quais os vetores de bits são criados. Nessa abordagem, cada bitmap agrupa um conjunto de valores, sendo que o bit “1” significa que o valor do atributo na tupla correspondente pertence àquele intervalo. Esse tipo de codificação é otimizado para responder eficientemente a consultas da forma “ $0 < X < 2$ ” (*two-sided range query*).

O método de codificação que produz o menor número de bitmaps é o *binary encoding*, o qual define uma palavra binária para cada valor do atributo indexado. Esse método utiliza somente $\log_2 C$ bitmaps.

Na literatura, as principais técnicas de codificação de índice bitmap são categorizadas em índices multicomponente e índices multinível (Chan and Ioannidis, 1999; Stockinger and Wu, 2007; Wu et al., 2006, 2010b). A codificação multicomponente quebra os valores do atributo indexado em múltiplos componentes, de maneira similar à forma como

os números são decompostos em seus dígitos decimais. A principal diferença é que as bases numéricas para cada componente podem ser diferentes. O principal exemplo dessa categoria é o *binary encoding*, o qual decompõe os valores em componentes de base dois. O *binary encoding* representa a decomposição máxima possível de um índice multicomponente. Ademais, de acordo com essa definição, fazem parte dessa categoria todas as técnicas simples de codificação, desde que elas podem ser tratadas como sendo do tipo 1-componente.

Na codificação multinível, é criada uma hierarquia de intervalos, sendo possível codificar cada nível separadamente utilizando quaisquer das técnicas de codificação simples. O nível inferior é o nível mais detalhado (*fine level*), o qual é capaz de responder a qualquer consulta com exatidão. Em contrapartida, o nível superior (*coarse level*) é criado para reduzir a quantidade de trabalho necessária para responder às consultas, por meio de intervalos mais abrangentes.

Na Figura 2.10 é ilustrado um índice de dois níveis para um atributo A, ambos codificados utilizando *equality encoding*. Nesse exemplo, os dados originais foram inicialmente agrupados em quatro bins com os seguintes intervalos $[0; 0,25[$, $[0,25; 0,5[$, $[0,5; 0,75[$ e $[0,75; 1[$, sendo esse o nível inferior do índice. O nível superior possui dois intervalos, os quais consistem da união (OR) entre os intervalos inferiores 0 e 1 e 2 e 3. Para resolver à consulta $A < 0.5$, são necessários os dois primeiros bitmaps do nível inferior ou apenas o primeiro bitmap do nível superior, evidenciando a vantagem oferecida pelo índice multinível.

Figura 2.10: Ilustração de um índice bitmap de dois níveis do tipo *equality-equality*. Adaptado de (Wu et al., 2010a).

De acordo com a análise encontrada em (Wu et al., 2010b), a codificação multinível mais eficiente deve possuir dois níveis, sendo que o nível inferior deve ser codificado de

acordo com o *equality-encoding* e os níveis superiores de acordo com *range-encoding* ou *interval-encoding* (Wu et al., 2010a,b). Dessa forma, os principais índices do tipo multi-nível são conhecidos como *equality-equality*, *interval-equality* e *range-equality*, sendo que o primeiro é eficiente para consultas com critérios de igualdade, e os demais apresentam melhores resultados para consultas do tipo *one-sided query* e *two-sided query*, respectivamente.

2.6.5 Índice bitmap de junção

Em ambientes de DWing, consultas OLAP demandam a realização de várias operações de junção entre a tabela de fatos e suas dimensões, o que caracteriza uma junção estrela. Devido ao grande volume de dados envolvido, o custo de junção torna-se um fator crítico de desempenho. Uma técnica que melhora o processamento dessas consultas consiste na construção de um índice bitmap para os atributos das tabelas de dimensão. Dessa forma, por meio do índice, é possível identificar o subconjunto de tuplas da tabela de fatos que satisfazem a uma consulta envolvendo os atributos indexados. De maneira geral, essa técnica permite a restrição das operações de junção apenas às tuplas que correspondem aos critérios da busca. Esse tipo de construção é denominado índice bitmap de junção (O’Neil and Graefe, 1995), o qual é amplamente utilizado em aplicações de DW.

Na Figura 2.11 é ilustrado um índice bitmap de junção para a aplicação enunciada na seção 2.3. Nesse exemplo, foi criado um índice para o atributo endereço da tabela de dimensão *Hospital*. A tabela de fatos *Exame* faz referência à tabela de dimensão pela chave estrangeira *e_HospitalKey*. As Figuras 2.11a e 2.11b ilustram uma amostra dos dados da dimensão *Hospital* e da tabela de fatos, respectivamente. Na Figura 2.11c é ilustrado o índice bitmap criado para o atributo *endereço* da tabela de dimensão *Hospital*. Por exemplo, a primeira tupla da tabela de fatos faz referência ao Hospital 1, o qual é localizado no endereço E1. Como resultado, a primeira posição do bitmap referente a E1 recebe o valor 1, enquanto as primeiras posições dos bitmaps referentes a E2 a E5 recebem o valor 0.

hospitalKey	Endereço	Cidade	e_HospitalKey	e_IdadeKey	Quantidade
1	E1	São Carlos	1	3	1
2	E2	Campinas	1	2	1
3	E3	Leme	1	1	1
4	E4	Araras	2	2	1
5	E5	São Paulo	3	2	1
			3	3	1
			4	5	1
			5	6	1

(a) Fragmento da tabela de dimensão
Hospital.

(b) Parte da tabela de fatos
Exame.

E1	E2	E3	E4	E5
1	0	0	0	0
1	0	0	0	0
1	0	0	0	0
0	1	0	0	0
0	0	1	0	0
0	0	1	0	0
0	0	0	1	0
0	0	0	0	1

(c) Índice bitmap para o atributo
endereço.

Figura 2.11: Exemplo de um índice bitmap de junção.

Para consultar os exames realizados em um determinado hospital, é necessário apenas considerar as entradas do vetor de bits referente ao endereço desse hospital que sejam iguais a “1” e recuperar as respectivas tuplas na tabela de fatos. Por exemplo, para o endereço E1, são recuperadas as três primeiras tuplas da tabela de fatos, de acordo com o índice bitmap referente a E1. Portanto, o índice bitmap de junção evita a junção entre as tabelas *Hospital* e *Exame*, possibilitando um processamento da consulta rápido por meio de operações lógicas bit a bit.

2.6.6 Biblioteca FastBit

O FastBit (Wu, 2005) é uma biblioteca de código aberto escrita em C++ que oferece implementações para a construção e manipulação de índices bitmap. Sua principal vantagem é o fato de disponibilizar técnicas de compressão, codificação e *binning*, oferecendo meios para criação de índices bitmap eficientes.

Atualmente, o FastBit possui a implementação da técnica de compressão WAH, bem como permite a criação de índices multinível e multicomponente. Ademais, é possível a criação de índices utilizando como parâmetro a quantidade de bins desejada.

Como um software independente de SGBD, o FastBit pode ser facilmente integrado a qualquer aplicação, tanto por meio da utilização direta da biblioteca no código, quanto por meio de ferramentas já desenvolvidas para acessar as principais funcionalidades oferecidas pelo FastBit, como criação dos índices e consultas. Atualmente, o software está em sua versão 1.3.9, entretanto, nesta dissertação, foi utilizada a versão 1.3.8 por ser mais estável.

2.7 Considerações Finais

Neste capítulo foram discutidos vários aspectos relacionados a DWs, desde as características dos dados armazenados, passando pela modelagem, até as operações OLAP para extração de informações relevantes ao processo de tomada de decisão. Mais especificamente, foi visto que um DW consiste de um repositório projetado para armazenar um grande volume de dados oriundos de diferentes provedores de informação e que um ambiente de DWing consiste de uma coleção de tecnologias utilizada para auxiliar especialistas no processo de tomada de decisão estratégica a partir de dados integrados, orientados a assunto, históricos e não voláteis armazenados no DW.

Foi discutido que a modelagem dos dados em um DW é feita de acordo com o esquema estrela, o qual, em sistemas ROLAP, consiste de uma tabela de fatos e de tabelas de dimensão. Ademais, foi visto que os dados podem ser organizados em diferentes níveis de agregação, do mais agregado (nível superior) até o menos agregado (nível inferior) e que essa forma de organização possibilita diferentes tipos de consultas analíticas, tais como *drill-down* e *roll-up*. Além dessas consultas, foram exemplificadas as consultas do tipo *slice-and-dice* e *pivot*.

Outra discussão realizada neste capítulo refere-se ao conceito de índice bitmap, o qual consiste da abordagem mais promissora para indexação em DWs. Foi abordado desde a sua versão mais simples, passando pelas principais técnicas de criação desses índices, até a aplicação de índice bitmap em DWs. Foi discutido que a técnica de *binning* reduz o tamanho do índice bitmap e também o torna viável para a indexação de atributos de alta cardinalidade. Além disso, técnicas de codificação do tipo multicomponente e multinível podem ser usadas para aumentar o desempenho desses índices, sendo que existem abordagens de codificação que otimizam consultas envolvendo condições de igualdade ou de intervalo. Sobre a compressão, foi citada a importância de uma técnica que leve em consideração o processo de descompressão, o qual impacta no desempenho do índice, e que a técnica WAH apresenta bons resultados nesse sentido. Ademais, foi discutido como

índices bitmap são promissores no processamento de consultas OLAP em DWs, tais como o índice bitmap de junção.

Finalmente, o software FastBit foi descrito, o qual possui implementações para a construção de índices bitmap, considerando as melhores técnicas de codificação, compressão e *binning* existentes na literatura.

No próximo capítulo são discutidos os conceitos relacionados a DW de imagens e os desafios envolvidos na incorporação de dados complexos, como imagens, em um ambiente de DWing.

Data Warehouse de Imagens

Ambientes de DWing convencionais possuem como limitação o fato de não oferecerem suporte a consultas envolvendo dados multimídia, tais como imagem, vídeo ou áudio. DWs convencionais lidam apenas com dados estruturados, os quais possuem relação de ordem total, podendo ser ordenados e pesquisados por meio de operadores relacionais (ex.: menor, maior, igual, diferente). Oferecer suporte a dados multimídia confere ao SSD uma capacidade maior de análise. Por essa razão, pesquisas recentes têm estudado a incorporação desse tipo de dado ao contexto de DW (Annibal et al., 2010; Arigon et al., 2007; Chen et al., 2008; Jin et al., 2010; Wong et al., 2004). Nesta dissertação de mestrado o foco é em um ambiente de DWing de imagens.

Ambientes de DWing de imagens manipulam não somente dados estruturados, mas também imagens, que são dados não estruturados. Pelo fato desse tipo de dado não possuir relação de ordem total, uma abordagem consiste em se utilizar técnicas de extração de características para viabilizar as comparações. Por meio dessas técnicas, o conteúdo intrínseco de uma imagem pode ser captado sob diferentes aspectos como cor, textura e forma. Sobre a representação numérica proporcionada por essas técnicas, a comparação de imagens torna-se possível por meio de consultas por similaridade. Nesta dissertação essa abordagem é utilizada, conforme discutido nas seções deste capítulo.

Esse capítulo é organizado da seguinte forma: Na seção 3.1 é feito um resumo dos principais trabalhos da literatura sobre ambientes de DWing multimídia e a definição de DW de imagens adotada nesta dissertação. Na seção 3.2 são descritos os principais conceitos relacionados a consultas por similaridade de imagens e à técnica Omni. Na seção 3.3 é descrito o conceito de camada perceptual. Na seção 3.4 é descrito o conceito de consulta IOLAP. Na seção 3.5, são descritos os principais componentes da arquitetura

de um ambiente de DWing de imagens, bem como um esquema estrela estendido genérico. As considerações finais são feitas na seção 3.6.

3.1 Introdução

Com relação especificamente ao armazenamento de imagens em ambientes de DWing, existem poucos trabalhos na literatura. Wong *et al.* (Wong et al., 2004) propõem um sistema de DW adaptado a dados multimídia que permite a realização de estudos clínicos e científicos sobre dados de pacientes. Arigon *et al.* (Arigon et al., 2007) propõem um modelo multidimensional capaz de lidar com dados multimídia representados por diferentes descritores. Chen *et al.* (Chen et al., 2008) propõem uma estrutura XML para organizar dados multimídia, especialmente vídeos, em um cubo de dados. Para isso, é descrito um esquema estrela particular de DW, bem como um algoritmo de consulta por abrangência. Jin *et al.* (Jin et al., 2010) descrevem um cubo visual e uma ferramenta OLAP multidimensional de coleções de imagens. Dessa forma, informações presentes em imagens *web*, indexadas por máquinas de busca ou fotos compartilhadas em redes sociais, podem ser analisadas estatisticamente.

Como pode ser observado, ainda não existe um consenso sobre a definição de um DW de imagens com relação ao esquema, dados e processamento de consultas OLAP. Neste trabalho, é utilizada a definição de DW de imagens proposta em (Annibal et al., 2010), a qual é descrita na Definição 1. O diferencial dessa definição com relação às demais é que ela é mais flexível, desde que considera a imagem como um dado intrínseco do DW.

Definição 1. *Um DW de imagens é projetado segundo um esquema estrela diferenciado que possui, além de tabelas de dimensão com dados convencionais, uma ou mais tabelas de dimensão específicas para a manipulação de dados sobre imagens.*

No trabalho de Annibal et. al (2010), é considerado que os dados de imagem que devem ser armazenados no DW são os vetores de características, bem como os dados necessários para o mecanismo de filtragem de imagens baseado na técnica Omni (seção 3.2.4). Por meio dessa abordagem, é possível a realização de consultas OLAP estendidas com predicados de similaridade de imagens.

3.2 Espaços Métricos e Consultas por Similaridade

Os dados que descrevem uma imagem são representados por vetores de características, que são *arrays* de dimensão n , onde n é o número de características obtidas. Para realizar

as consultas utilizando essas estruturas, é necessário definir uma função que meça o grau de semelhança entre esses vetores. Esse tipo de consulta, baseada em dados “parecidos” e não “iguais”, é conhecida como consulta por similaridade (Traina-Jr et al., 2007).

Relacionado a espaços métricos está o conceito de métodos de acesso métricos (MAMs). MAMs são índices ou técnicas de indexação projetados para atuarem como um caminho otimizado aos dados, evitando a análise exaustiva durante a recuperação dos elementos que satisfazem a uma consulta. Um método de acesso é definido por sua estrutura de dados, bem como por algoritmos de busca e de alteração. A pesquisa existente na literatura sobre MAM é bastante extensa, sendo que exemplos incluem a VP-tree (Yianilos, 1993); MVP-tree (Bozkaya and Ozsoyoglu, 1999); GNAT (Brin, 1995); M-tree e variações (Ciaccia et al., 1997; Skopal and Hoksza, 2007; Skopal and Lokoc, 2008; Zhou et al., 2003, 2005); Slim-tree e variações (Pola et al., 2009; Traina-Jr et al., 2002; Vieira et al., 2004); Onion-tree (Carélo et al., 2009, 2011) e Técnica Omni (Traina-Jr et al., 2007).

Na seção 3.2.1 são introduzidos os conceitos de espaços métricos e vetoriais. Na seção 3.2.2 são listadas as funções de distância mais utilizadas para realização de consultas por similaridade. Na seção 3.2.3 são explicadas algumas técnicas de extração de características de imagens. Na seção 3.2.4 é abordada a técnica Omni, a qual é utilizada na proposta desta dissertação de mestrado.

3.2.1 Espaços Métricos e Vetoriais

Na literatura, pelo menos duas técnicas têm sido estudadas para a modelagem dos vetores de características, o modelo de espaço métrico e o modelo de espaço vetorial (Traina-Jr et al., 2007). Um espaço métrico é um par ordenado $(\mathcal{U}, d)$, onde $\mathcal{U}$ é um conjunto de objetos e $d : \mathcal{U} \times \mathcal{U} \rightarrow \mathbb{R}^+$ é uma métrica, isto é, uma medida de distância que obedece às propriedades de identidade ($\forall x \in \mathcal{U}, d(x, x) = 0$); simetria ($\forall x, y \in \mathcal{U}, d(x, y) = d(y, x)$); não-negatividade ($\forall x, y \in \mathcal{U}, d(x, y) \geq 0$); e desigualdade triangular ($\forall x, y, z \in \mathcal{U}, d(x, y) \leq d(x, z) + d(z, y)$) (Ciaccia and Patella, 2002).

Diferente do espaço métrico, que não exige que os objetos possuam tamanho fixo, os objetos pertencentes a um espaço vetorial devem possuir dimensão fixa. Dessa forma, um vetor de características representado nesse modelo pode ser visto como um ponto em um espaço n -dimensional, onde n é o tamanho do vetor (Traina-Jr et al., 2007).

Espaços métricos são adequados para representar certos tipos de objetos como impressões digitais, pois o número de características que podem ser extraídas (deltas, terminações, etc.) variam entre as pessoas, e, portanto, não seria possível mapear um objeto desse domínio em um ponto no espaço n -dimensional (Traina-Jr et al., 2007). Já espa-

ços vetoriais são adequados para representar objetos com número fixo de características, como um histograma de níveis de cinza extraído de uma imagem, cuja dimensionalidade é o número de variações da escala de cinza.

As técnicas e algoritmos voltados à realização de consultas por similaridade descritos nesta dissertação de mestrado focam em operações sobre o modelo de espaço métrico, pois ele é menos restritivo e, munido de funções de distância adequadas, ainda inclui o espaço vetorial (Traina-Jr et al., 2007).

Em espaços métricos, há basicamente dois tipos de consultas de interesse: *range query* e *K-nearest neighbor query* (Chávez et al., 1999), as quais são ilustradas na Figura 3.1. Assim, sendo $q \in \mathcal{U}$ o elemento de consulta e r um número real que indica um raio de tolerância a partir de q (centro de consulta), esses tipos de busca são definidos da seguinte forma:

- **Range query** $R(q, r)$: Consulta por abrangência, a qual retorna todos os elementos que estão dentro da distância de r a q , isto é, $\{u \in \mathcal{U} \mid d(q, u) \leq r\}$.
- **K-nearest neighbor query** $Knn(q)$: Busca pelos vizinhos mais próximos, a qual retorna os k elementos que estão mais perto de q em $\mathcal{U}$, isto é, um conjunto $A \subseteq \mathcal{U}$, tal que $|A| = k$ e $\forall u \in A, v \in \mathcal{U} - A, d(q, u) \leq d(q, v)$. Sendo que em casos de empate, qualquer conjunto de k elementos que satisfaça a condição é aceitável.

Figura 3.1: Exemplos de consultas em espaços métricos. (a) Consulta por abrangência e (b) consulta pelos 4 vizinhos mais próximos. Adaptado de (Traina-Jr et al., 2007).

3.2.2 Funções de Distância

As funções de distância expressam numericamente o grau de dissimilaridade entre dois objetos. Dessa forma, a similaridade é inversamente proporcional ao resultado dessa função, isto é, quanto menor a distância obtida, maior a similaridade. Apesar de existirem diversas funções de distância para vários tipos de aplicações presentes na literatura, as distâncias da família Minkowski são as mais utilizadas (Wilson and Martinez, 1997). Conhecidas também como norma L_p , essa família de distâncias é válida para espaços vetoriais, nos quais os objetos são definidos por meio de tuplas $\{x_1, x_2, \dots, x_n\}$, onde n é o tamanho do vetor (quantidade de características). E essa família de distâncias é definida pela Equação 3.2.1

$$L_p((x_1, \dots, x_n), (y_1, \dots, y_n)) = \sqrt[p]{\sum_{i=1}^n |x_i - y_i|^p} \quad (3.2.1)$$

Alguns casos particulares desse grupo de funções podem ser obtidos a partir de variações do parâmetro p . Dessa forma, para $p = 2$, tem-se a distância L_2 , conhecida como distância Euclidiana, e para $p = 1$, tem-se a distância L_1 , também chamada de *City Block* ou distância Manhattan. Ao se calcular o limite da função L_p quando p tende ao infinito, obtém-se a função L_∞ , chamada Infinity ou Chebychev. Nesta dissertação de mestrado utiliza-se a distância Euclidiana.

Cada função de distância da família Minkowski gera, ao redor de um elemento de consulta q , uma forma geométrica diferente formada pelos pontos equidistantes a um dado raio de abrangência r_q . Assim, em uma *range query*, diferentes resultados podem ser obtidos dependendo da função de distância adotada. Na Figura 3.2, é ilustrada a representação espacial das formas geométricas geradas pelas distâncias L_1 , L_2 e L_∞ no espaço bidimensional. Pela figura, é possível notar que, de acordo com a distância escolhida, os elementos contidos no raio de abrangência r_q podem mudar.

Figura 3.2: Exemplos das formas geométricas geradas pelas distâncias L_p , considerando um espaço bidimensional. (a) Distância *Manhattan* (L_1), (b) Distância Euclidiana (L_2) e (c) Distância *Chebychev* (L_∞).

3.2.3 Extração de Características

Os principais desafios na implementação de sistemas de recuperação de imagens por conteúdo (*content-based image retrieval*, CBIR) são encontrar as representações do conteúdo intrínseco das imagens mais apropriadas e definir uma forma de calcular a similaridade entre duas imagens baseada nessa representação abstrata (Datta et al., 2008).

Nas áreas de processamento de imagens e visão computacional são estudados possíveis descritores de imagens, os quais representam as imagens com base principalmente nas propriedades de cor, textura e forma. Os algoritmos que implementam esses descritores, ou seja, realizam o processo de extração de características, são chamados de extratores. A seguir, são exemplificados alguns descritores de imagens.

- **Cor:** Dentre as técnicas existentes, o histograma de cor é o mais simples e o mais utilizado. Ele representa a distribuição das cores ou níveis de cinza presentes em uma imagem. Por exemplo, para uma imagem em tons de cinza, um histograma h consiste de um vetor de tamanho 255, onde cada posição i representa um nível de cinza, em que a posição $h[i]$ armazena o número de vezes que a intensidade de *pixel* i ocorre na imagem. As vantagens do uso de histogramas são sua facilidade de implementação, rapidez e a possibilidade de extração de várias informações probabilísticas, as quais também podem ser utilizadas como características da imagem. As desvantagens são que histogramas clássicos não captam informações espaciais dos *pixels* e um mesmo histograma pode ser gerado para imagens diferentes, embora cada imagem gere apenas um histograma.

- **Textura:** Não há uma definição formal do conceito de textura, mas pode-se dizer que por essa propriedade é possível identificar padrões repetitivos sobre a superfície da imagem, medindo propriedades tais como aspereza, suavidade e regularidade (Gonzalez and Woods, 2006). Em processamento de imagens, um meio para descrever essas propriedades é a abordagem estatística, sendo que nessa categoria os descritores de Haralick (Haralick, 1979) obtêm ótimos resultados. Os descritores de Haralick são obtidos por meio de matrizes de coocorrência da imagem, as quais representam as relações entre a posição dos *pixels* e seus valores. Dessa forma, uma matriz de coocorrência G é composta por valores g_{ij} que indicam o número de vezes que *pixels* de intensidade i e j aparecem na imagem de acordo com uma posição relativa pré-determinada, isto é, a partir de uma direção, tipicamente $0^\circ, 45^\circ, 90^\circ, 135^\circ$, e da distância entre *pixels*. Na Tabela 3.1, são ilustrados os principais descritores de Haralick utilizados na literatura.

Descritor	Equação
Variância	$\sum_i \sum_j (i - j)^2 G(i, j)$
Entropia	$\sum_i \sum_j G(i, j) \log(G(i, j))$
Energia	$\sum_i \sum_j G^2(i, j)$
Homogeneidade	$\sum_i \sum_j \frac{G(i, j)}{(1 + i - j)}$
3º momento	$\sum_i \sum_j (i - j)^3 G(i, j)$
Inversão da variância	$\sum_i \sum_j \frac{G(i, j)}{(i - j)^2}$

Tabela 3.1: Principais descritores de Haralick (Haralick et al., 1973).

- **Forma:** Existem vários descritores de forma existentes na literatura, os quais são categorizados em descritores baseados em contorno e descritores baseados em região. Enquanto no primeiro apenas a borda dos objetos é explorada, no segundo todos os *pixels* dentro de uma região são levados em conta no descritor, sendo, portanto, mais confiáveis para formas que possuam contornos mais complexos (Amanatiadis et al., 2011; Kim and Kim, 2000).

3.2.4 Técnica Omni

Consultas por similaridade entre objetos complexos são geralmente muito caras, e portanto, a redução do número de comparações é uma questão essencial para esse tipo de aplicação. Assim, para tornar essas operações mais rápidas, MAMs têm sido utilizados de maneira muito promissora, pois proporcionam acesso eficiente aos dados, por meio da redução do espaço de busca e, por conseguinte, da diminuição dos cálculos de distância

necessários. Dessa forma, com o auxílio dos MAMs, as buscas são restringidas às porções do conjunto de dados nas quais os objetos armazenados têm maior probabilidade de serem similares a um elemento de consulta (Carélo et al., 2009).

A técnica Omni consiste de um mecanismo de filtragem baseado no uso de representantes globais do conjunto de dados (elementos representativos). Para cada consulta, esses elementos especificam uma região do espaço métrico na qual garantidamente os objetos mais similares ao objeto de consulta residem. Dessa forma, uma vez que a poda no espaço métrico foi realizada, os objetos pertencentes à região encontrada são considerados candidatos à resposta da consulta. Apesar do método garantir que não haverá falsos negativos, devido à propriedade da desigualdade triangular, ele não assegura a ausência de falsos positivos (Traina-Jr et al., 2007). Portanto, apenas a definição da região não é suficiente, sendo necessária uma segunda etapa de refinamento.

De maneira mais detalhada, a principal ideia da técnica Omni consiste em selecionar um conjunto de h elementos como representantes globais e armazenar, para os demais objetos da base de dados, as respectivas distâncias a cada um dos representantes escolhidos. O conjunto dos representantes globais é descrito formalmente na Definição 2.

Definição 2. *Dado um espaço métrico $\mathbb{M} = \{\mathbb{S}, d()\}$, o conjunto $\mathcal{F}$ dos elementos representativos é definido por: $\mathcal{F} = \{f_1, f_2, \dots, f_h \mid f_g, f_h \in \mathbb{S}, f_g \neq f_j, \forall g \neq j\}$, onde cada f_g é um representante de $\mathbb{S}$ e h é o número de representantes.*

As consultas por abrangência realizadas utilizando essa técnica passam por duas etapas: etapa de filtragem e refinamento. A primeira consiste na seleção de objetos candidatos à resposta com o auxílio dos representantes globais. Esse processo é realizado por meio da especificação de uma *mbOr* (*minimum-bounding-Omni-region*). Dado um conjunto de representantes $\mathcal{F} = \{f_1, f_2, \dots, f_h\}$, $\mathcal{F} \subset \mathbb{S}$, a *mbOr* de um elemento de consulta $s_q \in \mathbb{S}$ com raio r_q , é definida por:

$$mbOr_{\mathcal{F}}(s_q, r_q) = \bigcap_{g=1}^h I_g \quad (3.2.2)$$

onde I_g é um subconjunto de objetos $s_i \in S (S \subseteq \mathbb{S})$ que pertencem ao intervalo fechado $[I_g^{inf}, I_g^{sup}]$ formado pelo representante g , onde $I_g^{inf} = d(f_g, s_q) - r_q$, se $d(f_g, s_q) \geq r_q$ ou $I_g^{inf} = 0$, caso contrário; e $I_g^{sup} = d(f_g, s_q) + r_q$, sendo $d(a, b)$ a distância entre os objetos. Formalmente:

$$I_g = \{s_i \mid I_g^{inf} \leq d(f_g, s_i) \leq I_g^{sup}, \forall s_i \in \mathbb{S}\} \quad (3.2.3)$$

Na Figura 3.3 é ilustrada a definição de uma *mbOr* considerando um conjunto de dados para o qual dois elementos representativos foram selecionados, f_1 e f_2 . De acordo com a figura, dada uma imagem de consulta s_q e um raio de abrangência r_q , a distância de s_q até f_1 e f_2 é representada, respectivamente, por d_1 e d_2 . Como pode ser observado, cada elemento representativo gera um anel no espaço métrico que contorna o objeto de consulta. Dessa forma, a *mbOr* corresponde à intersecção dos anéis gerados por todos os elementos representativos, como ilustrado pela área hachurada.

Figura 3.3: Definição de uma *mbOr* (área hachurada) para uma consulta ao elemento s_q , com raio r_q e dois elementos representativos, f_1 e f_2 . Adaptado de (Traina-Jr et al., 2007).

A etapa de refinamento consiste no cálculo efetivo das distâncias entre o elemento de consulta e os objetos contidos na *mbOr* para determinar a resposta da busca. Nota-se que a distância real entre os candidatos e o objeto de consulta é realizada apenas nessa etapa, quando a maioria dos objetos da base de dados já foi descartada (Traina-Jr et al., 2007).

Dentre os fatores que influenciam na delimitação de uma boa *mbOr* estão a quantidade de representantes globais e o posicionamento deles no espaço métrico. Assim, um conjunto de elementos representantes é considerado adequado quando ele reduz a *mbOr* o máximo possível, diminuindo a quantidade de distâncias a serem calculadas na etapa de refinamento.

Na proposta da técnica Omni, foi mostrado que o número h de representantes pode ser obtido de acordo com a dimensionalidade intrínseca do conjunto de dados, que consiste da quantidade mínima de atributos necessária para representar os objetos no conjunto. Uma maneira de encontrar uma aproximação para a dimensionalidade intrínseca é por meio da correlação da dimensão fractal D_2 (Belussi and Faloutsos, 1995; Traina-Jr et al., 2000). Ademais, foi mostrado que um valor adequado para h é o próximo inteiro que contém a dimensão intrínseca, ou seja, $\lceil D_2 \rceil + 1$.

Na Figura 3.4 é ilustrada a influência do posicionamento dos elementos representativos na eficiência da técnica Omni. No caso da Figura 3.4a, os três elementos representativos, f_1 , f_2 e f_3 , estão dispostos nas bordas do conjunto de dados, originando uma *mbOr* muito próxima da região da consulta. Por outro lado, na Figura 3.4b os representantes globais estão muito próximos entre si, gerando uma *mbOr* maior e, portanto, mais cálculos de distância devem ser realizados na etapa de refinamento.

Figura 3.4: Influência do posicionamento dos representantes globais na determinação de uma *mbOr* ótima. Na Figura 3.4a, os elementos estão dispostos nas bordas do conjunto de dados. Já na Figura 3.4b, os representantes estão muito próximos entre si. Adaptado de (Traina-Jr et al., 2007)

Como a redução máxima da *mbOr* nem sempre pode ser atingida (Traina-Jr et al., 2007), uma heurística na determinação do melhor posicionamento dos representantes consiste em selecionar aqueles elementos do espaço que se encontram mais próximos das bordas do conjunto de dados, o que é feito pelo algoritmo *Hull of Foci* (HF) (Traina-Jr et al., 2007).

3.3 Camadas Perceptuais

Nas áreas de processamento de imagens e visão computacional são pesquisados diversos descritores que representam o conteúdo intrínseco das imagens baseados principalmente nas propriedades de cor, textura e forma (Gonzalez and Woods, 2006).

No contexto de sistemas CBIR, um dos maiores desafios é a redução do *gap* semântico em consultas por similaridade, ou seja, a diferença entre o resultado produzido pelo sistema e o que era esperado pelos usuários (Traina et al., 2009). Isso ocorre, pois diferentes usuários podem escolher condições distintas para determinar a similaridade em um

conjunto de dados complexo. Portanto, a precisão das consultas por similaridade está sujeita à percepção dos usuários. O *gap* semântico pode ser reduzido pelo uso de diferentes descritores de imagens nesses sistemas (Ponciano-Silva et al., 2009), provendo ao usuário diversas opções, de modo que as consultas satisfaçam melhor às suas intenções.

Dessa forma, a opção por um ou mais descritores de imagens depende da aplicação e da preferência do usuário que realiza a análise das imagens. Visando a ampliar as possibilidades de consultas que um DW de imagens pode oferecer, minimizando os efeitos do *gap* semântico, nesta dissertação é aplicado o conceito de camada perceptual (Annibal, 2011), conforme descrito na Definição 3.

Definição 3. *Uma camada perceptual engloba os dados relacionados à representação do conteúdo intrínseco de uma imagem segundo um determinado descritor de imagem. Cada camada perceptual possui seu próprio espaço métrico, formado pelos vetores de características das imagens e seu próprio conjunto de elementos representativos, definidos de acordo com a técnica Omni.*

Nesta dissertação de mestrado, os dados referentes a uma camada perceptual são os vetores de características das imagens e as distâncias das imagens aos elementos representativos de acordo com a técnica Omni. Considera-se, portanto, que um DW de imagens pode ter uma ou mais camadas perceptuais, dependendo da necessidade do usuário, o qual pode escolher qual ou quais extratores são mais adequados, oferecendo melhores resultados à sua consulta. Por exemplo, certo usuário pode estar interessado em analisar imagens semelhantes segundo os descritores de Haralick ou segundo o Histograma, ou mesmo as imagens similares de acordo com os dois descritores.

3.4 Consulta IOLAP

Nesta dissertação, consultas analíticas com predicados de similaridade entre imagens são denominadas consultas IOLAP (*Image On-line Analytical Processing*). Essas consultas são compostas pelos critérios de similaridade de imagens e podem apresentar critérios sobre os predicados convencionais. Mais especificamente, nesta dissertação de mestrado, consideram-se apenas consultas OLAP do tipo *slice-and-dice* e consultas por similaridade do tipo *range query*. Na Definição 4 é descrito o conceito de consulta IOLAP.

Definição 4. *Uma consulta IOLAP é composta, obrigatoriamente, por predicados de similaridade entre imagens, parâmetros relacionados à consulta por similaridade implementada e, opcionalmente, por predicados convencionais.*

Os predicados de similaridade entre imagens referem-se às camadas perceptuais envolvidas na consulta e são executados sobre as tabelas de dimensão projetadas para armazenar os dados das imagens no esquema estrela. Os parâmetros são a imagem de consulta e o raio de abrangência. Já os predicados convencionais são os critérios da consulta sobre as dimensões convencionais. O exemplo a seguir ilustra uma consulta IOLAP válida no contexto desta dissertação de mestrado.

“Quantas imagens de encéfalo são similares a uma determinada imagem de consulta de acordo com as camadas perceptuais de Haralick e Histograma, que estejam em um raio de abrangência de 30% do raio total do conjunto de dados e que, concomitantemente, sejam oriundas do Hospital das Clínicas de Ribeirão Preto, ocorreram entre os anos de 1992 e 1994 em pacientes do sexo masculino com idade entre 40 e 50 anos?”

Para consultas IOLAP que consideram mais do que uma camada perceptual, a abordagem seguida nesta dissertação é a de que apenas as imagens que satisfazem concomitantemente a todas as camadas perceptuais são as procuradas. Se a consulta envolve predicados convencionais, a resposta final é a interseção dos conjuntos de resposta parciais.

Formalmente, considera-se um DW de imagens representado pelo conjunto de dados $\mathcal{S}$, onde cada elemento de $\mathcal{S}$ possui seus respectivos vetores de características, para cada camada perceptual, e dados convencionais. Dada uma consulta IOLAP definida pela imagem de consulta s_q , raio de abrangência r_q , um conjunto $\mathcal{P}$, composto pelos k identificadores das camadas perceptuais a serem consideradas na consulta, bem como pelos predicados convencionais, o conjunto de resposta dessa consulta é descrito na Definição 5.

Definição 5. *O conjunto resposta de uma consulta IOLAP é formado da seguinte maneira: $\forall p \in \mathcal{P}$, recuperam-se k subconjuntos $S_{p_i} \subseteq \mathcal{S} | i \leq k$, compostos pelas imagens semelhantes à s_q e contidas no raio r_q , de acordo com a similaridade dada pela camada perceptual p_i . Portanto, a resposta parcial da consulta, segundo apenas os predicados de similaridade entre imagens, é definida por $\mathcal{N} = \bigcap_{i=1}^k S_{p_i}$. Dado o conjunto $\mathcal{C} \subseteq \mathcal{S}$, formado pelos identificadores das tuplas que satisfazem aos predicados convencionais, a resposta final da consulta IOLAP é $\mathcal{R} = \mathcal{N} \cap \mathcal{C}$.*

3.5 Arquitetura

Nessa seção são descritos os componentes da arquitetura do ambiente de DWing de imagens adotados nesta dissertação de mestrado, conforme ilustrado na Figura 3.5. Para

exemplificar seu funcionamento, considera-se um DW de imagens sobre exames médicos realizados sobre pacientes de diferentes faixas etárias. A arquitetura descrita nessa seção envolve desde o processo de ETL até o acesso à informação integrada pelos usuários (e.g. médicos especialistas). Enfoca-se no processo de ETL realizado sobre as imagens, visto que os procedimentos realizados sobre os dados convencionais fogem ao escopo desta dissertação.

Figura 3.5: Arquitetura de um ambiente de DWing de imagens.

Conforme ilustrado na Figura 3.5, os dados convencionais sobre a idade dos pacientes estão armazenadas em um banco de dados relacional, enquanto que dados pessoais dos pacientes estão originalmente em arquivos texto. Com relação às imagens, elas podem estar armazenadas em um repositório específico. Os dados oriundos desses provedores de informação heterogêneos passam pela etapa de ETL, a qual realiza a integração desses dados em um DW projetado de acordo com um esquema estrela. Nessa figura, é ilustrado um esquema com duas dimensões convencionais, idade e paciente. As imagens médicas representam as tabelas de dimensão de imagens. Os dados integrados passam a ser disponíveis aos usuários para que eles possam realizar suas consultas analíticas.

Em um DW de imagens, a fase de ETL é estendida de modo que ela ofereça suporte ao processo de extração de características das imagens. Todo o processamento realizado sobre as imagens durante a fase de ETL denomina-se, nesta dissertação de mestrado, ETL de imagens, o qual é detalhado na Figura 3.6

Figura 3.6: Processo de armazenamento das imagens no DW de imagens. Adaptado de (Annibal et al., 2010).

A tarefa 1 consiste da extração de características da imagem de acordo com um ou mais extratores implementados nesse módulo. A escolha dos extratores implica na quantidade de camadas perceptuais adotadas para o DW de imagens. Os dados do conteúdo intrínseco das imagens são armazenadas em seus respectivos vetores de características. Na tarefa 2 os elementos representativos são identificados para cada camada perceptual, utilizando o algoritmo HF, conforme definido pela técnica Omni. Os vetores de características dos elementos representativos de cada camada perceptual são armazenados separadamente como metadados. Na tarefa 3 é realizado o cálculo das distâncias entre os vetores de características e os elementos representativos encontrados na etapa anterior. Por fim, o DW de imagens armazena os vetores de características das imagens e as distâncias das imagens até os elementos representativos, para cada camada perceptual.

A forma como as imagens são armazenadas no DW depende do esquema estrela estendido definido, o qual deve possuir uma ou mais dimensões específicas para os dados de imagens. Na Figura 3.7 é ilustrado um esquema estrela genérico de um DW de imagens, o qual possui uma tabela de fatos *Exame*, tabelas de dimensões convencionais *Hospital*, *Paciente*, *DataExame* e *Idade*, e uma nova tabela de dimensão chamada *Imagem*. Esse esquema apresenta apenas uma camada perceptual.

Figura 3.7: Esquema estrela estendido com a dimensão *Image*. Adaptado de (Annibal et al., 2010).

A dimensão *Imagem* é utilizada para fornecer suporte às consultas analíticas com predicados de similaridade entre imagens. Para isso, essa dimensão armazena os dados necessários para uma consulta por similaridade, tais como o vetor de características da imagem e distância aos elementos representativos $dist\_rep_1, \dots, dist\_rep_n$. Por exemplo, o atributo $dist\_rep_1$ armazena a distância dos vetores de características das imagens até o elemento representativo 1. Por meio da tabela de fatos, a imagem é relacionada aos demais dados convencionais do esquema. Essa tabela de fatos é caracterizada por ser uma tabela de fatos sem fatos, desde que ela contém como medida numérica um atributo artificial que sempre armazena o valor 1. Nessa modelagem, a medida numérica de interesse é representada pelo atributo *quantidade*.

Com relação ao processamento IOLAP, ele refere-se ao módulo responsável pela execução de consultas analíticas estendidas com predicados de similaridade entre imagens, o qual é o foco principal desta dissertação de mestrado. As consultas executadas por este módulo seguem a Definição 4. Os detalhes de como esse módulo é implementado nesta dissertação de mestrado são discutidos no Capítulo 5.

3.6 Considerações Finais

Neste capítulo foram discutidos os tópicos relacionados a consultas por similaridade, os quais são necessários para a compreensão do conceito de DW de imagens adotado nesta dissertação de mestrado. Mais especificamente, foram discutidos espaços métricos e vetoriais, funções de distância e como sua escolha influencia nos resultados obtidos, bem como consultas do tipo *range query* e *Knn*. Ademais, foram abordados alguns descritores de imagens utilizados para representar o conteúdo intrínseco das imagens para formar os vetores de características.

Foi detalhado o MAM da Família Omni, o qual consiste de um mecanismo de filtragem baseado no uso de representantes globais do conjunto de dados. Foi discutido como que esse MAM reduz a necessidade de cálculos de distâncias entre uma imagem de consulta e os elementos do conjunto de dados por meio da delimitação da *mbOr*. Essa região possui um conjunto de elementos candidatos à resposta, a qual passa por um processo de refinamento, gerando a resposta final da busca. Também foi discutido como que a quantidade e o posicionamento dos elementos representativos influenciam na qualidade da filtragem proporcionada pela técnica, e as heurísticas utilizadas para a determinação desses parâmetros. Esses tópicos iniciais serviram de base para a discussão da importância e dos desafios relacionados à incorporação de dados multimídia em ambientes de DWing.

Foi introduzido, especificamente, o conceito de DW de imagens e de ambientes de DWing de imagens. Um DW de imagens armazena, além de dados convencionais, dados relacionados à manipulação de imagens em uma ou mais dimensões do esquema estrela. Essa abordagem possui como diferencial o fato de considerar a imagem como um dado intrínseco do DW, diferentemente da tecnologia atualmente disponível para consultas analíticas envolvendo comparação de imagens, a qual considera o DW separado do banco de dados de imagens. Ambientes de DWing de imagens, por sua vez, referem-se ao conjunto de tecnologias empregadas para a manipulação de imagens e para a realização de consultas analíticas envolvendo predicados de similaridade entre imagens.

Os principais componentes da arquitetura de um ambiente de DWing de imagens foram detalhados, com destaque para o processo de ETL de imagens, o qual contempla a extração de características das imagens, identificação dos elementos representativos e cálculo das distâncias entre as imagens e esses elementos, de acordo com a técnica Omni. Esse processo é realizado para cada camada perceptual a ser criada. O conceito de camada perceptual foi abordado neste capítulo e refere-se ao conjunto de dados a ser armazenado no DW relacionado à representação do conteúdo intrínseco das imagens de acordo com um descritor de imagens. Por fim, foi abordado o conceito de consulta IOLAP, a qual

é caracterizada por envolver predicados de similaridade entre imagens e, opcionalmente, predicados convencionais.

No próximo capítulo são abordados os trabalhos correlatos a essa dissertação de mestrado, apresentando diferentes técnicas de indexação utilizando índice bitmap em um contexto de consultas por similaridade e uma técnica de processamento de consultas IOLAP em um DW de imagens. As limitações dos trabalhos correlatos também são discutidas.

Trabalhos Correlatos

Na literatura estudada existem diversos trabalhos que propõem índices para DWs e índices para o processamento de imagens, porém nenhum deles considera esses dois aspectos conjuntamente. Portanto, foram identificados como trabalhos correlatos a esta dissertação de mestrado, algumas propostas que envolvem adaptações de índices bitmap para o processamento de consultas por similaridade em bancos de dados de propósito geral e também uma proposta para o processamento de consultas em um DW de imagens.

Esse capítulo está organizado da seguinte maneira. Na seção 4.1 é abordada a parte do trabalho de Annibal et. al. (2010) que trata do processamento de consultas IOLAP em um DW de imagens. Nas seções 4.2, 4.3 e 4.4 são resumidos os seguintes trabalhos: proposta de um índice bitmap utilizando dimensões representativas dos objetos (Jeong and Nang, 2004), proposta do método HBI (Nang et al., 2010) e proposta do GB-index (Cha, 2004). Finalmente, na seção 4.5 são descritas as considerações finais do capítulo.

4.1 iCube: A Similarity-Based Data Cube for Medical Images

Em (Annibal et al., 2010) é proposto o iCube, um DW estendido que possui uma tabela de dimensão especialmente projetada para armazenar dados sobre imagens, como vetor de características e dados convencionais, tornando possível a realização de consultas OLAP com predicados de similaridade. Embora esse trabalho apresente um ambiente de DWing para o contexto médico, sua proposta é genérica o bastante para ser utilizada em outras aplicações de apoio à tomada de decisão.

A proposta do iCube mostra como os dados de imagens podem ser armazenados no DW considerando apenas uma camada perceptual, além de incluir uma forma de processar as consultas nesse ambiente em contraste com a tecnologia disponível, a qual trata os dados convencionais em um DW separadamente do banco de dados de imagens.

Como a definição de DW de imagens adotada nesta dissertação de mestrado segue a proposta do trabalho correlato em discussão, nessa seção o foco é apenas no processamento de consultas, visto que as principais definições já foram introduzidas no Capítulo 3. A estratégia de processamento de consultas apresentada nessa seção é utilizada como base na avaliação do desempenho das estratégias propostas nesta dissertação de mestrado. Essa discussão é feita no Capítulo 6.

As etapas realizadas no processamento de consultas IOLAP no iCube são ilustradas na Figura 4.1. Considerando o esquema estrela exemplificado na seção 3.5, tem-se a seguinte consulta IOLAP: “Quantas imagens são similares a uma dada imagem de mamografia, de acordo com o descritor Y e contidas em um raio de abrangência r_q , que ocorreram em pacientes do Estado de São Paulo, com idade entre 30 e 40 anos, nos anos de 1993 até 1994, e em hospitais da macro-região da grande São Paulo?”.

De acordo com esse trabalho correlato, a consulta é executada diretamente sobre o DW, sem o uso de estruturas de indexação. Primeiramente, na tarefa 1, o conjunto de dados é filtrado pelos predicados convencionais da consulta de modo a reduzir a quantidade de imagens que são submetidas à consulta por similaridade. Dessa forma, apenas as imagens que ocorreram de 1993 até 1994, em hospitais da Grande São Paulo, para pacientes do Estado de São Paulo com idade entre 30 e 40 anos, são selecionadas. Após essa primeira filtragem, o vetor de características da imagem de consulta é extraído na tarefa 2 e sua distância até os elementos representativos do conjunto de dados é calculada na tarefa 3.

Figura 4.1: Etapas do processamento de consultas no iCube. Adaptado de (Annibal et al., 2010).

Na tarefa 4, a consulta passa por um novo processo de filtragem, o qual é realizado por meio da definição da *mbOr*, isto é, a região que contém todas as imagens similares à imagem de consulta, identificada com o auxílio dos elementos representativos. Finalmente, na tarefa 5, após essa nova filtragem, os falsos positivos são eliminados calculando-se, efetivamente, as distâncias entre o elemento de consulta e as imagens retornadas, de acordo com o raio de consulta pré-definido. Na tarefa 6 é realizada a contagem dos identificadores das tuplas de resposta.

O trabalho de Annibal et. al. (2010) refere-se a um DW de imagens com apenas uma camada perceptual. Entretanto, de acordo com as Definições 3, 4 e 5 referentes, respectivamente, aos conceitos de camada perceptual, consulta IOLAP e geração do conjunto de resposta de uma consulta IOLAP, é possível estender as etapas do processamento de consultas descritas nesta seção, para contemplar a incorporação de n camadas perceptuais. Dessa forma, é possível comparar a estratégia de processamento de consultas deste trabalho correlato com as propostas desta dissertação de mestrado.

4.2 Índice Bitmap Baseado em Dimensões Representativas

Em (Jeong and Nang, 2004) é proposto um método de indexação baseado em bitmap para melhorar o desempenho de consultas por similaridade em bancos de dados de imagens. O mecanismo de indexação proposto consiste de uma técnica de filtragem que obtém

resultados significativos em consultas *Knn*, descartando objetos irrelevantes e reduzindo o número de cálculos de distância necessários.

De acordo com essa proposta, os vetores de características dos objetos são mapeados em vetores de bits, nos quais cada característica é representada por um bit que indica se aquele atributo é uma dimensão representativa do objeto, valor igual a “1”, ou não, valor igual a “0”. As dimensões representativas de um objeto são aquelas cujos valores são relativamente maiores do que os das outras dimensões. Dessa forma, cada objeto do conjunto de dados pode ser representado por um vetor de bits, onde cada bit indica se aquela é uma dimensão representativa ou não.

Em uma dada dimensão, se para um objeto ela é representativa e para outro não, considera-se que naquela dimensão os objetos estão distantes. Dessa forma, com esse mecanismo de indexação, uma aproximação do grau de dissimilaridade pode ser facilmente calculada aplicando-se a operação XOR entre os dois vetores de bits envolvidos e contando o número de “1”s no vetor binário resultante. Quanto maior a quantidade de “1”s após a operação XOR, maior a dissimilaridade entre os objetos comparados. Por outro lado, as operações que resultarem em vetores com baixa quantidade de “1”s indicam que os objetos são mais similares.

Considerando uma base de dados multimídia com n objetos O_k de dimensão m , onde $1 \leq k \leq n$, $O_k = \{h_k^1, h_k^2, h_k^i, \dots, h_k^m\}$ e h_k^i é o valor do atributo da i ésima dimensão referente ao k ésimo objeto, a geração do vetor binário é precedida por um processo de normalização dos vetores de características. Dessa forma, a determinação das dimensões representativas do objeto k é realizada sobre o vetor normalizado $N_k = \{n_k^1, n_k^2, n_k^i, \dots, n_k^m\}$, onde $n_k^i = h_k^i / q^i$, $\forall i, 1 \leq i \leq m$, sendo q^i o valor máximo do atributo da i ésima dimensão.

As dimensões representativas de O_k são obtidas com a definição do parâmetro T , o qual indica o valor ε_k que representa o T ésimo maior valor de atributo presente em N_k . A forma como se obtém o valor de T não é descrita na proposta desse índice. Dessa forma, as dimensões representativas de O_k são as que possuem valores maiores que ε_k em N_k . Portanto, o vetor bitmap B_k é gerado de acordo com a Equação 4.2.1.

$$B_k = b_k^1 \ b_k^2 \ b_k^3 \ \dots \ b_k^m, \begin{cases} b_k^i = 1, & \text{se } n_k^i \geq \varepsilon_k \\ b_k^i = 0, & \text{c.c} \end{cases} \quad (\forall i, 1 \leq i \leq m). \quad (4.2.1)$$

Na Figura 4.2 é exemplificado um índice bitmap de acordo com essa proposta, no qual a dimensionalidade do espaço de dados é 8 e o valor de T é 3. Nesse exemplo, o terceiro maior valor de atributo do objeto O_1 é igual a 0.7 (posição 5 do vetor de características).

Portanto, de acordo com a Equação 4.2.1, as dimensões 2, 4 e 5 são as representativas desse objeto, resultando no vetor bitmap $B_1 = "01011000"$. Para compor o índice bitmap da base de dados, os vetores de bits são calculados e armazenados para todos os objetos.

Figura 4.2: Índice bitmap para uma base de dados multimídia, com $T = 3$. Adaptado de (Jeong and Nang, 2004).

Em uma consulta do tipo Knn , primeiramente são extraídas as características e o vetor de bits do objeto de consulta, e em seguida o vetor bitmap é submetido a uma operação XOR com os vetores binários de todos os elementos da base de dados. Pelo fato de ser uma operação bit a bit, esse procedimento é realizado rapidamente. Após essa operação, os objetos que resultarem em vetores com a menor quantidade de “1”s, isto é, aqueles que apresentarem a menor quantidade de dimensões distantes do elemento de consulta, são considerados parte do conjunto de elementos candidatos à resposta. Após essa etapa de filtragem, as distâncias dos elementos candidatos ao objeto de consulta são calculadas de fato, sendo que apenas os k elementos mais próximos ao centro de consulta são retornados como resposta.

4.3 Hierarchical Bitmap Index (HBI)

Em (Nang et al., 2010) é proposto um mecanismo de indexação hierárquico baseado em filtragem para otimizar o processamento de consultas por similaridade no espaço n -dimensional. Por meio da proposta do *Hierarchical Bitmap Index (HBI)*, os objetos irrelevantes para uma consulta são filtrados do conjunto de dados, reduzindo a quantidade de cálculos de distância necessários para a busca.

Seja uma base de dados com n objetos representados no espaço d -dimensional, sendo $\Omega = [0, m]$ o intervalo de valores possíveis nesse espaço e a norma L_p a função de distância

utilizada, tal que $L_p(a, b) = (\sum_{j=1}^d |a^j - b^j|^p)^{1/p}$. A principal ideia do HBI é inspirada em uma característica da família L_p em que os componentes dominantes da função são os valores das dimensões em que os pontos estão mais afastados, ou seja, se a diferença entre dois objetos, o_p e o_q , em uma determinada dimensão j , $|o_p^j - o_q^j|$, é grande o bastante, então a distância L_p entre o_p e o_q é altamente dependente desse valor. As dimensões que satisfazem esse critério são chamadas de dimensões discriminativas. Nesse contexto, para a criação do índice, é definido um intervalo I que subdivide o espaço dos dados em três partições $[v_{min}, v_{low}]$, (v_{low}, v_{high}) , $[v_{high}, v_{max}]$, representados por I_{low} , I_{middle} , I_{high} , respectivamente, como ilustrado na Figura 4.3a, onde $v_{min} \leq v_{low} < v_{high} \leq v_{max}$. A escolha dos limiares, v_{low} e v_{high} , é feita baseada na distribuição dos valores dos atributos para todas as dimensões, visto que a definição de intervalos específicos para cada dimensão acarreta em maior gasto computacional. Na Figura 4.3b, é ilustrado um espaço bidimensional particionado dessa maneira.

Definido um intervalo I , de acordo com essa proposta, os vetores de características dos objetos da base são representados por vetores bitmaps, sendo cada atributo representado por um conjunto de 2 bits, indicando se o valor do atributo em determinada dimensão pertence ao intervalo I_{high} ('11'), I_{low} ('00'), ou nenhuma das duas opções ('01'). Por exemplo, seja $p = \langle 1, 8, 3, 9 \rangle$ e $\Omega = [1, 10]$, assumindo que $I_{low} = [1, 3]$, $I_{middle} = (3, 9)$, $I_{high} = [9, 10]$, então a representação binária de p em I é 00 01 00 11.

Figura 4.3: Particionamento do espaço de dados. Adaptado de (Nang et al., 2010).

Por meio desse particionamento e da representação binária dos vetores de características, é possível determinar uma aproximação da distância entre dois objetos. Se $o_p^j \in I_{low}$ e

$o_q^j \in I_{high}$, tem-se que a dimensão j é discriminativa entre o_p e o_q , e, portanto, a contribuição dessa dimensão na distância L_p pode ser aproximada a $|v_{high} - v_{low}|^p$. Assim, sendo C o número de dimensões discriminativas entre dois objetos, o qual pode ser obtido pela contagem do número de ‘11’s no resultado de uma operação XOR entre os vetores bitmap envolvidos, então a distância entre eles pode ser estimada em $(C \times |v_{high} - v_{low}|^p)^{1/p}$. Por exemplo, na Figura 4.3b, a distância L_2 estimada entre os objetos ‘a’ e ‘c’ é $(1 \times |v_{high} - v_{low}|^2)^{1/2}$, pois entre eles apenas a dimensão x é discriminativa. Por outro lado, a distância entre ‘a’ e ‘b’ é aproximadamente $(2 \times |v_{high} - v_{low}|^2)^{1/2}$, como destacado na figura pela seta tracejada.

O uso de apenas um intervalo pode acarretar na identificação de poucas dimensões discriminativas ($C \approx 0$), o que pode levar o valor da distância estimada a zero e, assim, reduzir a capacidade de filtragem do índice. Para resolver esse problema, o método utiliza uma hierarquia de intervalos de forma a viabilizar o cálculo aproximado das distâncias, que deve obrigatoriamente satisfazer à restrição das dimensões discriminativas. Para cada novo intervalo definido, é criado mais um vetor bitmap para os objetos, os quais são armazenados para compor o índice.

Considerando $I^1 = \Omega$ o intervalo inicial, há seis casos possíveis em que os valores de uma dimensão j de dois objetos, o_p^j e o_q^j , podem se enquadrar em um intervalo, como é mostrado na Tabela 4.1. De acordo com a Tabela 4.1, se os valores da dimensão j dos objetos se enquadram no caso 3, é possível estimar a distância entre eles, pois j seria uma dimensão discriminativa em I^1 . Para os demais casos, é necessária a formação de novos intervalos de maneira hierárquica. Assim, para lidar com os casos 1, 2 e 4, os intervalos I_{low}^1 e I_{middle}^1 são fundidos para compor o intervalo I^2 . Considera-se que a notação $I^1 \vee I^2$ significa a junção de dois intervalos adjacentes. Por exemplo, se $I^1 = [1, 5]$ e $I^2 = [5, 10]$, então $I^1 \vee I^2 = [1, 10]$. Por outro lado, para os casos 5 e 6, os intervalos I_{middle}^1 e I_{high}^1 também são fundidos, criando-se o intervalo I^3 . Seguindo essa lógica, uma estrutura de árvore semelhante à exemplificada na Figura 4.4 é gerada, a qual representa o particionamento do espaço em intervalos.

Casos	Intervalos			Tratamento
	I_{low}^1	I_{middle}^1	I_{high}^1	
1	o_p^j, o_q^j	-	-	$I^2 = I_{low}^1 \vee I_{middle}^1$
2	o_p^j	o_q^j	-	$I^2 = I_{low}^1 \vee I_{middle}^1$
3	o_p^j	-	o_q^j	<i>Dimensão discriminativa</i>
4	-	o_p^j, o_q^j	-	$I^2 = I_{low}^1 \vee I_{middle}^1$
5	-	o_p^j	o_q^j	$I^3 = I_{middle}^1 \vee I_{high}^1$
6	-	-	o_p^j, o_q^j	$I^3 = I_{middle}^1 \vee I_{high}^1$

Tabela 4.1: Casos possíveis em que os valores o_p^j e o_q^j podem se enquadrar no intervalo I^1 .

Figura 4.4: Árvore de particionamento em intervalos. Os intervalos com os mesmos limitantes estão representados com uma mesma textura. Adaptado de (Nang et al., 2010).

Com a criação da hierarquia de intervalos, a distância L_p aproximada entre dois objetos, $D(o_p, o_q)$, é calculada de acordo com a Equação 4.3.1, onde l é o número de bitmaps do índice e C_k é o número de dimensões discriminativas entre os objetos o_p e o_q no k ésimo bitmap.

$$D(o_p, o_q) = \left[\sum_{k=1}^l C_k \times \| I_{middle}^k \|^p \right]^{1/p} \quad (4.3.1)$$

Dado um elemento de consulta o_q , utilizando esse índice é possível responder a consultas por abrangência e do tipo *Knn*. No caso de uma consulta por abrangência com raio r , por exemplo, primeiramente é extraído o bitmap de o_q , em seguida são calculadas as distâncias aproximadas D_p entre o elemento de consulta e os objetos da base são calculadas. Aqueles objetos que apresentarem distância D_p maior que r são descartados e os demais são adicionados a um conjunto de elementos candidatos. Após esse processo, a distância

real L_p entre o centro de consulta e os elementos candidatos é calculada, retornando como resposta apenas os objetos cujas distâncias estejam dentro do raio r .

4.4 Grid Bitmap Index (GBI)

Em (Cha, 2004) é proposto o GB-index (*grid bitmap index*), um método de indexação para consultas por similaridade do tipo *Knn* em bases de dados multimídia. Esse método foca principalmente no problema da alta dimensionalidade do espaço de dados que, para algumas classes de funções de distância como a norma L_p , leva a uma perda de significado do conceito de proximidade (Beyer et al., 1999), o que afeta a eficiência dos métodos de acesso tradicionais. Dessa forma, o GB-index propõe uma função de similaridade na qual cada dimensão é tratada separadamente de modo que o aumento da dimensionalidade não impacte negativamente na eficiência do método de indexação.

A principal ideia do GB-index consiste na discretização dos valores de cada dimensão do vetor de características em K intervalos, onde cada um representa um *cluster* que contém uma fração dos dados da base. Dessa forma, dois pontos classificados em um mesmo intervalo são considerados similares naquela dimensão. Para essa finalidade, é possível utilizar qualquer método de agrupamento e aplicá-lo a cada dimensão, sendo que o algoritmo *K-means* (MacQueen, 1967) foi utilizado na proposta do método por ser uma técnica simples e eficiente.

Após o agrupamento dos objetos em cada uma das dimensões, para o intervalo j da dimensão i , $I_{i,j}$, é criado um bitmap $b_{i,j}$, o qual contém a informação binária que diz quais objetos da base possuem valores da dimensão i que pertencem ao intervalo $I_{i,j}$. Ademais, ao longo do bitmap $b_{i,j}$, são mantidas as coordenadas reais da dimensão i dos objetos que residem nesse intervalo.

De acordo com essa técnica, uma consulta *Knn* é transformada em uma consulta por abrangência, visto que os intervalos, nos quais residem os valores das dimensões de um objeto de consulta, automaticamente determinam uma região inicial na qual os objetos mais similares podem estar, sendo que, durante a busca, essa região pode ser estendida.

Exemplificando, considere um espaço bidimensional no qual as dimensões x e y foram divididas em 4 intervalos, como mostrado na Figura 4.5. Sendo o ponto de consulta identificado pelo sinal '+', os intervalos $I_{x,2}$ e $I_{y,1}$ determinam uma região inicial de busca para as dimensões x e y , respectivamente. Para a dimensão x , os objetos candidatos a similares são $S_x = \{b, d\}$, e para a dimensão y , os candidatos são $S_y = \{f, c, h\}$, visto que eles compartilham os mesmos intervalos que o objeto de consulta. Para obter os candidatos finais, é realizada a intersecção de S_x e S_y . Se o conjunto resultante obtiver

cardinalidade maior ou igual a k (quantidade de objetos a serem retornados), então esse conjunto é o candidato à resposta. Caso contrário, a região inicial é estendida para englobar os intervalos adjacentes a cada dimensão. Nesse exemplo, como a intersecção de S_x e S_y é menor do que k (ex.: $k = 3$), a busca é estendida para os intervalos $I_{x,1}, I_{x,2}, I_{x,3}$ na dimensão x e $I_{y,0}, I_{y,1}, I_{y,2}$ na dimensão y . Esse processo é repetido até que o conjunto final possua no mínimo k objetos.

Figura 4.5: Exemplo de uma consulta Knn . A região hachurada representa o conjunto final de elementos candidatos após a intersecção dos candidatos em x e em y . Adaptado de (Cha, 2004).

De modo a tratar cada dimensão independentemente, a Equação 4.4.1 foi proposta para calcular a distância entre dois objetos x e y na dimensão i que residem no mesmo intervalo j , onde $l_{i,j}$ e $u_{i,j}$ são os limites do intervalo.

$$D_i(x_i, y_i) = 1 \begin{cases} \frac{|x_i - y_i|}{u_{i,j} - l_{i,j}}, & \text{se } x_i \text{ e } y_i \text{ pertencem ao mesmo intervalo} \\ 0 & \text{c.c.} \end{cases} \quad (4.4.1)$$

O valor da distância total entre x e y é dado pela Função 4.4.2, onde d é a quantidade de dimensões e w_i é o peso dado pelo usuário à dimensão i .

$$D(x, y) = \left(\sum_{i=1}^d w_i D_i^p \right)^{1/p} \quad (4.4.2)$$

4.5 Considerações Finais

Neste capítulo foram descritos os principais trabalhos correlatos a essa dissertação de mestrado. Foi discutida a estratégia de processamento de consultas proposta para o iCube, detalhando como uma consulta analítica com predicados de similaridade entre imagens pode ser realizada diretamente sobre o DW de imagens. Entretanto, a proposta do iCube não contempla o uso de índices para melhorar o desempenho dessas consultas, o que motiva a proposta desta dissertação de mestrado.

Também foram abordadas diferentes técnicas de indexação baseadas em bitmap para consultas por similaridade em bancos de dados multimídia. Na Tabela 4.2 são resumidas as principais características das técnicas de indexação abordadas. A técnica descrita na seção 4.2 é identificada como IBBDR (Índice Bitmap Baseado em Dimensões Representativas) para facilitar a discussão.

Nome da proposta	Princípio	Técnica de filtragem	Construção do índice
IBBDR	Dimensões representativas dos objetos	Operação XOR para compor o conjunto de elementos candidatos	Normalização dos vetores de características e determinação do parâmetro T
HBI	Divisão dos dados em intervalos e identificação de dimensões discriminativas	Cálculo da distância aproximada para compor o conjunto de elementos candidatos	Especificação do intervalo inicial e posterior subdivisão hierárquica desses intervalos
GBI	Discretização dos dados em K intervalos para cada dimensão e uso de uma função de distância que trata as dimensões individualmente	Operação AND entre os vetores bitmaps dos intervalos de cada dimensão onde o elemento de consulta reside	Aplicação de um classificador para cada dimensão e construção de um vetor bitmap para cada intervalo encontrado

Tabela 4.2: Resumo das principais características das técnicas de indexação baseadas em bitmap para consultas por similaridade.

Apesar das vantagens introduzidas pelos métodos de indexação discutidos neste capítulo, eles apresentam limitações. Com relação ao IBDDR, sua acurácia e eficiência é altamente dependente do mecanismo de normalização para a escolha das dimensões representativas dos objetos, uma vez que uma escolha ruim pode impactar na capacidade de poda do índice. Os autores dessa proposta não apresentam um método específico para a determinação do número de dimensões que melhor representa os objetos indexados. Ademais, essa técnica possui como limitação o fato de ser voltada apenas a consultas do tipo *Knn* e ao fato de oferecer uma resposta não exata à consulta. Quanto à proposta do HBI, a escolha dos limiares da hierarquia de intervalos é feita levando-se em conta a distribuição dos valores de todas as dimensões. Apesar dessa opção de implementação ser motivada por razões de eficiência computacional, evitando a criação de hierarquias de intervalos específicas para cada dimensão, é possível que as diferenças de domínio existentes entre as dimensões impactem no tamanho do índice. Dessa forma, um determinado atributo, cujo domínio seja muito pequeno, pode demandar a criação de muitos intervalos para viabilizar o cálculo aproximado das distâncias, o que aumenta a complexidade de espaço do índice.

Uma limitação em comum às propostas do IBDDR e HBI, dificultando a aplicação dessas técnicas em ambientes de DW, é que seriam necessárias adaptações estruturais para que elas possam ser utilizadas juntamente com índices bitmap sobre dados convencionais. Por exemplo, em índices bitmap de junção cada bit representa uma tupla da base de dados, enquanto que nessas propostas os bits representam as dimensões do vetor de características. Finalmente, com relação ao GB-index, sua principal limitação refere-se à complexa etapa de construção do índice, devido à necessidade de um classificador aplicado a todas as dimensões da base. Além disso, a diferença no comportamento dos dados de cada dimensão pode levar a uma classificação ruim dependendo da escolha do algoritmo de *clustering*.

Além das limitações individuais descritas anteriormente, nenhum desses trabalhos explora as características específicas de um ambiente de DWing, tais como a organização diferenciada dos dados (multidimensional) e o grande volume de dados a ser manipulado. Portanto, as limitações existentes nos trabalhos correlatos evidenciam a necessidade do desenvolvimento de uma estratégia de indexação específica para aplicações de DWing que seja voltada ao processamento eficiente de consultas OLAP com predicados de similaridade entre imagens, que é o resultado desta dissertação de mestrado.

Processamento de Consultas IOLAP

Neste capítulo são apresentadas as propostas desta dissertação de mestrado para o processamento de consultas analíticas com predicados de similaridade entre imagens em ambientes de DWing.

São introduzidas as seguintes contribuições principais:

- Proposta do ImageDW-index, um mecanismo para a otimização de consultas analíticas estendidas com predicados de similaridade entre imagens, o qual é detalhado na seção 5.1.
- Definição de diferentes estratégias de processamento de consultas usando o ImageDW-index. São definidas quatro estratégias, as quais são apresentadas na seção 5.2.

Para validar as soluções propostas, foram introduzidas as seguintes contribuições secundárias:

- Desenvolvimento do ImageDW-Gen, um gerador de dados com o objetivo de povoar o DW de imagens. O ImageDW-Gen é detalhado na seção 5.3.
- Especificação de diferentes classes de consultas, as quais enfocam em diferentes custos do processamento de predicados de similaridade em ambientes de DWing de imagens. As quatro classes propostas são definidas na seção 5.4.

Após a descrição das contribuições, o capítulo é finalizado na seção 5.5, com as considerações finais.

5.1 Proposta do ImageDW-index

O ImageDW-index é um mecanismo para a otimização de consultas que utiliza as vantagens introduzidas por duas técnicas já existentes e bastante eficientes: a técnica Omni e o índice bitmap. A técnica Omni proporciona mecanismos de filtragem e de refinamento do conjunto de dados de imagens em consultas por similaridade, enquanto que índices bitmap são amplamente usados em aplicações de DWing convencionais, por apresentarem melhor desempenho em grandes volumes dados, visto que as respostas são obtidas por meio de operações lógicas bit a bit.

Antes de detalhar o funcionamento do ImageDw-index na seção 5.1.2, é importante destacar que, para que o ImageDW-index seja utilizado, o ambiente de DWing de imagens deve seguir algumas diretrizes, as quais são descritas na seção 5.1.1.

5.1.1 Diretrizes para o Ambiente de DWing de Imagens

No Capítulo 3 foram introduzidas definições importantes relacionadas a DW de imagens, camadas perceptuais e consultas IOLAP. Com base naquelas definições, as diretrizes a seguir têm o objetivo de sumarizar os requisitos de dados necessários para que o ImageDW-index possa ser utilizado em ambientes de DWing de imagens.

- O esquema estrela do DW de imagens deve possuir tabelas de dimensão que armazenem os seguintes dados de imagem como atributos: vetor de características das imagens, segundo um ou mais extratores de características, e as distâncias dos vetores de características das imagens até os elementos representativos, segundo a técnica Omni.
- Os seguintes dados de imagem devem ser armazenados como metadados no DW de imagem: quantidade de camadas perceptuais, vetores de características dos elementos representativos e sua dimensionalidade, e diâmetro dos conjuntos de dados formados pelos vetores de características das camadas perceptuais. O valor do diâmetro é utilizado em uma consulta para determinar o valor do raio de abrangência, o qual é dado em termos da porcentagem do raio do conjunto.

O esquema ilustrado na Figura 5.9, discutido na seção 5.3.2, e usado como base nesta dissertação de mestrado, segue a primeira diretriz. A segunda diretriz é seguida na implementação do ambiente de DWing de imagens adotada nesta dissertação para a realização das consultas IOLAP.

5.1.2 Detalhamento do ImageDW-index

De acordo com a proposta do ImageDW-index, o índice bitmap é utilizado para a indexação das distâncias entre os vetores de características das imagens e os elementos representativos para cada camada perceptual. Em uma consulta utilizando essa técnica, um conjunto de elementos candidatos à resposta é gerado em uma etapa de filtragem, por meio de operações lógicas AND e OR sobre o índice. Essas operações lógicas referem-se à formação da *mbOr* de acordo com a definição da técnica Omni, a qual é responsável por uma grande poda no conjunto de dados original. Posteriormente, na etapa de refinamento, os elementos candidatos à resposta têm suas distâncias efetivamente calculadas até a imagem de consulta original, removendo assim os falsos positivos.

Uma ilustração da estrutura do ImageDW-index para um DW de imagens com N camadas perceptuais é feita na Figura 5.1. Essa figura mostra uma estrutura genérica do ImageDW-index para um ambiente de DWing de imagens desenvolvido de acordo com as diretrizes apresentadas na seção 5.1.1. Dessa forma, utilizando os dados contidos no DW de imagens e no repositório de metadados, o índice bitmap é construído.

ImageDW-Index						
Camadas Perceptuais						
PL0			...	PLN		
Distância aos elementos representativos						
PL0DistRep_1	PL0DistRep_2	PL0DistRep_3	...	PLNDistRep_1	...	PLNDistRep_K
Índice Bitmap						

Figura 5.1: Estrutura genérica do ImageDW-index, a qual representa o índice bitmap construído sobre os atributos de distância entre as imagens armazenadas e os elementos representativos, segundo a técnica Omni, para cada camada perceptual.

Conforme a Figura 5.1, para as camadas perceptuais de *PL0* até *PLN*, as distâncias das imagens aos elementos representativos são indexadas utilizando índice bitmap. Por exemplo, considerando a camada perceptual *PL0*, que possui três elementos representativos, as distâncias referentes a ela, representadas por *PL0DistRep_1*, *PL0DistRep_2* e *PL0DistRep_3*, são indexadas em uma única estrutura de índice bitmap, onde as demais distâncias relativas às outras camadas perceptuais também são indexadas.

Para exemplificar a estrutura e funcionamento do ImageDW-index, considera-se um DW de imagens com duas camadas perceptuais, *PL0* e *PL1*, as quais são definidas pelos espaços métricos ilustrados na Figura 5.2. A primeira camada possui três elementos

representativos (Rep_1 , Rep_2 e Rep_3), enquanto que a segunda possui dois elementos representativos (Rep_1 e Rep_2). Ressalta-se que ambas as camadas perceptuais são constituídas sobre o mesmo DW de imagens, sendo que cada camada representa os dados intrínsecos das imagens segundo extratores de características diferentes.

Figura 5.2: Ilustração de duas camadas perceptuais de um DW de imagens em seus espaços métricos e com seus respectivos elementos representativos, os quais são definidos de acordo com a técnica Omni. Em (a) é representado $PL0$, com três elementos representativos. Em (b) é ilustrado $PL1$ com dois elementos representativos.

Para facilitar a compreensão, o ImageDW-index é representado a seguir para as camadas perceptuais separadamente, mas, na realidade, o índice é construído considerando todos os atributos indexados em uma mesma estrutura. Na Figura 5.3 é ilustrado o ImageDW-index para uma amostra do DW de imagens considerando $PL0$, já na Figura 5.4, o mesmo é feito para a camada $PL1$. Em ambas as figuras, a parte em cinza representa apenas informações adicionais para melhorar a qualidade do exemplo. Nesse exemplo, considera-se que o índice bitmap é gerado utilizando a técnica de *binning*, ou seja, são criados vetores de bits para agrupamentos de dados representados por intervalos de valores. São criados três bins que dividem o intervalo $[0,10]$, o qual representa os possíveis valores de distância desse cenário. Por exemplo, na Figura 5.3, a imagem identificada por 2 está a uma distância contida no intervalo de $[8-10]$ do elemento representativo Rep_1 (bin #3), conforme o índice do atributo $PL0DistRep_1$.

IDs	Camada perceptual PL0								
	Elementos representativos								
	PL0DistRep_1			PL0DistRep_2			PL0DistRep_3		
	Bitmap bins - Rep. 1			Bitmap bins - Rep. 2			Bitmap bins - Rep. 3		
	#1 [0 - 3]	#2 [4 - 7]	#3 [8 - 10]	#4 [0 - 3]	#5 [4 - 7]	#6 [8 - 10]	#7 [0 - 3]	#8 [4 - 7]	#9 [8 - 10]
0	1	0	0	0	0	1	1	0	0
1	0	1	0	0	1	0	1	0	0
2	0	0	1	1	0	0	0	1	0

Figura 5.3: ImageDW-index para uma amostra dos dados do DW de imagens, considerando a camada perceptual *PL0*. Para cada imagem identificada por um *ID*, o índice bitmap é ilustrado.

IDs	Camada perceptual PL1					
	Elementos representativos					
	PL1DistRep_1			PL1DistRep_2		
	Bitmap bins - Rep. 1			Bitmap bins - Rep. 2		
	#1 [0 - 3]	#2 [4 - 7]	#3 [8 - 10]	#4 [0 - 3]	#5 [4 - 7]	#6 [8 - 10]
0	1	0	0	0	0	1
1	1	0	0	0	0	1
2	0	1	0	1	0	0

Figura 5.4: ImageDW-index para uma amostra dos dados do DW de imagens, considerando a camada perceptual *PL1*. Para cada imagem identificada por um *ID*, o índice bitmap é ilustrado.

Para exemplificar o funcionamento do ImageDW-index, utilizando o DW de imagens e o índice de exemplo, considera-se uma imagem de consulta img_q e um raio de abrangência $r_q = 2$ válido para ambas as camadas perceptuais. Os vetores de características de img_q , segundo os descritores de imagens definidos pelas camadas perceptuais *PL0* e *PL1*, são dados por $fv_{PL0}(img_q)$ e $fv_{PL1}(img_q)$. Os vetores de características dos elementos representativos de *PL0* são dados por $fv_{PL0}(Rep_1)$, $fv_{PL0}(Rep_2)$ e $fv_{PL0}(Rep_3)$. Os vetores de características dos elementos representativos de *PL1* são dados por $fv_{PL1}(Rep_1)$ e $fv_{PL1}(Rep_2)$.

As distâncias entre a imagem de consulta e os elementos representativos de *PL0* são informadas na Tabela 5.1, enquanto que as distâncias referentes à camada perceptual

$PL1$ estão na Tabela 5.2. Esses valores de distâncias são utilizados para formação dos intervalos em torno de cada elemento representativo, de acordo com a técnica Omni, os quais também são informados nas tabelas. Os intervalos possuem a seguinte forma geral $[d(fv(img_q), fv(Rep)) - r_q, d(fv(img_q), fv(Rep)) + r_q]$.

Distâncias	Intervalos
$d(fv_{PL0}(img_q), fv_{PL0}(Rep_1)) = 3$	$I_{PL0Rep_1} = [1, 5]$
$d(fv_{PL0}(img_q), fv_{PL0}(Rep_2)) = 5$	$I_{PL0Rep_2} = [3, 7]$
$d(fv_{PL0}(img_q), fv_{PL0}(Rep_3)) = 3$	$I_{PL0Rep_3} = [1, 5]$

Tabela 5.1: Distâncias entre a imagem de consulta img_q e os elementos representativos, bem como os intervalos obtidos para a camada perceptual $PL0$, considerando um raio $r_q = 2$

Distâncias	Intervalos
$d(fv_{PL1}(img_q), fv_{PL1}(Rep_1)) = 3$	$I_{PL1Rep_1} = [1, 5]$
$d(fv_{PL1}(img_q), fv_{PL1}(Rep_2)) = 8$	$I_{PL1Rep_2} = [6, 10]$

Tabela 5.2: Distâncias entre a imagem de consulta img_q e os elementos representativos, bem como os intervalos obtidos para a camada perceptual $PL1$, considerando um raio $r_q = 2$.

As imagens que fazem parte do conjunto de elementos candidatos são encontradas, primeiramente, pela operação OR entre os bins do índice bitmap que contém os intervalos obtidos pela consulta, para cada elemento representativo. Por exemplo, de acordo com a Figura 5.3, para o elemento representativo 1 na camada perceptual $PL0$ ($PL0DistRep_1$), os bins #1 e #2 contém o intervalo I_{PL0Rep_1} . Dessa forma, a operação OR entre eles resulta na cadeia de bits “110”, significando que as entradas com ID 0 e 1 são o resultado dessa operação. Seguindo a mesma lógica em $PL0$, o resultado para o representativo 2 é “011”, e, para o representativo 3, o resultado da operação OR é “111”. O conjunto de elementos candidatos para essa camada perceptual é gerado por meio da operação AND entre os resultados parciais, o que resulta na cadeia de bits “010”, significando que apenas a imagem com ID=1 faz parte desse conjunto.

Aplicando a mesma lógica na camada perceptual $PL1$, segundo a Figura 5.4, o resultado da operação OR para o elemento representativo 1 é igual a “111” e o resultado para o elemento representativo 2 é “110”. Finalmente, após uma operação AND, a cadeia de bits “110” é obtida, significando que as imagens com ID 0 e 1 fazem parte do conjunto de elementos candidatos para essa camada perceptual.

Como a consulta foi realizada considerando as duas camadas perceptuais, o conjunto final de elementos candidatos é gerado por meio da intersecção dos resultados parciais obtidos. Dessa forma, $010 \text{ AND } 110 = 010$, ou seja, apenas a imagem com ID 1 é candidata à resposta da consulta. Na etapa de refinamento, considerando que $d(fv_{PL0}(img_q), fv_{PL0}(img_{ID=1})) < r_q$ e que $d(fv_{PL1}(img_q), fv_{PL1}(img_{ID=1})) < r_q$, ou seja, que as distâncias dos vetores de características da imagem 1 até os vetores de características da imagem de consulta sejam menores do que o raio de abrangência ($r_q = 2$), para ambas as camadas perceptuais, essa imagem é a resposta final dessa consulta IOLAP.

No exemplo anterior o processo foi detalhado apenas para fins didáticos, entretanto, os elementos candidatos das camadas perceptuais envolvidas na consulta são encontrados acessando-se o índice uma só vez, ou seja, apenas um conjunto de elementos candidatos é recuperado, o qual possui as imagens pertencentes a todas as *mbOrs* envolvidas. Em outras palavras, não é necessário calcular os elementos candidatos para cada camada perceptual individualmente, visto que a própria estrutura do índice bitmap, por meio de operações lógicas AND e OR, retornam a intersecção das *mbOrs*.

A maneira como o índice bitmap é construído, bem como a forma com que as consultas são processadas, dependem da estratégia de processamento adotada. Nesta dissertação, foram propostas quatro estratégias, as quais são detalhadas na seção 5.2. As estratégias propostas determinam quais atributos devem ser indexados, a ordem com que as etapas de filtragem e de refinamento devem ser executadas, e o momento em que o índice ou o DW devem ser acessados.

5.2 Estratégias de Processamento de Consultas

Nessa seção são apresentadas as estratégias de processamento de consultas propostas nesta dissertação de mestrado, utilizando o ImageDW-index. São definidas quatro estratégias, as quais são baseadas em duas abordagens, denominadas Especialista e Unificada, cada qual com duas configurações distintas, denominadas ACESSADW e UsaBitmap.

As estratégias apresentadas nessa seção foram criadas com o objetivo de avaliar a ordem em que as etapas do processamento de consultas IOLAP devem ser executadas considerando o uso do ImageDW-index. Aliás, a definição dessas estratégias permite que diversos cenários de consultas analíticas com predicados de similaridade entre imagens possam ser explorados de modo que determinada abordagem de processamento de consultas pode ser mais vantajosa em um cenário do que em outro. Dessa forma, um processador de consultas IOLAP poderia identificar, dada uma consulta de entrada, qual seria a es-

estratégia mais eficiente para executá-la. A avaliação do desempenho das estratégias em diferentes cenários é realizada no Capítulo 6.

A principal diferença entre as abordagens propostas refere-se à forma como os predicados convencionais e de similaridade entre imagens são tratados. Enquanto a abordagem Especialista realiza a filtragem considerando somente os predicados de similaridade entre imagens, a abordagem Unificada realiza a filtragem considerando os atributos convencionais e de similaridade entre imagens conjuntamente.

A diferença entre as configurações diz respeito à forma como os vetores de características das imagens são recuperados em uma consulta, desde que devido a questões de dimensionalidade, em geral esses vetores representam um alto custo ao processamento das consultas IOLAP. Na configuração *AcessaDW*, os vetores de características são acessados diretamente no DW, enquanto que na configuração *UsaBitmap* os vetores de características são indexados usando-se o índice bitmap e, portanto, recuperados por meio desse índice.

As estratégias propostas e suas respectivas características são resumidas conforme segue:

- **Estratégia EspecialistaAcessaDW:** Filtragem pelos predicados de similaridade entre imagens usando o ImageDW-index, depois filtragem pelos predicados convencionais acessando o DW e recuperando os vetores de características das imagens resultantes, depois refinamento Omni. Indexação apenas dos atributos de distância usando o índice bitmap.
- **Estratégia EspecialistaUsaBitmap:** Filtragem pelos predicados de similaridade entre imagens usando o ImageDW-index, com recuperação direta dos vetores de características das imagens resultantes pelo índice, depois refinamento Omni, e, por fim, filtragem pelos predicados convencionais. Indexação apenas dos atributos de distância e também dos vetores de características usando o índice bitmap.
- **Estratégia UnificadaAcessaDW:** Filtragem pelos atributos convencionais e de similaridade entre imagens conjuntamente por meio do ImageDW-index, depois refinamento Omni, recuperando os vetores de características das imagens resultantes por meio de acesso ao DW. Indexação de todos os atributos usando o índice bitmap, com exceção dos vetores de características.
- **Estratégia UnificadaUsaBitmap:** Filtragem pelos atributos convencionais e de similaridade entre imagens conjuntamente, com recuperação direta dos vetores de características das imagens resultantes pelo ImageDW-index, depois refinamento

Omni. Indexação de todos os atributos usando o índice bitmap, incluindo os vetores de características.

Para facilitar a compreensão das estratégias, cada passo executado durante o processamento das consultas é representado por módulos, cada qual responsável por determinada tarefa. Nas seções 5.2.1, 5.2.2 são descritas as configurações de processamento *AcessaDW* e *UsaBitmap*, de acordo com as abordagens Especialista e Unificada, respectivamente. Já na seção 5.2.3 são detalhados os algoritmos executados em cada um dos módulos, de acordo com as estratégias propostas.

5.2.1 Abordagem Especialista

De acordo com a abordagem Especialista, uma consulta IOLAP é particionada de modo que os predicados convencionais sejam processados independentemente dos predicados de similaridade entre imagens. Para essa estratégia, foram definidos os seguintes módulos:

- Módulo de filtragem dos dados pelos predicados de similaridade entre imagens (*NonConveFiAlg*).
- Módulo de filtragem dos dados pelos predicados convencionais (*ConveFiAlg*).
- Módulo de refinamento Omni (*OmRefAlg*).

De acordo com essa abordagem, o índice bitmap é criado apenas para os atributos de distâncias, os quais representam a distância entre os vetores de características das imagens e seus elementos representativos, para cada camada perceptual. Os atributos convencionais não são indexados. Ademais, com relação à estratégia *EspecialistaAcessaDW*, apenas as distâncias são indexadas, enquanto que para a estratégia *EspecialistaUsaBitmap* as distâncias e os vetores de características são indexados, conforme detalhado a seguir.

Estratégia EspecialistaAcessaDW

A Figura 5.5 ilustra a estratégia *EspecialistaAcessaDW*. O módulo *NonConveFi* recebe os predicados de similaridade entre imagens e realiza uma consulta por similaridade por meio do acesso ao índice bitmap, representado pelo componente *ImageDW-index* na figura. O resultado obtido é passado para o componente *ConveFi*, juntamente com os predicados convencionais da consulta original. Então, uma nova consulta é submetida, dessa vez ao

DW, de modo a recuperar os identificadores das tuplas que satisfazem aos predicados convencionais e que estejam contidas no conjunto de resposta obtido por NonConveFi. Uma vez que o ImageDW-index foi usado para recuperar imagens candidatas, a resposta final da consulta é encontrada por meio da etapa de refinamento, a qual é representada pelo módulo OmRef, que remove os elementos que não estão contidas dentro do raio da consulta IOLAP original. Finalmente, a última etapa consiste do processo de contagem dos identificadores retornados, gerando a resposta final da consulta.

Figura 5.5: Processamento de consultas IOLAP de acordo com a Estratégia EspecialistaAcessaDW.

Uma desvantagem dessa estratégia é que o acesso ao DW é feito mesmo quando a consulta IOLAP não envolve predicados convencionais. Isso ocorre para que os vetores de características sejam recuperados para a próxima etapa do processamento da consulta. Pelo fato do refinamento ser a última etapa realizada, a filtragem pelos predicados convencionais em ConveFi é mais custosa do que poderia ser, visto que os elementos que são descartados ao final do processo ainda pertencem ao conjunto de resposta parcial obtido em NonConveFi. Em contrapartida, o refinamento Omni é realizado sobre um conjunto ainda menor de elementos, considerando as filtrações realizadas previamente.

Estratégia EspecialistaUsaBitmap

A Figura 5.6 ilustra a estratégia EspecialistaUsaBitmap. O módulo NonConveFi recebe os predicados de similaridade entre imagens e realiza uma consulta por similaridade por meio do acesso ao índice bitmap, representado pelo componente ImageDW-index na figura. Diferentemente da estratégia EspecialistaAcessaDW, os vetores de características são recuperados diretamente pelo índice bitmap, ou seja, as imagens candidatas, oriundas da etapa de refinamento realizada em NonConveFi, são obtidas com seus respectivos

vetores de características, que são representados pela expressão “ $+FVs$ ” na figura. Uma vez que os vetores de características já estão disponíveis, a consulta passa para o processo de refinamento Omni, o qual ocorre no módulo OmRef, que remove os elementos que não estão contidos dentro do raio da consulta IOLAP original. Se a consulta envolve predicados convencionais, a próxima etapa do processamento consiste do acesso ao DW para realização da filtragem pelos atributos convencionais, o que ocorre no módulo ConveFi. O refinamento Omni é realizado antes do processamento de ConveFi para que a consulta ao DW seja restrita à menor quantidade de elementos possível.

Figura 5.6: Processamento de consultas IOLAP de acordo com a Estratégia EspecialistaUsaBitmap.

Por um lado, a estratégia EspecialistaUsaBitmap introduz o custo de recuperar os vetores de características diretamente do índice bitmap, tirando a necessidade de acessar o DW para recuperar esses dados, conforme a estratégia EspecialistaAcessaDW. Dessa forma, em consultas envolvendo predicados convencionais, de acordo com EspecialistaUsaBitmap, o tempo no processamento realizado em ConveFi pode ser reduzido pela realização prévia da filtragem pelos predicados de similaridade entre imagens, a qual é realizada em etapas anteriores nos módulos NonConveFi e OmRef. Por outro lado, o custo do processamento realizado no módulo OmRef, de acordo com a estratégia EspecialistaUsaBitmap, é maior do que o de EspecialistaAcessaDW, visto que ele é realizado sobre um conjunto maior de dados, ou seja, sem a filtragem prévia dos atributos convencionais. Aliás, a dimensionalidade dos vetores de características pode impactar negativamente no desempenho dessa estratégia, visto que na estratégia EspecialistaUsaBitmap todos os elementos que satisfazem aos predicados de similaridade entre imagens são recuperadas juntamente com seus vetores de características, enquanto que, na estratégia EspecialistaAcessaDW, os vetores de características são recuperados para um conjunto mais restrito de dados, filtrado pelos predicados convencionais.

Outra diferença da estratégia EspecialistaUsaBitmap com relação a EspecialistaAcessaDW é que na primeira o DW só é acessado caso a consulta envolva predicados convencionais, visto que os vetores de características são obtidos por meio de acesso ao índice bitmap. Já na segunda, o acesso ao DW é feito independentemente da consulta apresentar ou não predicados convencionais.

Em suma, a principal diferença entre as duas estratégias da abordagem Especialista é que em EspecialistaUsaBitmap, toda a filtragem pelos atributos não convencionais é realizada antes do processamento dos predicados convencionais. Já em EspecialistaAcessaDW, os predicados convencionais filtram os dados recuperados pelo processamento dos predicados de similaridade entre imagens.

5.2.2 Abordagem Unificada

De acordo com a abordagem Unificada, a consulta IOLAP é processada inteiramente por um único componente, sendo que o índice bitmap é criado tanto para os atributos convencionais, quanto para os não convencionais. Para essa estratégia, foram definidos os seguintes módulos:

- Módulo de filtragem por todos os predicados da consulta IOLAP (ConveNonConveFi).
- Módulo de refinamento Omni (OmRef).

Com relação à estratégia UnificadaAcessaDW, ela indexa as distâncias e os atributos convencionais. Já a estratégia UnificadaUsaBitmap indexa os atributos convencionais, os atributos não convencionais e os vetores de características. Essas estratégias são detalhadas a seguir.

Estratégia UnificadaAcessaDW

De acordo com a estratégia UnificadaAcessaDW, a consulta IOLAP é filtrada pelos atributos convencionais e não convencionais, acessando apenas o índice bitmap, no módulo ConveFiNonConveFi. Para a etapa de refinamento, recuperam-se os vetores de características dos elementos do conjunto de candidatos acessando o DW, no módulo OmRef. Após esse processo, o refinamento é realizado e a contagem dos elementos de resposta é feita. Na Figura 5.7 essa estratégia é ilustrada.

Figura 5.7: Processamento de consultas IOLAP de acordo com a Estratégia UnificadaAcessaDW.

A desvantagem dessa abordagem é a necessidade de acessar o DW para recuperar os vetores de características em todas as consultas IOLAP, visto que esses dados devem ser recuperados para a próxima etapa, realizada no módulo OmRef. Entretanto, apenas os vetores das imagens candidatas são recuperados e a etapa de refinamento é realizada sobre um conjunto menor de dados, visto que a filtragem pelos predicados convencionais e de similaridade entre imagens é feita anteriormente.

Estratégia UnificadaUsaBitmap

De acordo com essa estratégia, a consulta IOLAP é filtrada pelos atributos convencionais e não convencionais, acessando apenas o índice bitmap, no módulo ConveFiNonConveFiAlg. Os elementos de resposta já são recuperadas juntamente com seus respectivos vetores de características. Dessa forma, a etapa de refinamento é realizada sem o acesso ao DW, no módulo OmRefAlg. Após esse processo, a contagem dos elementos de resposta é realizada. Na Figura 5.8 essa estratégia é ilustrada.

Figura 5.8: Processamento de consultas IOLAP de acordo com a Estratégia UnificadaUsaBitmap.

Intuitivamente, essa estratégia pode oferecer o melhor desempenho, visto que não há acesso ao DW. Toda a consulta pode ser respondida diretamente pelo índice bitmap,

sendo que a etapa de refinamento pode ser realizada em seguida já com os vetores de características recuperados.

5.2.3 Algoritmos Executados pelos Módulos de Processamento de Consultas IOLAP

Nessa seção são descritos os algoritmos referentes às tarefas executadas pelos seguintes módulos: Filtragem pelos predicados convencionais (ConveFi), Filtragem pelos predicados de similaridade entre imagens (NonConveFi), Filtragem por todos os predicados (ConvAndNonConveFi) e Refinamento Omni (OmRef). Os algoritmos 1, 2 e 3 englobam o funcionamento de todos os esses módulos. Com relação ao módulo NonConveFi, sua implementação está contida no algoritmo 2.

No Algoritmo 1 é realizada a recuperação dos identificadores das tuplas que satisfazem aos predicados convencionais da consulta por meio de acesso ao DW, dado um conjunto parcial de resposta. Os parâmetros que devem ser passados são os predicados convencionais da consulta IOLAP, um conjunto de resposta preliminar, contendo tuplas candidatas que foram recuperadas anteriormente, e um tipo booleano que diz se a consulta ao DW deve ou não retornar os vetores de características. De acordo com o algoritmo, a consulta SQL (*Structured Query Language*) é construída na linha 2 utilizando os parâmetros de entrada *convPred*, *inSet* e *selectFV*. O conjunto de resposta preliminar é utilizado na cláusula IN, o qual é intersectado com o resultado da consulta ao DW, o que ocorre na linha 3.

Algoritmo 1: ConveFi (*convPred*, *inSet*, *selectFV*)

Entrada: *convPred* {Lista de predicados convencionais},
inSet {Conjunto de dados para a cláusula IN da consulta SQL},
selectFV {Se a consulta deve retornar os vetores de características}

Saída: *resultSet* {Tuplas que satisfazem à consulta}

1 **inicio**

2   *sqlQuery* $\leftarrow$ buildQuery(*convPred*, *inSet*, *selectFV*);

3   *resultSet* $\leftarrow$ executeQuery(*sqlQuery*);

4   **retorna** *resultSet*;

5 **fin**

O Algoritmo 2 recupera os identificadores das tuplas referentes aos elementos candidatos que satisfazem aos predicados de similaridade entre imagens da consulta IOLAP e, dependendo da utilização, também considera os predicados convencionais. Para que esse algoritmo seja válido, pressupõe-se que os atributos convencionais da consulta sejam

indexados utilizando o índice bitmap. Esse algoritmo recebe como entrada uma lista com os predicados convencionais da consulta, outra com os identificadores das camadas perceptuais utilizadas, a imagem e o raio de consulta. O acesso aos dados é feito por meio do índice bitmap de acordo com a técnica do ImageDW-index. Para recuperar as imagens acessando o índice bitmap, é necessário encontrar as distâncias delas até os elementos representativos em cada camada perceptual considerada.

Primeiramente, na linha 2, inicializa-se uma lista que armazenará os valores de distância. Para simplificar, considera-se que as distâncias armazenadas nessa estrutura sejam identificadas por camada perceptual. O laço entre as linhas 4 e 10 é responsável pelo processamento necessário para o cálculo das distâncias em cada camada perceptual. Para isso, na linha 5, são extraídas as características da imagem de consulta, segundo o descritor de imagens da camada perceptual corrente. Na linha 6 são carregados em memória os vetores de características dos elementos representativos, segundo o mesmo descritor. Por fim, nas linhas 7 e 8, são calculadas as distâncias entre o vetor de características da imagem de consulta e os vetores referentes aos elementos representativos e o resultado é armazenado em *distRepList*, respectivamente. Com as distâncias disponíveis, na linha 11 a consulta ao índice bitmap é construída. Caso o parâmetro *convPred* não seja vazio, no passo da linha 11 a consulta construída considerará os predicados convencionais, caso contrário, apenas os predicados de similaridade entre imagens serão utilizados. Na linha 12, a consulta ao índice é executada, retornando os identificadores dos elementos candidatos.

Algoritmo 2: ConvAndNonConveFi ($convPred$, $percL[1..n]$, Img_q , r_q)

Entrada: $convPred$ {Lista de predicados convencionais},
 $percL[1..n]$ {Lista de camadas perceptuais},
 Img_q {Imagem de consulta},
 r_q {Raio da consulta}

Saída: $resultSet$ {Tuplas que satisfazem à consulta}

```

1 inicio
2    $distRepList \leftarrow \emptyset$ ;
3    $i \leftarrow 0$ ;
4   repita
5      $featVec \leftarrow \text{extractFeatures}(Img_q, percL[i])$ ;
6      $repFeatVecSet \leftarrow \text{loadRepFeaturesVectors}(percL[i])$ ;
7      $distRep \leftarrow \text{calcDistances}(featVec, repFeatVecSet)$ ;
8      $distRepList.add(distRep)$ ;
9      $i++$ ;
10  até fim de }  $percL$ ;
11   $query \leftarrow \text{buildIndexQuery}(distRepList, convPred)$ ;
12   $resultSet \leftarrow \text{executeQuery}(query)$ ;
13  retorna  $resultSet$ ;
14 fin

```

Por fim, o Algoritmo 3 descreve os passos realizados na etapa de refinamento. Os parâmetros de entrada são o conjunto de elementos candidatos disponíveis, as camadas perceptuais e os respectivos vetores de características de seus elementos representativos, bem como o raio de abrangência da consulta IOLAP. Ressalta-se que apenas um raio de abrangência é especificado na consulta, o qual é considerado para todas as camadas perceptuais, e que seu valor é calculado em termos da porcentagem do raio do conjunto de dados (metadado) formado pelos vetores de características em cada camada perceptual.

Para cada elemento candidato à resposta, o laço entre as linhas 5 e 22 é responsável por verificar se ele está contido no raio de abrangência especificado para todas as camadas perceptuais da consulta. Na linha 6 é recuperado um elemento candidato a ser analisado, considerando a princípio que ele seja uma resposta válida, conforme a linha 7. Para cada camada perceptual, é calculada a distância entre o vetor de características do elemento candidato e o da imagem de consulta, de acordo com o descritor de imagens da camada perceptual corrente. Se para qualquer camada perceptual, o elemento candidato não estiver contido em seu respectivo raio de abrangência, o elemento já é descartado. Caso

contrário, conforme as linhas de 17 a 19, esse elemento é considerado uma resposta válida.

Algoritmo 3: OmRef ($candidateSet, percL[1..n], featVec_q[1..n], r_q$)

Entrada: $candidateSet$ {Conjunto dos elementos candidatos},
 $percL[1..n]$ {Camadas perceptuais},
 $featVec_q[1..n]$ {Vetores de características da imagem de consulta},
 r_q {Raio da consulta}

Saída: $resultSet$ {Resultado do refinamento}

```

1 inicio
2    $resultSet \leftarrow \emptyset$ ;
3   if  $candidateSet$  está vazio then
4     | retorna  $resultSet$ ;
5   repita
6     |  $candidate \leftarrow candidateSet.pop()$  {Obtém um elemento e o remove do conjunto};
7     |  $isResp \leftarrow true$ ;
8     |  $i \leftarrow 0$ ;
9     | repita
10      |  $dist \leftarrow calcDistance(candidate.featVec[i], featVec_q[i])$ ;
11      | if  $dist > r_q$  then
12        | |  $isResp \leftarrow false$ ;
13      | else
14        | | continue;
15      | end if
16      | até fim de  $percL$  E  $isResp$  igual a  $true$ ;
17      | if  $isResp$  igual a  $true$  then
18        | |  $resultSet.add(candidate)$ ;
19      | else
20        | | continue;
21      | end if
22    | até fim de  $candidateSet$ ;
23    Retorne  $resultSet$ ;
24 fin

```

Um detalhe referente ao Algoritmo 3 é que para as estratégias EspecialistaAcessaDW, EspecialistaUsaBitmap e UnificadaUsaBitmap o conjunto de elementos candidatos a resposta já estão com seus respectivos vetores de características recuperados. Entretanto, na estratégia UnificadaUsaDW o algoritmo precisa acessar o DW para que esses dados sejam obtidos.

5.3 Gerador de Dados ImageDW-Gen

No melhor de nosso conhecimento, não existe na literatura um DW de imagens que contenha um grande volume de dados. Entretanto, os testes projetados para validar o ImageDW-index necessitavam de uma base dessa natureza, para avaliar o desempenho das estratégias de indexação propostas. Esse fato motivou o desenvolvimento do ImageDW-Gen, um gerador de dados voltado especificamente para a criação e o povoamento de um DW de imagens.

O ImageDW-Gen possui as seguintes características: (i) ele utiliza uma ferramenta externa para a realização do ETL de imagens; (ii) ele gera dados convencionais sintéticos inseridos em um contexto médico; (iii) ele segue as diretrizes especificadas na seção 5.1.1, utilizando um esquema estrala estendido para o armazenamento dos dados convencionais e de imagens; e (iv) ele provê a distribuição das tuplas na tabela de fatos de maneira uniforme por meio da combinação das imagens com os dados convencionais.

Na seção 5.3.1 são descritas as ferramentas utilizadas na etapa de ETL de imagens, bem como a origem dos dados utilizados. Na seção 5.3.2 é descrito o esquema estrela adotado para a construção do DW de imagens. Finalmente, na seção 5.3.3, é descrita a arquitetura do ImageDW-Gen.

5.3.1 Origem dos Dados e Ferramentas Utilizadas para o ETL de Imagens

Com relação aos dados não convencionais, obteve-se acesso a um banco de imagens médicas reais do Hospital das Clínicas de Ribeirão Preto (HCRP) para a extração dos vetores de características. As imagens obtidas são do tipo DICOM e referem-se a várias modalidades (cérebro, mama, tórax, etc), sendo em torno de 10 milhões de imagens no total. Para os testes apresentados nesta dissertação de mestrado, no entanto, são utilizadas 1.409.848 imagens. Com relação aos dados convencionais, eles foram gerados sinteticamente por um gerador de dados desenvolvido para essa finalidade. A integração dos dados convencionais com os de imagens também foi realizada por esse gerador.

A extração das características, procedimento realizado na fase de ETL de imagens, foi feita utilizando o FMI-SiR (*user-defined Features, Metrics and Indexes for Similarity Retrieval*) (Kaster et al., 2010), que é um módulo de banco de dados que permite executar consultas por similaridade integradas às demais operações do SGBD, mais especificamente, sobre o SGBD Oracle. Como essa ferramenta oferece suporte ao tipo DICOM, foi possível processar as imagens do HCRP, as quais estão armazenadas nesse formato.

O recurso do FMI-SiR que foi aproveitado nesta dissertação de mestrado refere-se à extração de características das imagens armazenadas no banco de dados. Utilizando essa ferramenta, foi possível inserir os métodos de extração de características desejados no núcleo do SGBD, e assim, por meio de procedimentos em PL/SQL, realizar a extração de características das imagens diretamente do servidor Oracle do HCRP.

Nesta dissertação, foi adotada a implementação dos métodos de extração de características da biblioteca Artemis. A Artemis foi desenvolvida pelo Grupo de Bases de Dados e Imagens (GBDI) do ICMC-USP e consiste de um protótipo para a extração de características segundo vários descritores de imagens. Compõem essa biblioteca implementações de diversos índices, bem como de extratores de características. Tanto o FMI-SiR quanto a Artemis são implementados na linguagem C++. Dessa forma, para esse projeto, foi feita a integração de parte do código fonte da Artemis ao código fonte do FMI-SiR.

Em suma, é possível utilizar a ferramenta FMI-SiR para extração de características de imagens DICOM armazenadas em um SGBD Oracle. Ademais, é possível adicionar tantos extratores de imagens quantos forem necessários ao FMI-SiR para que, por meio de procedimentos em PL/SQL, seja possível extrair os vetores de características desejados.

Na próxima seção é apresentado o esquema estrela utilizado como base para os experimentos desta dissertação de mestrado e, portanto, utilizado pelo ImageDW-Gen para a geração do DW de imagens. São detalhados os dados armazenados nas tabelas de dimensão convencionais e não convencionais. Dessa forma, são descritos quais extratores de imagens que foram utilizados para compor as camadas perceptuais do DW de imagens, bem como as características de cada uma das camadas.

5.3.2 Esquema Estrela Adotado

Na Figura 5.9, é ilustrado o esquema estrela usado como base para o desenvolvimento desta dissertação de mestrado. Esse esquema é projetado para um contexto médico e consolida, em um DW de imagens, exames de diferentes modalidades como cabeça, mama, joelho, etc., realizados em pacientes de diferentes hospitais, com diversas faixas etárias ao longo de um determinado período de tempo. A medida numérica de interesse é definida na tabela de fatos *Exame* pelo atributo *quantidade*, referindo-se à quantidade de tuplas que atendem a determinada consulta analítica. Essa medida numérica armazena sempre o valor igual a 1 no nível mais inferior do DW. Para facilitar a execução de consultas IOLAP e deixar o esquema mais semântico, armazena-se o atributo *quantidade*.

Sobre as dimensões não convencionais, esse esquema possui cinco camadas perceptuais representadas pelas tabelas de dimensão *PL0*, *PL1*, *PL2*, *PL3* e *PL4*. Optou-se por

Figura 5.9: Esquema Estrela do DW de imagens.

manter os nomes das camadas perceptuais de maneira genérica (e.g. *perceptual layer 0*, *PL0*). Nesta dissertação, elas referem-se, respectivamente, ao histograma de cor (Hist) e aos descritores de Haralick variância (HV), entropia (HE), uniformidade (HU) e homogeneidade (HH). As características de cada uma dessas camadas perceptuais são descritas na Tabela 5.3. São detalhadas a dimensionalidade dos vetores de características, a quantidade de elementos representativos considerados usando o HF, bem como o diâmetro do conjunto de dados formado por cada camada perceptual, medido aproximadamente pela maior distância entre dois elementos do conjunto.

Nome	Descritor	Dimensionalidade	Qtde. rep.	Diâmetro
PL0	Hist.	256	3	594.402,88
PL1	HV	4	3	80,51
PL2	HE	4	3	4,02
PL3	HU	4	3	1,98
PL4	HH	4	4	1,25

Tabela 5.3: Características das camadas perceptuais.

Os dados armazenados nas dimensões não convencionais são as distâncias das imagens aos elementos representativos da referida camada perceptual. Por exemplo, os atributos

pl4distrep_1, *pl4distrep_2*, *pl4distrep_3* e *pl4distrep_4* armazenam as distâncias entre as imagens e os quatro elementos representativos da camada perceptual PL4. Os vetores de características são armazenados em uma única tabela de dimensão denominada *VetCaract*, nos atributos *vetCaractPl0*, *vetCaractPl1*, *vetCaractPl2*, *vetCaractPl3* e *vetCaractPl4*, os quais referem-se aos vetores de características segundo os descritores de imagens das cinco camadas perceptuais.

Esse esquema possui ainda as seguintes dimensões convencionais *Hospital*, *DataExame*, *Paciente*, *Idade* e *DadosExame*. A primeira tabela de dimensão armazena dados referentes ao hospital em que o exame foi realizado, a segunda refere-se à tabela de dimensão temporal dessa aplicação, a terceira possui dados sobre o paciente, a quarta armazena dados da idade do paciente, e, por fim, a última dimensão refere-se aos dados do exame, com destaque para o atributo *parteCorpo*, o qual informa a modalidade do exame.

A principal característica dessa modelagem é reservar uma tabela de dimensão especificamente para os vetores de características, enquanto que são criadas outras tabelas de dimensão para cada camada perceptual, as quais armazenam apenas as distâncias das imagens até os elementos representativos. Na dissertação de mestrado (Annibal, 2011) são discutidas várias abordagens para a modelagem de um DW de imagens com n camadas perceptuais, bem como são apresentados resultados referentes ao impacto da modelagem no processamento de consultas da seção 4.1, levando em consideração a dimensionalidade dos vetores de características, a variação do raio de abrangência da consulta e o envolvimento de predicados convencionais. De acordo com os resultados obtidos naquela dissertação de mestrado, a modelagem mais flexível e que apresentou, em geral, os melhores resultados de desempenho para camadas perceptuais com vetores de características de baixa ou alta dimensionalidade foi a discutida nessa seção. Por essa razão essa modelagem foi a adotada nesta dissertação de mestrado.

5.3.3 Estrutura do ImageDW-Gen

Na Figura 5.10 são ilustrados os principais componentes da estrutura do ImageDW-Gen. A fase de ETL de imagens foi desenvolvida utilizando as implementações da biblioteca Artemis na ferramenta FMI-SiR, conforme abordado na seção 5.3.1. Nessa fase, foram extraídos os vetores de características das imagens disponibilizadas pelo HCRP, segundo as cinco camadas perceptuais, e armazenados em seus respectivos arquivos, PL0.txt, PL1.txt, etc. Ainda nessa fase, os elementos representativos foram identificados e seus vetores de características armazenados nos arquivos PL0_Rep.txt, PL1_Rep.txt, etc. Finalizando essa etapa, as distâncias entre as imagens e os elementos representativos foram

calculadas e os resultados foram armazenados nos arquivos *PL0_Dist.txt*, *PL1_Dist.txt*, etc. Outra saída da etapa de ETL de imagens, nesse caso, refere-se à modalidade da imagem sendo processada. No caso de imagens DICOM, esse dado foi extraído diretamente dos metadados da imagem, sendo armazenado no arquivo *BodyType.txt*.

Figura 5.10: Principais componentes da estrutura do ImageDW-Gen.

A criação e o povoamento das tabelas do DW são realizados por meio de *scripts* SQL predefinidos, os quais carregam a base de dados seguindo o modelo de esquema estrela descrito na seção 5.3.2. Para isso, no ImageDW-Gen, são gerados arquivos CSV (*Comma-separated values*) para tabelas de dimensão convencionais, por meio do componente CSVGenerator-ConvDim, para tabelas não convencionais, por meio do componente CSVGenerator-NonConvDim, e por fim, para a tabela de fatos, por meio do componente CSVGenerator-FactTable.

O CSVGenerator-ConvDim recebe como entrada os arquivos *BodyType.txt* e um arquivo predefinido com dados sobre hospitais, *HospitalData.txt*, além dos parâmetros *QtdePacientes* e *IntervaloAnosExame*. O primeiro parâmetro representa a quantidade de pacientes desejada para o DW, e o segundo refere-se ao intervalo de tempo, em anos, aos quais os exames gerados vão se relacionar.

O CSVGenerator-NonConvDim recebe como entrada os arquivos relacionados aos dados de imagens. Sua função é, basicamente, relacionar os dados dos arquivos de entrada de

acordo com o esperado pelas tabelas de dimensão não convencionais, levando em consideração a consistência com os dados convencionais gerados pelo componente CSVGenerator-ConvDim.

Finalmente, o CSVGenerator-FactTable realiza a conexão entre todos os dados gerados para as tabelas de dimensão do esquema estrela na tabela de fatos. Basicamente, esse componente associa os identificadores de cada tupla das tabelas de dimensão na tabela de fatos, considerando as restrições de referência.

5.4 Classes de Consultas IOLAP

Nessa seção são definidas quatro classes de consultas IOLAP, as quais enfocam em diferentes custos de processamento dos predicados de similaridade entre imagens. Essas classes agrupam consultas semelhantes em termos da quantidade de camadas perceptuais e tamanho dos vetores de características, isto é, a dimensão dos vetores.

As classes de consultas foram definidas da seguinte maneira:

- **Classe 1:** Consultas IOLAP pertencentes a essa classe possuem apenas uma camada perceptual envolvendo vetores de características com baixa dimensionalidade. Por exemplo, camada perceptual de Haralick variância.
- **Classe 2:** Consultas IOLAP dessa classe possuem mais do que uma camada perceptual envolvendo vetores de características com baixa dimensionalidade. Por exemplo, todas as camadas perceptuais de Haralick.
- **Classe 3:** Consultas IOLAP pertencentes a essa classe possuem apenas uma camada perceptual envolvendo vetores de características com alta dimensionalidade. Por exemplo, camada perceptual de histograma.
- **Classe 4:** Consultas IOLAP dessa classe possuem mais do que uma camada perceptual envolvendo vetores de características com baixa e alta dimensionalidade. Por exemplo, todas as camadas perceptuais de Haralick e histograma.

O objetivo dessa classificação é facilitar a análise do processamento das consultas considerando as especificidades das consultas IOLAP, as quais envolvem o processamento de vetores de características, camadas perceptuais e dados convencionais, cada qual gerando um determinado custo de processamento. As classes de consultas propostas nessa seção não são fixas, isto é, elas podem ser adaptadas de acordo com as características

do DW de imagens a ser utilizado (e.g. quantidade de camadas perceptuais, tamanho dos vetores de características, etc.). Para essa dissertação de mestrado, no entanto, as quatro classes propostas foram utilizadas no Capítulo 6 para avaliação do desempenho do ImageDW-index.

5.5 Considerações Finais

Neste capítulo foram apresentadas as contribuições desta dissertação de mestrado. Foi mostrado que o ImageDW-index consiste de um mecanismo para a otimização de consultas analíticas estendidas com predicados de similaridade entre imagens, o qual utiliza conceitos bem difundidos na literatura referentes aos índices bitmap e ao MAM da família Omni.

Foram apresentadas as estratégias de processamento de consultas que utilizam o ImageDW-index. Elas permitem o processamento de consultas IOLAP baseadas em uma ou mais camadas perceptuais, utilizando os processos de filtragem e de refinamento definidos pela técnica Omni, bem como fazendo uso do índice bitmap na indexação das distâncias aos elementos representativos. Foram especificados também os requisitos principais para que um ambiente de DW de imagens possa utilizar as estratégias propostas nesta dissertação de mestrado.

Como contribuição secundária, foi descrito neste capítulo o gerador de dados ImageDW-Gen. Esse gerador utiliza ferramentas de extração de características de imagens, como o FMI-SiR e a biblioteca Artemis, bem dados convencionais sintéticos para a geração de uma base de dados de grande porte em um DW de imagens. Foi descrito que os dados gerados referem-se ao contexto médico, consolidando imagens de diferentes modalidades em um DW para análise de exames. A estrutura desse gerador de dados foi descrita, com destaque para sua flexibilidade em termos da quantidade de camadas perceptuais, quantidade de pacientes e o intervalo de tempo em que os exames ocorreram.

Outra contribuição secundária refere-se à definição de quatro classes de consultas, as quais enfocam em diferentes custos de processamento, como tamanho dos vetores de características e quantidade de camadas perceptuais. Essas classes foram criadas para auxiliar no processo de validação da proposta do ImageDW-index.

No próximo capítulo são descritos os testes de desempenho realizados para validação das estratégias de processamento de consultas propostas, do gerador de dados ImageDW-Gen e das classes de consultas especificadas nesta dissertação de mestrado.

Testes de Desempenho

Neste capítulo são descritos os testes para a investigação das vantagens introduzidas pelas estratégias de processamento de consultas usando o ImageDW-index. Foram utilizadas consultas que representam as quatro classes definidas na seção 5.4, as quais foram executadas sobre cada uma das configurações propostas na seção 5.2. O DW de imagens utilizado para a realização das baterias de testes descritas neste capítulo segue o modelo de esquema estrela discutido na seção 5.3.2.

Cada consulta foi executada cinco vezes e o tempo médio em segundos do processamento total das consultas foi calculado. As estratégias propostas nesta dissertação de mestrado foram comparadas com o trabalho de Annibal et. al. (2010), no qual é definida uma forma de processamento de consultas IOLAP, conforme discutido na seção 4.1.

Nos testes realizados, o índice bitmap foi construído usando a técnica de codificação por intervalo com cinco bins. Na literatura, existem diversas técnicas que poderiam ser usadas no índice bitmap, como codificação por igualdade, *range encoding*, *binary encoding*, entre outras, conforme discutido na seção 2.6. Por meio de testes experimentais, descritos no Apêndice A, identificou-se que a codificação por intervalo com cinco bins corresponde à estrutura que proporciona o melhor tempo de acesso ao índice bitmap, para o conjunto de dados de teste, dentre as técnicas analisadas. Assim, essa técnica foi empregada nos testes de desempenho descritos neste capítulo.

Os testes foram executados em uma máquina equipada com um processador Intel Core i7-4770 com 3.40GHz, 32 GB de memória RAM, rodando o sistema operacional Linux CentOS 6.5. A base de dados foi criada utilizando o gerador de dados ImageDW-Gen, apresentado na seção 5.3, o qual populou o DW de imagens com 1.409.848 de tuplas.

Os testes foram realizados considerando a seletividade das consultas, dada pelos predicados de similaridade entre imagens e também pelos predicados convencionais. Se uma consulta recupera uma quantidade pequena de tuplas, diz-se que essa consulta é muito seletiva, caso contrário, trata-se de uma consulta pouco seletiva. Em detalhes, foram realizadas duas baterias de testes. A primeira bateria refere-se à análise de consultas definidas somente considerando-se predicados de similaridade entre imagens, e é discutida na seção 6.1. A segunda bateria refere-se à análise das estratégias considerando consultas definidas usando-se tanto predicados de similaridade entre imagens quanto predicados convencionais, e é discutida na seção 6.2.

A justificativa para a realização de testes de desempenho considerando primeiro somente os predicados de similaridade entre imagens deve-se ao fato de que a proposta desta dissertação de mestrado é justamente focar na melhoria de desempenho do processamento de consultas IOLAP. Investigando-se apenas os predicados de similaridade entre imagens, é possível identificar como as estratégias propostas comportam-se. Entretanto, consultas IOLAP em geral envolvem também um ou mais predicados convencionais. Nesse sentido, justifica-se a realização de testes de desempenho considerando consultas com predicados de similaridade entre imagens e predicados convencionais. Investigando essas consultas, é possível identificar se o processamento dos predicados convencionais impacta no comportamento das estratégias propostas.

Após a descrição dos testes de desempenho, o capítulo é finalizado na seção 6.3, com as considerações finais.

6.1 Testes Baseados na Seletividade dos Predicados de Similaridade entre Imagens

O objetivo desses testes foi avaliar o desempenho do ImageDW-index em um cenário envolvendo consultas definidas apenas em termos de predicados de similaridade entre imagens. Para isso, dentre as estratégias de processamento propostas, foi necessário avaliar apenas as estratégias EspecialistaAcessaDW e EspecialistaUsaBitmap, desde que as demais estratégias enfocam também a indexação de atributos convencionais, os quais são desconsiderados nessa bateria de testes.

De acordo com as classes de consultas definidas, foram submetidas as seguintes consultas ao DW de imagens:

- Consulta Q1: “Qual a quantidade de imagens similares à imagem de consulta de encéfalo, de acordo com o raio r_{q1} , para a camada perceptual $PL1$?”

- Consulta Q2: “Qual a quantidade de imagens similares à imagem de consulta de encéfalo, de acordo com o raio r_{q_2} , para as camadas perceptuais $PL1$, $PL2$, $PL3$ e $PL4$?”
- Consulta Q3: “Qual a quantidade de imagens similares à imagem de consulta de encéfalo, de acordo com o raio r_{q_3} , para a camada perceptual $PL0$?”
- Consulta Q4: “Qual a quantidade de imagens similares à imagem de consulta de encéfalo, de acordo com o raio r_{q_4} , para as camadas perceptuais $PL0$, $PL1$, $PL2$, $PL3$ e $PL4$?”

Essas consultas buscam imagens médicas de encéfalo, considerando as diferentes camadas perceptuais do DW de imagens para a realização das consultas por similaridade. Pelo fato do atributo convencional *parteCorpo* ser parte essencial das consultas, ou seja, $\text{parteCorpo} = \text{encéfalo}$, nessa bateria de testes também foi utilizado um índice bitmap para indexá-lo.

Para cada consulta, o raio de abrangência r_q variou de modo que as seguintes seletividades fossem obtidas, em termos da quantidade de tuplas de respostas recuperadas ao fim da consulta: 10, 100, 1.000, 10.000 e 100.000. A escolha dessa faixa de valores de seletividades deve-se ao pressuposto de que usuários de SSD que utilizam um DW de imagens não estão interessados em um conjunto amplo de respostas, como 10 e 100. Por outro lado, também foram consideradas quantidades maiores de respostas para completude. Os testes realizados nesta dissertação de mestrado utilizaram, portanto, seletividades menores do que 8% do volume de dados da base.

Na Figura 6.3 são exibidos os gráficos com os resultados obtidos utilizando EspecialistaAcessaDW e EspecialistaUsaBitmap. A consulta Q1 considera uma camada perceptual com vetores de baixa dimensionalidade, os quais possuem tamanho quatro. Conforme a Figura 6.1a, para esse cenário, a estratégia EspecialistaUsaBitmap apresentou resultados melhores. Pelo fato da baixa dimensionalidade dos vetores de características, o custo de recuperá-los diretamente pelo índice não impactou negativamente no desempenho da estratégia. Em Q2, foram consideradas quatro camadas perceptuais com vetores de dimensionalidade quatro. De acordo com a Figura 6.1b, o desempenho de EspecialistaUsaBitmap foi apenas um pouco inferior ao de EspecialistaAcessaDW. Pode-se concluir, portanto, que a indexação dos vetores de características com baixa dimensionalidade proveu, em geral, melhor desempenho no processamento das consultas.

Figura 6.1: Resultados da execução de Q1, Q2, Q3 e Q4 utilizando as estratégias EspecialistaAcessaDW, EspecialistaUsaBitmap e o trabalho correlato, para as seletividades consideradas.

O desempenho de Q3, que considera uma camada perceptual com vetores de dimensionalidade igual a 256, evidencia o alto custo causado pela recuperação das tuplas com vetores dessa natureza. Esse fato impactou negativamente no desempenho da estratégia EspecialistaUsaBitmap. Conforme a Figura 6.1c, a estratégia EspecialistaAcessaDW apresentou os melhores resultados, mostrando que acessar o DW para selecionar os vetores de características apenas das imagens do conjunto de candidatos é mais vantajoso do que utilizar o índice bitmap para recuperar esses dados. O resultado apresentado na Figura 6.1d, em que a consulta Q4 é executada, considerando camadas perceptuais com vetores de alta e baixa dimensionalidades, reforça esse resultado. Pode-se concluir, portanto, que a indexação dos vetores de características, conforme a estratégia EspecialistaUsaBitmap,

com alta dimensionalidade e com alta e baixa dimensionalidades misturados proveu, em geral, pior desempenho no processamento das consultas. Ou seja, para esses tipos de consultas, foi mais vantajoso acessar o DW para recuperar esses dados de imagens, conforme a estratégia EspecialistaAcessaDW.

Pelos resultados apresentados nessa seção, o uso do ImageDW-Index mostrou-se muito mais eficiente do que o trabalho correlato de Annibal et al. (2010). Em todos os casos, seu uso proporcionou ganhos expressivos. Em Q1, a estratégia EspecialistaUsaBitmap proporcionou o melhor desempenho, com ganho variando entre 45,31% e 85,96%, nas seletividades 10 e 100.000, respectivamente. Em Q2, o melhor desempenho foi obtido pela estratégia EspecialistaUsaBitmap, com ganho variando entre 62,80% e 88,07%, nas seletividades 10 e 100.000, respectivamente. Já em Q3, o melhor desempenho foi proporcionado pela estratégia EspecialistaAcessaDW, a qual obteve ganhos entre 53,36% e 80,50%, nas seletividades 10 e 100.000, respectivamente. Por fim, em Q4, a estratégia EspecialistaAcessaDW obteve o melhor desempenho, proporcionando ganhos entre 62,83% e 81,43%, nas seletividades 1.000 e 100.000, respectivamente.

Pelo desempenho do trabalho correlato, percebe-se que conforme a seletividade da consulta diminui, isto é, muitas tuplas são recuperadas, em todos os casos analisados, seu desempenho é deteriorado, devido ao alto custo oriundo do acesso a disco para recuperar uma grande quantidade de tuplas envolvendo dados como vetores de características. Esse fato leva a mais acessos a disco para recuperar todas as tuplas de resposta.

Na Figura 6.2 é exibido um gráfico com a média da execução das quatro consultas, apresentando o comportamento médio das estratégias EspecialistaAcessaDW, EspecialistaUsaBitmap e do trabalho correlato. Conforme abordado nos resultados individuais, a média dos resultados evidenciou o melhor desempenho da estratégia EspecialistaAcessaDW. Ademais, conforme atestado pelos resultados individuais, todas as estratégias propostas apresentaram ganhos de desempenho se comparadas ao trabalho correlato. De acordo com o gráfico da Figura 6.2, a estratégia EspecialistaAcessaDW apresentou um ganho de 55,57% até aproximadamente 82,16% com relação ao trabalho correlato.

Figura 6.2: Gráfico com a média dos resultados obtidos pela execução das consultas Q1, Q2, Q3 e Q4.

6.2 Testes Baseados na Seletividade dos Predicados Convencionais e dos Predicados de Similaridade entre Imagens

O objetivo dessa bateria de testes foi avaliar o desempenho do ImageDW-index em um cenário envolvendo consultas definidas considerando-se não somente predicados de similaridade entre imagens, mas também predicados convencionais. Para isso, todas as estratégias propostas foram avaliadas. Diferentemente da bateria de testes anterior, o raio de abrangência das consultas IOLAP foi fixado de modo que a seletividade dos predicados de similaridade entre imagens fosse de 100.000 imagens. Já a parte convencional das consultas variou de cinco a apenas um predicado. Ou seja, foram consideradas consultas com um predicado convencional, consultas com dois predicados convencionais, e assim sucessivamente até cinco predicados convencionais.

Os atributos usados nos predicados convencionais das consultas são descritos na Tabela 6.1, juntamente com a tabela de dimensão de origem, valor e seletividade individual, ou seja, a quantidade de tuplas da tabela de fatos com o valor especificado. Na tabela, os atributos são apresentados do mais seletivo para o menos seletivo. As consultas Q1, Q2, Q3 e Q4, referentes às classes de consultas, foram executadas considerando, primeiramente, cinco predicados convencionais. Para formar as demais consultas, os predicados mais seletivos foram removidos sucessivamente, até que apenas um predicado convencional permanecesse.

Dimensão	Atributo	Valor	Seletividade
Hospital	MacroRegiao	Baixada santista	13.849
DataExame	Ano	Entre 1.990 e 1.995	192.598
DadosExame	RazaoInvestigacao	Suspeita de tumor	246.658
Idade	AnosQtde	Entre 0 e 60 anos	433.116
Paciente	Estado	SP	438.393

Tabela 6.1: Características dos atributos convencionais utilizados nessa bateria de testes.

As consultas submetidas ao DW de imagens são definidas a seguir. Pelo fato do atributo convencional *parteCorpo* ser parte essencial das consultas (ou seja, *parteCorpo* = encéfalo), nessa bateria de testes também foi utilizado um índice bitmap para indexá-lo.

- **Consulta Q1:** “Qual a quantidade de imagens similares à imagem de consulta de encéfalo, de acordo com o raio r_{q1} , para a camada perceptual *PL1*, e que satisfaçam aos predicados convencionais *PC*?”.
- **Consulta Q2:** “Qual a quantidade de imagens similares à imagem de consulta de encéfalo, de acordo com o raio r_{q2} , para as camadas perceptuais *PL1*, *PL2*, *PL3*, e *PL4* e que satisfaçam aos predicados convencionais *PC*?”.
- **Consulta Q3:** “Qual a quantidade de imagens similares à imagem de consulta de encéfalo, de acordo com o raio r_{q3} , para a camada perceptual *PL0*, e que satisfaçam aos predicados convencionais *PC*?”.
- **Consulta Q4:** “Qual a quantidade de imagens similares à imagem de consulta de encéfalo, de acordo com o raio r_{q4} , para as camadas perceptuais *PL0*, *PL1*, *PL2*, *PL3*, e *PL4* e que satisfaçam aos predicados convencionais *PC*?”.

Onde *PC* varia entre os seguintes predicados convencionais:

- **5Conv:** *MacroRegiao* = “Baixada Santista” AND *Ano* entre 1990 e 1995 AND *RazaoInvestigacao* = “suspeita de tumor” AND *anosQtde* entre 0 e 60 anos AND *Estado* = “SP”.
- **4Conv:** *Ano* entre 1990 e 1995 AND *RazaoInvestigacao* = “suspeita de tumor” AND *AnosQtde* entre 0 e 60 anos AND *Estado* = “SP”.
- **3Conv:** *RazaoInvestigacao* = “suspeita de tumor” AND *AnosQtde* entre 0 e 60 anos AND *Estado* = “SP”.

- **2Conv:** *AnosQtde* entre 0 e 60 anos AND *Estado* = “SP”.
- **1Conv:** *Estado* = “SP”.

A seletividade das consultas Q1, Q2, Q3 e Q4 considerando 5Conv, 4Conv, 3Conv, 2Conv e 1Conv é informada na Tabelas 6.2. Nessa tabela são exibidas a quantidade de tuplas recuperada ao final das consultas.

Cons.	Q1	Q2	Q3	Q4
5Conv	8	11	7	9
4Conv	906	931	3.669	922
3Conv	4.810	4.803	4.931	4.772
2Conv	19.141	19.327	19.805	19.125
1Conv	44.185	44.658	44.605	44.511

Tabela 6.2: Seletividade das consultas Q1, Q2, Q3 e Q4.

Os resultados dessa bateria de testes são exibidos na Figura 6.3, em que o desempenho de cada estratégia proposta nesta dissertação de mestrado é ilustrado para cada consulta, variando-se os predicados convencionais.

Figura 6.3: Desempenho das estratégias propostas, considerando as consultas em que a seletividade dos predicados convencionais é variada.

Com relação ao desempenho da estratégia EspecialistaAcessaDW, a qual depende do acesso ao DW para o processamento dos predicados convencionais e recuperação dos vetores de características, observado na Figura 6.3a, as consultas Q2 e Q4 executaram em menos tempo do que as demais. Essas consultas foram menos custosas por serem as mais restritivas em termos dos predicados de similaridade entre imagens, pois envolvem mais do que uma camada perceptual. Esse fato favoreceu a filtragem pelos predicados convencionais de modo que a recuperação dos vetores de características, processo mais custoso, fosse realizada sobre um conjunto menor de dados. Dessa forma, o tempo de acesso ao índice bitmap ocorreu mais rapidamente e a quantidade de tuplas a ser recuperada do DW foi menor. A consulta Q1 demonstrou, basicamente, o mesmo comportamento de Q2 e Q4, entretanto, uma explicação para o seu maior tempo de execução é o fato dessa consulta ser menos restritiva, envolvendo apenas uma camada perceptual. Com relação à

Q3, essa consulta apresentou o maior tempo de processamento. Isso se deve ao fato de que Q3 envolve uma camada perceptual com vetores de características de dimensionalidade alta, sendo que essa consulta é pouco restritiva, conforme os dados da Tabela 6.2, em que pode ser observado que, para a maioria dos casos, a consulta Q3 é a que retornou mais tuplas de resposta. Comparando Q3 com Q4, ambas as consultas envolvem vetores de características com dimensionalidade alta, entretanto o fato de Q4 ser mais restritiva do que Q3, foi determinante para o seu melhor desempenho.

Com relação ao desempenho da estratégia EspecialistaUsaBitmap, as consultas Q3 e Q4 executaram em mais tempo do que as demais. Pelo fato dessa estratégia indexar os vetores de características, a dimensionalidade dos vetores é um fator determinante para o aumento do custo de processamento, devido ao custo de recuperar esses dados. Portanto, o fato de Q3 e Q4 envolverem camadas perceptuais com vetores de características de alta dimensionalidade pode ser uma explicação para o maior custo no processamento dessas consultas. Em contrapartida, observa-se que as consultas Q1 e Q2 são as que executam em menos tempo, por apresentarem os menores vetores de características. Observa-se pelo gráfico da Figura 6.3b que, quanto mais predicados convencionais são considerados, maior o custo para a execução das consultas. Esse comportamento pode ser explicado pelo fato dessa estratégia não apresentar um índice para os atributos convencionais. Dessa forma, o processamento dos predicados convencionais é realizado sobre o DW por meio de operações de junção entre a tabela de fatos e as tabelas de dimensão envolvidas. Como cada predicado convencional está armazenado em uma tabela de dimensão distinta, quanto mais predicados convencionais a consulta envolver, mais operações de junção devem ser realizadas. Outro detalhe importante desse gráfico refere-se ao tempo praticamente constante entre a execução de 2Conv e 1Conv. Esse fato pode ser explicado pela seletividade dos atributos *AnosQtde* e *Estado* terem valores próximos, conforme a Tabela 6.1.

A abordagem Unificada realiza a indexação de todos os atributos envolvidos na consulta, tanto dos atributos convencionais quanto dos atributos de similaridade entre imagens. Com relação à estratégia UnificadaAcessaDW, cujos resultados são exibidos na Figura 6.3c, percebe-se as vantagens introduzidas pelo ImageDW-index, para consultas mais restritivas, envolvendo mais predicados convencionais. Conforme o número de predicados convencionais diminuiu, a capacidade de poda do índice também ficou mais restrita, de modo que o acesso ao DW para a recuperação dos vetores de características se tornou maior, impactando negativamente no desempenho das consultas.

Pelo fato da estratégia UnificadaUsaBitmap realizar a indexação dos vetores de características, percebe-se que, pelo desempenho de Q3 e Q4 mostrado na Figura 6.3d, que, novamente, a dimensionalidade dos vetores de características das consultas influenciou

negativamente nessa abordagem. Uma explicação para o comportamento praticamente constante a partir de 4Conv dado pelas consultas Q1 e Q2, deve-se à seletividade dos atributos convencionais a partir de 4Conv, os quais não apresentam grande variação.

Com relação ao trabalho correlato de Annibal et al. (2010), na Figura 6.4 são exibidos os resultados obtidos por essa abordagem na execução das consultas Q1, Q2, Q3 e Q4. Pelo gráfico, quanto mais restritiva a consulta, mais rápida ela é executada. Esse fator é devido ao menor custo na recuperação dos vetores de características, visto que, nessa abordagem, não é utilizado índice.

Figura 6.4: Desempenho da estratégia de Annibal et. al. (2010), considerando a execução das consultas variando a seletividade dos predicados convencionais.

Na Tabela 6.3 são mostrados os tempos médios da execução das consultas para cada estratégia, inclusive a do trabalho correlato, considerando as consultas dessa bateria de testes. Ademais, nessa tabela, o tempo médio de execução de todas as consultas foi calculado. Pelos dados apresentados, a estratégia com melhor desempenho foi UnificadaAcessaDW, apresentando um ganho médio de 76,76% em relação ao desempenho do trabalho correlato. Mais detalhadamente, o desempenho dessa estratégia variou desde uma perda de desempenho com relação ao trabalho correlato, o qual obteve um ganho de 8,48% na consulta 5Conv sobre a estratégia UnificadaAcessaDW, até um ganho de 85,42% na consulta 1Conv.

Cons.	Annibal	EspAcDW	EspUsaBtm	UniAcDW	UniUsaBtm
5Conv	7,18s	159,28s	119,39	7,84s	10,34s
4Conv	26,40s	166,30s	119,70s	13,19s	23,53s
3Conv	78,38s	160,96s	119,40s	18,30s	25,58s
2Conv	86,66s	85,49s	32,32s	17,42s	26,01s
1Conv	122,24s	119,80s	32,68s	17,82s	26,60s
Média	64,17s	138,36s	84,70s	14,91s	22,41s

Tabela 6.3: Tempo médio (em segundos) do desempenho das estratégias para as cinco consultas realizadas.

Para essa bateria de testes ficou evidente o desempenho inferior das configurações da abordagem Especialista (estratégias EspecialistaAcessaDW e EspecialistaUsaBitmap) sobre a abordagem Unificada (estratégias UnificadaAcessaDW e UnificadaUsaBitmap). Esse comportamento deve-se ao fato de que a abordagem Unificada considera a indexação de todos os atributos da consulta, evitando assim o acesso ao DW para a realização da filtragem pelos predicados convencionais.

Adicionalmente, pelos dados descritos na Tabela 6.3, pode-se identificar que a estratégia que apresentou o melhor desempenho foi a UnificadaAcessaDW, conforme destacado em negrito na coluna correspondente. Esse comportamento deve-se ao fato de que o acesso ao DW, nessa estratégia, foi realizada sobre um conjunto de dados previamente filtrado com o objetivo apenas de recuperar os vetores de características. Essa estratégia mostrou-se mais eficiente do que a recuperação desses vetores por meio do índice bitmap. De acordo com a Tabela 6.3, se comparada com o trabalho correlato de Annibal et. al. (2010), a estratégia UnificadaAcessaDW apresentou um ganho médio de 76,76 %, sendo que, na consulta 5Conv, o trabalho correlato obteve um ganho de 8,48%, enquanto que na consulta 1Conv, a estratégia UnificadaAcessaDW proporcionou um ganho de 85,42%. Esse resultado confronta a intuição inicial de que a indexação de todos os atributos, conforme a estratégia UnificadaUsaBitmap, seria mais vantajosa para o ImageDW-index, o que foi demonstrado não ser verdadeiro.

Na bateria de testes anterior, que considerava apenas predicados de similaridade entre imagens, as abordagens Especialista e Unificada apresentaram ganhos expressivos com relação ao trabalho correlato em todos os cenários observados. Diferentemente, nessa bateria de testes com predicados convencionais envolvidos, o uso do ImageDW-index proporcionou ganhos, com relação ao trabalho correlato, para a grande maioria dos testes de realizados.

6.3 Considerações Finais

Neste capítulo foram apresentados os testes de desempenho das estratégias de processamento de consultas propostas nesta dissertação de mestrado. Foram realizadas baterias de testes que enfocaram nos predicados de similaridade entre imagens das consultas IOLAP, bem como nos predicados convencionais. Na bateria de testes envolvendo apenas predicados de similaridade entre imagens, as abordagens Especialista e Unificada apresentaram ganhos expressivos com relação ao trabalho correlato em todos os cenários observados. Com relação aos testes envolvendo predicados convencionais, o uso do ImageDW-index proporcionou ganhos, com relação ao trabalho correlato, para a grande maioria dos testes de realizados.

Com relação à bateria de testes envolvendo apenas predicados de similaridade entre imagens, isto é, sem predicados convencionais, a estratégia EspecialistaAcessaDW proporcionou um melhor desempenho no processamento de consultas, com um ganho que variou de 55,57% até 82,16% com relação ao trabalho correlato de Annibal et. al. (2010). Nesse cenário, foi mostrado que para consultas das classes 1 e 2, as quais envolvem camadas perceptuais com vetores de características de baixa dimensionalidade, a estratégia EspecialistaUsaBitmap apresentou um bom desempenho, evidenciando o fato de que a indexação de vetores de características só é vantajosa para baixas dimensionalidades. Foi mostrado também que o trabalho correlato apresentou perda de desempenho à medida que a seletividade da consulta aumentou (mais tuplas de resposta).

Em um cenário envolvendo consultas especificadas em termos de predicados de similaridade entre imagens e de predicados convencionais, foi mostrado que a estratégia UnificadaAcessaDW proporcionou, em média, o melhor desempenho. O ganho médio com essa estratégia foi de 76.76%, se comparada ao trabalho correlato. As consultas da classe 1 e 2, entretanto, foram executadas mais rapidamente de acordo com a estratégia UnificadaUsaBitmap, evidenciando mais uma vez que a indexação dos vetores de características só é vantajosa em cenários de baixa dimensionalidade.

Ainda considerando o cenário envolvendo predicados convencionais nas consultas IOLAP, foi mostrado que para consultas muito restritivas, como consultas envolvendo cinco predicados convencionais (5Conv), as estratégias propostas nesta dissertação não apresentaram ganhos se comparadas ao trabalho correlato. De acordo com os resultados, a estratégia UnificadaAcessaDW foi a que mais se aproximou do desempenho do trabalho correlato para a consulta 5Conv. Por outro lado, à medida que a consulta se torna pouco restritiva, percebe-se, pelos resultados obtidos, que o desempenho do trabalho correlato torna-se inferior ao das estratégias propostas nesta dissertação de mestrado.

Os resultados apresentados neste capítulo evidenciam a eficiência das estratégias de processamento de consultas IOLAP utilizando o ImageDW-index. Para a grande maioria dos cenários avaliados, as estratégias propostas apresentaram ganhos expressivos se comparadas ao trabalho correlato a essa dissertação de mestrado.

No próximo capítulo são feitas as considerações finais desta dissertação de mestrado.

Conclusões

Esta dissertação de mestrado enfocou no processamento de consultas analíticas estendidas com predicados de similaridade entre imagens em ambientes de DWing. As estratégias propostas inserem-se no contexto de DWs de imagens, os quais armazenam não apenas dados convencionais, mas também dados de imagens. Para viabilizar consultas sobre imagens é necessário a extração do conteúdo intrínseco das imagens, representando esses dados em vetores de características, os quais podem ser comparados por meio de consultas por similaridade. Para incorporar essas modificações, segundo adotado nesta dissertação de mestrado, um DW de imagens deve ser projetado segundo um esquema estrela diferenciado, o qual possui dimensões específicas para a manipulação dos dados de imagens. Por meio do conceito de camadas perceptuais, um DW de imagens pode oferecer consultas que enfocam em diferentes aspectos da imagem, como cor, textura ou forma, de acordo com as preferências e necessidades do usuário de SSD.

Ambientes de DWing de imagens podem oferecer suporte a uma nova gama de consultas analíticas, denominadas de IOLAP. Nesta dissertação, essas consultas envolvem operações OLAP do tipo *slice-and-dice* juntamente com consultas por similaridade do tipo *range query*. Consultas dessa natureza são muito custosas, devido ao grande volume de dados que o DW deve lidar e à complexidade das consultas por similaridade.

Nesta dissertação de mestrado foi proposto o ImageDW-index, um mecanismo para a otimização de consultas que consolida duas técnicas bem difundidas na literatura e muito eficientes, que são índices bitmap e a técnica Omni. A principal ideia do ImageDW-index consiste em utilizar os mecanismos de filtragem e de refinamento proporcionados pela técnica Omni, aplicando-se índices bitmap na indexação das distâncias dos vetores de características das imagens até seus elementos representativos, para cada camada percep-

tual. Com essa técnica, a formação da *mbOr* é realizada de maneira muito mais rápida, por meio de operações lógicas bit a bit.

Foram propostas duas abordagens de processamento de consultas IOLAP que utilizam o ImageDW-index. A primeira, denominada Especialista, trata cada parte da consulta em módulos distintos, um especificamente para o processamento dos predicados convencionais e outro para os predicados de imagens. A segunda, denominada Unificada, processa a consulta como um todo. A principal diferença entre as abordagens refere-se à construção do índice bitmap. Para a abordagem Especialista, o índice é construído apenas para as distâncias, enquanto que para a abordagem Unificada, o índice é construído para todos os atributos envolvidos na consulta. Uma vez que os vetores de características introduzem um custo extra no processamento das consultas, devido ao tamanho desse atributo, foi analisado, para cada abordagem, o impacto de se utilizar o DW (Configuração *AcessaDW*) ou o índice bitmap (Configuração *UsaBitmap*) para recuperar esse dado. A combinação das duas abordagens, cada qual com duas configurações, gerou a proposta de quatro diferentes estratégias de processamento, a saber: *EspecialistaAcessaDW*, *EspecialistaUsaBitmap*, *UnificadaAcessaDW* e *UnificadaUsaBitmap*.

Além dessas contribuições principais, nessa dissertação de mestrado foram feitas duas contribuições secundárias. A primeira refere-se ao gerador de dados ImageDW-Gen, uma ferramenta para construção de um DW de imagens de grande porte, a qual foi utilizada na geração do ambiente de testes para esta dissertação de mestrado. A segunda contribuição refere-se à definição de quatro classes de consultas IOLAP, as quais enfocam em diferentes custos do processamento de consultas analíticas que envolvam predicados de similaridade entre imagens, tais como o tamanho dos vetores de características e a quantidade de camadas perceptuais utilizadas na consulta. Essas classes foram utilizadas na definição das consultas escolhidas para as baterias de testes sobre o ImageDW-index.

Foram realizadas várias baterias de testes para investigar as vantagens introduzidas pelas estratégias de processamento de consultas utilizando o ImageDW-index. Primeiramente, foram realizados testes experimentais para identificar qual a melhor configuração do índice bitmap para as estratégias propostas, de modo que o tempo no gasto para processar os predicados de similaridade entre imagens fosse o menor possível em termos de acesso ao índice. Por meio desses testes, descobriu-se que a técnica de codificação por intervalo com cinco bins consiste na melhor configuração para o índice, considerando o conjunto de dados de teste. A partir desse resultado, as estratégias propostas foram analisadas utilizando essa configuração e os resultados foram comparados com a proposta de Annibal et. al (2010). De acordo com os resultados, para consultas que envolvem apenas predicados de similaridade entre imagens, houve um ganho médio de 55.57% até

aproximadamente 82.16% utilizando a estratégia EspecialistaAcessaDW. Com relação às consultas que envolvem tanto predicados de similaridade entre imagens quanto predicados convencionais, em uma bateria de testes, que sucessivamente removia os predicados convencionais mais seletivos da consulta, o ganho médio, considerando o processamento de todas as consultas, foi de aproximadamente 76,76%. Esse ganho foi proporcionado pela estratégia UnificadaAcessaDW.

7.1 Contribuições

As contribuições principais introduzidas por esta dissertação de mestrado são:

- Proposta do ImageDW-index, em termos do mecanismo de otimização de consultas, o qual utiliza a técnica Omni e índice bitmap.
- Proposta de diferentes estratégias de processamento de consultas IOLAP e investigação acerca de seus benefícios.

As seguintes contribuições secundárias foram apresentadas:

- Proposta do gerador de dados ImageDW-Gen.
- Proposta de diferentes classes de consultas com diferentes custos de processamento dos predicados de similaridade entre imagens.

7.2 Trabalhos Futuros

Como continuidade à pesquisa desenvolvida para essa dissertação de mestrado, os seguintes trabalhos futuros podem ser explorados:

- Incorporação de outras medidas numéricas adotadas na construção do DW de imagens.
- Proposta de uma taxonomia para as consultas IOLAP envolvendo outras operações analíticas.
- Incorporação de outros tipos de consultas por similaridade ao contexto de DW de imagens.

- Incorporação de técnicas de realimentação de relevância para o refinamento dos resultados obtidos pelas consultas por similaridade, no contexto de DW de imagens.
- Desenvolvimento de um *benchmark* para DWs de imagens.

Referências Bibliográficas

- Amanatiadis, A., Kaburlasos, V., Gasteratos, A., and Papadakis, S. (2011). Evaluation of shape descriptors for shape-based image retrieval. *Image Processing, IET*, 5(5):493–499.
- Annibal, L. (2011). *iStar: Um Esquema Estrela Otimizado para Image Data Warehouses Baseado em Similaridade*. Dissertação de mestrado em ciências da computação, Centro de Ciências Exatas e de Tecnologia, Universidade Federal de São Carlos.
- Annibal, L., Felipe, J., Ciferri, C., and Ciferri, R. (2010). icube: A similarity-based data cube for medical images. In *CBMS*, pages 321–326. IEEE.
- Antoshenkov, G. (1995). Byte-aligned bitmap compression. *DCC*, 0:476.
- Arigon, A.-M., Miquel, M., and Tchounikine, A. (2007). Multimedia data warehouses: a multiversion model and a medical application. *Multimedia Tools Appl.*, 35(1):91–108.
- Ariyachandra, T. and Watson, H. (2010). Key organizational factors in data warehouse architecture selection. *DSS*, 49(2):200–212.
- Belussi, A. and Faloutsos, C. (1995). Estimating the selectivity of spatial queries using the ‘correlation’ fractal dimension. In *VLDB*, pages 299–310, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.
- Beyer, K. S., Goldstein, J., Ramakrishnan, R., and Shaft, U. (1999). When is ‘nearest neighbor’ meaningful? In *ICDT*, pages 217–235, London, UK, UK. Springer-Verlag.
- Bozkaya, T. and Ozsoyoglu, M. (1999). Indexing large metric spaces for similarity search queries. *TODS*, 24(3):361–404.
- Brin, S. (1995). Near neighbor search in large metric spaces. In *VLDB*, pages 574–584, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

- Carélo, C. C., Pola, I. R., Ciferri, R., Traina, A. J. M., Traina-Jr, C., and Ciferri, C. (2009). The onion-tree: Quick indexing of complex data in the main memory. In *ADBIS*, pages 235–252, Berlin, Heidelberg. Springer-Verlag.
- Carélo, C. C. M., Pola, I. R. V., Ciferri, R. R., Traina, A. J. M., Traina-Jr, C., and Ciferri, C. (2011). Slicing the metric space to provide quick indexing of complex data in the main memory. *JIS*, 36(1):79–98.
- Cha, G.-H. (2004). Efficient and flexible bitmap indexing for complex similarity queries. In Lee, Y., Li, J., Whang, K.-Y., and Lee, D., editors, *DASFAA*, volume 2973, pages 708–720. Springer Berlin Heidelberg.
- Chan, C.-Y. and Ioannidis, Y. E. (1999). An efficient bitmap encoding scheme for selection queries. *SIGMOD Rec.*, 28(2):215–226.
- Chaudhuri, S. and Dayal, U. (1997). An overview of data warehousing and olap technology. *SIGMOD Rec.*, 26(1):65–74.
- Chaudhuri, S., Dayal, U., and Narasayya, V. (2011). An overview of business intelligence technology. *Commun. ACM*, 54(8):88–98.
- Chávez, E., Navarro, G., Baeza-yates, R., and Marroquín, J. L. (1999). Searching in metric spaces. *CSUR*, 33:273–321.
- Chen, M., Song, Y., Sun, Z., Chen, H., and Sang, A. (2008). Multimedia database retrieval based on data cube. In *ICALIP*, pages 1265–1269. IEEE.
- Ciaccia, P. and Patella, M. (2002). Searching in metric spaces with user-defined and approximate distances. *TODS*, 27:398–437.
- Ciaccia, P., Patella, M., and Zezula, P. (1997). M-tree: An efficient access method for similarity search in metric spaces. In *VLDB*, pages 426–435. Morgan Kaufmann Publishers Inc.
- Codd, E. F., Codd, S. B., and Salley, C. T. (1993). *Providing olap (on-line analytical processing) to user-analysts: an it mandate*.
- Datta, R., Joshi, D., Li, J., and Wang, J. Z. (2008). Image retrieval: Ideas, influences, and trends of the new age. *CSUR*, 40(2):5:1–5:60.
- Elmasri, R. and Navathe, S. B. (2011). *Fundamentals of Database Systems*. Pearson, 6nd edition.

- Golfarelli, M. and Rizzi, S. (2009). A survey on temporal data warehousing. *IJDWM*, 5(1):1–17.
- Gonzales, M., Bagchi, K., Udo, G., and Kirs, P. (2011). Diffusion of business intelligence and data warehousing: An exploratory investigation of research and practice. In *HICSS*, pages 1–9. IEEE.
- Gonzalez, R. C. and Woods, R. E. (2006). *Digital Image Processing*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 3rd edition.
- Haralick, R. (1979). Statistical and structural approaches to texture. *IEEE*, 67(5):786–804.
- Haralick, R. M., Shanmugam, K., and Dinstein, I. (1973). Textural features for image classification. *Systems, Man and Cybernetics, IEEE Transactions on*, SMC-3(6):610–621.
- Harinarayan, V., Rajaraman, A., and Ullman, J. D. (1996). Implementing data cubes efficiently. *SIGMOD Rec.*, 25(2):205–216.
- Inmon, W. H. (2005). *Building the Data Warehouse*. Wiley Publishing, 4th edition.
- Jeong, J. and Nang, J. (2004). An efficient bitmap indexing method for similarity search in high dimensional multimedia databases. In *ICME*, volume 2, pages 815–818. IEEE.
- Jin, X., Han, J., Cao, L., Luo, J., Ding, B., and Lin, C. X. (2010). Visual cube and on-line analytical processing of images. In *CIKM*, pages 849–858, New York, NY, USA. ACM.
- Johnson, T. (1999). Performance measurements of compressed bitmap indices. In *VLDB*, pages 278–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.
- Kaster, D. S., Bugatti, P. H., Traina, A. J. M., and Traina-Jr, C. (2010). Fmi-sir: A flexible and efficient module for similarity searching on oracle database. *JIDM*, 1:229–244.
- Kim, W.-Y. and Kim, Y.-S. (2000). A region-based shape descriptor using zernike moments. *Signal Processing: Image Communication*, 16(1-2):95–102.
- Kimball, R. and Ross, M. (2002). *The data warehouse toolkit: the complete guide to dimensional modeling*. John Wiley & Sons.

- MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. *Fifth Berkeley Symposium on Mathematical Statistics and Probability*, 1:281–296.
- Mohania, M., Samtani, S., Roddick, J., and Kambayashi, Y. (1999). Advances and research directions in data-warehousing technology. *AJIS*, 7(1):41–59.
- Nang, J., Park, J., Yang, J., and Kim, S. (2010). A hierarchical bitmap indexing method for similarity search in high-dimensional multimedia databases. *JISE*, 26:393–407.
- O’Neil, P. and Graefe, G. (1995). Multi-table joins through bitmapped join indices. *SIGMOD Rec.*, 24(3):8–11.
- Pola, I. R., Traina, A. J. M., and Traina-Jr, C. (2009). Easing the dimensionality curse by stretching metric spaces. In *SSDBM*, pages 417–434, Berlin, Heidelberg. Springer-Verlag.
- Ponciano-Silva, M., Traina, A. J. M., Azevedo-Marques, P., Felipe, C., and Traina-Jr, C. (2009). Including the perceptual parameter to tune the retrieval ability of pulmonary cbir systems. In *CBMS*, pages 1–8.
- Shoshani, A. (1997). Olap and statistical databases: similarities and differences. In *SIGACT-SIGMOD-SIGART*, pages 185–196, New York, NY, USA. ACM.
- Skopal, T. and Hoksza, D. (2007). Improving the performance of m-tree family by nearest-neighbor graphs. In Ioannidis, Y., Novikov, B., and Rachev, B., editors, *ADBIS*, volume 4690, pages 172–188. Springer Berlin Heidelberg.
- Skopal, T. and Lokoc, J. (2008). Nm-tree: Flexible approximate similarity search in metric and non-metric spaces. In *DEXA*, pages 312–325, Berlin, Heidelberg. Springer-Verlag.
- Stabno, M. and Wrembel, R. (2007). Rlh: bitmap compression technique based on run-length and huffman encoding. In *DOLAP*, pages 41–48, New York, NY, USA. ACM.
- Stockinger, K. and Wu, K. (2007). *Bitmap Indices for Data Warehouses*, chapter 7, pages 157–178. IGI Global.
- Traina, A. J. M., Traina-Jr, C., Ciferri, C., Ribeiro, M. X., and Azevedo-Marques, P. M. (2009). How to cope with the performance gap in content-based image retrieval systems. *IJHISI*, pages 47–67.

- Traina-Jr, C., Filho, R. F., Traina, A. J. M., Vieira, M. R., and Faloutsos, C. (2007). The omni-family of all-purpose access methods: a simple and effective way to make similarity search more efficient. *VLDB*, 16(4):483–505.
- Traina-Jr, C., Traina, A. J. M., and Faloutsos, C. (2000). Distance exponent: A new concept for selectivity estimation in metric trees. *ICDE*, 0:195.
- Traina-Jr, C., Traina, A. J. M., Faloutsos, C., and Seeger, B. (2002). Fast indexing and visualization of metric data sets using slim-trees. *TKDE*, 14(2):244–260.
- Vieira, M. R., Traina-Jr, C., Chino, F. J. T., and Traina, A. J. M. (2004). Dbm-tree: A dynamic metric access method sensitive to local density data. In *SBBD*, pages 163–177.
- Wilson, D. R. and Martinez, T. R. (1997). Improved heterogeneous distance functions. *JAIR*, 6:1–34.
- Wong, H. K. T., Liu, H.-F., Olken, F., Rotem, D., and Wong, L. (1985). Bit transposed files. In *VLDB*, volume 11 of *VLDB*, pages 448–457. VLDB Endowment.
- Wong, S. T. C., Hoo, K. S., Cao, X., Tjandra, D., Fu, J. C., and Dillon, W. P. (2004). A neuroinformatics database system for disease-oriented neuroimaging research. *Acad Radiol*, 11(3):345–58.
- Wu, K. (2005). Fastbit: an efficient indexing technology for accelerating data-intensive science. *JPCS*, page 556.
- Wu, K., Madduri, K., and Canon, S. (2010a). Multi-level bitmap indexes for flash memory storage. In *IDEAS*, pages 114–116, New York, NY, USA. ACM.
- Wu, K., Otoo, E. J., and Shoshani, A. (2006). Optimizing bitmap indices with efficient compression. *TODS*, 31(1):1–38.
- Wu, K., Shoshani, A., and Stockinger, K. (2010b). Analyses of multi-level and multi-component compressed bitmap indexes. *TODS*, 35(1):2:1–2:52.
- Yianilos, P. N. (1993). Data structures and algorithms for nearest neighbor search in general metric spaces. In *SODA*, SODA’93, pages 311–321, Philadelphia, PA, USA. Society for Industrial and Applied Mathematics.
- Zhou, X., Wang, G., Yu, J. X., and Yu, G. (2003). M+-tree: a new dynamical multi-dimensional index for metric spaces. In *ADC*, volume 17 of *ADC’03*, pages 161–168, Darlinghurst, Australia, Australia. Australian Computer Society, Inc.

Zhou, X., Wang, G., Zhou, X., and Yu, G. (2005). Bm+ -tree: A hyperplane-based index method for high-dimensional metric spaces. In Zhou, L., Ooi, B., and Meng, X., editors, *DASFAA*, volume 3453, pages 398–409. Springer Berlin Heidelberg.

Análise das Principais Técnicas de Índice Bitmap

O objetivo dos testes apresentados nesse apêndice foi determinar a estrutura do índice bitmap utilizada pelas estratégias de processamento de consultas IOLAP propostas nesta dissertação de mestrado. Mais especificamente, buscou-se determinar qual combinação de técnicas de *binning* e de codificação que oferece um maior ganho no tempo de acesso ao índice durante o processamento dos predicados de similaridade entre imagens.

Esse apêndice está organizado da seguinte forma. Na seção A.1 são descritos os detalhes dos testes de desempenho realizados. Na seção A.2 são apresentados os resultados obtidos. Finalmente, na seção A.3, são feitas as considerações finais.

A.1 Descrição dos Testes

As seguintes técnicas de codificação foram analisadas: *interval encoding* (InEnc), *interval-equality encoding* (IEEnc) e *binary encoding* (BiEnc). As técnicas que envolvem codificação por intervalo foram escolhidas pelo fato das consultas submetidas ao DW de imagens serem do tipo *two-sided range query*. Com relação ao *binary encoding*, ele foi considerado por apresentar bons resultados na categoria multicomponente, gerando um índice com o número mínimo de bitmaps sem considerar a compressão, e por ser utilizado comercialmente (Wu et al., 2010b).

A base de dados foi criada utilizando o gerador de dados ImageDW-Gen apresentado na seção 5.3, o qual populou o DW de imagens com 1.409.848 de tuplas. As consultas

de interesse são as que envolvem apenas predicados de similaridade entre imagens e o tempo medido em segundos refere-se apenas ao acesso ao índice bitmap, sem considerar as demais etapas do processamento da consulta IOLAP. Dessa forma, foram definidas quatro consultas, uma para cada classe apresentada na seção 5.4, as quais são descritas a seguir:

- Consulta Q1: “Qual a quantidade de imagens similares à imagem de consulta de encéfalo, de acordo com o raio r_q , para a camada perceptual *PL1*.”
- Consulta Q2: “Qual a quantidade de imagens similares à imagem de consulta de encéfalo, de acordo com o raio r_q , para as camadas perceptuais *PL1*, *PL2*, *PL3* e *PL4*.”
- Consulta Q3: “Qual a quantidade de imagens similares à imagem de consulta de encéfalo, de acordo com o raio r_q , para a camada perceptual *PL0*.”
- Consulta Q4: “Qual a quantidade de imagens similares à imagem de consulta de encéfalo, de acordo com o raio r_q , para as camadas perceptuais *PL0*, *PL1*, *PL2*, *PL3* e *PL4*.”

Para cada consulta, o raio de abrangência variou de modo que as seguintes seletividades fossem obtidas em termos da quantidade de tuplas de respostas recuperadas pelo índice: 10, 100, 1.000, 10.000 e 100.000. Cada consulta foi executada cinco vezes e o tempo médio foi calculado. Pelo fato da seletividade ser fixa para as quatro consultas, os resultados apresentados na seção A.2 referem-se à média dos tempos de acesso ao índice.

Assim como justificado na seção 6.1, a escolha dessa faixa de valores de seletividades, entre 10 e 100.000, deve-se ao pressuposto de que usuários de SSD que utilizam um DW de imagens não estão interessados em um conjunto de respostas muito grande, dessa forma os testes de performance realizados nesta dissertação de mestrado utilizam seletividades que não ultrapassem 8% do volume de dados da base.

Os testes foram executados em uma máquina equipada com um processador Intel Core i7-4770 com 3.40GHz, 32 GB de memória RAM, rodando o sistema operacional Linux CentOS 6.5.

A.2 Resultados

A metodologia adotada nessa bateria de testes foi avaliar o desempenho do índice bitmap construído de acordo com cada técnica de codificação isoladamente e em conjunto com

a técnica de *binning*. O número de bins variou entre 5 e 50. Por exemplo, considerando a técnica InEnc, foram executados testes sem *binning* (InEnc), com 5 bins (InEnc-5bins), 10 bins (InEnc-10bins), e assim por diante. Para cada estrutura avaliada (e.g InEnc-10bins), foi construído um índice bitmap apenas sobre os atributos de similaridade entre imagens.

Na seção A.2.1, são descritos os resultados dos testes considerando apenas as técnicas de codificação, enquanto que na seção A.2.2, são discutidos os testes efetuados considerado também o *binning*.

A.2.1 Comparação das Técnicas de Codificação sem *Binning*

As consultas Q1, Q2, Q3 e Q4 foram submetidas ao índice de acordo com as técnicas InEnc, BiEnc e IEEnc, as quais não consideram o uso de *binning*. Os resultados obtidos são sumarizados na Figura A.1. O gráfico ilustra a média dos tempos de acesso ao índice referentes à execução das quatro consultas, considerando cada seletividade.

De acordo com o gráfico da Figura A.1, a técnica de codificação BiEnc apresentou os melhores resultados para todas as seletividades analisadas, considerando o conjunto de dados de testes e as consultas realizadas. Seu comportamento aproximadamente constante pode ser explicado pela própria estrutura do *binary encoding*, o qual, em geral, necessita que todos os bitmaps sejam acessados para responder a qualquer consulta. O fato de IEnc ter apresentado um desempenho inferior ao IEEnc deve-se ao fato de que o segundo é um índice multinível, o qual apresenta uma estrutura hierárquica capaz de responder a consultas com maior rapidez do que índices bitmap simples, como o IEnc.

Figura A.1: Gráfico com o tempo médio obtido da execução das consultas Q1, Q2, Q3 e Q4 em diferentes seletividades, utilizando as técnicas de codificação InEnc, BiEnc e IEEnc isoladamente.

Ressalta-se que, de acordo com a literatura, técnicas multinível são mais rápidas do que as demais técnicas multicomponente (Wu et al., 2010b). Entretanto, conforme os resultados obtidos nesses testes, o desempenho de IEEnc foi inferior ao BiEnc. Esse comportamento atípico pode ser explicado pela seletividade das consultas executadas nessa bateria de testes, a qual considerou uma faixa de valores relativamente pequena se comparada à seletividade adotada nos testes apresentados no *survey* de Wu et. al. 2010, a qual é da ordem de 10^7 .

A.2.2 Comparação das Técnicas de Codificação com *Binning*

Para cada técnica de codificação em análise, foram gerados índices bitmap utilizando também a técnica de *binning*. Dessa forma, foram gerados índices com 5, 10, 15, 20, 25, 30, 35, 40, 45 e 50 bins para IEnc, IEEnc e BiEnc. As consultas Q1, Q2, Q3 e Q4 foram submetidas a cada uma dessas configurações e o tempo médio de execução foi calculado para cada seletividade. Os resultados obtidos são ilustrados na Figura A.2.

Nas Figuras A.2a e A.2c são mostrados os resultados referentes a InEnc e IEEnc. Em ambas as técnicas, a aplicação do *binning* provocou ganhos expressivos de desempenho. Com relação ao InEnc, o melhor desempenho foi obtido utilizando 5 bins, provendo um ganho máximo de aproximadamente 77,3% para a seletividade 1.000, se comparado com

sua versão sem *binning*. Já com relação ao IEEnc, o maior ganho foi proporcionado pela aplicação da técnica com 10 bins, provendo até 45% de ganho de desempenho se comparado à sua versão sem *binning*. Diferente das demais técnicas, o particionamento dos dados para a geração de bins não proveu ganhos de desempenho expressivos para a codificação BiEnc, conforme ilustrado na Figura A.2b. Apenas a opção com 5 bins proveu um pequeno ganho de aproximadamente 3% se comparada à sua versão sem *binning*.

Na Figura A.2d o desempenho das configurações que obtiveram os melhores resultados são comparadas. Conforme o gráfico, a aplicação da codificação InEnc com 5 bins proveu o melhor desempenho. O maior ganho obtido com InEnc-5bins foi para a seletividade 10, que proporcionou um ganho de aproximadamente 20,8% se comparado à BiEnc-5bins.

Figura A.2: Gráficos com os resultados obtidos pela execução de Q1, Q2, Q3 e Q4 sobre índices bitmap configurados de acordo com as técnicas de codificação IEnc, IEEnc e BiEnc com *binning*.

A.3 Considerações Finais

Conforme as discussões apresentadas nesse apêndice, concluiu-se que a melhor configuração do índice bitmap para as estratégias de processamento de consultas IOLAP, utilizando o ImageDW-index, é a aplicação da técnica de codificação *interval-encoding* com cinco bins. Dessa forma, os testes para a validação dessas estratégias, descritos no Capítulo 6, utilizam esses resultados.

A avaliação das técnicas referentes à estrutura do índice bitmap é importante, pois permite que as estratégias de processamento propostas nesta dissertação de mestrado possam ser avaliadas considerando o maior ganho de tempo possível na etapa de análise dos predicados de similaridade entre imagens. Ressalta-se, no entanto, que os resultados apresentados nesse apêndice são dependentes dos dados gerados para os experimentos. Entretanto, em qualquer ambiente de DWing em que se aplique o ImageDW-index, os passos seguidos nesse apêndice podem ser replicados de modo que a mesma eficiência possa ser obtida em outros cenários.