**UNIVERSIDADE FEDERAL DE SANTA CATARINA**
**PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO**

Karina Fasolin

# UMA PROPOSTA PARA EXECUÇÃO DE CONSULTAS COMPLEXAS EM UMA GRANDE BASE DE DADOS DE IMAGENS HORIZONTALMENTE FRAGMENTADA

Florianópolis

2014

**UNIVERSIDADE FEDERAL DE SANTA CATARINA**
**PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO**

# UMA PROPOSTA PARA EXECUÇÃO DE CONSULTAS COMPLEXAS EM UMA GRANDE BASE DE DADOS DE IMAGENS HORIZONTALMENTE FRAGMENTADA

Monografia submetida ao Programa de Pós-Graduação em Ciência da Computação como parte dos requisitos para a obtenção do Grau de Mestre em Ciência da Computação.
Orientador: Prof. Dr. Renato Fileto
Candidato: Karina Fasolin

Florianópolis

2014

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Fasolin, Karina
Uma Proposta para Execução de Consultas Complexas em uma Grande Base de Dados de Imagens Horizontalmente Fragmentada / Karina Fasolin ; orientador, Renato Fileto - Florianópolis, SC, 2014.
85 p.

Dissertação (mestrado) - Universidade Federal de Santa Catarina, Centro Tecnológico. Programa de Pós-Graduação em Ciência da Computação.

Inclui referências

1. Ciência da Computação. 2. Fragmentação horizontal. 3. Buscas por similaridade. 4. Recuperação de informação. 5. Dados complexos. I. Fileto, Renato. II. Universidade Federal de Santa Catarina. Programa de Pós-Graduação em Ciência da Computação. III. Título.

Karina Fasolin

**UMA PROPOSTA PARA EXECUÇÃO DE CONSULTAS COMPLEXAS EM UMA GRANDE BASE DE DADOS DE IMAGENS HORIZONTALMENTE FRAGMENTADA**

Esta Monografia foi julgada aprovada para a obtenção do Título de "Mestre em Ciência da Computação", e aprovada em sua forma final pelo Programa de Pós-Graduação em Ciência da Computação.

Florianópolis, 24 de fevereiro 2014.

---

Prof Dr. Ronaldo dos Santos Mello
Coordenador do Curso

---

Prof. Dr. Renato Fileto
Orientador

**Banca Examinadora:**

---

Prof. Dr. Robson Leonardo Ferreira Cordeiro

Prof. Dr. Eros Comunello

Prof Dr. Mario Antonio Ribeiro Dantas

## RESUMO

Sistemas de recuperação de informação têm se tornado cada vez mais populares e eficientes. Porém, a recuperação de objetos complexos (e.g., imagens, vídeos, séries temporais) ainda apresenta enormes desafios, principalmente quando envolve similaridade de conteúdo. O problema se torna ainda mais intrincado se as condições de busca incluem predicados convencionais conectados logicamente à predicados baseados em similaridade. A otimização de tais consultas é um problema em aberto hoje em dia. Este trabalho valida uma proposta para melhorar o desempenho de consultas que podem ser expressas por conjunções de predicados convencionais e baseados em similaridade. Tal proposta utiliza fragmentação de dados, segundo predicados diversos e compatíveis com predicados utilizados em consultas. A validação da proposta é feita sobre uma grande base de dados chamada CoPhIR a respeito de imagens, com dados convencionais a elas relacionados. Esta base é manipulada em um sistema de banco de dados relacional com extensões para o tratamento de predicados baseados em similaridade, caracterizada segundo a distribuição do seu conteúdo, fragmentada e indexada, com métodos de acesso convencionais e métricos. Verificou-se um melhor desempenho na execução de algumas consultas com cláusulas conjuntivas para filtragem de dados utilizando os fragmentos propostos do que sobre a base completa.


**Palavras-chave:** Recuperação de informação. Dados complexos. Buscas por similaridade. Fragmentação horizontal. Espaços métricos. Métodos de acesso.

## ABSTRACT

Information retrieval systems are growing in popularity and efficiency. However, the retrieval of complex data (e.g., images, video, temporal series) presents huge challenges yet, particularly when it involves content similarity. The problem becomes even more intricate if the search condition includes conventional predicates logically connected to similarity-based predicates. The optimization of such queries is an open problem nowadays. This work validates a proposal for improving the performance of queries that can be expressed by conjunctions of conventional predicates and similarity-based predicates. This proposal employs data fragmentation, according to diverse predicates, that are compatible with the predicates used in queries. The validation of this proposal is done on a large image database, named CoPhIR with conventional data associated with the images. This database is handled in a relational database system with extensions for coping with similarity-based predicates, characterized according to contents distribution, fragmented and indexed, for efficient access with conventional methods and metric methods. The result of the experiments shows that for some queries with conjunctive filtering clauses were executed more efficiently on fragments than by accessing the complete database.


**Keywords:** Information retrieval. Complex data. Similarity search. Horizontal fragmentation. Metric spaces. Access methods.

## LISTA DE FIGURAS

## LISTA DE TABELAS

## LISTA DE ABREVIATURAS E SIGLAS

## SUMÁRIO

# 1 INTRODUÇÃO

Objetos complexos como dados multimídia (imagens, áudio e vídeo), informações geográficas, séries temporais, informações de genoma e sequências de proteínas, entre outros, estão ganhando espaço nos sistemas de informação. O processamento eficiente de consultas sobre grandes coleções de dados complexos é um dos grandes problemas em aberto na computação (BAEZA-YATES; RIBEIRO-NETO, 2011). Além disso, o tratamento da informação em grandes volumes de dados multimídia distribuídos foi definido como um dos "Grandes Desafios da Pesquisa em Computação no Brasil" para o período de 2006 a 2016 (TORRES, 2006).

De acordo com Darmont et al. (2005) dados complexos apresentam ao menos uma das seguintes características: (i) representação complexa com falta de ordenação que faça sentido para todos os usuários e aplicações, por exemplo imagens, sons e vídeos; (ii) coleções estruturadas de várias formas, como por exemplo, bibliotecas digitais e dados em grafos; (iii) várias fontes de dados, por exemplo, sensores, redes sociais; (iv) percepção através de diferentes pontos de vista, por exemplo, diagnóstico médico baseado em imagens e (v) mudança com o tempo, como por exemplo, séries temporais. Estas características combinadas com o grande tamanho e o crescimento considerável de coleções de dados complexos trazem novos desafios para o gerenciamento e a recuperação dos mesmos.

Vários tipos de dados convencionais (i.e., dados de tipos simples, como números e textos), podem ser relacionados a dados complexos para ajudar a descrevê-los e recuperá-los. Alguns dados convencionais podem ser obtidos pelo mesmo dispositivo usado para capturar dados complexos (e.g., tempo, localização espacial e estado de sensores de câmeras modernas quando capturam uma foto) (SINHA; JAIN, 2008), ou então pelas pessoas que os utilizam (e.g., descrições de objetos que aparecem em uma foto). Outros dados relacionados podem ser obtidos por uma série de técnicas que vêm sendo propostas na literatura para suportar anotação automática ou melhorar a qualidade das anotações produzidas por grupos de pessoas (MELUCCI; BAEZA-YATES, 2011).

Gigantescas coleções de dados complexos com dados convencionais relacionados podem ser acumuladas por uma série de sistemas de informação (e.g., coleções compartilhadas de fotos na Web, imagens médicas, etc). Algumas requisições de recuperação de informação podem ser satisfeitas apenas pesquisando dados convencionais relacionados aos

dados complexos (e.g., título, descrição, rótulos, palavras-chaves, laudos). A recuperação por similaridade de conteúdo (*CBIR - Content Based Information Retrieval*) em espaços métricos (ZEZULA et al., 2006; DATTA et al., 2008; SKOPAL, 2010), por outro lado, usa características (*features*) extraídas do próprio conteúdo dos dados complexos (e.g., vetores descrevendo a distribuição de cor, textura ou formas encontradas em imagens). O problema é que as características relevantes e as medidas de similaridade apropriadas para comparar dados complexos variam com a natureza das coleções de dados e as consultas realizadas sobre elas, entre outros fatores. Assim, uma variedade de características e métricas de similaridade têm sido propostas para apoiar CBIR em domínios e aplicações específicos.

Uma proposta teórica para suportar a recuperação eficiente de informação de grandes coleções de dados complexos com dados convencionais a eles associados vem sendo estudada. Tal proposta se apoia na Teoria Relacional estendendo algumas noções semânticas, tais como dependências funcionais, para caracterizar o conteúdo de dados (contexto de dados) de fragmentos horizontais. Predicados que ocorrem com frequência em consultas e são satisfeitos por uma porção considerável de tuplas em um fragmento e por poucos ou nenhum par composto por tuplas de diferentes fragmentos são utilizados para caracterizar o conteúdo de alguns fragmentos. Consultas envolvendo conjunções de predicados podem então ser solucionadas mediante a determinação dos fragmentos cujos contextos de dados (expressos por predicados) tenham sobreposição lógica com alguns predicados explícitos ou implícitos dessas consultas. A busca por tuplas satisfazendo os demais predicados de consulta é executada somente nesses fragmentos. Tal proposta permite inclusive o uso de técnicas distintas (características, métricas de similaridade e métodos de acesso) apropriadas para melhor solucionar tais predicados em fragmentos de dados com propriedades específicas. A nossa hipótese é que tal abordagem tem potencial para resolver consultas sobre os fragmentos, com menor tempo de execução e gerando melhores resultados.

Esta dissertação de mestrado visa contribuir na validação de tal hipótese mediante a realização de experimentos sobre grandes bases de dados reais com vetores de características extraídas de dados complexos e dados convencionais associados. As consultas executadas sobre esta base envolvem conjunções de predicados sobre os dados convencionais com predicados de similaridade. Com a fragmentação e indexação dos fragmentos segundo os seus contextos de dados, espera-se uma melhoria no desempenho das buscas com conjunções de predicados convencionais

e baseados em similaridade, em comparação à execução sobre todo o conjunto de dados. Alguns exemplos das possíveis consultas são apresentados a seguir.

## 1.1 EXEMPLOS MOTIVADORES

Os exemplos a seguir ilustram consultas sobre uma base de dados real, cuja solução pode ser mais eficiente em fragmentos com contexto de dados que se sobrepõe a alguns predicados de consulta.

Considere as seguintes buscas em uma grande coleção de fotos compartilhadas, disponíveis na Web. Essa coleção contém imagens e dados convencionais a elas associados (e.g., tags):

**Q1:** Recupere as $k$ imagens mais similares a uma dada imagem $q_{(im1)}$, e anotada com a tag *"filhote"*.

Existem algumas alternativas para resolver esta consulta. A primeira delas é procurar pelas $k$ imagens mais similares a $q_{(im1)}$ desconsiderando a informação de tag fornecida. Essa abordagem irá recuperar as imagens mais similares, porém se o descritor utilizado não for apropriado para este tipo de imagem corre-se o risco de recuperar imagens que não condizem com a intenção de busca do usuário. Dependendo do tamanho da base de dados, pode ser demorado efetuar todos os cálculos de similaridade para retornar uma resposta ao usuário.

Agora, supondo que essa base já esteja previamente fragmentada pelas tags associadas às imagens. A recuperação pode ser feita utilizando apenas o fragmento que foi construído com a tag *"filhote"*. Utilizar esse fragmento pode otimizar o resultado da consulta, dependendo da razão entre o tamanho da base e o tamanho do fragmento, e também recuperar apenas imagens que estejam no conjunto de interesses do usuário, uma vez que outras pessoas anotaram aquelas imagens com a mesma tag que o usuário está buscando. Isso não é garantido, dado que a mesma palavra pode ter significados distintos para coisas distintas dependendo do contexto, mas provavelmente o resultado será melhorado ao procurar apenas entre as imagens que estão anotadas pela mesma tag utilizada.

Uma evolução natural para recuperar mais fragmentos que possam contribuir para a busca é utilizar não apenas o valor fornecido mas também sinônimos da tag utilizada, ou a tradução do valor deste atributo em outras línguas.

**Q2:** Selecione as $k$ imagens mais similares a uma dada imagem $q_{(im2)}$,

cuja localização é Florianópolis, datadas com o ano 2012 e anotadas com a tag *cão perdido*.

Na execução dessa consulta, para seleção do fragmento a ser utilizado deve-se levar em consideração todas as informações de contexto mencionadas. Cada fragmento considerado na recuperação deve conter imagens que se refiram a *Florianópolis*, *datadas* de *2012* e que contenham a *tag cão perdido*. Após a seleção do fragmento via metadados resolve-se a consulta por similaridade, utilizando uma função de dissimilaridade que recupere as imagens mais similares à $q_{(im2)}$. Note que na seleção dos fragmentos a tag utilizada pode não ser exclusiva, no caso mencionado, diversas outras tags podem ter sido mencionadas para anotar as imagens desejadas, por exemplo, *cão, cão perdido, cão florianópolis*, todas as imagens com essas tags devem fazer parte do fragmento a ser utilizado na recuperação.

## 1.2 FRAGMENTOS HORIZONTAIS E DESCRIÇÃO DO SEU CONTEÚDO

Os exemplos apresentados na seção 1.1 sugerem que a recuperação de informação pode ser feita de maneira mais eficiente pela fragmentação horizontal das coleções de dados, i.e, agrupando os itens dos dados de grandes coleções de dados em fragmentos apropriados (de acordo com os valores de dados) pode-se melhorar o desempenho de consultas. Essa abordagem é crucial para o gerenciamento de algumas grandes coleções de dados que vêm sendo criadas e as coleções de dados gigantescas que virão num futuro próximo. Além disso, se a fragmentação horizontal é feita de acordo com o conteúdo dos dados, é possível usar métodos distintos de recuperação, mais apropriados para cada fragmento. Desta forma pode-se prover métodos de acesso adequados para conteúdos específicos e auxiliar os sistemas de recuperação de informação a melhorarem a precisão dos seus resultados.

Entretanto, para resolver consultas como as apresentadas na seção 1.1 de maneira eficiente, sobre uma grande coleção de dados é preciso fragmentá-la. Simplesmente usar métodos de acesso eficientes para cada tipo de fragmento não é suficiente. Também é necessário descrever e indexar os fragmentos de acordo com o seu conteúdo de modo a encontrar rapidamente os fragmentos e os métodos de acesso apropriados para o contexto de busca.

Dessa forma, é necessário um nível maior de indexação, i.e., indexar os fragmentos horizontais de dados de acordo com seu conteúdo,

data, anotações e outros critérios pertinentes. Além disso, alguns modelos clássicos de dados podem ser úteis para gerenciar fragmentos de dados complexos e dados convencionais relacionados.

Os exemplos da seção 1.1 sugerem que a fragmentação horizontal baseada nas propriedades do conteúdo e predicados usados em consultas pode permitir a otimização do processamento de consultas e estratégias eficientes para sua execução muitas vezes paralelizada. Além disso, essa abordagem tem um grande potencial para produzir resultados de consultas mais completos e precisos, para um conjunto de buscas muito mais vasto do que os suportados pelos sistemas de recuperação de informação atuais.

## 1.3 OBJETIVOS

### 1.3.1 Objetivo Geral

A hipótese a ser avaliada neste trabalho é que haverá uma melhoria de desempenho de consultas complexas sobre grandes coleções de dados se tais coleções forem fragmentadas horizontalmente, de acordo com predicados apropriados, e a recuperação de informação for feita sobre fragmentos da coleção, ao invés de considerá-la no todo. O objetivo deste trabalho é validar essa hipótese na execução de consultas expressas como conjunções de predicados convencionais e baseados em similaridade, sobre bases de dados que possuem objetos complexos e informações convencionais a eles relacionadas.

### 1.3.2 Objetivos Específicos

- Efetuar revisão bibliográfica atualizada sobre o tema;
- Preparar bases de dados a serem usadas como estudo de caso, conhecer a distribuição dos seus valores de dados e se habilitar a manipulá-las de forma adequada nos experimentos;
- Propor fragmentações dessas bases de dados, caracterizar e indexar fragmentos com predicados compatíveis aos utilizados em consultas;
- Executar consultas com conjunções de predicados convencionais e baseados em similaridade sobre essas bases, utilizando índices

convencionais e métricos;

- Avaliar possíveis ganhos de desempenho (tempo de execução, número de acessos a disco, número de cálculos de similaridade, uso de memória, além de precisão e cobertura, na medida do possível) no processamento dessas consultas sobre os fragmentos caracterizados ao invés da base completa.

## 1.4 METODOLOGIA

Primeiramente, é feita uma revisão bibliográfica sobre recuperação de objetos complexos através de dados a eles associados e similaridade.

Em um segundo momento, a base de dados CoPhIR (BOLETTIERI et al., 2009) é estudada, para entender a distribuição de dados encontrada na mesma. Esse entendimento se faz necessário para a definição dos fragmentos de dados, descritores, dados associados, métricas de similaridade, consultas e métodos de acesso a serem empregados em experimentos.

Após essa análise os fragmentos podem ser criados e os experimentos executados. Os fragmentos são definidos por predicados que sejam bons caracterizadores para esses fragmentos e possam ser usados em consultas conjuntivas, tais como as apresentadas na seção 1.1.

Para suportar a resolução eficiente de predicados baseados em similaridade com índices métricos é utilizado o FMI-SiR (KASTER et al., 2009), uma extensão do Oracle para suportar a recuperação de dados complexos por similaridade de conteúdo. O tempo de execução das consultas foi medido utilizando os fragmentos propostos e sobre toda a coleção de dados, para efeito de comparação. Além do tempo de execução, outros fatores como memória utilizada e espaço em disco são considerados.

## 1.5 ORGANIZAÇÃO DO TRABALHO

O capítulo 2 apresenta os conceitos básicos de recuperação de dados complexos, e utilização de similaridade na recuperação de informação, bem como as medidas de desempenho e similaridade utilizadas na recuperação de informação. O capítulo 4 define em detalhe a hipótese que será testada. O capítulo 5 valida a proposta apresentada, detalhando sua arquitetura, apresentando a base de dados utilizada e

os resultados obtidos com a fragmentação das mesmas e execução dos experimentos. O capítulo 3 apresenta os trabalhos correlatos à esta proposta. O capítulo 6 fecha o trabalho, apresentando as conclusões e os trabalhos futuros.

# 2 RECUPERAÇÃO DE DADOS COMPLEXOS

## 2.1 RECUPERAÇÃO DE INFORMAÇÃO

Um dos problemas fundamentais em computação é efetuar recuperação de informação sobre grandes repositórios de documentos ou outros objetos complexos (BAEZA-YATES; RIBEIRO-NETO, 1999). Este ainda é um problema em aberto quando os dados a serem recuperados são dados multimídia (TORRES, 2006). O objetivo da recuperação de informação é recuperar eficientemente todos (cobertura) e somente (precisão) os objetos (documentos, imagens, registros, etc) que satisfaçam os critérios de busca do usuário.

Recentemente houve um grande aumento tanto na quantidade quanto na complexidade dos dados gerados por artefatos sensores e sistemas de coleta de dados. Cada vez mais dados complexos, tais como multimídia, imagens, vídeo, áudio e textos longos são armazenados e de alguma forma precisam ser recuperados (BARIONI, 2006). Para a recuperação desse tipo de dados pode-se utilizar dados convencionais a eles associados, ou características do próprio conteúdo dos objetos a serem recuperados. Essas abordagens serão melhor discutidas na seção 2.2, após a descrição de algumas medidas de desempenho e relevância dos resultados de sistemas recuperação de informação, na seção 2.1.

### 2.1.1 Medidas de Desempenho e de Relevância

A metodologia padrão para avaliação de um sistema de recuperação de informação consiste em avaliar medidas de desempenho (tempo de execução, número de acessos a disco, número de operações efetuadas) para executar consultas e medidas de relevância dos resultados recuperados por tal sistema (BAEZA-YATES; RIBEIRO-NETO, 1999). Medidas de desempenho são relativamente simples, fáceis de coletar e comparar. Usualmente, basta instrumentar o código do sistema para coletá-las e usar algum *benchmark* (base de dados e consultas padronizadas) para permitir a comparação do desempenho de diferentes sistemas. Diversos *benchmarks* têm sido propostos para a recuperação de informação em geral (BAEZA-YATES; RIBEIRO-NETO, 2011; ZEZULA et al., 2006; MANNING; RAGHAVAN; SCHüTZE, 2008) e alguns para dados complexos (LEW et al., 2006; HUISKES; LEW, 2008; OVER et al., ). Vários deles incluem a avaliação da relevância. Porém, os experimentos realizados neste tra-

balho focaram principalmente na análise de desempenho do método proposto uma vez que não existe *benchmark* disponível para grandes coleções de dados complexos e a criação de um *benchmark* para avaliação por si só tem a complexidade de um trabalho. Para melhor avaliar o desempenho, preferimos usar uma grande base de dados complexos com dados convencionais associados, ao invés de um *benchmark* somente com dados convencionais e/ou um volume relativamente pequeno de dados complexos, como a maioria das coleções atualmente disponíveis.

As medidas de relevância dos resultados dependem do conhecimento prévio do conjunto de resultados (regra ouro) a ser recuperado em resposta a cada consulta de um dado conjunto (e.g., benchmark). Algumas das medidas de desempenho de sistemas de recuperação de informação mais usadas na literatura são: precisão, revocação (também chamada cobertura) e medida F, detalhadas nas seções seguintes.

De maneira resumida, precisão é a fração dos documentos recuperados que são relevantes, revocação é a fração dos documentos relevantes observada dentre os documentos recuperados e a medida F associa as duas medidas anteriores para compor uma medida única.

Além dessas medidas, pode-se avaliar o tempo de execução das consultas, o número de acessos a disco (ou de outras operações relevantes, como cálculos de dissimilaridade, por exemplo), a quantidade de disco e memória utilizados.

## PRECISÃO

A precisão mede quanto da informação retornada pelo sistema está correta. Este cálculo é feito dividindo a quantidade de documentos relevantes recuperados pelo número total de documentos recuperados (BAEZA-YATES; RIBEIRO-NETO, 1999). Por exemplo, se para uma busca realizada por um sistema de recuperação de informação forem recuperados 6 documentos e destes apenas 3 forem realmente relevantes, a precisão do sistema é 0,5 ou 50%. A polissemia[1] pode resultar em baixas taxas de precisão, pois pode provocar a recuperação de documentos irrelevantes (referentes a um significado diferente da intenção do usuário).

$$\text{Precisão} = \frac{\sharp \textit{ respostas corretas retornadas pelo sistema}}{\sharp \textit{ total de respostas retornadas pelo sistema}}$$

[1] Polissemia é a propriedade que uma mesma palavra tem de apresentar mais de um significado nos múltiplos contextos em que aparece.

Precisão pode ser esquematicamente observadas na figura 1.

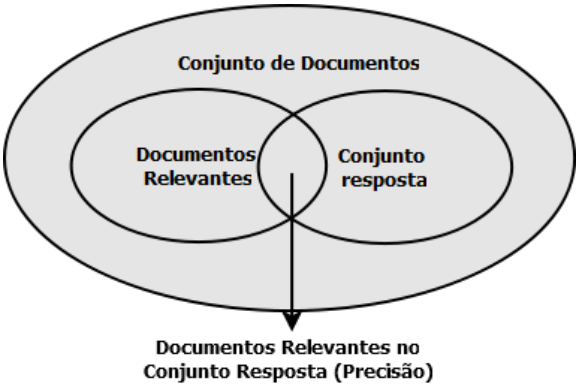


Figura 1 – Precisão. Fonte:(BAEZA-YATES; RIBEIRO-NETO, 1999)

### REVOCAÇÃO

A revocação mede o quanto da informação relevante é recuperada (cobertura da resposta) (BAEZA-YATES; RIBEIRO-NETO, 1999). O usuário do sistema de recuperação de informação pode considerar diversos documentos presentes na base de dados relevantes para a sua consulta, mas é possível que o sistema consiga recuperar apenas alguns deles a partir da consulta fornecida. A taxa de revocação de uma consulta é dada pelo número de documentos relevantes recuperados pelo sistema dividido pelo número total de documentos relevantes existente na base de dados.

$$\text{Revocação} = \frac{\sharp\ respostas\ corretas\ retornadas\ pelo\ sistema}{\sharp\ total\ de\ respostas\ corretas\ no\ sistema}$$

### MEDIDA F

Precisão e revocação normalmente são medidas opostas, enquanto uma aumenta a outra diminui. A Medida F combina esses dois valores em uma média harmônica com peso (BAEZA-YATES; RIBEIRO-NETO, 1999).

$$Medida\ F = \frac{(1+\beta^2) * \boldsymbol{Precisão} * \boldsymbol{Revocação}}{\beta^2 * \boldsymbol{Precisão} + \boldsymbol{Revocação}}$$

Onde $\beta$ é um número real não negativo. Quando $\beta = 1$, precisão e cobertura tem o mesmo peso no cálculo da medida F, reduzindo a equação acima a:

$$Medida\ F = \frac{2 * \boldsymbol{Precisão} * \boldsymbol{Revocação}}{\boldsymbol{Precisão} + \boldsymbol{Revocação}}$$

Dessa forma, quando $\beta > 1$, a precisão é favorecida e consequentemente quando $\beta < 1$, a revocação é favorecida.

Além dessas medidas clássicas, diversas outras medidas de relevância de resultados têm sido propostas na literatura sobre recuperação de informação, incluindo medidas da qualidade do ranking dos resultados (QIN et al., 2010; LIU et al., 2007).

## 2.2 RECUPERAÇÃO DE DADOS COMPLEXOS

Atualmente, tem crescido a demanda por técnicas de recuperação de informação sobre sistemas de bancos de dados, devido à crescente utilização de dados complexos em aplicações. Os SGBDs mais utilizados atualmente são construídos segundo o Modelo Relacional (CODD, 1970), os quais modelam todos os elementos por meio de dois construtores semânticos: atributos e relações, sendo os atributos definidos por tipos de dados textuais ou numéricos. Os dados complexos são normalmente armazenados em um atributo do tipo BLOB (*Binary Large Object*) que é um grande bloco de memória não interpretável, sendo que sua recuperação é feita por atributos numéricos ou textuais armazenados na mesma relação (BARIONI, 2006).

Dados complexos, como mencionado anteriormente, são dados que não têm uma ordenação própria que faça sentido para todas as aplicações. Usualmente não faz sentido em aplicações comparar objetos complexos (faces de pessoas, imagens de raio-X, etc.) usando operadores de ordem ($<, >, \leq, \geq$), como se faz com dados numéricos e textuais em consultas SQL. Então, duas estratégias têm sido utilizadas para permitir a recuperação de informação das coleções de dados complexos: (i) recuperação por metadados ou dados convencionais associados e (ii) recuperação por similaridade de conteúdo (DATTA et al., 2008; BAEZA-YATES; RIBEIRO-NETO, 2011; BAEZA-YATES; MELUCCI, 2011; ZEZULA et al., 2006).

A primeira requer dados textuais ou numéricos para descrever, indexar e recuperar dados complexos. Esta estratégia tem sido amplamente utilizada em máquinas de busca por palavras-chaves na Web. Porém seu sucesso é limitado por dificuldades na anotação manual ou automatizada dos objetos complexos, fenômenos linguísticos e, em certos casos, pela própria dificuldade de identificar e expressar verbalmente o conteúdo relevante em uma imagem.

Na recuperação por similaridade de conteúdo (*Content Based Image Retrieval* - CBIR) (ZEZULA et al., 2006), por outro lado, o usuário pode indicar um objeto complexo ou uma porção do mesmo para especificar os resultados desejados em uma consulta (referindo-se a algum conteúdo similar ao fornecido). Neste caso, é necessário extrair automaticamente um conjunto de características relevantes dos objetos e, a partir dessas características, definir a dissimilaridade (i.e., distância) entre objetos da coleção (BARIONI, 2006).

A figura 2 exemplifica de maneira genérica o funcionamento de uma aplicação com recuperação baseada em conteúdo, considerando o domínio de imagens (SMEULDERS et al., 2000). Como pode ser observado, o sistema tem quatro componentes principais (BARIONI, 2006):

- O componente **Processamento de Imagens** é responsável pela extração das características que representam o conteúdo complexo;
- O componente **Similaridade** define um conjunto de métricas capazes de avaliar similaridade entre objetos complexos;
- O componente **Interação** faz a interface com o usuário, e dá suporte tanto à definição de parâmetros para a consulta sobre objetos complexos quanto à visualização dos resultados;
- E ainda existe o mecanismo de busca, representado pelo componente **Armazenamento e indexação** que realiza as operações sobre o conjunto de dados armazenados.

As técnicas baseadas em conteúdo utilizam características extraídas dos objetos a serem comparados para calcular a distância entre eles, essas características são estruturadas em vetores de características. Tais vetores usualmente contêm valores numéricos extraídos de algum aspecto da imagem (e.g., cores, texturas e formas em imagens) (SMEULDERS et al., 2000).

Para suportar a execução eficiente de consultas com predicados baseados em similaridade é necessário definir uma métrica de dissimilaridade, calculada a partir de vetores de características dos objetos.

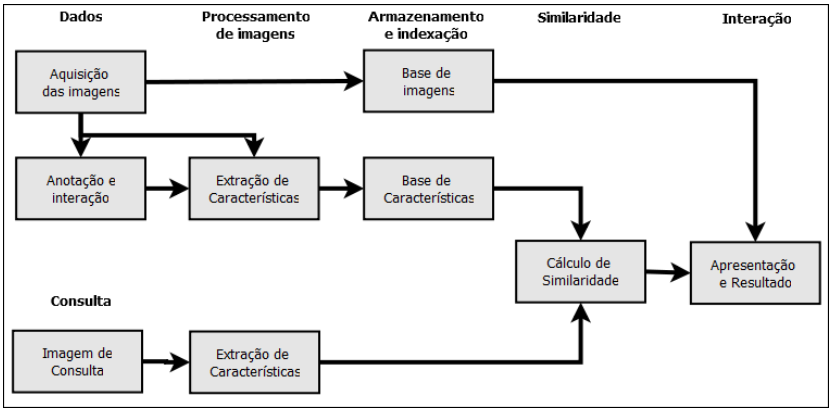


Figura 2 – Ilustração de um esquema genérico de recuperação por conteúdo. Adaptada de (SMEULDERS et al., 2000)

Índices métricos (ZEZULA et al., 2006; SAMET, 2005) são utilizados para acelerar a execução dessas consultas.

## 2.3 EXTRATORES DE CARACTERÍSTICAS

A extração de características é o passo inicial no processamento de dados complexos para recuperação por conteúdo (KASTER et al., 2010). Seu objetivo é extrair os vetores de características dos objetos, para posterior indexação e recuperação por conteúdo baseando-se nessas características.

O processo de extração das características consiste em calcular representações numéricas que caracterizem o conteúdo a ser representado (BARIONI, 2006). Existem duas formas básicas de executar essa extração, a primeira é sobre os dados brutos *(raw data)* e a segunda é sobre os dados transformados, i.e., as representações numéricas são obtidas a partir de um domínio de transformação comprimido (TRAINA; JR., 2003).

Para a representação numérica de imagens normalmente os extratores utilizam os dados brutos (*pixels*) na extração, sendo os dados transformados mais utilizados em ambientes multimídia. Este trabalho tem seu estudo de caso em recuperação imagens. Assim, os descritores de imagens são detalhados a seguir.

### 2.3.1 Domínio de dados de Imagem

Gudivada e Raghavan (1995) utilizam três aspectos para classificar as características normalmente utilizadas em sistemas de recuperação de imagens por conteúdo:

- **Tipo da característica:** Essa divisão compreende a divisão entre primitivas e lógicas. As características primitivas são características de baixo nível (e.g. borda e cor) que podem ser extraídas automaticamente de uma imagem. Por outro lado, as características lógicas (e.g., o significado de cada objeto presente na imagem) são representações abstratas que podem apresentar elementos de uma imagem em diferentes níveis de detalhes (BARIONI, 2006).

- **Nível de Abstração:** Segundo Aslandogan e Yu (1999) o conteúdo visual das imagens pode ser classificado hierarquicamente em 4 níveis de abstração: (i) *pixels* da imagem, que representam informações sobre cor e brilho; (ii) características como bordas, cantos, linhas, curvas e regiões de cores; (iii) características do nível anterior combinadas e interpretadas como objetos e seus atributos. E (iv) conceitos relacionados à percepção humana dos relacionamentos existentes entre um ou mais objetos de uma imagem.

- **Grau de independência do domínio:** As características extraídas podem ser de cunho visual geral, como cor, textura e forma ou de conteúdo visual específico de um domínio. As últimas dependem do domínio da aplicação. Por exemplo, imagens de faces humanas e impressões digitais (LONG; ZHANG; FENG, 2002) requerem a extração de características específicas para se obter precisão e cobertura satisfatórias na sua recuperação.

A seguir, é apresentada uma breve visão geral das características mais comumente utilizadas em CBIR (BARIONI, 2006). Essas características serão melhor detalhadas posteriormente, na descrição do formato de dados utilizado para a descrição de conteúdo de imagens neste trabalho.

- **Cor:** É a característica mais amplamente utilizada, representada normalmente através de um histograma de cores. Esse histograma registra a distribuição da quantidade de *pixels* de uma imagem para cada cor e pode ser comparado a outro histograma de cor

pela soma das diferenças absolutas ou quadráticas do número de *pixels* de cada cor (ASLANDOGAN; YU, 1999).

- **Textura:** Embora seja de fácil percepção para o ser humano, não é fácil defini-la (BARIONI, 2006). Não existem definições formais para essa característica. Segundo Traina (2001) textura é *"um padrão visual em que há um grande número de elementos visíveis arranjados de forma equânime com densidades variadas"*, onde um elemento corresponde a *"uma região de intensidade uniforme de formas simples que se repetem dentro de um intervalo"*.

- **Forma:** A recuperação por forma tem maior custo computacional para CBIR. Segundo Aslandogan e Yu (1999) isso acontece devido à dificuldade de se obter uma segmentação precisa dos objetos de interesse da imagem. A maioria dos descritores de forma existentes utiliza informações geométricas bem definidas no reconhecimento.

## 2.4 MÉTRICAS DE DISSIMILARIDADE E ESPAÇOS MÉTRICOS

Para recuperação de informações por similaridade, o sistema precisa ser capaz de calcular a distância entre dois objetos, para tanto são utilizadas funções de distância capazes de mensurar o quão dissimilares são dois objetos. Quanto maior o valor calculado dessa distância menos similares são os objetos. Se a distância calculada for zero, significa que os objetos comparados são iguais (BARIONI, 2006).

Dado um domínio $\mathbb{D}$ de objetos complexos, uma função de dissimilaridade $\delta : \mathbb{D} \times \mathbb{D} \rightarrow \mathbb{R}^+$ é chamada métrica se e somente se satisfaz as seguintes propriedades (ZEZULA et al., 2006), $\forall x, y, z \in \mathbb{D}$:

1. Simetria: $\delta(x, y) = \delta(y, x)$,
2. Não-negatividade: $\delta(x, y) \geq 0$,
3. Identidade: $\delta(x, y) = 0$, se e somente se $x = y$,
4. Desigualdade triangular: $\delta(x, z) \leq \delta(x, y) + \delta(y, z)$.

Um espaço métrico $\mathbb{M}$ (BOZKAYA; OZSOYOGLU, 1999; CIACCIA; PATELLA, 2002) é definido pelo par $(\mathbb{D}, \delta())$ no qual $\mathbb{D}$ refere-se a um domínio ou coleção de dados e $\delta()$ é uma métrica de distância. O espaço métrico $(\mathbb{D}, \delta())$ permite organizar os objetos complexos de uma coleção $\mathbb{D}$ segundo a métrica de dissimilaridade $\delta$.

Funções de distância/dissimilaridade para dados complexos são usualmente expressas sobre descritores na forma de vetores de características. Quando descritores usados nas comparações são vetores de coordenadas numéricas em um espaço $n$-dimensional com uma distância métrica definida, tem-se um caso particular do espaço métrico, denomidado espaço vetorial (ou espaço vetorial de dimensão finita). Espaços vetoriais e métricos podem ser indexados com estruturas de dados multidimensionais e métricas, respectivamente, para acelerar a execução de consultas baseadas em predicados espaciais e/ou baseadas em similaridade (ZEZULA et al., 2006; SAMET, 2005).

Existem várias famílias de funções de distância que podem ser utilizadas na recuperação de informação. Uma das famílias mais utilizadas é a Minkowski ($L_p$) (ZEZULA et al., 2006). Esta família de métricas pode ser aplicada a domínios vetoriais e é definida matematicamente na Equação 2.1.:

$$d(x, y) = \sqrt[p]{\sum_{i=1}^{n} |x_i - y_i|^p} \tag{2.1}$$

na qual $n$ é a dimensão do espaço vetorial e os valores inteiros p $> 1$ definem as métricas desta família. Os valores p $= 1$, 2 e $\infty$ correspondem às métricas $L_1$ (Manhattan), $L_2$ (Euclidiana) e $L_\infty$ (Chebychev) respectivamente (BARIONI, 2006).

Algumas vezes é necessário normalizar os vetores de características para fazer com que os valores de todas as dimensões dos vetores variem na mesma faixa. Essa normalização assegura que cada dimensão influencie igualmente na determinação da similaridade entre esses vetores (BARIONI, 2006).

Outros exemplos de funções de distância utilizadas em domínios vetoriais são: Mahalanobis (ZEZULA et al., 2006), Camberra (LONG; ZHANG; FENG, 2002) e Kullback-Leibler (WILSON; MARTINEZ, 1997). Diferentes métricas têm resultados com maior ou menor acurácia para diferentes conjuntos de dados, características extraídas dos dados e categorias das consultas (BUGATTI; TRAINA; JR., 2008).

Embora existam várias funções de distância disponíveis na literatura, sua aplicação normalmente é específica. A função de distância que mais se adapta ao domínio é dependente das características extraídas dos objetos complexos e da natureza desses objetos (BARIONI, 2006).

A definição dos espaços métricos com funções de distância permite a execução de consultas por similaridade, detalhadas na próxima

seção.

## 2.5 CONSULTAS POR SIMILARIDADE

Consultas por similaridade usam um objeto complexo de um dado domínio como centro de consulta e uma medida de similaridade entre objetos de tal domínio para recuperar os objetos mais similares ao objeto de consulta (BARIONI, 2006). Tais consultas podem ser expressas com o uso de predicados baseados em uma métrica de similaridade (BARIONI, 2006). Os dois principais tipos de predicados de consulta por similaridade em um espaço métrico $(\mathbb{D}, \delta())$ são:

- **Predicado de consulta por abrangência *(Range Query)*:** Dado um objeto de consulta $s_q \in \mathbb{D}$ e uma distância máxima de busca $\xi \in \mathbb{R}^+$, o predicado de consulta $Range_q(s_q, \xi)$ recupera todos os objetos $s \in \mathbb{D}$ tais que $\delta(s, s_q) \leq \xi$, i.e., objetos a uma distância de $s_q$ menor ou igual a $\xi$.

  A figura 3 (a) exemplifica uma consulta por abrangência em um conjunto de objetos, utilizando diferentes distâncias da família $L_P$.

- **Predicado de consulta aos $k$-vizinhos mais próximos *(KNN Query)*:** Dado um objeto de consulta $s_q \in \mathbb{D}$ e um número natural $k$, a consulta $kNN_q(s_q, k)$ recupera os $k$ objetos de $\mathbb{D}$ mais próximos de $s_q$, i.e.,

  $$\mathbb{D}' = \{s_i \in \mathbb{D} | \forall s_j \in (\mathbb{D} - \mathbb{D}'), |\mathbb{D}'| = k, \delta(s_q, s_i) \leq \delta(s_q, s_j)\}$$

  .

  Um exemplo desse tipo de consulta pode ser observado na figura 3 (b), com $k$=4.

Assim, dada uma imagem de consulta $im_q$ em um espaço métrico $(\mathbb{D}, \delta())$, pode-se recuperar as imagens a até 5 unidades de distância de $im_q$ e as 10 imagens mais próximas de $im_q$ com os predicados $Range_q(im_q, 5)$ e $kNN_q(im_q, 10)$, respectivamente.

Para melhorar o tempo de execução das consultas por similaridade é necessária a indexação dos objetos complexos. Essa indexação é semelhante à indexação de tuplas simples para recuperação de informação, porém requer outras estruturas para indexação, essas estruturas são apresentadas a seguir.

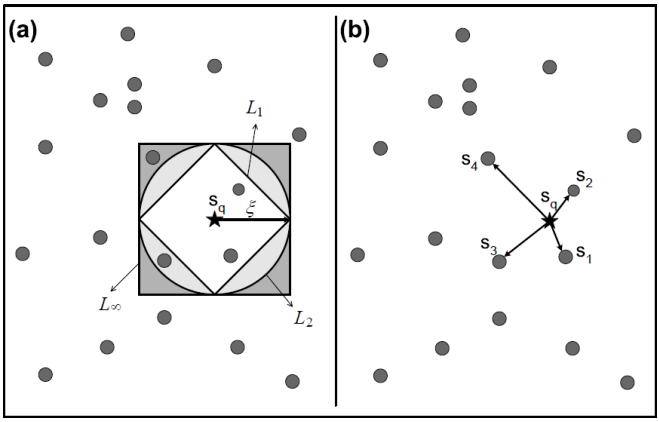


Figura 3 – Exemplos dos predicados de consulta por similaridade (a) Consulta por abrangência, considerando $L_1$, $L_2$ e $L_\infty$ (b) Consulta KNN considerando $k$=4, para distância euclidiana. Fonte: (BARIONI, 2006)

## 2.6 ESTRUTURAS DE INDEXAÇÃO

As consultas apresentadas na seção 2.5 podem ser respondidas através de uma busca sequencial na base de dados, porém essa estratégia não é a mais adequada para grandes conjuntos de dados devido ao alto custo computacional. Dessa forma, outro aspecto importante em CBIR é a utilização de estruturas de indexação apropriadas para espaços métricos. Essas estruturas podem agilizar a realização das consultas por similaridade, minimizando o número de cálculos de distância necessários à execução de uma consulta (BARIONI, 2006).

Estruturas de indexação básicas, como a B-tree, suas variantes e estruturas de *hashing* são normalmente fornecidas pelos SGBDs. Porém essas estruturas são apenas suficientes para trabalhar com dados tradicionais. Elas não são adequadas para recuperação por conteúdo, pois esses dados apresentam alta dimensionalidade e não possuem relação de ordem.

Os Métodos de Acesso Métricos (Metric Access Methods - MAM) são adequados (BARIONI, 2006) para a execução de consultas por similaridade em espaços métricos genéricos (dados espaciais com dimensão definida e dados adimensionais). Segundo Skopal (2010) MAM é um conjunto de algoritmos e estruturas de dados que provê uma busca por

similaridade eficiente (rápida) no modelo de espaço métrico. Os MAMs podem ser utilizados para indexar praticamente qualquer tipo de dado, sendo necessária e suficiente a definição de uma métrica adequada (KASTER, 2012).

Essas estruturas utilizam apenas funções de distância, como as definidas anteriormente, para organizar os objetos na base de dados. Existe uma variedade de MAMs propostos na literatura. As primeiras estruturas propostas com essa finalidade são as chamadas BK-trees (*Burkhard-Keller-trees*). A proposta básica dessas estruturas consiste na escolha de um objeto arbitrário central e na aplicação de uma função de distância para dividir o conjunto de objetos a indexar em vários subconjuntos (BARIONI, 2006). Esse procedimento é executado recursivamente para cada subconjunto não vazio, preenchendo a árvore.

Vários trabalhos têm apresentado propostas de MAMs, incluindo estruturas estáticas e dinâmicas (KASTER, 2012). MAMs estáticos, são aqueles que não podem sofrer novas atualizações/remoções sem degenerar as estruturas criadas. Os mais utilizados são: VP-tree (*Vantage Point tree*) (YIANILOS, 1993) e a MVP-tree (*Multi-Vantage Point tree*) (BOZKAYA; OZSOYOGLU, 1999). O primeiro MAM dinâmico desenvolvido foi a M-tree (CIACCIA; PATELLA; ZEZULA, 1997), que pode ser visto como uma adaptação do método de acesso espacial R-tree para indexar dados em domínios métricos. A M-tree é uma árvore balanceada com crescimento *bottom-up* e com dois tipos de nodos (internos e folha), que permite a inserção de elementos a qualquer momento, mantendo-se sempre balanceada, sem a necessidade de reorganizações periódicas. A Slim-tree (JR. et al., 2002b) é uma evolução da M-tree, que traz como melhorias a avaliação e minimização do grau de sobreposição entre seus nodos e um novo algoritmo de split, baseado na árvore de cobertura minimal (Minimal Spanning Tree – MST) . Outros exemplos de MAMs dinâmicos são a DBM-tree (Density-Based Metric tree) (VIEIRA et al., 2007) e os índices da família Omni (JR. et al., 2007).

Neste trabalho foi utilizado o MAM Slim-tree para a indexação dos objetos complexos. Este MAM foi escolhido por já ter integração com o *FMI-SiR*$_O$ (KASTER et al., 2010) e já ter sido utilizado em diversas avaliações de desempenho, mostrando um desempenho satisfatório. A avaliação de outros MAMs é um dos trabalhos futuros a serem desenvolvidos. O funcionamento da estrutura Slim-tree é detalhado a seguir.

### 2.6.1 Slim-tree

A Slim-tree (JR. et al., 2000) é uma árvore dinâmica balanceada que cresce a partir das folhas em direção a raíz (*bottom-up*). Assim como outras estruturas de acesso métrico (e.g., M-tree e DBM-tree) ela agrupa os objetos de um conjunto de dados em páginas cujo tamanho é fixo, e cada uma dessas páginas é um nodo da árvore (BARIONI, 2006).

Os elementos são armazenados nas folhas, organizados em uma estrutura hierárquica que utiliza um elemento representante como centro de uma região de cobertura dos elementos de uma subárvore, delimitada por um raio máximo de cobertura. Para reduzir os cálculos de distância durante as buscas, as distâncias entre cada elemento de um nodo e o seu respectivo representante são calculadas no momento da inserção e armazenadas na árvore (KASTER, 2012). Sendo assim, cada nodo da árvore (exceto o raiz) mantém registro de um objeto representante, um raio de cobertura e do conjunto de dados que estão cobertos pela região do nodo.

A Slim-tree é composta por nodos de dados (ou folhas) e nodos índice, sendo que cada tipo de nodo armazena até um número máximo pré-definido de objetos. A figura 4 ilustra a organização de nodos, com capacidade máxima de 3 objetos, em um espaço bi-dimensional utilizando a função de distância Euclidiana ($L_2$). Nessa figura os círculos brancos representam nodos folha, os de cor cinza nodos índice. Os elementos representantes de cada nodo são mostrados em preto, e os demais elementos indexados em cor cinza (KASTER, 2012).

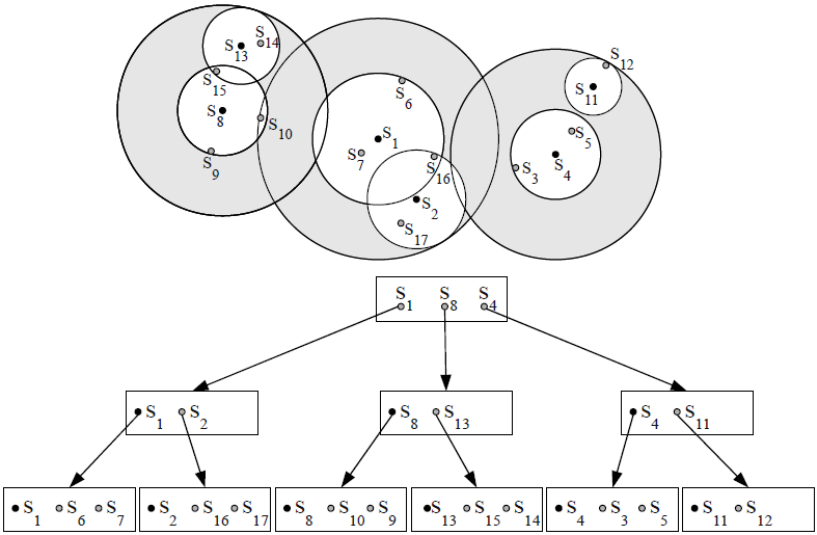


Figura 4 – Representação visual de uma Slim-tree com capacidade nos nodos igual a 3. Fonte: (KASTER, 2012)

# 3 TRABALHOS RELACIONADOS

A revisão bibliográfica efetuada neste trabalho seguiu a técnica de revisão sistemática apresentada em (KITCHENHAM, 2004). Esta técnica permite explorar as fontes de informação existentes de maneira a não fazer uma revisão influenciada por apenas algum aspecto. Kitchenham (2004) coloca que revisões sistemáticas tem por objetivo apresentar uma avaliação justa do tópico de pesquisa, utilizando uma metodologia confiável, rigorosa e que posssa ser auditada.

## 3.1 CONTEXTUALIZAÇÃO

A questão primária avaliada neste trabalho é que haverá uma melhoria de desempenho de consultas complexas sobre grandes coleções de dados se tais coleções forem fragmentadas horizontalmente, de acordo com predicados apropriados.

Dada a complexidade do problema tratado as perguntas consideradas na aplicação da metodologia de revisão bibliográfica utilizada foram quebradas em aspectos do mesmo, para que mais trabalhos relacionados fossem encontrados. Por questões de viabilidade do trabalho, a revisão bibliográfica, tal como os experimentos, focou principalmente em bases de dados de imagens e na análise dos ganhos de desempenho (esforço funcional). Seguem as perguntas utilizadas bem como as palavras-chaves buscadas.

1. **Como efetuar recuperação de informação de grandes volumes de dados complexos e dados convencionais a eles relacionados.**

   Palavras-chave pesquisadas: Multimedia image retrieval; Content-based image retrieval; Cross-media retrieval, Cross-modal retrieval, Image and Text Retrieval, Concept-based Retrieval, Semantic Image Search.

2. **Como melhorar a eficiência (esforço computacional) da recuperação de informação de grandes volumes de imagens com dados convencionais a eles associados.**

   Palavras-chave pesquisadas: efficient image retrieval

Como este é o primeiro trabalho visando a validação da abordagem proposta e devido a limitações de tempo, preferimos deixar outros

aspectos específicos e mesmo a análise da qualidade dos resultados retornados como trabalhos futuros.

## 3.2 MÉTODO DE REVISÃO

### 3.2.1 Fontes Utilizadas e Estratégia de Busca

As bases bibliográficas utilizadas nessa revisão foram as que possuem conteúdo científico mais confiável e acesso aos estudantes da UFSC: IEEE, ACM, Springer Digital Libraries, Google Scholar, BDBComp e DBLP.

Cada palavra chave foi buscada com o seu conteúdo completo, sendo descartados estudos que possuíam apenas uma palavra quando a palavra-chave era composta. Também foram filtrados estudos anteriores ao ano 2000 uma vez que o objetivo é entender o estado da arte atual do tema.

Os resultados obtidos em todas as bases de dados foram agrupados de maneira a remover as duplicatas e então os resumos de todos os artigos foram lidos para uma primeira categorização. Os resultados descartados nessa primeira análise por não se adequarem ao tema discutido estão mencionados em cada uma das palavras chaves na seção seguinte.

## 3.3 DISCUSSÃO

O atual estado da arte na área de CBIR vem sendo estudado e apresentado em alguns trabalhos (DATTA et al., 2008; JAIN, 2008; SKOPAL, 2010). A investigação de descritores de dados apropriados e métodos de acesso para diferentes situações é tema discutido na área (LEW et al., 2006; BLANKEN et al., 2007; TORRES et al., 2009; SKOPAL, 2010). Contudo, apesar de estarem relacionados a abordagem desenvolvida, esses trabalhos são muito específicos, pois normalmente os métodos de acesso desenvolvidos não servem para imagens de cunho geral.

Vários outros trabalhos têm proposto a recuperação eficiente de informação de conjuntos de dados complexos através da exploração de várias propriedades do conjunto de dados e das consultas efetuadas (GOKER; DAVIES; GRAHAM, 2007; DATTA et al., 2008; BAEZA-YATES; RIBEIRO-NETO, 2011; BAEZA-YATES; MELUCCI, 2011; FERREIRA et al., 2011) porém estes trabalhos normalmente limitam a recuperação de

informação a um determinado aspecto do dado, muitas vezes não utilizando recuperação por conteúdo ou as informações contextuais associadas ao mesmo.

Outros trabalhos que utilizam técnicas de modelagem relacional ajudaram a conceber a idéia da fragmentação horizontal, de maneira genérica como foi apresentada no capítulo 4 (CODD, 1979; DATE, 2009; ALASHQUR, 2009, 2010; LIU et al., 2012; SONG; CHEN; YU, 2011; SONG; CHEN, 2011). Conceitos e critérios análogos podem ajudar a conceber fragmentações horizontais apropriadas das relações, i.e., definir fragmentos como seleções de tuplas satisfazendo predicados específicos (RASIWASIA; VASCONCELOS, 2009; CHBEIR; LAURENT, 2010; SONG; CHEN; YUAN, 2011).

Pode-se também avaliar as principais características e propriedades do conjunto de fragmentos gerados por diferentes técnicas de fragmentação, de acordo com os predicados satisfeitos pelo conteúdo desses fragmentos. Isso permite então descrever e indexar grandes coleções de dados fragmentadas horizontalmente com predicados compatíveis aos utilizados em consultas (RASIWASIA et al., 2010; SOARES; KASTER, 2013). Rasiwasia et al. (2010) propõem uma técnica para modelar componentes de texto e imagem em documentos multimídia. São levantadas duas hipóteses: (i) existe benefício em modelar explicitamente correlações entre os componentes de texto e imagem e (ii) a modelagem é mais efetiva quando os níveis de abstração são mais altos. As correlações entre os dois componentes imagem e texto são aprendidas com análise de correlações canônicas e a abstração é alcançada pela representação de textos e imagens em um nível mais geral, semântico. Tal modelagem permite ao sistema recuperar imagens mais próximas a um determinado texto e textos mais próximos a uma determinada imagem fornecida na consulta. A conclusão é que ambos, correlações cross-modais e abstração semântica podem aumentar a acurácia da recuperação (RASIWASIA et al., 2010). Costa Pereira et al. (2013) continuam este trabalho e concluem que ambas as hipóteses melhoram os resultados nas consultas por imagens, por texto e vice-versa. Ambas as hipóteses são complementares, porém existe uma evidência em favor da hipótese de abstração sobre a de correlação. Estes trabalhos apresentam uma maneira diferente de recuperação, utilizando correlações ao invés da recuperação baseada em similaridade de conteúdo. Porém quando os descritores das imagens já estão disponíveis, como é o caso da base de dados utilizada nos experimentos apresentados, é demorado gerar essa nova camada de modelagem de componentes uma vez que o número de entradas no banco de dados é muito grande.

A combinação da similaridade de conteúdo e textual também tem sido investigada para melhorar a recuperação de informação em dados complexos(MURTHY et al., 2009; SANTOS et al., 2009; PEDRONETTE; TORRES, 2011; ESCALANTE; GÓMEZ; SUCAR, 2012; WANG et al., 2008; YANG et al., 2012; GAO et al., 2011).

Alguns trabalhos também utilizam paralelismo e técnicas como MapReduce para acelerar a recuperação de informação e, grandes conjuntos de dados(HIEMSTRA; HAUFF, 2010; ALIPANAH et al., 2011; WU; MAO; CAO, 2011). Entretanto, nessas propostas, o conteúdo dos fragmentos não é definido de acordo com os predicados de consulta, i.e., não é explorada a especificação do conteúdo dos fragmentos para a melhorida da eficiência na execução das consultas.

A proposta aqui apresentada combina diversas ideias dessas propostas anteriores para melhorar o desempenho da recuperação de informação para grandes coleções de dados complexos, com dados convencionais associados. Até onde conhecemos, essa é a primeira proposta geral a tirar vantagem de fragmentos horizontais, definidos conforme predicatos típicos de consulta para acelerar a execução das mesmas. Ela promove a customização de técnicas de recuperação de informação de acordo com o conteúdo de cada fragmento de uma coleção de dados, possivelmente gigantesca, com conteúdo heterogêneo.

# 4 RESOLUÇÃO DE CONSULTAS USANDO FRAGMENTOS HORIZONTAIS

Este capítulo apresenta em detalhes a proposta a ser validada neste trabalho, que foi introduzida na seção 1.2. A hipótese é que se pode conseguir considerável melhoria de desempenho na execução de algumas consultas expressas por conjunções de predicados convencionais e baseados em similaridade, sobre grandes bases de dados complexos com dados convencionais a eles associados, quando a coleção de dados é fragmentada horizontalmente. Isso leva a uma nova abordagem para o processamento de consultas expressas por composições de predicados, a qual consiste na seleção de fragmentos compatíveis com certos predicados, para posterior resolução de outros predicados utilizando métodos de acesso apropriados para o conteúdo dos fragmentos selecionados. A necessidade de fragmentação acontece em grandes bases de dados, uma vez que os índices utilizados atualmente, principalmente os índices métricos, não apresentam desempenho satisfatório para volumes de dados muito grandes.

## 4.1 VISÃO GERAL DA ABORDAGEM PROPOSTA

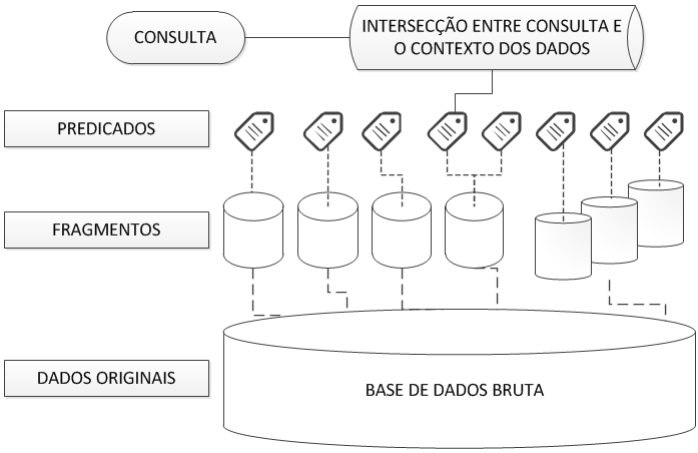


Figura 5 – A abordagem a ser validada

A Figura 5 ilustra de maneira esquemática a abordagem pro-

posta. Sobre uma base de dados com dados complexos e convencionais associados, representada como Base de Dados Bruta na figura, define-se uma série de fragmentos. Por simplificação, neste trabalho não são utilizados fragmentos cujo predicado de criação é desconhecido. Juntamente com os fragmentos definidos sempre será disponibilizado o predicado utilizado para a sua criação. Esses predicados são utilizados na indexação desses fragmentos e na escolha do conjunto de fragmentos a serem utilizados em uma busca.

## 4.2 RELAÇÃO COMPLEXA

A abordagem proposta neste trabalho pode ser implementada sobre bancos de dados relacionais, com algumas extensões para o tratamento de dados complexos e similaridade. Tais extensões incluem relações complexas, descritores de conteúdo de dados complexos, medidas de (dis)similaridade e métodos de acesso eficientes para o tratamento de predicados baseados em similaridade. Uma relação complexa $\mathfrak{R}(S, A)$ mantém dados complexos e dados convencionais a eles relacionados. Ela inclui no seu esquema um conjunto de atributos complexos $S = \{S_1, S_2, \ldots S_m\}$ e um conjunto de atributos convencionais $A = \{A_1, A_2, \ldots A_n\}$. Cada tupla $t \in \mathfrak{R}(S, A)$ relaciona um objeto complexo (e.g., imagem) ou uma série de de tais objetos a seus dados convencionais, tais como tags, título, descrição, horário de carregamento e a posição geográfica onde esses objetos foram obtidos, por exemplo.

Consultas sobre relações complexas podem ser especificadas utilizando predicados convencionais e baseados em similaridade. Predicados convencionais são utilizados para comparar valores dos atributos convencionais com constantes e/ou entre si (e.g., para recuperar as tuplas com um dado valor de tag). Dados complexos (e.g., imagens), por outro lado, normalmente não são comparados por igualdade ($=$, $\neq$), desigualdade ($<$, $\leq$, $\geq$, $>$) ou até mesmo predicados espaciais (e.g., $INSIDE$). Dados complexos são normalmente comparados por similaridade. A noção de similaridade utilizada para Recuperação de Informação (RI) pode variar com o tipo de dados e sua aplicação. Diferentes descritores (e.g., cor, textura e forma) podem ser extraídos dos dados complexos e representados como vetores de diferentes tamanhos. Esses descritores e/ou dados convencionais podem ser comparados utilizando várias funções de similaridade ou dissimilaridade.

## 4.3 FRAGMENTAÇÃO HORIZONTAL

Uma relação complexa $\mathfrak{R}(S, A)$ pode ser fragmentada para o propósito de recuperação de informação pelo uso de uma varidade de métodos. A abordagem para recuperação de informação utilizada neste trabalho permite a fragmentação de qualquer relação complexa pela função:

$$\mathbb{H} : \mathfrak{R}(S, A) \rightarrow 2^{\left(2^{\mathfrak{R}(S,A)} - \{\varnothing\}\right)}$$

A função de fragmentação $\mathbb{H}$ recebe como entrada uma relação complexa $\mathfrak{R}(S, A)$ e sua saída é um conjunto $\mathbb{H}(\mathfrak{R}(S, A))$ de fragmentos horizontais, i.e., subconjuntos de tuplas em $\mathfrak{R}(S, A)$, que respeita as seguintes condições:

1. $|\mathbb{H}(\mathfrak{R}(S, A))| \geq 1$, i.e., $\mathbb{H}(\mathfrak{R}(S, A))$ tem ao menos um fragmento horizontal.

2. $\forall \mathfrak{I}(S, A) \in \mathbb{H}(\mathfrak{R}(S, A)) : |\mathfrak{I}(S, A)| \geq 1$, i.e., cada fragmento horizontal $\mathfrak{I}(S, A)$ tem ao menos uma tupla.

3. Cada fragmento $\mathfrak{I}(S, A) \in \mathbb{H}(\mathfrak{R}(S, A))$ tem o mesmo esquema de $\mathfrak{R}(S, A)$ e contém um subconjunto das suas tuplas.

Um fragmento horizontal $\mathfrak{I}(S, A)$ é composto por um conjunto de tuplas. Cada uma dessas tuplas é uma relação complexa $\mathfrak{R}(S, A)$, tal qual a definida em 4.2. O conteúdo do fragmento $\mathfrak{I}(S, A)$ é caracterizado pelo predicado $\Phi$, o qual pode ter sido utilizado na criação do fragmento, ou apenas descreve seu conteúdo.

Note que os fragmentos nem sempre são partições do conjunto de dados. Os fragmentos podem se sobrepor, ou seja, serem caracterizados por dois predicados que possuem uma intersecção e o conjunto de todos os fragmentos pode não ser a base por completo

Funções de fragmentação podem se basear somente nos valores de subconjuntos de atributos. Uma função de fragmentação com relações complexas $\mathbb{H}_X(\mathfrak{R}(S, A))$ gera subconjuntos de $\mathfrak{R}(S, A)$ pela checagem apenas dos valores da projeção $\Pi_X(\mathfrak{R}(S, A))$. Se $X \subseteq A$ então diz-se que $\mathbb{H}_X(\mathfrak{R}(S, A))$ é baseada na projeção dos atributos convencionais, e se $X \subseteq S$ diz-se que é baseada na projeção dos seus atributos complexos. Todavia, está fora do escopo deste trabalho determinar a melhor forma de fragmentar uma base de dados para fins de recuperação de informação.

Note que é permitido que uma tupla $t \in \mathfrak{R}(S, A)$ apareça em mais de um fragmento $\mathfrak{I}(S, A) \in \mathbb{H}(\mathfrak{R}(S, A))$. Isso é permitido porque o conteúdo de qualquer tupla pode ser interessante para diferentes objetivos de recuperação. Em outras palavras, mesmo que dois fragmentos $\mathfrak{I}$, $\mathfrak{I}' \in \mathbb{H}(\mathfrak{R}(S, A))$ referenciem grupos de dados distintos, algumas vezes eles se sobrepõem, i.e., existem algumas tuplas $t \in \mathfrak{R}$ com $t \in \mathfrak{I}$ e $t \in \mathfrak{I}'$, que permitem sua recuperação de acordo com diferentes pontos de vista.

Embora diferentes fragmentos sirvam para atender diferentes necessidades de informação ou diferentes públicos, eles podem compartilhar conteúdos. Por exemplo, fotos de praias podem ser relevantes para diferentes tipos de pessoas (pescadores, surfistas, viajantes, geólogos, oceanógrafos, etc.). Essas comunidades podem ter interesses distintos e usar noções de similaridade diferentes, que utilizam diferentes aspectos para comparar o conteúdo de dados, sendo assim a mesma tupla pode ser do interesse de pessoas de diferentes comunidades, por razões distintas (e.g., pescadores podem estar interessados em uma textura particular causada pela aproximação dos peixes na superfície da água, os surfistas podem estar procurando por ondas com um formato particular, enquanto alguns viajantes estão apenas procurando por águas cristalinas).

Também é possível que o processo de fragmentação deixe algumas tuplas $t \in \mathfrak{R}$ fora de qualquer fragmento $\mathfrak{I} \in \mathbb{H}(\mathfrak{R})$, i.e., $\exists t \in \mathfrak{R} : (\forall \mathfrak{I} \in \mathbb{H}(\mathfrak{R}) : t \notin \mathfrak{I})$. Isso pode acontecer, por exemplo se $t$ é estranho com respeito ao critério considerado em $\mathbb{H}$ para fragmentar $\mathfrak{R}$ e/ou se $t$ não é do interesse de nenhum foco da recuperação de informação $\mathfrak{I} \in \mathbb{H}$.

## 4.4 CONSULTAS CONJUNTIVAS COM OBJETOS COMPLEXOS

Dada uma relação complexa $\mathfrak{R}(S, A)$, como descrito na seção 4.2, uma consulta conjuntiva com objetos complexos em $\mathfrak{R}(S, A)$ é expressa como uma conjunção de $l$ predicados $\phi_{X_1} \wedge \phi_{X_2} \wedge \ldots \wedge \phi_{X_l}$. Cada predicado $\phi_{X_i} : \mathfrak{R}(S, A) \rightarrow \{TRUE, FALSE\}$ recebe uma tupla $t \in \mathfrak{R}(S, A)$ e retorna $TRUE$ ou $FALSE$ dependendo dos valores $t[X]$ dos atributos $X \subseteq S \cup A$ na tupla $t$. Um tal predicado pode usar operadores de igualdade, desigualdade, espaciais e baseados em similaridade.

Por simplificação, nos experimentos realizados neste trabalho apenas são consideradas consultas conjuntivas com dois predicados atômicos: (i) uma igualdade de atributos convencionais com uma constante (e.g., $tag$ = "*lost dog*", $tag$ = "*dog*"), e (ii) um operador baseado em

similaridade (e.g., $Range_q$ ou $kNN_q$). Por exemplo, a consulta conjuntiva com objetos complexos a seguir recupera tuplas da relação complexa `PhotoSharingData` associadas com a tag `"dog"`, e cujo conteúdo do atributo `picture` tem distância de no máximo `5` unidades da imagem dada `img1.jpg`. Esta consulta utiliza a sintaxe de SQL extendida proposta por (BARIONI et al., 2009) e, para calcular a similaridade, o descritor *scalabe color*, Ite-vil (2009), utilizando a métrica métrica $L_1$ (Manhattan).

```
SELECT R.*
FROM PhotoSharingData R NAT JOIN Tag
WHERE Tag.value = "dog" AND
    R.picture NEAR "D:/images/img1.jpg"
      BY ScalableColor RANGE 5;
```

## 4.5 INDEXAÇÃO DA COLEÇÃO DE FRAGMENTOS HORIZONTAIS

Quando o número de fragmentos horizontais $|\mathbb{H}(\mathfrak{R}(S, A))|$ criados para suportar a recuperação de informação de uma relação complexa $\mathfrak{R}(S, A)$ por vezes é grande, pode ser necessário indexar a coleção de fragmentos $\mathbb{H}(\mathfrak{R}(S, A))$ (e.g., uma coleção com fragmentos definidos pelo valor do atributo *tag*, para um grande número de valores de *tag*) para encontrar de forma eficiente o(s) fragmento(s) que podem resolver os predicados da consulta. O método de indexação para este objetivo pode variar com a natureza dos predicados que definem os fragmentos. Por exemplo, coleções de fragmentos horizontais definidas pelos valores de *tag* podem ser indexadas por um índice convencional (e.g., uma B-tree) ou por um arquivo invertido. Todavia, está fora do escopo deste trabalho determinar a melhor forma de indexar os fragmentos criados.

## 4.6 INDEXAÇÃO INTRA-FRAGMENTO HORIZONTAL

O conteúdo de cada fragmento também deve ser indexado para acelerar a aplicação de filtros adicionais nos dados do fragmento de acordo com outros predicados. Por exemplo, para suportar o processamento eficiente das consultas baseadas em operadores de similaridade (e.g., $Range_q$, $kNN_q$) no conteúdo de fragmentos grandes, um Método de Acesso Métrico (MAM) (ZEZULA et al., 2006) pode ser usado.

Um MAM indexa o conteúdo dos fragmentos em um espaço mé-

trico, definido pelo descritor extraído do conteúdo de algum(ns) atributo(s) e por uma métrica para comparar os dados descritos por similaridade. Vários MAMs têm sido propostos na literatura (ZEZULA et al., 2006; SAMET, 2005), e muitos deles estão disponíveis em SGBDs e ferramentas de RI bem conhecidas (KASTER et al., 2010). O descritor apropriado, a métrica de similaridade e o MAM para suportar o acesso eficiente ao conteúdo dos fragmentos dependem da natureza do conteúdo dos dados e dos predicados de consulta a serem aplicados ao conteúdo do fragmento (DATTA et al., 2008; BUGATTI; TRAINA; JR., 2008; SKOPAL, 2010).

## 4.7 PROCESSAMENTO DE CONSULTAS CONJUNTIVAS UTILIZANDO FRAGMENTOS HORIZONTAIS

A nossa proposta para acelerar a execução das consultas conjuntivas sobre grande bases de dados com objetos complexos e dados convencionais a eles relacionados é utilizar fragmentos horizontais de relações complexas. Esses fragmentos são criados de acordo com os predicados em comum com as consultas. Por exemplo, considere uma base de dados com fotos de diferentes fontes e sobre vários temas (como cidades, casas, escritórios, paisagens, flores, árvores, animais, pessoas, comida, etc.). Essas fotos podem ser organizadas em grupos. Em consequência, a recuperação de uma foto em um grupo específico não precisa necessariamente considerar todas as fotos na base de dados. Ao invés disso, uma abordagem mais eficiente é identificar o(s) fragmento(s) que pode(m) auxiliar a resolver predicado(s) da consulta (i.e., cujo conteúdo satisfaça tal(is) predicado(s), mesmo que parcialmente) e executar a busca considerando apenas o conteúdo desse(s) fragmento(s).

A Figura 6 ilustra a abordagem proposta. Supondo que a base de dados foi dividida em quatro fragmentos, de acordo com os temas `Filhotes`, `Carros`, `Cães` e `Gatos`. O usuário posta uma consulta, e essa consulta é tratada para a identificação dos predicados contidos na mesma. Após a extração desses predicados é idenficado o fragmento, ou fragmentos, a utilizar na execução da busca. De posse desse(s) fragmento(s), a execução da consulta dentro do(s) fragmento(s) depende do seu conteúdo e da natureza dos predicados contidos na consulta.

Na Figura 6 podem ser observados os passos para a execução das consultas propostas na seção 1.1. Na figura essas consultas estão representadas pelo círculo amarelo com o rótulo Q1 e Q2. Relembrando,

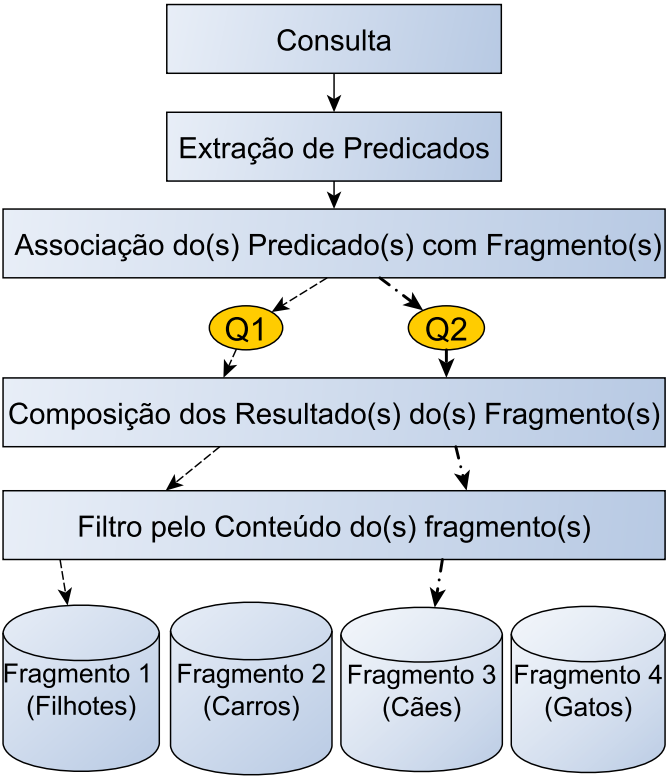


Figura 6 – Estratégias de execução das consultas

as consultas propostas são:

Q1: Recupere as $k$ imagens mais similares a uma dada imagem $q_{(im1)}$, e anotada com a tag *"filhote"*.

Q2: Selecione as $k$ imagens mais similares a uma dada imagem $q_{(im2)}$, cuja localização é Florianópolis, datadas com o ano 2012 e anotadas com a tag *cão perdido*.

Na execução de *Q1* deve ser utilizado o atributo convencional *tag* mencionado na consulta, com o valor *filhote*. Nesse caso, uma busca apenas utilizando as tags deve ser suficiente, e é essa busca que é sugerida na figura. A consulta deve ser executada sobre um fragmento

que possua todas as imagens de filhotes, na falta de um fragmento tão específico, deve ser utilizado um fragmento que possua apenas imagens de cachorros. Para a recuperação por similaridade deve ser usada uma função de distância, a princípio `LP2` (distância euclideana – $L_2$) para recuperar as $k$ imagens mais similares. A figura 7 ilustra o SQL necessário para a execução da consulta Q1 nos experimentos propostos.

```
SELECT frag_name INTO fragment
FROM cophir_frag_catalog
WHERE tag='filhote';
EXECUTE IMMEDIATE
SELECT coeff FROM (
SELECT coeff, ROW_NUMBER()
OVER (ORDER BY MANHATTAN_DIST(coeff, (
SELECT coeff FROM || fragment ||
      WHERE PHOTO_ID=123456)
      )) AS rn
FROM fragment
) WHERE rn <= 5;
```

Figura 7 – Exemplo da consulta Q1 com objetos complexos no Oracle com FMI-SiR$_O$

Na consulta *Q2* é necessária uma combinação de métodos na execução. As informações de localização *Florianópolis* e data *2012* podem ser utilizadas sobre os atributos convencionais uma vez que não devem ser retornadas imagens que não sejam de Florianópolis e do ano de 2012. O fragmento a ser utilizado nessa consulta deve conter imagens de cães perdidos, anotados, pela tag "perdido". Após a busca pela localização e data pode-se efetuar uma busca por similaridade utilizando-se a imagem dada na consulta. A representação SQL da consulta Q2 pode ser observada na figura 8.

Consultas conjuntivas com objetos complexos como as apresentadas anteriormente, quando propostas sobre grandes bases de dados podem ser respondidas de maneira muito mais eficiente pelo acesso apenas do(s) fragmento(s) cujo conteúdo satisfaz algum dos seus predicados, ao invés de buscar em toda a base de dados. Além disso, os dados em cada fragmento podem ser examinados pelo uso de um método de acesso diferente, adaptado ao fragmento em questão. Isso também pode ajudar a obter resultados mais precisos.

Os maiores desafios para implementar as estratégias eficientes de

```
SELECT frag_name INTO fragment
FROM cophir_frag_catalog
WHERE tag='lost', year='2012', place='Florianópolis';
EXECUTE IMMEDIATE
SELECT coeff FROM (
SELECT coeff, ROW_NUMBER()
OVER (ORDER BY MANHATTAN_DIST(coeff, (
SELECT coeff FROM || fragment ||
      WHERE PHOTO_ID=123456)
      )) AS rn
FROM fragment
) WHERE rn <= 5;
```

Figura 8 – Exemplo da consulta Q2 com objetos complexos no Oracle com FMI-SiR$_O$

execução de consultas com vários predicados usando fragmentos horizontais de dados proposta são:

- Particionar a base de dados em fragmentos horizontais que façam sentido para suportar a execução das consultas;
- Encontrar maneiras eficientes de identificar o(s) fragmento(s) que respondem o(s) predicado(s) de consulta;
- Indexar de maneira apropriada o conteúdo de cada fragmento cujo tamanho requeira métodos de acesso eficientes para suportar a verificação dos predicados de consulta;
- Desenvolver estratégias adequadas para otimizar o processamento de consultas utilizando fragmentos apropriados.

A subseção a seguir descreve mais formalmente, uma estratégia genérica para processamento eficiente de consultas usando fragmentos horizontais.

### 4.7.1 Execução das Consultas

O algoritmo 1 descreve a abordagem para processar eficientemente as consultas conjuntivas em grandes bases de dados, utilizando os fragmentos horizontais dessa base de dados e indexação multi-nível.

O usuário, que não sabe da fragmentação da base de dados e dos métodos de acesso, entra com um consulta referenciando toda a base de dados. Essa consulta é recebida no parâmetro *c_query* na linha 1. Primeiramente o sistema de recuperação de informação extrai os predicados da consulta, chamando a função EXTRACT_PREDICATES (linha 2). Então, o sistema escolhe os fragmentos adequados para o processamento da consulta, i.e., os fragmentos cujas tuplas satisfaçam, ainda que parcialmente, algum(uns) do(s) predicado(s) da consulta, através da chamada da função SELECT_FRAGMENTS (linha 3). Um índice construído sobre a coleção de fragmentos pode acelerar a seleção desses fragmentos.

O próximo passo é filtrar as tuplas do(s) fragmento(s) escolhidos, de acordo com o(s) predicado(s) da consulta, chamando a função FILTER_DATA (linha 6). Esta função recebe através do seu segundo parâmetro todos os *predicates* da consulta para verificar o(s) outro(s) predicado(s) no conteúdo do fragmento. Cada fragmento escolhido deve ser menor do que a base completa. Se o tamanho do fragmento ainda é grande, seu conteúdo pode ser indexado e/ou melhor fragmentado para permitir o processamento eficiente de predicados de consulta em particular.

Finalmente, se o processamento da consulta usou mais de um fragmento, o sistema de RI combina os resultados obtidos para cada fragmento, utilizando a função APPEND_RESULTS (linha 7). A combinação de resultados pode usar uniões ou intersecções, dependendo de como a consulta está estruturada e do critério utilizado para escolher os fragmentos, além de outros detalhes da estratégia de execução de cada tipo de consulta, os quais transcendem o escopo deste trabalho e são deixados para trabalhos futuros.

**Algorithm 1** Execução de consultas utilizando fragmentos horizontais

```
1:  function EXECUTE_QUERY(c_query)
2:      predicates = EXTRACT_PREDICATES(c_query)
3:      fragments = SELECT_FRAGMENTS(predicates)
4:      results = ∅
5:      for each f in fragments do
6:          f_results = FILTER_DATA(f, predicates)
7:          results.APPEND_RESULTS(f_results)
8:      end for
9:      return results
10: end function
```

A abordagem aqui proposta é generalista em termos de número, natureza e conexões lógicas dos predicados em uma consulta complexa. Apesar disso, por simplicidade, na implementação atual e nos experimentos apenas são consideradas consultas com predicados convencionais na forma $tag = value$ combinados conjuntivamente com consultas baseadas em similaridade (i.e., $Range_q$ or $kNN_q$). Acredita-se que isso é suficiente para mostrar os benefícios potenciais da abordagem proposta. O capítulo 5 descreve a implementação de um protótipo de sistema de recuperação de informação baseado nesta estratégia e relata experimentos realizados com algumas consultas conjuntivas sobre uma grande base de dados extraídos de mídia social

# 5 VALIDAÇÃO DA ESTRATÉGIA PROPOSTA

Este capítulo descreve a implementação de um protótipo para validar a abordagem proposta e experimentos realizados sobre ele. Sua arquitetura utiliza a ferramenta de recuperação por similaridade *FMI-SiR$_O$* (KASTER et al., 2010) sobre o Oracle. Os experimentos envolvem a execução de consultas conjuntivas sobre a base de dados *CoPhIR*[1] (BOLETTIERI et al., 2009)

## 5.1 FMI-SIR$_O$

O *FMI-SiR$_O$* (user-defined **F**eatures, **M**etrics and **I**ndexes for **Si**milarity **R**etrieval) (KASTER et al., 2010) é um módulo acoplado ao Oracle para executar consultas com predicados baseados em similaridade e funciona de maneira transparente. Esta abordagem permite buscas por similaridade em SGBDs extendendo a linguagem SQL utilizada com operadores baseados em similaridade, tais como $Range_q$ e $KNN_q$.

Este módulo suporta os dois tipos de operadores de consulta por similaridade mencionados na seção 2.5 ($Range_q$ e $kNN_q$), e utiliza MAMs para processar esses predicados em grandes volumes de dados.

Para ser capaz de suportar a base de dados CoPhIR(BOLETTIERI et al., 2009) utilizada nos experimentos o FMI-SiR$_O$ precisou ser extendido para utilizar vetores de características no padrão MPEG-7 disponíveis no CoPhIR.

## 5.2 ARQUITETURA

A figura 9 ilustra a arquitetura implementada. O módulo *Application* recebe a consulta conjuntiva vinda do usuário e repassa a mesma para o módulo *Extract Predicates*. Este módulo é responsável pelo parsing da consulta conjuntiva escrita em SQL extendido suportado pelo *FMI-SiR$_O$*.

Os predicados suportados pela implementação atual podem ser categorizados como:

1. Comparação de igualdade de um atributo convencional com uma constante (e.g., *tag* = "*dog*").

[1] http://cophir.isti.cnr.it

2. Predicados baseados em similaridade em dados complexos ou convencionais ($Range_q$ ou $kNN_q$).

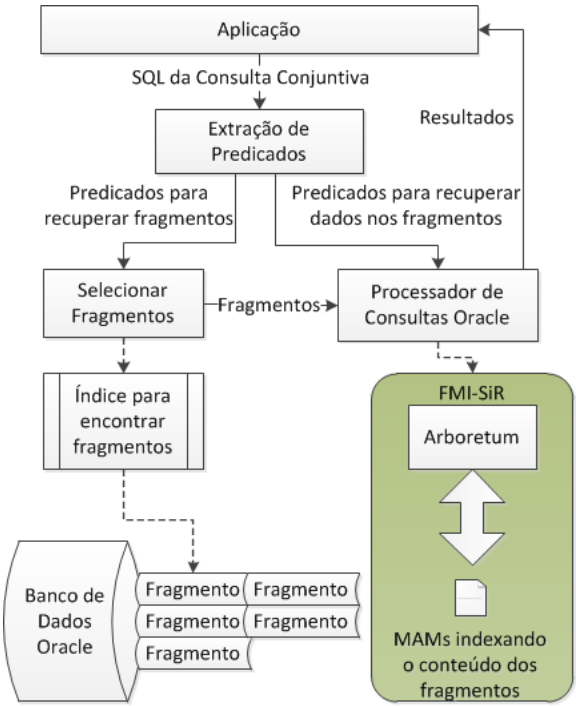


Figura 9 – Arquitetura do Protótipo

Um índice B-tree permite ao módulo *Select Fragments* encontrar eficientemente fragmentos cujas tuplas satisfaçam os predicados de igualdade usado na comparação quando a cardinalidade do atributo é alta, e assim existem muitos fragmentos horizontais para os diferentes valores do atributo (e.g., quando utiliza-se o valor de *tag* para a geração dos fragmentos, cada valor gera um fragmento, se existirem muitos valores de *tag* o número de fragmentos será grande).

Uma vez selecionado(s) o(s) fragmento(s) que se adequa(m) à consulta (i.e., as tuplas que satisfazem algum(ns) predicado(s) da primeira categoria), o *Oracle Query Processor* executa os predicados remanescentes da consulta conjuntiva no conteúdo desse(s) fragmento(s). O FMI-SiR$_O$ faz parte do módulo *Oracle Query Processor* e executa os predicados baseados em similaridade sobre o conteúdo dos fragmen-

tos utilizando a biblioteca de MAM chamada Arboretum[2] (CHINO et al., 2005) para melhorar a performance dessas operações para grandes bases de dados.

Nos experimentos apresentados a seguir foi utilizada a Slim-tree (JR. et al., 2002a), apresentada na seção 2.6.1 como MAM para recuperação eficiente do conteúdo baseado em similaridade nos fragmentos horizontais da base de dados.

Após a recuperação das tuplas que respondem à consulta postada essas são compostas utilizando os operadores da consulta conjuntiva e retornadas para o usuário através do módulo *Application*.

## 5.3 EXTENSÕES AO SQL

A ferramenta SIREN (BARIONI et al., 2006) apresentada a seguir foi cogitada para ser utilizada nos experimentos propostos. Porém devido a dificuldades de instalação, configuração e operação do SIREN optou-se pela utilização do *FMI-SiR*$_O$ (KASTER et al., 2010). Alguns aspectos do funcionamento do SIREN são aqui apresentados a título de comparação e porque alguns também se aplicam ao *FMI-SiR*$_O$, sendo necessários para o entendimento das consultas avaliadas nos experimentos.

A alternativa proposta por Barioni et al. (2009) para suportar consultas por similaridade é extender a linguagem SQL para suportar novos operadores, de similaridade. Nessa extensão os autores incluem novos tipos para dados complexos, instruções de definição de dados para associar atributos de tipos complexos a extratores de características e funções de distância em uma sintaxe enxuta para representação de consultas por similaridade.

Existem dois novos tipos de dados complexos nessa proposta: monolíticos e particulados. Os tipos monolíticos armazenam dados em formato binário, os tipos `STILLIMAGE` e `AUDIO`, por exemplo são utilizados para armazenar imagens e trilhas de áudio. Já o tipo particulado (`PARTICULATE`) é usado na agregação de atributos de uma tabela que representam um dado complexo. Um exemplo de caso do tipo particulado acontece quando tem-se coordenadas geográficas de um elemento armazenadas em dois atributos distintos, definindo-se um atributo do tipo `PARTICULATE` que agrega os dois atributos torna-se possível tratalos em conjunto do processo de avaliação da similaridade (BARIONI, 2006).

[2] http://www.gbdi.icmc.usp.br/arboretum

A instrução `CREATE METRIC` associa um tipo de dado complexo e a instância do espaço métrico desejada (essa instância é formada pelo vetor de características e a função distância). O exemplo a seguir ilustra a definição de uma métrica de similaridade sobre o tipo `PARTICULATE` que utiliza seis vetores de características no cálculo da dissimilaridade, utilizando a função de distância Euclidiana.

```
CREATE METRIC ColorLayoutMetric
  USING LP2 FOR PARTICULATE(
    ydc REAL[], cbd REAL[], crd REAL[],
    yac REAL[], cba REAL[], cra REAL[],
  )
```

onde ColorLayoutMetric é o nome da métrica a ser criada, LP2 indica a distância Euclidiana e os parâmetros: ydc, cdb, crd, yac, cba e cra indicam os vetores de características a serem utilizados, todos com o mesmo peso.

Após a criação da métrica é necessário associá-la a atributos complexos, por meio de restrições do tipo `METRIC`. Seguindo o exemplo anterior, a seguinte instrução cria uma tabela com um atributo `PARTICULATE` associado à métrica *ColorLayoutMetric*:

```
CREATE TABLE ColorLayoutType (
  uri VARCHAR(30) REFERENCES Media,
  YDCCoeff REAL[], CbDCCoeff REAL[], CrDCCoeff REAL[],
  YACCoeff5 REAL[], CbACCoeff2 REAL[], CrACCoeff2 REAL[],
  colorlayout PARTICULATE,
  METRIC REFERENCES (
   YDCCoeff as ydc, CbDCCoeff as cbd, CrDCCoeff as crd,
   YACCoeff5 as yac, CbACCoeff2 as cba, CrACCoeff2 as cra)
  USING (ColorLayoutMetric default))
```

Ao criar a tabela `ColorLayoutType` é feita a referência ao descritor `ColorLayoutMetric`, definido anteriormente, indicando os dados desta tabela a serem utilizados no cálculo da métrica. Nesse caso os vetores de características estão armazenados na tabela, porém também é suportada a utilização apenas das imagens armazenadas, como *BLOB*. Nesse caso, são utilizados extratores para o cálculo das características, por exemplo, histogramas de cor, sem a necessidade de persistência desses vetores em tabelas (BARIONI, 2006).

Após a criação das métricas, e sua devida associação com os dados complexos, é possível a realização de consultas. Essas consultas são representadas em instruções `SELECT` utilizando novas contruções sintáticas. Na cláusula `WHERE`, utiliza-se a seguinte construção:

```
<atributo> NEAR  <Q> [STOP AFTER <k>] [RANGE <x>]
```

onde `<atributo>` é o nome de um atributo complexo a ser comparado, `NEAR` indica que trata-se de uma operação por similaridade, `<Q>` é o conjunto de elementos de consulta e os termos `[STOP AFTER <k>]` e `[RANGE <x>]` indicam, respectivamente, que a consulta é baseada em vizinhos mais próximos e/ou abrangência. A seguinte consulta mostra um exemplo de $k$-vizinhos mais próximos:

```
SELECT * FROM media
  WHERE uri in (
    SELECT uri FROM ColorLayoutType
      WHERE colorlayout NEAR (0.5 as ydc, 64.0 as cbd,
        10 as yac, 5.2 as cba, 8.6 as cra) STOP AFTER 9
```

Esta consulta utiliza o predicado $kNN_q$, (stop after 9) para recuperar os vizinhos mais próximos à imagem cujos valores das características são fornecidos.

Uma consulta muito semelhante pode ser feita utilizando-se abrangência:

```
SELECT * FROM media
  WHERE uri in (
    SELECT uri FROM ColorLayoutType
      WHERE colorlayout NEAR (0.5 as ydc, 64.0 as cbd,
        10 as yac, 5.2 as cba, 8.6 as cra) RANGE 0.1
```

Nessa consulta, todos os objetos cujo $\epsilon = 0.1$ serão recuperados. Também é possível a realização dessas consultas indicando um centro de consulta, ao invés de valores para cada um dos parâmetros da métrica a ser calculada.

Consultas híbridas filtrando os dados também são suportadas. No exemplo a seguir os resultados são filtrados pelo atributo *tag*, e depois é aplicado o predicado $kNN$.

```
SELECT * FROM media, ColorLayoutType, Tags
  WHERE media.media_uri = ColorLayoutType.uri
    and media.media_uri = tag.uri
    and tag.raw = 'tower'
    and colorlayout near (0.5 as ydc, 64.0 as cbd,
      10 as yac, 5.2 as cba, 8.6 as cra) stop after 9
```

Todas essas construções foram feitas no protótipo desenvolvido chamado SIREN[3] (*SImilarity Retrieval ENgine*) (BARIONI et al., 2006).

[3] http://gbdi.icmc.usp.br/siren

Essa ferramenta é capaz de interpretar o SQL extendido e efetuar buscas sobre as bases de dados Oracle e PostgreSQL.

## 5.4 EXPERIMENTOS

O objetivo dos experimentos é avaliar os benefícios da estratégia proposta no desempenho das consultas, i.e., o ganho de desempenho na execução das consultas sobre um fragmento ou sobre um conjunto de fragmentos ao invés da base como um todo. Para tanto é necessário verificar o tempo de execução, o número de acessos a disco, o número de cálculos de funções de (dis)similaridade e o uso de memória e disco no processamento das consultas e apresentação dos seus resultados.

### 5.4.1 Bases de dados

Como a abordagem de fragmentação apenas faz sentido para bases de dados grandes, um dos problemas encontrados foi o levantamento de uma base que tivesse o potencial de melhoria de desempenho esperado. Dentre as bases encontradas, destacam-se:

- ImageCLEF[4] é uma iniciativa mantida desde 2003 como parte do CLEF (*Cross Language Evaluation Forum*) com o objetivo de prover um conjunto de dados para anotação e recuperação de imagens que seja multi-linguagem. A motivação para a criação do conjunto de dados é a necessidade de suportar usuários multilingues da comunidade global acessando um número de informações visuais cada vez maior. O principal objetivo do ImageCLEF é suportar avanços na área de análise visual de mídias, indexação, classificação e recuperação desenvolvendo a infraestrutura necessária para a avaliação da recuperação visual de informação em diferentes tipos de contextos de linguagem. Uma nova versão desse conjunto de dados é lançada a cada ano, na versão de 2012, avaliada no presente trabalho, o ImageCLEF contém 300.000 imagens e 75.000 artigos de biomedicina abertos para análise e redistribuição (MÜLLER et al., 2012).

- O *CoPhIR* (BOLETTIERI et al., 2009) (Content-based Photo Image Retrieval) é uma coleção de metadados multimídia criada para avaliar técnicas escaláveis de busca por conteúdo. Ela contém a

[4] http://www.imageclef.org/

descrição de imagens (vetores de característica MPEG-7) e informação textual (tags, título, descrição, data de upload, usuário, localização) de 106 milhões de imagens extraídas do *FLICKR*[5]. *CoPhIR* não inclui a própria imagem, apenas seus vetores de características MPEG-7 e URLs para a imagem original no FLICKR e seu thumbnail no website do CoPhIR. Os vetores de característica bem como as imagens convencionais associadas são disponibilizadas em um documento *XML* para cada imagem.

Devido ao tamanho da base de dados, e por possuir os vetores de características extraídos, optou-se por utilizar o CoPhIR para a realização dos experimentos. A cada registro existente no Cophir existe uma série de vetores associados, além de uma URI que pode ser utilizada para visualizar a imagem descrita pelos vetores. Esses vetores estão no formato MPEG-7, detalhado na seção a seguir.

O processo de extração, transformação e carga (ETL) dos arquivos XML do CoPhir está fora do escopo desta dissertação. Este processo foi tema de um trabalho de conclusão de curso, do aluno Marcelo Krüger. Esses dados foram colocados em uma base de dados Oracle, e os experimentos descritos nessa dissertação foram executados sobre essa base, e não mais sobre o conjunto de XMLs.

### 5.4.2 MPEG-7

MPEG-7 é um padrão de descrição de conteúdo multimídia. Foi padronizado na ISO/IEC 15938 (*Multimedia content description interface*). Esta descrição deve estar associada ao conteúdo, para permitir a busca rápida e eficiente de dados que interessem ao usuário. Formalmente, o MPEG-7 é chamado de *Multimedia Content Description Interface*, entretanto esse padrão não trata vídeos e audios, como o MPEG-1, MPEG-2 e MPEG-4. Esse padrão usa *XML* para armazenar metadados.

Entre os vários vetores de características disponíveis para descrever as imagens no CoPhIR, foi utilizado o *Scalable color* (ITE-VIL, 2009). Este descritor é derivado do histograma de cores, definido em HSV *(Hue-Saturation-Value)*. Os valores extraídos do histograma são normalizados e mapeados para uma representação não linear de quatro bits. Depois disso uma transformação Haar é aplicada.

Várias funções de distância podem ser utilizadas para recuperar imagens descritas pelos vetores de característica do MPEG-7 (EIDEN-

[5] http://www.flickr.com

BERGER, 2003). Nos experimentos executados foi utilizada a métrica $L_1$ (Manhattan), pois a mesma normalmente produz resultados mais precisos do que as outras métricas simples da família Minkowski, como mencionado na literatura (DORAIRAJ; NAMUDURI, 2004). Este comportamento foi observado nos experimentos preliminares.

### 5.4.3 Configuração dos Experimentos

Os experimentos realizados utilizam uma base de dados relacional, carregada no Oracle, resultante da conversão dos documentos XML disponibilizados pelo CoPhIR. A eficiência das consultas executadas é suportada pelo Oracle (utilizando métodos convencionais de acesso para recuperar os dados convencionais) e pelo FMI-SiR$_O$ (utilizando Slim-trees para indexar o conteúdo das relações).

Os experimentos foram executados em um servidor equipado com um processador Intel®Core™i7 3.8Ghz e and 8GB de memoria RAM. Utilizou-se o Oracle Database 11g com o sistema operacional Debian 7.0 "wheezy" (Kernel 3.2.0 x86-64).

Na realização desses experimentos só foram usados fragmentos cujos predicados utilizados na construção eram conhecidos. Como passo inicial na validação da proposta, foi utilizado para a fragmentação o campo *"tag"* disponível no CoPhir. Essa não é uma limitação da arquitetura, porém, é uma das formas mais simplificadas de caracterizarmos esse conjunto de dados. A medida que esses experimentos forem evoluindo em trabalhos futuros, novos predicados podem ser usados na fragmentação e caracterização, ou até mesmo serem identificados a partir de dados fragmentados a priori.

### 5.4.4 Criação dos Fragmentos

Para o planejamento da fragmentação da base de dados, utilizou-se um histograma que mostra o número de registros existentes na base de dados anotados com cada uma das tags existentes. A figura 10 mostra o histograma para a base completa. Ao todo são 334.254.683 instâncias de tag utilizadas para anotar as imagens, dessas 4.666.256 são tags distintas. Uma tag pode ser utilizada para anotar várias imagens e uma imagem pode ser anotada por várias tags. Pode-se notar nesse histograma que o número de registros para cada uma das tags não segue uma média, ele apresenta muito características de uma lei de potência.

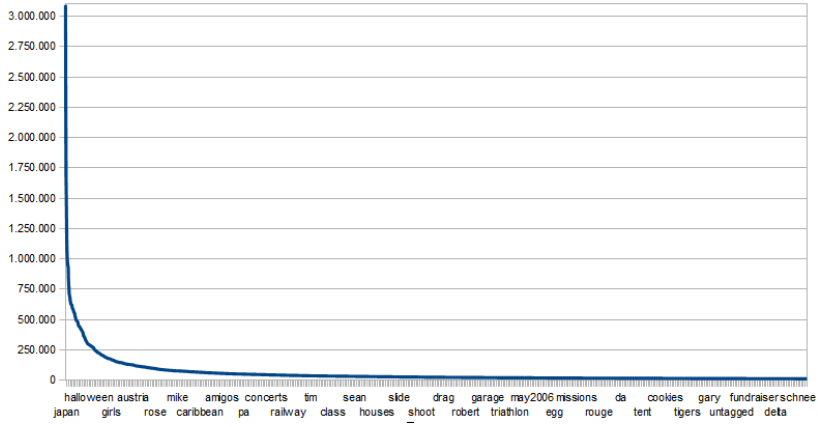


Figura 10 – Histograma de distribuição das tags

A seguinte estratégia foi utilizada para gerar os fragmentos para os experimentos. Primeiramente as tags foram filtradas para eliminar aquelas utilizadas para anotar apenas uma imagem. Esse primeiro filtro deixou 2.111.554 valores de tag distintos, i.e., 45.25% do total de tags distintas. Sobre as tags resultantes se aplicou um segundo filtro baseado nas palavras contidas na WordNet[6] (ontologia de termos da língua inglesa). Este segundo filtro manteve apenas as tags cujos valores existem na língua inglesa, valores de tags como datas ou palavras de outras línguas foram filtrados. A aplicação de tal filtro resultou em 68.767 valores de tags, i.e. apenas 3.25% das tags resultantes do primeiro filtro.

Apesar do conjunto de dados resultante parecer pequeno frente ao total de tags esses filtros são importantes para que os experimentos não sejam mascarados por problemas advindos dos dados brutos. A figura 11 mostra a distribuição dos valores de tag selecionados entre todas as descrições de imagens do CoPhIR. As tags mais frequentes são *"wedding"* (usada para anotar 1.678.711 imagens), *"party"* (1.334.741 imagens) e *"travel"* (1.154.688 imagens). No outro extremo da seleção estão as tags *"algonkin"*, *"precognitive"*, e *"chamberlains"* que anotam apenas 2 imagens cada uma. Essa distribuição foi divida em quartis, como pode ser visto na figura 11, levando a quatro regiões (marcadas com os rótulos R1, R2, R3 e R4).

Foram utilizados 5 valores nos limites de cada região (linha ver-

[6] http://en.wikipedia.org/wiki/WordNet

tical tracejada), perfazendo 10 fragmentos em cada limite de região. Além disso foram escolhidos aleatoriamente 10 valores de tags distintas dentro de cada região para a construção dos fragmentos. Isso dá um total de 80 fragmentos horizontais da base de dados CoPhIR, cada um com as imagens anotadas com um valor de tag escolhido.

O objetivo do experimento é comparar o desempenho da consulta nesses fragmentos com a base completa, contendo 106 milhões de imagens. Para a identificação do fragmento a ser utilizado na busca são utilizadas seleções como *tag* = *"dog"* em meio a consulta a ser executada. Essa é uma limitação grande da validação, porém é a forma mais simples de validação da proposta.

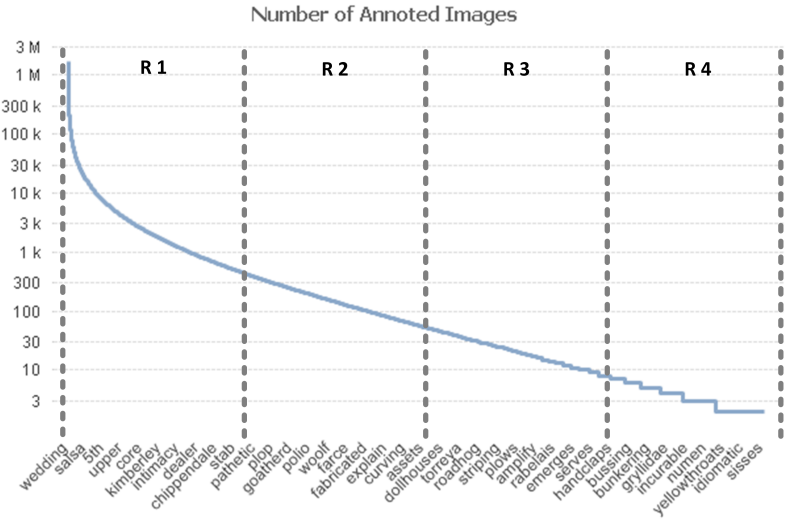


Figura 11 – Distribuição de frequência de valores de tags nas imagens anotadas pelo CoPhIR

### 5.4.5 Consultas Executadas

O próximo passo foi executar as consultas com predicados de igualdade com valores das tags e predicados baseados em similaridade com o conteúdo das imagens. Os valores das tags usados nos predicados de igualdade são os mesmos utilizados para construir os fragmentos para os experimentos (Seção 5.4.4).

Uma imagem, escolhida aleatoriamente, é utilizada como centro da consulta no predicado baseado em similaridade. Após isso, são comparados a média de tempo de execução das consultas em cada um dos fragmentos com a mesma consulta sendo executada na base completa. A figura 12 mostra um exemplo de uma consulta conjuntiva complexa que recupera imagens similares a uma dada imagem no fragmento cujos descritores e metadados das imagens está rotulado como `'puppy'`. Esta figura utiliza a sintaxe do FMI-SiR$_O$ Oracle (KASTER et al., 2010).

```
SELECT frag_name INTO fragment
FROM cophir_frag_catalog
WHERE tag='puppy';
EXECUTE IMMEDIATE
 'SELECT * FROM ' || fragment ||
 ' WHERE MANHATTAN_DIST(coeff,
    (SELECT coeff FROM ' || fragment ||
     ' WHERE PHOTO_ID=123456)) <= 50';
```

Figura 12 – Exemplo de uma consulta com objetos complexos no Oracle com FMI-SiR$_O$

### 5.4.6 Resultados

A seguir são apresentados os resultados obtidos na execução das consultas sobre os fragmentos. Por demandar elevado tempo de processamento também são apresentadas as curvas de tempo de criação dos fragmentos.

#### 5.4.6.1 Criação dos Fragmentos

A figura 13 mostra o tamanho em disco dos índices de cada fragmento criado e também o tempo gasto na criação dos fragmentos horizontais de uma relação com metadados e descritores retirados do CoPhIR.

Cada fragmento é definido por um valor de tag. Os tamanhos dos fragmentos variam com o número de ocorrências de cada tag. A criação do fragmento inclui selecionar as tuplas que referenciam imagens anotadas com determinada tag, a criação de uma tabela para o fragmento

com todos os atributos relacionados a tal imagens, e a construção do índice Slim-tree para suportar a recuperação eficiente das imagens por similaridade de conteúdo. O tamanho em disco do índice de cada fragmento refere-se ao tamanho da Slim-tree criada para a indexação do conteúdo do fragmento. Essa Slim-tree é utilizada pelo *FMI-SiR*$_O$ para a recuperação por conteúdo.

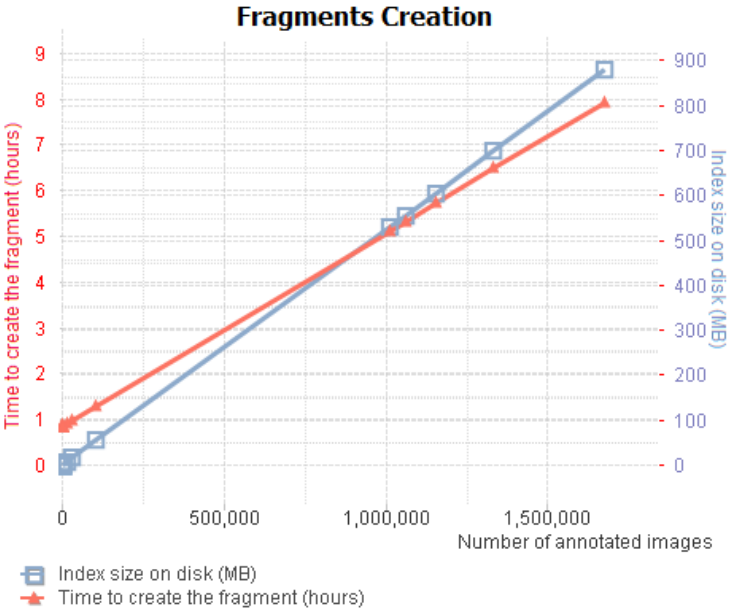


Figura 13 – Tamanho do índice do fragmento em disco e tempo gasto na sua criação.

No gráfico da figura 13 nota-se um comportamento linear, quanto maior o fragmento maior o tamanho da sua Slim-tree em disco. Para a execução das consultas também foi criada a Slim-tree para base completa, com 106 milhões de entradas. A criação desse índice tomou 20 dias de processamento e o arquivo gerado tem aproximadamente 56 GB. Dessa forma é notável que, além da diminuição do tempo de execução nas consultas complexas, a indexação de bases de dados muito grandes requer muito tempo e espaço em disco, o que pode inviabilizar sua construção.

#### 5.4.6.2 Execução das consultas

A figura 14 apresenta o número de acessos a disco e o número de cálculos de similaridade feitos na execução de uma consulta análoga a apresentada na Figura 12 nos fragmentos criados. O tempo de execução engloba (i) a busca na B-tree para encontrar o fragmento que contém as tuplas anotadas com a tag que aparece no predicado convencional, e (ii) a solução do predicado baseado em similaridade em uma Slim-tree que indexa apenas o conteúdo do fragmento.

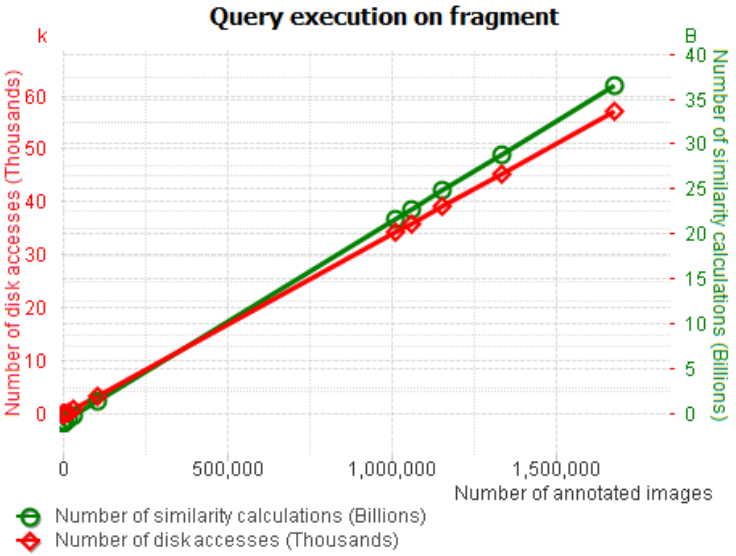


Figura 14 – Número de Acessos a Disco e Número de Cálculos de Similaridade na execução das consultas com fragmentos de diferentes tamanhos

Quando utilizam-se buscas por similaridade, as medidas de desempenho são análogas às de buscas convencionais (tempo de execução, cobertura, precisão, etc.). Cabe observar apenas que o cálculo de algumas métricas complexas (e.g., Mahalanobis) pode demandar muito tempo de CPU e que os métodos de acesso métricos ainda estão em evolução, além de serem tipicamente menos eficientes que métodos convencionais tais como índices baseados em B-Trees. Infelizmente, o descritor da imagem, a função de similaridade, índices e fragmentos utilizados nesses experimentos não garantem um crescimento sub-linear do tempo

gasto na execução dos predicados baseados em similaridade conforme o crescimento do tamanho dos fragmentos.

A figura 15 mostra o tempo de execução da consulta em cada fragmento bem como a quantidade de memória utilizada em cada execução. Como trata-se de uma consulta Range a memória utilizada varia de acordo com o número de respostas retornadas pela consulta. Quanto mais imagens próximas ao centro de consulta maior a memória utilizada.

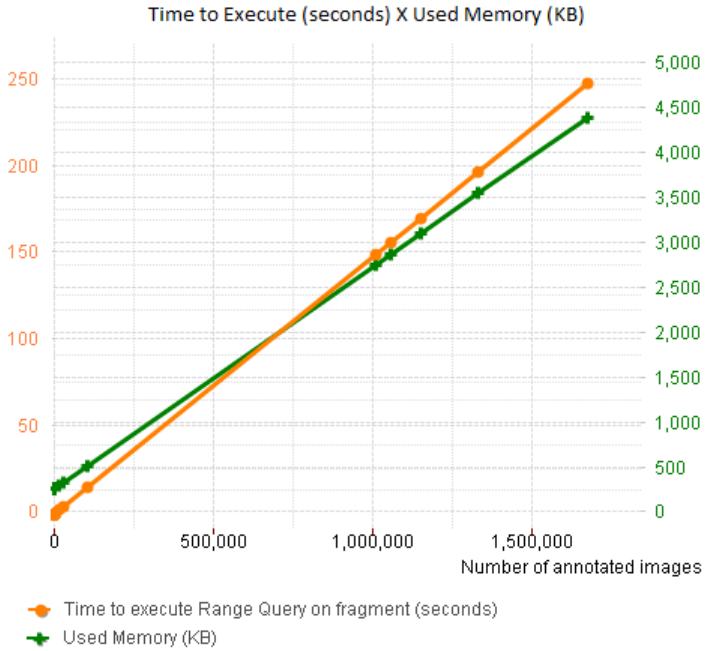


Figura 15 – Tempo de execução de cada consulta no fragmento e memória utilizada na execução

A execução das consultas utilizando a Slim-tree que indexa o conteúdo do fragmento é em torno de uma ordem de magnitude mais rápida que a utilizando a Slim-tree que indexa a base completa, para a maioria dos fragmentos. Ainda é 10 vezes mais rápido executar as consultas no maior fragmento do que na base completa. Pode-se observar na tabela 1 a comparação do tempo de execução no fragmento com o tempo de execução na base completa para diferentes tamanhos de fragmentos.

Tabela 1 – Tempo de execução das consultas no fragmento e na base completa

| Tag | Número de tuplas do Fragmento | Execução no Fragmento (segundos) | Execução na Base Completa (segundos) |
|---|---|---|---|
| wedding | 1.678.711 | 1.200 | 18.577 |
| party | 1.334.741 | 655 | 18.481 |
| travel | 1.154.688 | 672 | 17.597 |
| japan | 1.061.237 | 304 | 14.831 |
| family | 1.011.848 | 258 | 15.531 |
| puppy | 105.570 | 108 | 13.166 |
| bush | 31.851 | 1 | 7.847 |
| gaming | 16.508 | 1 | 10.064 |
| provo | 2.670 | 0,046 | 8.132 |
| precognitive | 2 | 0,027 | 10.956 |
| chamberlains | 2 | 0,029 | 12.514 |

Por exemplo, a execução de uma consulta para recuperar as imagens anotadas com a tag *"wedding"* (1.678.711 imagens) com um raio de distância igual a 50 de uma dada imagem gasta em torno de 1.200 segundos utilizando a Slim-tree para o fragmento, a mesma consulta gasta 18.577 segundos na execução na base completa. Por outro lado, uma consulta para recuperar imagens com a tag *"chamberlains"* (apenas 2 imagens) e com o mesmo raio de distância, 50, de uma dada imagem gasta menos de um segundo utilizando seu respectivo fragmento, e 12.514 segundos utilizando o índice Slim-tree para a base completa.

A tabela 2 apresenta a média dos resultados apresentados para cada um dos quartis utilizados. Para melhor análise dos dados o primeiro quartil, R1, foi quebrado em duas faixas, entre 1 e 2 milhões de tuplas e abaixo de 100.000 tuplas. O primeiro quartil compreende 20 tags, o limite superior (5 tags) todas com mais de 1 milhão de aparições. Já as tags escolhidas aleatoriamente começam sua distribuição em 100.000 aparições e no limite inferior do quartil as tags aparecem 462 vezes. Isso demonstra o quanto a frequência de anotações com cada tag cai rapidamente. Uma melhor análise desse quartil deve ser feita para que se analise a partir de qual número de tuplas é necessário a fragmentação.

As 5 tags mais utilizadas (wedding, party, travel, japan e fa-

mily) todas tem mais de 1 milhão de tuplas. Elas estão representadas na primeira coluna da tabela. Na segunda coluna estão representadas as outras tags de R1 (15 tags). Nas colunas posteriores estão representados os próximos quartis, que não tem uma distribuição tão desigual quanto o primeiro quartil.

Tabela 2 – Médias dos resultados por quartil

| Médias | R1: 1-2 M tuplas | R1: 100.000 a 462 tuplas | R2: 400 - 50 tuplas | R3: 58 - 9 tuplas | R4: 9 - 2 tuplas |
|---|---|---|---|---|---|
| Execução no fragmento (s) | 617,6 | 9,27 | 0,0511 | 0,0519 | 0,048 |
| Execução na base completa (s) | 17003 | 8995,2 | 8467,1 | 8456,2 | 8102,3 |
| Criação dos fragmentos (h) | 6,034 | 0,926 | 0,8675 | 0,873 | 0,891 |
| Número de cálculos de similaridade | 2.08e+10 | 5206,06 | 680,04 | 104,02 | 32,09 |
| Tamanho do índice em disco (MB) | 654,24 | 5,69 | 0,11 | 0,029 | 0,015 |

Nesta tabela pode-se notar o quanto a abordagem com fragmentação é eficiente na consulta sobre o fragmento ao invés da base completa. Nos maiores fragmentos, entretanto, o tempo de resposta não é aceitável e novas otimizações devem ser estudadas para chegar num tempo de resposta razoável. Nota-se que a execução na base completa tem um pico na primeira coluna e para as próximas colunas é praticamente constante. Isso deve-se principalmente ao fato da execução de uma range query. Como os primeiros fragmentos são maiores a tendência é que muito mais resultados sejam retornados, ja em fragmentos menores, como menos comparações são executadas o tempo quase não é alterado.

As outras medidas apresentadas se comportam da maneira esperada, para os fragmentos gigantes o tempo de criação, espaço em disco

utilizado pelo índice e número de cálculos de similaridade são expressivamente maiores, o que faz sentido dado a diferença do número de tuplas utilizada.

Finalmente, a figura 16 apresenta os resultados de uma consulta conjuntiva com objetos complexos com o predicado de igualdade $tag$ = *"puppy"* e um predicado $kNN_q$ com centro na imagem apresentada no canto esquerdo no topo (destacada pelo quadrado vermelho). Estas 4 imagens foram ranqueadas nos resultados da esquerda para a direita e do topo para a base. A execução desta consulta utilizando o fragmento referente a tag *"puppy"*, que contém as descrições de 105.570 imagens, levou 108 segundos utilizando uma B-tree para encontar o fragmento e uma Slim-tree para processar o predicado baseado em similaridade neste fragmento.

Figura 16 – Resultados da consulta executada utilizando o fragmento que descreve apenas imagens com a tag *"puppy"*

Como o predicado $kNN$ não é comutativo com outros predicados de filtro de dados (FERREIRA et al., 2011) são apresentados na figura 17 os resultados de uma consulta por um predicado $Range_q$ com raio igual a 50 e centro na imagem no canto superior esquerdo. Estes resultados foram produzidos utilizando a Slim-tree que indexa a base completa. Esta consulta levou 13.176 segundos para ser executada o que significa quase 100 vezes mais tempo de processamento do que a consulta na figura 16. Filtrar seus resultados pela tag *"puppy"* para produzir os

resultados mostrados na figura 16 requer maior processamento, mas o tempo para executar o predicado $Range_q$ na Slim-tree que indexa a base completa é dominante.

Todas as imagens apresentadas nas consultas executadas foram obtidas através da URL do FLICKR disponível na base de dados do CoPhIR.


Figura 17 – Resultados do predicado $Range_q$ na base de dados completa

A maior dificuldade encontrada até o momento é o tamanho da base de dados que vem sendo utilizada. Devido ao seu tamanho, tanto em número de entradas quanto em espaço em disco qualquer processamento que necessite mudanças na maioria das tuplas torna-se demorado.

# 6 CONCLUSÕES E TRABALHOS FUTUROS

Este trabalho apresenta e inicia a validação de uma abordagem para processar eficientemente consultas expressas como composições de predicados em grandes bases de dados utilizando fragmentos horizontais apropriados dessa base de dados e indexação multinível. Embora a abordagem proposta possa ser aplicada a consultas com diversos tipos de composições de predicados, este trabalho focou na validação da estratégia para o processamento de consultas conjuntivas, sendo a mesma simplificada em três passos:

1. Encontrar fragmentos com dados que satisfazem algum(ns) predicado(s) de consulta;

2. Filtrar os dados no(s) fragmento(s) escolhidos de acordo com outros predicado(s) conectados conjuntivamente ao anterior;

3. Compor os resultados obtidos de cada fragmento.

A revisão bibliográfica atualizada sobre o tema mostrou que ainda existe uma grande lacuna na teoria de fragmentação guiada horizontalmente, levando em consideração quais os predicados se quer responder e não apenas o dado em si.

Foi preparada e estudada a base de dados CoPhIR para utilização nos experimentos. Conhecer a distribuição dos seus valores de dados e dessa forma estar apto a manipulá-la de forma adequada nos experimentos foi o requisito proposto e executado neste trabalho.

Para uma validação inicial da abordagem a fragmentação proposta foi simples, utilizando as tags existentes na base de dados. Essas mesmas tags foram utilizadas nas consultas executadas e dessa forma uma indexação direta com uma B-tree possibilitou a recuperação dos fragmentos. Ainda não foi possível extender os experimentos para fragmentos quaisquer onde o predicado de criação é desconhecido.

Os experimentos apresentados nesse trabalho apenas consideram consultas conjuntivas com predicados de igualdade e predicados baseados em similaridade. A abordagem proposta pode ser empregada para executar eficientemente consultas com um número arbitrário de predicados, de vários tipos, e logicamente conectados de diferentes modos. Esta abordagem abre novos caminhos de pesquisa para executar consultas eficientemente em grandes coleções de dados complexos.

Os resultados experimentais demonstam que a abordagem proposta aumenta drasticamente a velocidade da execução das consultas

avaliadas. Eles também mostram que não é viável executar predicados baseados em similaridade sobre a base de dados CoPhIR completa (que descreve em torno de 106 milhões de imagens), mesmo utilizando o índice métrico Slim-tree para acelerar a execução dos predicados baseados em similaridade no conteúdo dos descritores das imagens. De fato, mesmo em fragmentos grandes (que descrevem mais de 100 mil imagens, aproximadamente) é necessário uma maior fragmentação para garantir tempos de resposta aceitáveis.

Entre os desafios envolvidos para a maior exploração da abordagem proposta, pode-se mencionar os seguintes para trabalhos futuros:

1. Avaliar o custo da manutenção dos fragmentos com inserções, alterações e remoções de objetos de dados;

2. Desenvolver técnicas automatizadas para criar fragmentos horizontais em grandes bases de dados para a execução eficiente de consultas;

3. Indexar a coleção de fragmentos para eficientemente encontrar os fragmentos que resolvem diferentes tipos de predicados;

4. Criar técnicas de otimização de consultas que exploram apropriadamente os fragmentos da base de dados e os métodos de acesso.

5. Validar a abordagem proposta na execução de diferentes tipos de consulta sobre bases de dados diversas.

6. Avaliar outros métodos de acesso métrico, e compará-los com a Slim-tree para decidir qual método tem o melhor desempenho.

7. Armazenar não apenas os descritores e imagens reduzidas disponíveis no CoPhIR mas também as imagens originais disponíveis no FLICKR para que se, eventualmente, essas imagens sejam retiradas do FLICKR ainda tenhamos acesso às mesmas.

Os resultados obtidos neste trabalho foram publicados em forma de artigo em um evento da área:

*Efficient Execution of Conjunctive Complex Queries on Big Multimedia Databases* - IEEE International Symposium on Multimedia - ISM 2013. (FASOLIN et al., 2013)

# REFERÊNCIAS

ALASHQUR, A. Expressing database functional dependencies in terms of association rules. *European J Scientific Research*, v. 32, n. 2, p. 260–267, 2009.

ALASHQUR, A. Rdb-miner: A sql-based algorithm for mining true relational databases. *JSW*, v. 5, n. 9, p. 998–1005, 2010.

ALIPANAH, N. et al. Ontology-driven query expansion using map/reduce framework to facilitate federated queries. In: *Proceedings of the 2011 IEEE International Conference on Web Services*. Washington, DC, USA: IEEE Computer Society, 2011. (ICWS '11), p. 712–713. ISBN 978-0-7695-4463-2. <http://dx.doi.org/10.1109/ICWS.2011.21>.

ASLANDOGAN, Y. A.; YU, C. T. Techniques and systems for image and video retrieval. *IEEE Trans. Knowl. Data Eng.*, v. 11, n. 1, p. 56–63, 1999.

BAEZA-YATES, R.; MELUCCI, M. (Ed.). *Advanced Topics in Information Retrieval*. [S.l.]: Springer, 2011.

BAEZA-YATES, R. A.; RIBEIRO-NETO, B. *Modern Information Retrieval*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1999. ISBN 020139829X.

BAEZA-YATES, R. A.; RIBEIRO-NETO, B. A. *Modern Information Retrieval - the concepts and technology behind search, Second edition*. [S.l.]: Pearson Education Ltd., Harlow, England, 2011. ISBN 978-0-321-41691-9.

BARIONI, M. C. N. Operacoes de consulta por similaridade em grandes bases de dados complexos. *Tese de Doutorado - ICMC, USP*, Biblioteca Digital de Teses e Dissertações da USP, 2006. <http://www.teses.usp.br/teses/disponiveis/55/55134/tde-28092006-151225/>.

BARIONI, M. C. N. et al. SIREN: A similarity retrieval engine for complex data. In: DAYAL, U. et al. (Ed.). *Proceedings of the 32nd International Conference on Very Large Data Bases (VLDB), Seoul, Korea*. [S.l.]: ACM, 2006. p. 1155–1158. ISBN 1-59593-385-9.

BARIONI, M. C. N. et al. Seamlessly integrating similarity queries in SQL. *Software: Practice and Experience*, v. 39, n. 4, p. 355–384, 2009.

BLANKEN, H. et al. (Ed.). *Multimedia Retrieval*. Heidelberg: Springer Verlag, 2007. (Data-Centric Systems and Applications). ISBN=978-3-540-72894-8.

BOLETTIERI, P. et al. CoPhIR: a test collection for content-based image retrieval. *CoRR*, abs/0905.4627v2, 2009. <http://cophir.isti.cnr.it>.

BOZKAYA, T.; OZSOYOGLU, M. Indexing large metric spaces for similarity search queries. *ACM Trans. Database Syst.*, ACM, New York, NY, USA, v. 24, p. 361–404, September 1999. ISSN 0362-5915. <http://doi.acm.org/10.1145/328939.328959>.

BRAYNER, A. (Ed.). *XXIV Simpósio Brasileiro de Banco de Dados, 05-09 de Outubro, Fortaleza, Ceará, Brasil, Anais*. [S.l.]: SBC, 2009.

BUGATTI, P. H.; TRAINA, A. J. M.; JR., C. T. Assessing the best integration between distance-function and image-feature to answer similarity queries. In: *23rd Annual ACM Symposium on Applied Computing (SAC2008)*. Fortaleza, Ceará - Brazil: ACM Press, 2008. p. 1225–1230.

CHBEIR, R.; LAURENT, D. Enhancing multimedia data fragmentation. *JMPT*, v. 1, n. 2, p. 112–130, 2010.

CHINO, F. J. T. et al. Mamview: a visual tool for exploring and understanding metric access methods. In: *Proceedings of the 2005 ACM Symposium on Applied computing*. New York, NY, USA: ACM, 2005. (SAC '05), p. 1218–1223. ISBN 1-58113-964-0. <http://doi.acm.org/10.1145/1066677.1066952>.

CIACCIA, P.; PATELLA, M. Searching in metric spaces with user-defined and approximate distances. *ACM Trans. Database Syst.*, ACM, New York, NY, USA, v. 27, p. 398–437, December 2002. ISSN 0362-5915. <http://doi.acm.org/10.1145/582410.582412>.

CIACCIA, P.; PATELLA, M.; ZEZULA, P. M-tree: An efficient access method for similarity search in metric spaces. In: JARKE, M. et al. (Ed.). *VLDB'97, Proceedings of 23rd International Conference on Very Large Data Bases, August 25-29, 1997, Athens, Greece*. [S.l.]: Morgan Kaufmann, 1997. p. 426–435. ISBN 1-55860-470-7.

CODD, E. F. A relational model of data for large shared data banks. *Commun. ACM*, v. 13, n. 6, p. 377–387, 1970.

CODD, E. F. Extending the database relational model to capture more meaning. *ACM Trans. Database Syst.*, v. 4, n. 4, p. 397–434, 1979.

Costa Pereira, J. et al. On the Role of Correlation and Abstraction in Cross-Modal Multimedia Retrieval. *IEEE transactions on pattern analysis and machine intelligence*, p. 1–15, ago. 2013. ISSN 1939-3539. <http://www.ncbi.nlm.nih.gov/pubmed/23917421>.

DARMONT, J. et al. An architecture framework for complex data warehouses. In: *ICEIS (1)*. [S.l.: s.n.], 2005. p. 370–373.

DATE, C. J. *SQL and Relational Theory - How to Write Accurate SQL Code*. [S.l.]: O'Reilly, 2009. I-XIX, 1-404 p. ISBN 978-0-596-52306-0.

DATTA, R. et al. Image retrieval: Ideas, influences, and trends of the new age. *ACM Comput. Surv.*, ACM, New York, NY, USA, v. 40, n. 2, p. 1–60, 2008. ISSN 0360-0300.

DORAIRAJ, R.; NAMUDURI, K. R. Compact combination of mpeg-7 color and texture descriptors for image retrieval. In: IEEE. *Signals, Systems and Computers, 2004. Conference Record of the Thirty-Eighth Asilomar Conference on*. [S.l.], 2004. v. 1, p. 387–391.

EIDENBERGER, H. Distance measures for mpeg-7-based retrieval. In: *Proceedings of the 5th ACM SIGMM international workshop on Multimedia information retrieval*. New York, NY, USA: ACM, 2003. (MIR '03), p. 130–137. ISBN 1-58113-778-8. <http://doi.acm.org/10.1145/973264.973286>.

ESCALANTE, H. J.; GÓMEZ, M. M. y; SUCAR, L. E. Multimodal indexing based on semantic cohesion for image retrieval. *Inf. Retr.*, v. 15, n. 1, p. 1–32, 2012.

FASOLIN, K. et al. Efficient execution of conjunctive complex queries on big multimedia databases. In: *Multimedia (ISM), 2013 IEEE International Symposium on*. [S.l.: s.n.], 2013. p. 536–543.

FERREIRA, M. R. P. et al. Algebraic properties to optimize knn queries. In: *Proceedings of the 26th Brazilian Symposium on Databases (SBBD)*. [S.l.: s.n.], 2011.

GAO, Y. et al. Tag-based social image search with visual-text joint hypergraph learning. In: *Proceedings of the 19th ACM international conference on Multimedia - MM '11*. New York, New York, USA: ACM Press, 2011. p. 1517. ISBN 9781450306164. <http://dl.acm.org/citation.cfm?id=2072298.2072054>.

GOKER, A.; DAVIES, J.; GRAHAM, M. *Information Retrieval: Searching in the 21st Century*. [S.l.]: John Wiley & Sons, 2007. ISBN 0470027622.

GUDIVADA, V. N.; RAGHAVAN, V. V. Content-based image retrieval systems - guest editors' introduction. *IEEE Computer*, v. 28, n. 9, p. 18–22, 1995.

HIEMSTRA, D.; HAUFF, C. Mapreduce for information retrieval evaluation: "let's quickly test this on 12 tb of data". In: *Proceedings of the 2010 international conference on Multilingual and multimodal information access evaluation: cross-language evaluation forum*. Berlin, Heidelberg: Springer-Verlag, 2010. (CLEF'10), p. 64–69. ISBN 3-642-15997-4, 978-3-642-15997-8. <http://dl.acm.org/citation.cfm?id=1889174.1889186>.

HUISKES, M. J.; LEW, M. S. The mir flickr retrieval evaluation. In: *Proceedings of the 1st ACM International Conference on Multimedia Information Retrieval*. New York, NY, USA: ACM, 2008. (MIR '08), p. 39–43. ISBN 978-1-60558-312-9. <http://doi.acm.org/10.1145/1460096.1460104>.

ITE-VIL, L. C. (mitsubishi E. *The MPEG-7 Color Descriptors Jens-Rainer Ohm (RWTH Aachen, Institute of Communications Engineering)*. 2009.

JAIN, R. Multimedia information retrieval. In: *Proceeding of the 1st ACM international conference on Multimedia information retrieval - MIR '08*. New York, New York, USA: ACM Press, 2008. p. 229. ISBN 9781605583129. <http://dl.acm.org/citation.cfm?id=1460096.1460135>.

JR., C. T. et al. The omni-family of all-purpose access methods: A simple and effective way to make similarity search more efficient. *The International Journal on Very Large Databases*, v. 16, n. 4, p. 483–505, 2007.

JR., C. T. et al. Fast indexing and visualization of metric datasets using slim-trees. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, v. 14, n. 2, p. 244–260, 2002.

JR., C. T. et al. How to improve the pruning ability of dynamic metric access methods. In: *International Conference on Information and Knowledge Management (CIKM)*. McLean, VA, USA: ACM Press, 2002. p. 219–226.

JR., C. T. et al. Slim-trees: High performance metric trees minimizing overlap between nodes. In: ZANIOLO, C. et al. (Ed.). *International Conference on Extending Database Technology (EDBT)*. Konstanz, Germany: Springer Verlag, 2000. (Lecture Notes in Computer Science, v. 1777), p. 51–65.

KASTER, D. dos S. Tratamento de condies especiais para busca por similaridade em bancos de dados complexos. *Tese de Doutorado - ICMC, USP*, Biblioteca Digital de Teses e Dissertações da USP, 2012. <http://www.teses.usp.br/teses/disponiveis/55/55134/tde-23072012-164717/>.

KASTER, D. S. et al. Incorporating metric access methods for similarity searching on oracle database. In: BRAYNER, A. (Ed.). *SBBD*. [S.l.]: SBC, 2009. p. 196–210.

KASTER, D. S. et al. Fmi-sir: A flexible and efficient module for similarity searching on oracle database. *JIDM*, v. 1, n. 2, p. 229–244, 2010.

KITCHENHAM, B. Procedures for performing systematic reviews. *Technical Report TR/SE-0401*, 2004.

LEW, M. S. et al. Content-based multimedia information retrieval: State of the art and challenges. *ACM Trans. Multimedia Comput. Commun. Appl.*, ACM, New York, NY, USA, v. 2, p. 1–19, February 2006. ISSN 1551-6857. <http://doi.acm.org/10.1145/1126004.1126005>.

LIU, J. et al. Discover dependencies from data - a review. *IEEE Trans. Knowl. Data Eng.*, v. 24, n. 2, p. 251–264, 2012.

LIU, T. Y. et al. Letor: Benchmark dataset for research on learning to rank for information retrieval. In: *SIGIR '07: Proceedings of the Learning to Rank workshop in the 30th annual international ACM SIGIR conference on Research and development in information retrieval*. [s.n.],

2007. <http://research.microsoft.com/ junxu/papers/SGIR2007-LR4IR

LONG, F.; ZHANG, H.; FENG, D. Fundamentals of content-based image retrieval. *Multimedia Information Retrieval and Management*, Springer, 2002.

MANNING, C. D.; RAGHAVAN, P.; SCHüTZE, H. *Introduction to Information Retrieval*. New York, NY, USA: Cambridge University Press, 2008. ISBN 0521865719, 9780521865715.

MELUCCI, M.; BAEZA-YATES, R. *Advanced Topics in Information Retrieval*. Springer, 2011. (The Information Retrieval Series). ISBN 9783642209451. <http://books.google.com.br/books?id=yFD0YjXoiiYC>.

MÜLLER, H. et al. Overview of the imageclef 2012 medical image retrieval and classification tasks. In: FORNER, P.; KARLGREN, J.; WOMSER-HACKER, C. (Ed.). *CLEF (Online Working Notes/Labs/Workshop)*. [S.l.: s.n.], 2012. ISBN 978-88-904810-3-1.

MURTHY, U. et al. Superimposed image description and retrieval for fish species identification. In: AGOSTI, M. et al. (Ed.). *ECDL*. [S.l.]: Springer, 2009. (Lecture Notes in Computer Science, v. 5714), p. 285–296. ISBN 978-3-642-04345-1.

OVER, P. et al. Multimedia retrieval benchmarks. *IEEE MultiMedia*, n. 2, p. 80–84.

PEDRONETTE, D. C. G. a.; TORRES, R. da S. Exploiting contextual spaces for image re-ranking and rank aggregation. In: *Proceedings of the 1st ACM International Conference on Multimedia Retrieval*. New York, NY, USA: ACM, 2011. (ICMR '11), p. 13:1–13:8. ISBN 978-1-4503-0336-1. <http://doi.acm.org/10.1145/1991996.1992009>.

QIN, T. et al. Letor: A benchmark collection for research on learning to rank for information retrieval. *Information Retrieval*, Springer Netherlands, v. 13, n. 4, p. 346–374, 2010. ISSN 1386-4564. <http://dx.doi.org/10.1007/s10791-009-9123-y>.

RASIWASIA, N. et al. A new approach to cross-modal multimedia retrieval. In: BIMBO, A. D.; CHANG, S.-F.; SMEULDERS, A. W. M. (Ed.). *ACM Multimedia*. [S.l.]: ACM, 2010. p. 251–260. ISBN 978-1-60558-933-6.

RASIWASIA, N.; VASCONCELOS, N. Holistic context modeling using semantic co-occurrences. In: *CVPR*. [S.l.]: IEEE, 2009. p. 1889–1895. ISBN 978-1-4244-3992-8.

SAMET, H. *Foundations of Multidimensional and Metric Data Structures (The Morgan Kaufmann Series in Computer Graphics and Geometric Modeling)*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2005. ISBN 0123694469.

SANTOS, K. C. L. et al. Recuperação de imagens da web utilizando múltiplas evidências textuais e programação genética. In: BRAYNER, A. (Ed.). *SBBD*. [S.l.]: SBC, 2009. p. 91–105.

SINHA, P.; JAIN, R. Semantics in digital photos: A contenxtual analysis. *International Conference on Semantic Computing*, IEEE Computer Society, Los Alamitos, CA, USA, v. 0, p. 58–65, 2008.

SKOPAL, T. Where are you heading, metric access methods?: a provocative survey. In: CIACCIA, P.; PATELLA, M. (Ed.). *SISAP*. [S.l.]: ACM, 2010. p. 13–21. ISBN 978-1-4503-0420-7.

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

SOARES, L. C.; KASTER, D. S. cx-sim: A metric access method for similarity queries with additional conditions. *JIDM*, v. 4, n. 3, p. 437–452, 2013. <http://dblp.uni-trier.de/db/journals/jidm/jidm4.htmlSoaresK13>.

SONG, S.; CHEN, L. Differential dependencies: Reasoning and discovery. *ACM Trans. Database Syst.*, v. 36, n. 3, p. 16, 2011.

SONG, S.; CHEN, L.; YU, P. S. On data dependencies in dataspaces. In: ABITEBOUL, S. et al. (Ed.). *ICDE*. [S.l.]: IEEE Computer Society, 2011. p. 470–481. ISBN 978-1-4244-8958-9.

SONG, S.; CHEN, L.; YUAN, M. Materialization and decomposition of dataspaces for efficient search. *IEEE Trans. Knowl. Data Eng.*, v. 23, n. 12, p. 1872–1887, 2011.

TORRES, R. d. S. Sistemas de Informação para o Gerenciamento de Imagens: Aplicações e Desafios de Pesquisa. In: *Grandes Desafios da Sociedade Brasileira de Computaçcão*. São Paulo, SP, Brasil: [s.n.], 2006.

TORRES, R. d. S. et al. A genetic programming framework for content-based image retrieval. *Pattern Recognition*, v. 42, n. 2, p. 283 – 292, 2009. ISSN 0031-3203. Learning Semantics from Multimedia Content. <http://www.sciencedirect.com/science/article/pii/S0031320308001623>.

TRAINA, A. J. M. Suporte à visualização de consultas por similaridade em imagens médias através de estruturas de indexação métricas. *Tese de Livre-Docência em Computação - ICMC/USP*, Biblioteca Digital de Teses e Dissertações da USP, p. 104, 2001.

TRAINA, A. J. M.; JR., C. T. Similarity search in multimedia databases. In: _____. *Handbook of Video Databases - Design and Applications*. CRC Press, 2003. (Internet and Communications, v. 1), p. 711–738. <http://www.crcpress.com/shopping$_c$*art*/*products*/*product*$_d$*etail.asp*?*sku* = 7006&*af* = *W*1129>.

VIEIRA, M. R. et al. Estimating suitable query radii to boost knearest neighbor queries. In: *19th International Conference on Scientific and Statistical Database Management (SSDBM 2007)*. Banff, Canada: ACM Press, 2007. p. 1–10.

WANG, M. et al. PictureBook: A Text-and-Image Summary System for Web Search Result. In: *2008 IEEE 24th International Conference on Data Engineering*. IEEE, 2008. v. 00, p. 1612–1615. ISBN 978-1-4244-1836-7. <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4497634>.

WILSON, D. R.; MARTINEZ, T. R. Improved heterogeneous distance functions. *Journal of Artificial Intelligence Research*, v. 6, p. 1–34, 1997.

WU, Z.; MAO, B.; CAO, J. Mrgir: Open geographical information retrieval using mapreduce. In: *Geoinformatics, 2011 19th International Conference on*. [S.l.: s.n.], 2011. p. 1–5. ISSN 2161-024X.

YANG, L. et al. A unified context model for web image retrieval. *ACM Transactions on Multimedia Computing, Communications, and Applications*, ACM, v. 8, n. 3, p. 1–19, jul. 2012. ISSN 15516857. <http://dl.acm.org/citation.cfm?id=2240136.2240141>.

YIANILOS, P. N. Data structures and algorithms for nearest neighbor search in general metric spaces. In: *Proceedings of the fourth annual ACM-SIAM Symposium on Discrete algorithms*. Philadelphia, PA, USA: Society for Industrial and Applied

Mathematics, 1993. (SODA '93), p. 311–321. ISBN 0-89871-313-7. <http://dl.acm.org/citation.cfm?id=313559.313789>.

ZEZULA, P. et al. *Similarity Search - The Metric Space Approach*. [S.l.]: Springer, 2006. 220 p.