UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DAINF - DEPARTAMENTO ACADÊMICO DE INFORMÁTICA
CURSO DE ENGENHARIA DE COMPUTAÇÃO

CRISTIANO JOSÉ MENDES MATSUI

CONSULTAS POR SIMILARIDADE EM BASES DE DADOS COMPLEXAS UTILIZANDO TÉCNICAS OMNI EM SGBDR

TRABALHO DE CONCLUSÃO DE CURSO

PATO BRANCO
2017

CRISTIANO JOSÉ MENDES MATSUI

CONSULTAS POR SIMILARIDADE EM BASES DE DADOS COMPLEXAS UTILIZANDO TÉCNICAS OMNI EM SGBDR

Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia de Computação da Universidade Tecnológica Federal do Paraná, como requisito parcial para a obtenção do título de Bacharel em Engenharia de Computação.

Orientador: Dr. Ives Renê Venturini Pola

Coorientadora: Dra. Fernanda Paula Barbosa Pola

PATO BRANCO
2017

When life gives you lemons, don't make lemonade. Make life take the lemons back! Get mad! I don't want your damn lemons, what the hell am I supposed to do with these? Demand to see life's manager! Make life rue the day it thought it could give Cave Johnson lemons! Do you know who I am? I'm the man who's gonna burn your house down! With the lemons! I'm gonna get my engineers to invent a combustible lemon that burns your house down!
(JOHNSON, Cave. Portal 2).

RESUMO

MATSUI, Cristiano. Consultas por similaridade em bases de dados complexas utilizando técnicas OMNI em SGBDR. 2017. 26 f. Trabalho de Conclusão de Curso – Curso de Engenharia de Computação, Universidade Tecnológica Federal do Paraná. Pato Branco, 2017.

A necessidade de armazenamento de mídias cada vez maiores em termos de tamanho de armazenamento e complexas é uma tendência que aumentou consideravelmente com os avanços da tecnologia e comunicação. Estes dados conhecidos como dados complexos exigem uma complexidade estrutural de armazenamento e análise maior do que dados simples como palavras ou números, além de requererem operadores especiais de consultas, como a consulta por abrangência (Rq) e a consulta aos k-vizinhos mais próximos (kNNq). Dentre o conjunto de dados complexos destacam-se as imagens, que precisam ser comparadas de acordo com características extraídas como cor, forma ou textura. Esta comparação é realizada na forma de um cálculo de distância entre o valor da característica da imagem central da consulta em relação a todas as outras imagens da base de dados. O tempo de consulta pode aumentar significativamente com o aumento da base de dados. Para contornar o problema da maldição da cardinalidade, este trabalho tem como proposta aplicar uma técnica (Omni) utilizada para promover uma etapa de filtragem do número de imagens a terem as suas distâncias calculadas, evitando a comparação com toda a base de dados. Após a modelagem e construção de um banco de dados que suporte esta técnica, será implementado um sistema de recuperação de imagens (CBIR).

Palavras-chave: Dados Complexos. OMNI. CBIR. Rq. kNNq.

LISTA DE FIGURAS

LISTA DE ABREVIATURAS E SIGLAS

GBDI	Grupo de Bases de Dados e Imagens
IOid(s_i)	Identificador do objeto s_i
kNNq	Consulta aos k-vizinhos mais próximos
mbOr	Região Omni de delimitação mínima
Rq	Consulta por abrangência
SGBDR	Sistema Gerenciador de Banco de Dados Relacional
SQL	*Structured Query Language*

LISTA DE SÍMBOLOS

$\in$	Pertence
$\forall$	Para todo
s_q	Elemento central da consulta
ξ	Raio da consulta por abrangência
$\mathbb{S}$	Domínio de dados
S	Conjunto de elementos pertencentes ao domínio $\mathbb{S}$
k	Número de vizinhos desejados em uma *kNNq*
s_i	Elemento pertencente ao conjunto S
$d : \mathbb{S} \times \mathbb{S} \rightarrow \mathbb{R}^+$	Métrica ou função de distância
$M<S,d>$	Espaço métrico
$\mathscr{F}$	Base de focos Omni
f_k	Foco pertencente a base focal
D	Dimensão intrínseca da base de dados

LISTA DE ALGORITMOS

SUMÁRIO

1 INTRODUÇÃO

Neste capítulo será apresentada uma contextualização do problema, assim como o estado da arte abordado por este trabalho. Também será introduzido um tipo de consulta não-nativo ao SGBDR, a consulta por similaridade, assim como os tipos de consultas por similaridade que serão utilizados neste trabalho como a consulta por abrangência e a consulta por k-vizinhos mais próximos. Finalmente, será apresentada a organização deste documento.

1.1 CONSIDERAÇÕES INICIAIS

Nos anos recentes foi notado um grande aumento no tráfego e armazenamento de diferentes aplicações e dados multimídias, como imagens, áudio, vídeo, impressões digitais, séries temporais, sequências de proteínas, etc. Estes tipos de dados, que apresentam muito mais atributos do que simples numerais ou pequenas cadeias de caracteres, são conhecidos como dados complexos (ZIGHED et al., 2008).

Quando tratados por um Sistema Gerenciador de Banco de Dados Relacional (SGBDR), não suportam comparações com os operadores conhecidos como "big six" da linguagem SQL: $=$, $\neq$, $<$, $>$, $\leq$, $\geq$. Esse fato limita muito as comparações entre dados complexos inseridos em um SGBDR, ocasionando um grande problema no contexto de base de dados, uma vez que os principais sistemas de gerenciamento de base de dados são relacionais (DB-ENGINES, 2017). Com isso, tornou-se necessária a concepção de novos tipos de comparadores, como buscas por similaridade.

Estas consultas por similaridade se aplicam de maneira geral a muitos dos tipos de dados complexos (BARIONI et al., 2009). Exemplos na área médica são encontrados nos trabalhos de (MARCHIORI et al., 2001), (BUGATTI et al., 2014) e (LEHMANN et al., 1999). Também é possível encontrar trabalhos no campo de reconhecimento facial (GUTTA; WECHSLER, 1997) e sistemas de identificação por biometria (CHORAS, 2007).

1.2 TIPOS DE CONSULTAS

Dentre os operadores de consulta por similaridade os mais comuns são as consultas por abrangência (*range query: Rq*) e consulta aos k-vizinhos mais próximos (*k-nearest neighbor querry: kNNq*).

As consultas por abrangência *Rq*(s_q, ξ) recebem como parâmetro um elemento s_q do domínio de dados (elemento central da consulta) e um limite máximo de dissimilaridade ξ. O resultado é o conjunto de elementos da base que diferem do elemento central da consulta por no máximo a dissimilaridade indicada.

Definição 1 *Seja S um atributo complexo de um domínio $\mathbb{S}$ ($S \subset \mathbb{S}$) sobre o qual a condição de similaridade é expressada, seja d uma função de distância, seja ξ o limiar de dissimilaridade e seja $s_q \in \mathbb{S}$ o elemento central de consulta. A consulta Rq(s_q , ξ) retorna todos os elementos $s_i \in \mathbb{S}$ que possuem o valor do atributo S distantes até um máximo de ξ deste atributo referente ao elemento central da consulta:*

$$Rq(s_q, \xi) : S = \{s_i \in S \mid d(s_i, s_q) \leq \xi\} \tag{1}$$

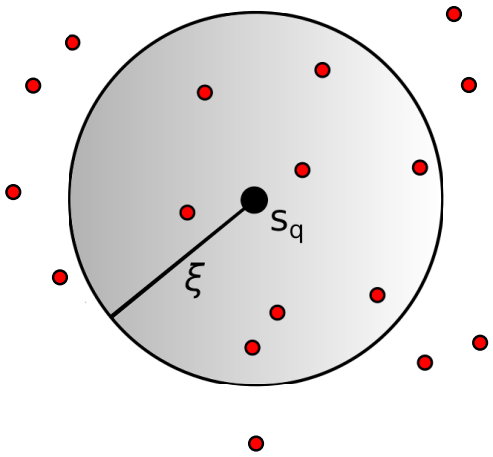


Figura 1 – Exemplo de consulta por abrangência

Fonte: Autoria Própria

A Figura 1 exemplifica este tipo de consulta. Tomando s_q e ξ como parâmetros da consulta, todos os elementos contidos pelo raio de abrangência fazem parte do conjunto resposta da consulta. O elemento s_q não necessariamente precisa pertencer ao conjunto de dados de pesquisa, devendo apenas pertencer ao mesmo domínio de dados.

Uma consulta aos k-vizinhos mais próximos *kNNq(s_q, k)* também recebe como um de seus parâmetros um elemento central da consulta s_q, e um número inteiro k de vizinhos desejados, e retorna como resultado o conjunto dos k elementos com a menor dissimilaridade em relação ao elemento central da consulta s_q (POLA, 2010).

Definição 2 *Seja S um atributo complexo de um domínio $\mathbb{S}$ sobre o qual a condição de similaridade é expressada, seja d uma função de distância, seja $k \in \mathbb{N}^*$ a quantidade de elementos desejados e seja $s_q \in \mathbb{S}$ o elemento central de consulta. A consulta kNNq(s_q , k) retorna k elementos $s_i \in \mathbb{S}$ que possuem o valor do atributo S menos distantes do valor deste atributo referente ao elemento central da consulta(FERREIRA et al., 2009):*

$$kNNq(s_q, k) : S = \{s_i \in S \mid \forall\ s_j \in S - S^{'}, d(s_i, s_q) \leq d(s_j, s_q)\}, \tag{2}$$

onde $S^{'} = 0$, se $i = 1$ e $S^{'} = \{s_1, ..., s_{(i-1)}\}$, se $1 < i \leq k$.

A figura 2 exemplifica este tipo de operação. Nesta consulta, os parâmetros foram o elemento central da consulta s_q e o número de elementos k a serem encontrados igual a 4. Os elementos ligados ao elemento central foram os mais próximos a este, portanto apenas eles fazem parte do conjunto resposta da consulta. Note que o elemento central da consulta não precisa pertencer ao conjunto de dados, como é o caso exemplificado.

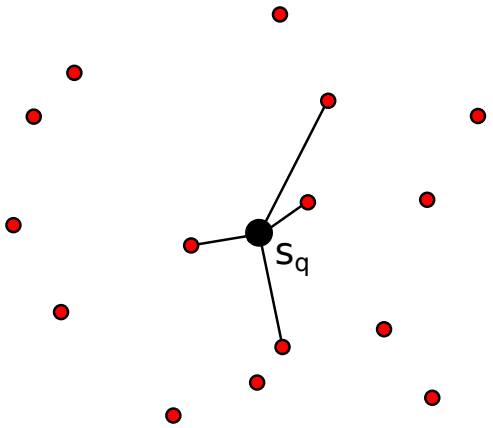


Figura 2 – Exemplo de consulta por k-vizinhos mais próximos

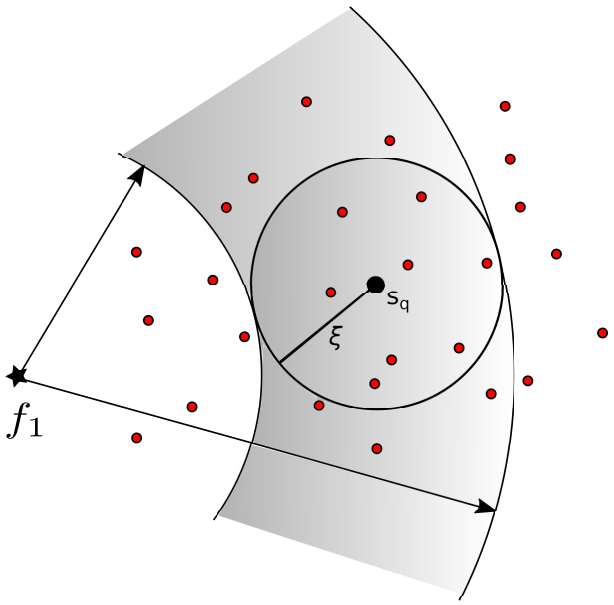
Fonte: Autoria Própria

O SGBDR não possui suporte nativo a estes tipos de consulta, mas é possível construir estas consultas utilizando ferramentas existentes em um banco de dados relacional (como a estrutura de dados *B-tree*).

A solução abordada por esta proposta é a do uso de técnicas Omni, presentes no trabalho de (SANTOS FILHO et al., 2001). Um número calculado de elementos do conjunto de dados são selecionados como "focos", e utilizados para evitar cálculos desnecessários de distâncias, fazendo uso da desigualdade triangular.

A técnica aqui aplicada tem base em calcular previamente as distâncias de todos os elementos para todos os focos selecionados, armazenando estas distâncias no banco. Quando uma consulta por similaridade (como uma consulta por abrangência) é realizada, são conhecidas as distâncias entre o elemento central da consulta s_q e o raio de abrangência ξ. Considerando um foco f_1 e utilizando a desigualdade triangular, elementos que possuem uma distância entre o foco escolhido menor do que a distância de s_q até o foco menos o valor ξ serão descartados do conjunto de elementos necessários para os cálculos de distância com o elemento central. Simetricamente, elementos cuja distância até o foco seja maior do que a distância de s_q até o foco mais o valor do raio de abrangência ξ também serão descartados. Com isso, ocorre uma grande redução do número de cálculos necessários para fornecer o conjunto resposta. Essa poda também pode ser realizada por mais de um foco.

Figura 3 – Consulta por abrangência pela técnica Omni utilizando um foco



Fonte: Autoria Própria

A Figura 3 ilustra a poda no número de cálculos. Apenas os elementos na área sombreada terão as suas distâncias em relação ao centro da consulta calculadas, pois estão no conjunto de elementos que não foram descartados utilizando a desigualdade triangular com as distâncias previamente calculadas em relação ao foco. O armazenamento das distâncias de cada foco f_i para cada outro elemento s_k será feito utilizando uma estrutura de indexação que implementa os conceitos da técnica Omni com a estrutura da B-tree, originando uma nova estrutura chamada de OmniB-Tree. As distâncias serão armazenadas em l OmniB-Trees, sendo l o número de focos criados para a base de dados (SANTOS FILHO et al., 2001).

1.3 OBJETIVOS

Os objetivos encontram-se divididos em objetivos gerais, referentes ao resultado obtido com a conclusão deste trabalho e objetivos específicos, ilustrando etapas intermediárias necessárias para alcançar o objetivo geral.

1.3.1 OBJETIVOS GERAIS

O principal foco deste trabalho é a construção de um sistema de consultas em SGBDR por similaridade em uma base de imagens utilizando técnicas da família Omni para reduzir o custo computacional das operações de consulta.

1.3.2 OBJETIVOS ESPECÍFICOS

- Modelar o banco de dados para atender a problemática apresentada;

- Aplicar os extratores de características das imagens utilizadas;
- Inserir no banco de dados as imagens e os valores de suas características;
- Criar a estrutura Omni necessária para a filtragem dos cálculos;
- Analisar e comparar os resultados obtidos.

1.4 ORGANIZAÇÃO DO TRABALHO

A estrutura deste trabalho apresenta mais detalhes sobre os tipos de consulta por similaridade utilizados, assim como os seus algoritmos computacionais. Posteriormente, explana os conceitos e técnicas da família Omni utilizada para melhorar a performance das consultas. Também realiza a abordagem sobre os extratores de características utilizados para a base de imagens utilizada neste trabalho.

2 CONCEITOS

O presente capítulo abordará os conceitos necessários para uma melhor compreensão dos artifícios utilizados para tratar o problema apresentado no Capítulo 1, assim como uma explanação sobre os extratores de características de imagens que serão utilizados e o estado da arte atual.

2.1 ESPAÇOS MÉTRICOS

Uma métrica em um domínio $\mathbb{S}$ é uma função $d : \mathbb{S} \times \mathbb{S} \rightarrow \mathbb{R}^+$ que associa a cada par ordenado de elementos (s_1, s_2) um número real $d(s_1,s_2)$ chamado de distância de s_1 a s_2, e que atenda as propriedades definidas no espaço métrico(LIMA, 1977).

Um espaço métrico M é um par $<S, d>$ no qual S é um conjunto de elementos e d é uma métrica (ou função de distância). Esta distância $d(s_1,s_2)$ pode ser compreendida como uma medida de dissimilaridade entre dois elementos. Quanto menor esta distância entre dois elementos, mais semelhantes eles são.

Definição 3 *Seja S um conjunto não-vazio de elementos e $d(s_1,s_2)$ uma métrica definida sobre $\mathbb{S} \times \mathbb{S}$. O par $<S, d>$ é chamado de espaço métrico desde que d satisfaça as seguintes propriedades:*

1. $d(s_1,s_1) = 0$ *(identidade);*
2. *Se* $s_1 \neq s_2$ *então* $d(s_1,s_2) > 0$ *(não-negatividade);*
3. $d(s_1,s_2) = d(s_2,s_1)$ *(simetria);*
4. $d(s_1,s_3) \leq d(s_1,s_2) + d(s_2,s_3)$ *(desigualdade triangular)*

onde s_1, s_2, $s_3 \in \mathbb{S}$.

O conjunto de dados complexos a ser utilizado geralmente apresenta dimensões elevadas, mas também podem não ter dimensão fixa. Para a utilização computacional destes conceitos torna-se necessário o conceito de bola, empregado em espaços métricos. A seguinte definição é baseada no livro de (SHIRALI; VASUDEVA, 2005).

Definição 4 *Seja $<\mathbb{S}, d>$ um espaço métrico. O conjunto*

$$S(s_0,r) = x \in \mathbb{S} : d(s_0, x) \leq r, onde\ r > 0\ e\ s_0 \in \mathbb{S}, \tag{3}$$

é chamado de bola fechada de raio r e centro em s_0.

A aplicação direta das propriedades do espaço métrico neste trabalho reside fortemente na quarta propriedade apresentada pela Definição 3. A desigualdade triangular é fundamentada na geometria euclidiana, na qual a soma do comprimento de dois lados de um triângulo não pode ser superior ao comprimento do terceiro lado, como ilustra a Figura 4.

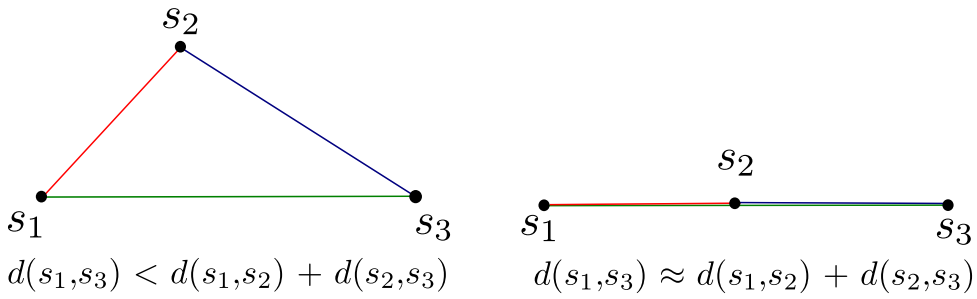


Figura 4 – Exemplo geométrico da desigualdade triangular

Fonte: Autoria Própria

O emprego da desigualdade triangular neste trabalho será detalhado no Capítulo 3.

2.2 FUNÇÕES DE DISTÂNCIA

Para verificar a similaridade entre dois elementos de um domínio, é utilizada uma função de distância. Esta função recebe como parâmetro um par de elementos do conjunto e retorna o valor da dissimilaridade entre eles. Quanto mais próximo de zero, mais similares os elementos são.

A importância do uso de uma função de distância para este trabalho é fornecer uma métrica de comparação entre os elementos complexos. Além de necessária para a consulta por similaridade, sua propriedade de desigualdade triangular é utilizada para evitar cálculos desnecessários, fornecendo uma métrica empregada pela técnica Omni.

2.2.1 DISTÂNCIA MINKOWSKI

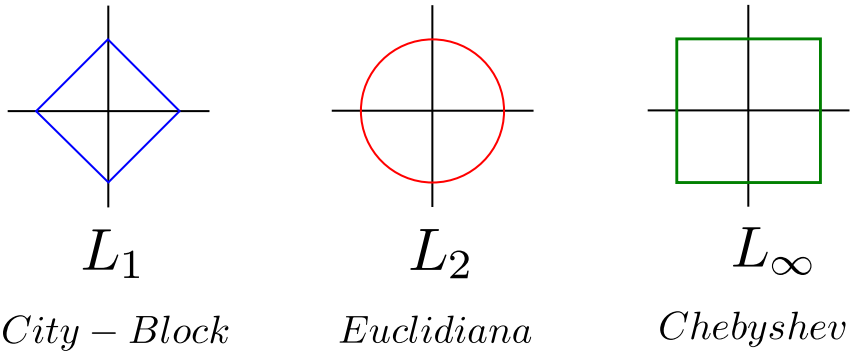
Dentre as funções de distância, será abordada a métrica Minkowski. Esta métrica é a mais utilizada para cálculos de índice de similaridade pois é independente da origem do espaço do conjunto de dados, e tem como resultado o valor da dissimilaridade entre dois elementos (JAIN; DUBES, 1988).

Definição 5 *Sejam* $s_1 = \{s_{11},s_{12},...,s_{1n}\}$ *e* $s_2 = \{s_{21},s_{22},...,s_{2n}\}$ *dois vetores de dimensionalidade* n *pertencentes ao conjunto de elementos* $\mathbb{S}$*, a distância Minkowski entre esses dois elementos é dada por:*

$$d(s_1,s_2) = \sqrt[p]{\sum_{i=1}^{n} |s_{1i} - s_{2i}|^p} \tag{4}$$

A Figura 5 ilustra as diferentes formas geométricas geradas de acordo com a métrica L_p utilizada. Para estes determinados valores de p, a distância Minkowski recebe nomes próprios (*distância City-Block* para $p = 1$, *distância Euclidiana* para $p = 2$ e *distância Chebyshev* para $p = \infty$).

Figura 5 – Círculos unitários em espaços bi-dimensionais para diferentes valores de p



Fonte: Autoria Própria

2.2.2 DISTÂNCIA CANBERRA

A distância Minkowski nem sempre é a melhor opção de métrica. A diferença da distância em cada dimensão é elevada a uma potência p geralmente ≥ 1 coloca uma enorme ênfase para os casos nos quais a distância dimensional é grande (KOKARE; BISWAS; CHATTERJI, 2007). Uma alternativa é a distância Canberra. Ela apresenta um cálculo menos custoso do que a distância Minkowski, e atribui um peso menor para a diferença de distâncias, mas é sensível à origem do espaço de dados.

Definição 6 *Sejam* $s_1 = \{s_{11},s_{12},...,s_{1n}\}$ *e* $s_2 = \{s_{21},s_{22},...,s_{2n}\}$ *dois vetores de dimensionalidade* n *pertencentes ao conjunto de elementos* $\mathbb{S}$*, a distância Canberra entre esses dois elementos é dada por:*

$$d(s_1,s_2) = \sum_{i=1}^{n} \frac{|s_{1i} - s_{2i}|}{|s_{1i}| + |s_{2i}|} \tag{5}$$

2.3 ESTRUTURAS DE INDEXAÇÃO

Dados complexos costumam apresentar tamanho físico muito elevado quando comparados com dados numéricos ou pequenas cadeias de caracteres, que são os tipos de dados mais comuns em bancos de dados tradicionais. Para responder consultas que envolvam dados complexos, são necessários mais acessos a disco em relação a uma consulta que envolva tipos de dados mais simples, como os previamente mencionados. Estes acessos são custosos e precisam ser reduzidos para um melhor desempenho do banco.

Uma maneira de tornar o acesso a disco mais eficiente é evitar movimentar grandes porções do banco de dados do disco para a memória, fazendo o uso de índices dentro do banco de dados. Um índice em um SGBDR funciona de maneira semelhante a um índice de um livro. Para procurar um tópico específico, é possível consultar o índice no fim do livro e descobrir qual o número da página correspondente ao tópico, contornando a necessidade da leitura sequencial do livro até o tópico procurado. Os índices são armazenados em ordem e apresentam um

tamanho muito menor do que um capítulo do livro, reduzindo o esforço necessário para a sua consulta (SILBERSCHATZ; KORTH; SUDARSHAN, 2011).

Mas índices também podem ser empregados para uso em memória RAM. Enquanto as estruturas de indexação orientadas a disco são armazenadas em disco e apresentam um custo elevado para serem acessadas, índices orientados a memória estão contidos na memória RAM, assim não existem acessos a disco para serem minimizados. Assim, uma estrutura de indexação em memória busca reduzir o tempo de computação geral enquanto usa o mínimo de memória possível. Estas estruturas armazenam ponteiros para as tuplas, que são menos custosos de serem manipulados do que as tuplas (LEHMAN; CAREY, 1986).

Dentre as estruturas de indexação em disco para dados tradicionais, destacam-se a B^+-Tree e índices bitmap. Para indexação em memória, podem ser utilizadas estruturas como B-Trees e arrays para preservar a ordenação natural dos dados, enquanto estruturas de hashing (*Chained Bucket Hashing, Linear Hashing e Extendible Hashing*) aleatorizam os dados dentro do índice.

2.4 EXTRATORES DE CARACTERÍSTICAS

O principal foco deste trabalho é o emprego destas técnicas para bases de dados constituídas por imagens. Para isto, torna-se necessário o uso de uma miríade de extratores de características das imagens, para um maior refinamento do uso de consultas por similaridade. Estas características podem se referir a: atributos visuais (cor, forma, textura), atributos lógicos (identificação de elementos) e atributos semânticos (identificação de emoções humanas).

As características visuais podem ser utilizadas como histogramas de cores para a análise de cor, matrizes de co-ocorrência para a análise de textura e métodos baseados em contorno para a análise de forma. Geralmente, consultas são feitas utilizando uma combinação destas características, e não apenas uma delas.

Dado uma palheta discreta de cores definida por alguns eixos de cor, o histograma de cores é obtido através da discretização das cores da imagem e contagem do número de vezes que cada cor discreta ocorre na matriz da imagem (SWAIN; BALLARD, 1991). As vantagens do uso do histograma de cores é a sua simplicidade computacional e pouca sensibilidade a alterações na imagem (rotação e translação), particularmente útil para a representação de objetos tridimensionais. Entretanto, duas imagens completamente diferentes podem apresentar o mesmo histograma de cores.

Para a análise de textura, o objetivo e conseguir distinguir regiões que apresentam cores similares (como folhagem e grama), analisando o padrão de variação dessas cores. A técnica mais utilizada analisa conjunto de pares de pixels da imagem em tons de cinza e monta estruturas com informações características. A principal estrutura utilizada nesta técnica é a "Matriz de co-ocorrência", e a sua utilização é a identificação de padrões em uma imagem, sendo considerada crucial para pesquisas por similaridade em imagens médicas (GLATARD; MONTAGNAT; MAGNIN, 2004).

Diversas medidas que podem ser extraídas pela análise de textura estão presentes no trabalho de (HARALICK; SHANMUGAM; DINSTEIN, 1973). Algumas das métricas extraídas da análise das matrizes de co-ocorrência são relacionadas com características específicas da textura da imagem como homogeneidade, contraste e a presença de estruturas organizadas dentro da imagem. Outras métricas caracterizam a complexidade e a natureza das transições dos tons de cinza presentes na imagem, como a entropia.

Os extratores de características das imagens utilizados neste trabalho são do framework Arboretum, desenvolvido pelo Grupo de Bases de Dados e Imagens (GBDI) da Universidade de São Paulo - campus São Carlos.

3 TÉCNICA OMNI

Neste capítulo será explanada a técnica Omni, mencionada previamente nos Capítulos 1 e 2. A técnica Omni, sua concepção, construção e uso é retirada do trabalho de (SANTOS FILHO et al., 2001) e (TRAINA JUNIOR et al., 2007).

3.1 CONCEITOS E DEFINIÇÕES OMNI

Como apresentado anteriormente, as técnicas da família Omni baseiam-se em em uma série de elementos chamada de Omni-focos. Estes focos são elementos presentes na base de dados previamente escolhidos, e possuem como função servirem de marco para o cálculo das coordenadas Omni de cada elemento presente no banco.

Definição 7 *Seja um espaço métrico* $M =< S,d >$*, uma base de focos Omni é um conjunto* $\mathscr{F} = \{f_1, f_2, ..., f_l | f_k \in S, f_k \neq f_j, l \leq N\}$ *onde cada* f_k *é um foco (ou ponto focal) de* S*,* l *é o número de focos da base de focos e* N *é o número de elementos da base de dados.*

Definição 8 *Dado um objeto* $s_i \in S$ *e a base de focos Omni* $\mathscr{F}$*, as coordenadas Omni* C_i *do objeto é o conjunto de distâncias de* s_i *para cada foco em* $\mathscr{F}$*:*

$$C_i = \{< f_k, d(f_k, s_i) >, \forall f_k \in \mathscr{F}\} \tag{6}$$

Para distinguir a distância $d(f_k, s_i)$ como uma coordenada, é utilizada a notação $df_k(s_i) = d(f_k, s_i)$.

Quando um novo objeto é inserido na base de dados, as suas coordenadas Omni são calculadas e armazenadas. Em uma pesquisa por similaridade, estas coordenadas são utilizadas para podar o número de cálculos de distância, como ilustrado pela Figura 3.

O custo associado a utilização da técnica Omni provém de duas fontes: o tempo para calcular as coordenadas Omni (cálculo da distância entre o elemento inserido e cada um dos focos) e o espaço físico necessário para armazenar a estrutura Omni, geralmente armazenada em disco. Como o número de focos necessários usualmente é baixo, estes custos atrelados a técnica Omni são menores do que o ganho de desempenho nas consultas.

3.2 USO DA BASE DE FOCOS

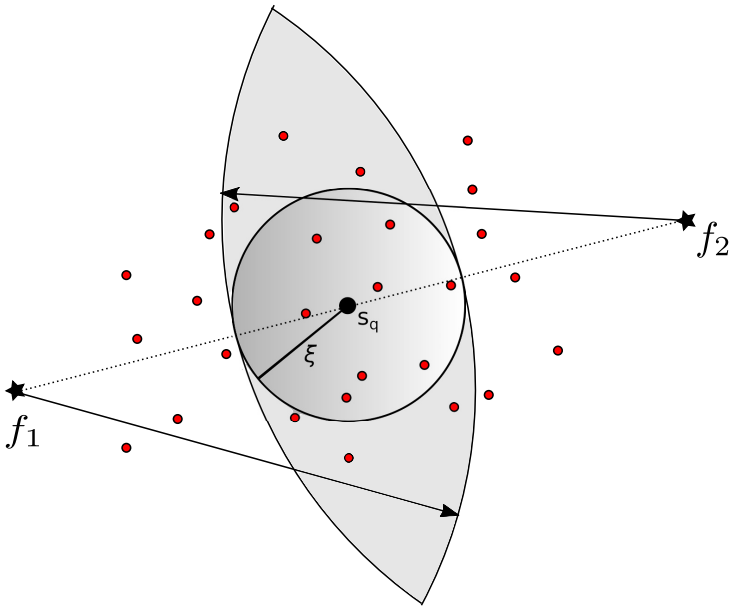
Os problemas entre a escolha da base de focos $\mathscr{F}$ e a sua cardinalidade l são intimamente relacionados. Quanto mais focos, mais espaço e tempo para processamento é necessário. Por isso, é necessário maximizar o ganho de desempenho com a menor quantidade possível de focos.

Definição 9 *Dado uma base de focos Omni* $\mathscr{F} = \{f_1, f_2, ..., f_l\}$ *e uma coleção de objetos* $A = \{x_1,x_2,...,x_n\} \subset \mathbb{S}$*, a região Omni de delimitação mínima (minimum bounding Omni region - mbOr) de* A *é definido como a interseção dos intervalos métricos* $R_A = |\,{}_1^l I_i$*, onde* $I_i = [min(d(f_i,x_j)), max(d(f_i,x_j))]$, $1 \leq i \leq l$, $1 \leq j \leq n$.

A ideia gráfica de uma *mbOr* foi previamente apresentada na Figura 3 para o caso de uma consulta utilizando um único foco. É possível ver que a *mbOr* sempre inclui todos os elementos do conjunto-resposta, mas pode incluir elementos que não são pertinentes ao conjunto-resposta (alarmes falsos). Embora seja necessária mais uma etapa de cálculo de distâncias (etapa de refinamento), o número de cálculos é reduzido drasticamente com o uso da *mbOr*.

Uma *mbOr* pode ser reduzida ainda mais com o uso de múltiplos focos, como ilustra a Figura 6.

Figura 6 – Consulta por abrangência pela técnica Omni utilizando dois focos



Fonte: Autoria Própria

Considerando a família Minkowski de distâncias (métricas L_p), um número de focos correspondente ao valor da dimensão intrínseca da base de dados acrescido de um ($\lceil D \rceil + 1$) seria o suficiente para maximizar a performance da técnica Omni. O uso de mais focos do que o necessário traria pouca ou nenhuma redução à mbOr.

3.3 ESCOLHA DOS FOCOS

Para a escolha dos focos Omni, o elemento candidato a foco deve pertencer a base de dados. Isso se deve ao fato de que algumas vezes é impossível sintetizar um objeto de uma base de dados, como uma impressão digital. O algoritmo utilizado para o encontro dos focos é

o algoritmo HF. Esse algoritmo procura aleatoriamente um elemento s_1, encontra o elemento f_1 mais distante daquele e o seleciona como o primeiro foco. Após isso, procura pelo elemento f_2 mais longe de f_1 e o seleciona como o segundo foco, e armazenando a distância entre eles como $borda$.

O próximo foco é o elemento com as distâncias mais similares aos outros focos previamente escolhidos. Para cada objeto s_i não escolhido como foco ainda, o erro da distância em relação à borda é:

$$erro_i = \sum_{k}^{kfoco} |borda - d(f_k, s_i)| \tag{7}$$

Algoritmo 1: Algoritmo HF

Input: a base de dados $\mathbb{S}$ e o número de focos l
Output: base de focos Omni $\mathscr{F}$
Início
1.Selecione aleatoriamente um objeto $s_i \in \mathbb{S}$.
2.Encontre o objeto f_1 mais longe de s_i. Insira f_1 em $\mathscr{F}$.
3.Encontre o objeto f_2 mais longe de f_1. Insira f_2 em $\mathscr{F}$.
4.Defina $borda = d(f_1,f_2)$, usada para calcular $erro_i$.
while *existirem focos a serem determinados* **do**
  5.Para cada $s_i \in \mathbb{S}$, $s_i \notin \mathscr{F}$: calcule $erro_i$.
  6.Selecione $s_i \in \mathbb{S}$ de tal modo que $erro_i$ seja mínimo.
  7.Insira s_i em $\mathscr{F}$.
end

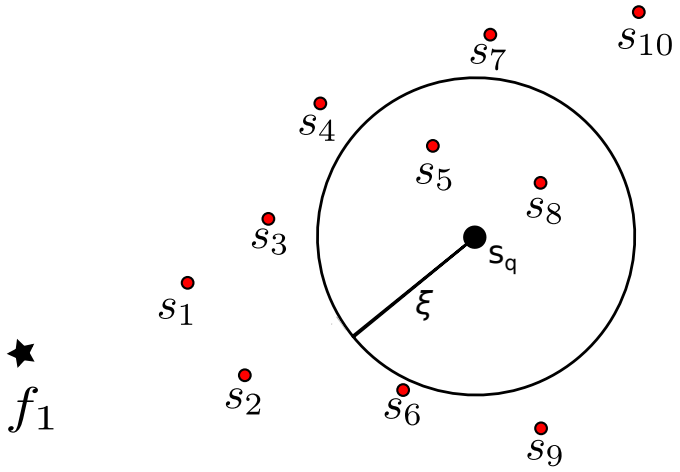
O algoritmo HF requer $l * N$ cálculos de distância. O Algoritmo 1 mostra que as etapas 3 e 5 são cálculos necessários para a determinação das coordenadas Omni de cada objeto. Isso mostra que o algoritmo HF também prepara as coordenadas Omni da base de dados. Outro fato importante é o da base de focos Omni ser invariável a operações de inserção ou remoção na base.

3.4 INDEXAÇÃO DAS COORDENADAS OMNI

Objetos em um espaço métrico não são ordenáveis, portanto métodos de acesso baseados na propriedade de ordenação total como B-Trees não podem ser utilizados diretamente para a sua indexação. No entanto, as distâncias $df_k(s_i)$ de cada foco f para a base de objetos pode ser ordenada e indexada por estruturas B-Tree. Assim, é possível armazenar as coordenadas Omni em um conjunto de h B-Trees, sendo h o número de focos da base. Cada nó da *k-ésima* B-Tree é composto pela distância $df_k(s_i)$ e o identificador interno do objeto *IOid*(s_i). Este conjunto de B-Trees é chamado de OmniB-Forest, e fornece um suporte efetivo para bases de dados imersas em um espaço métrico, utilizando recursos nativos a SGBDRs.

Um exemplo do uso das coordenadas Omni para a filtragem inicial do cálculo de distância dos objetos da base em relação ao elemento central da consulta no caso de uma consulta por abrangência pode ser visto abaixo.

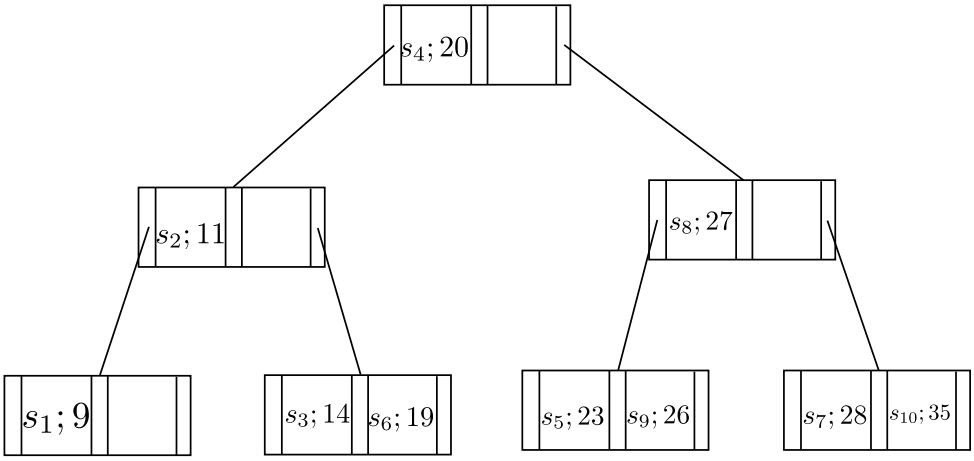
Figura 7 – Base de elementos de exemplo para uma consulta por abrangência utilizando um foco



Fonte: Autoria Própria

A Figura 8 representa a OmniB-Tree utilizada para armazenar as coordenadas Omni dos objetos da base em relação ao foco f_1. Nela foram armazenadas a distância do objeto s_i até o foco, e um identificador do objeto.

Figura 8 – OmniB-Tree utilizada para o exemplo

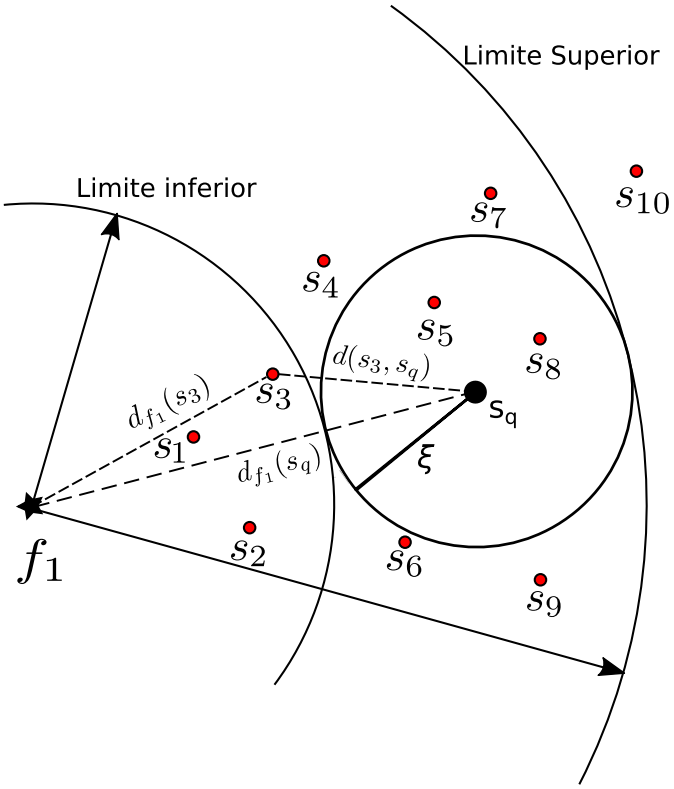


Fonte: Autoria Própria

Analisando geometricamente, é possível perceber que um elemento candidato a estar presente no conjunto de resposta deve ter a sua distância em relação ao foco maior ou igual a distância do elemento central de consulta menos o valor do raio da consulta, formando, assim, um limite inferior. Analogamente, o limite superior pode ser definido utilizando a distância do elemento central da consulta até o foco mais o valor do raio da consulta.

$$|d(f_k, s_q) - \xi| \leq d(f_k, s_i) \leq |d(f_k, s_q) + \xi| \tag{8}$$

Figura 9 – Análise geométrica dos limites dos elementos candidatos



Fonte: Autoria Própria

A Figura 9 ilustra a utilização da propriedade da desigualdade triangular para a etapa de filtragem. Tomando o elemento s_3 como referência e a equação da desigualdade triangular apresentada na Definição 3, é possível definir que:

$$d_{f_1}(s_3) \leq d_{f_1}(s_q) + d(s_3, s_q) \tag{9}$$

$$d(s_3, s_q) \geq |d_{f_1}(s_3) - d_{f_1}(s_q)| \tag{10}$$

Utilizando o conceito de bola apresentado pela Definição 4, se o elemento s_3 só pode pertencer a bola de consulta se a distância $d(s_3, s_q)$ for menor do que o raio de consulta ξ. Com a Equação 10, é possível definir que o elemento está fora do conjunto de elementos candidatos se:

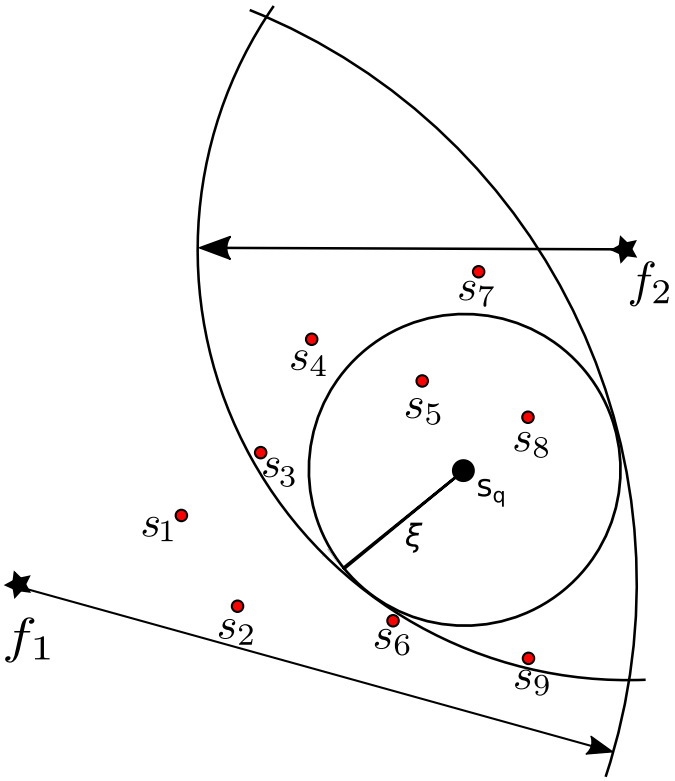
$$|d_{f_1}(s_3) - d_{f_1}(s_q)| \geq \xi \tag{11}$$

Sabendo que a distância $df_1(s_q)$ é de 23 unidades e o raio da consulta por abrangência é de 8 unidades, utilizando a Equação 8 é possível determinar que a distância $d(f_k, s_i)$ do elemento candidato precisa ser ≥ 15 e ≤ 31. Utilizando a OmniB-Tree criada para o foco f_1, é possível encontrar os elementos candidatos que prosseguirão para a etapa de refinamento, sendo eles: $s_4, s_5, s_6, s_7, s_8, s_9$. É possível notar que estes candidatos estão em conformidade com o que foi previsto utilizando uma análise geométrica fornecida pela Figura 9. Também é

possível verificar a validade da Equação 11, pois esta fornece o mesmo conjunto de elementos encontrados com a análise geométrica.

É importante notar que nem todos os elementos selecionados durante a etapa de filtragem da técnica Omni pertencem ao conjunto resposta. Na *mbOr*, pode ocorrer a presença de elementos que não estão dentro da bola da consulta por abrangência (alarmes falsos). Para minimizar o número de alarmes falsos, múltiplos focos podem ser utilizados, como demonstrado pela Figura 10. Para este exemplo, o elemento s_{10} é escolhido como um novo foco f_2. O elemento s_6 é descartado antes da etapa de refinamento, o que não acontece para uma filtragem utilizando um único foco.

Figura 10 – Etapa de filtragem com dois focos



Fonte: Autoria Própria

Referências

BARIONI, M. C. N. et al. Seamlessly integrating similarity queries in sql. **Software: Practice and Experience**, John Wiley & Sons, Ltd., v. 39, n. 4, p. 355–384, 2009. ISSN 1097-024X. Disponível em: <http://dx.doi.org/10.1002/spe.898>.

BUGATTI, P. H. et al. Prosper: Perceptual similarity queries in medical cbir systems through user profiles. **Computers in biology and medicine**, Elsevier, v. 45, p. 8–19, 2014.

CHORAS, R. S. Image feature extraction techniques and their applications for cbir and biometrics systems. **International journal of biology and biomedical engineering**, v. 1, n. 1, p. 6–16, 2007.

DB-ENGINES. **DB-Engines Ranking**. 2017. Disponível em: <https://db-engines.com/en/ranking>. Acesso em: 30 de agosto de 2017.

FERREIRA, M. R. P. et al. Identifying algebraic properties to support optimization of unary similarity queries. v. 450, 05 2009.

GLATARD, T.; MONTAGNAT, J.; MAGNIN, I. E. Texture based medical image indexing and retrieval: Application to cardiac imaging. In: **Proceedings of the 6th ACM SIGMM International Workshop on Multimedia Information Retrieval**. New York, NY, USA: ACM, 2004. (MIR '04), p. 135–142. ISBN 1-58113-940-3. Disponível em: <http://doi.acm.org/10.1145/1026711.1026734>.

GUTTA, S.; WECHSLER, H. Face recognition using hybrid classifiers. **Pattern Recognition**, Elsevier, v. 30, n. 4, p. 539–553, 1997.

HARALICK, R. M.; SHANMUGAM, K.; DINSTEIN, I. Textural features for image classification. **IEEE Transactions on Systems, Man, and Cybernetics**, SMC-3, n. 6, p. 610–621, Nov 1973. ISSN 0018-9472.

JAIN, A. K.; DUBES, R. C. **Algorithms for Clustering Data**. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1988. ISBN 0-13-022278-X.

KOKARE, M.; BISWAS, P.; CHATTERJI, B. Texture image retrieval using rotated wavelet filters. **Pattern Recognition Letters**, v. 28, n. 10, p. 1240 – 1249, 2007. ISSN 0167-8655. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0167865507000608>.

LEHMAN, T. J.; CAREY, M. J. A study of index structures for main memory database management systems. In: **Proceedings of the 12th International Conference on Very Large Data Bases**. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1986. (VLDB '86), p. 294–303. ISBN 0-934613-18-4. Disponível em: <http://dl.acm.org/citation.cfm?id=645913.671312>.

LEHMANN, T. M. et al. Content-based image retrieval in medical applications: a novel multistep approach. In: INTERNATIONAL SOCIETY FOR OPTICS AND PHOTONICS. **Storage and Retrieval for Media Databases 2000**. [S.l.], 1999. v. 3972, p. 312–321.

LIMA, E. **Espaços métricos**. [S.l.]: Instituto de Matemática Pura e Aplicada, CNPq, 1977. (Projeto Euclides).

MARCHIORI, A. et al. Cbir for medical images - an evaluation trial. In: **Proceedings IEEE Workshop on Content-Based Access of Image and Video Libraries (CBAIVL 2001)**. [S.l.: s.n.], 2001. p. 89–93.

POLA, I. R. V. **Explorando conceitos da teoria de espaços métricos em consultas por similaridade sobre dados complexos**. Agosto 2010. Tese (Doutorado em Ciência da Computação) — Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2010.

SANTOS FILHO, R. F. et al. Similarity search without tears: The omni family of all-purpose access methods. In: **Proceedings of the 17th International Conference on Data Engineering**. Washington, DC, USA: IEEE Computer Society, 2001. p. 623–630. ISBN 0-7695-1001-9. Disponível em: <http://dl.acm.org/citation.cfm?id=645484.656543>.

SHIRALI, S.; VASUDEVA, H. **Metric Spaces**. Springer London, 2005. ISBN 9781846282447. Disponível em: <https://books.google.com.br/books?id=MXUbkAhMjLQC>.

SILBERSCHATZ, A.; KORTH, H. F.; SUDARSHAN, S. **Database system concepts**. 6. ed. [S.l.]: McGraw-Hill, 2011.

SWAIN, M. J.; BALLARD, D. H. Color indexing. **International Journal of Computer Vision**, v. 7, n. 1, p. 11–32, Nov 1991. ISSN 1573-1405. Disponível em: <https://doi.org/10.1007/BF00130487>.

TRAINA JUNIOR, C. et al. The omni-family of all-purpose access methods: a simple and effective way to make similarity search more efficient. v. 16, p. 483–505, 08 2007.

ZIGHED, D. A. et al. **Mining Complex Data**. 1st. ed. [S.l.]: Springer Publishing Company, Incorporated, 2008. ISBN 3540880666, 9783540880660.