Universidade de São Paulo
Instituto de Matemática e Estatística
Bacharelado em Ciência da Computação

Rodrigo Guerrero Zerbini

Complex Network Spectrum Signature - Um estudo sobre um descritor de formas via análise espectral de grafos

São Paulo
Novembro de 2016

Complex Network Spectrum Signature - Um estudo sobre um descritor de formas via análise espectral de grafos

Monografia final da disciplina
MAC0499 – Trabalho de Formatura Supervisionado.

Supervisor: Prof. Dr. Paulo A. V. de Miranda

São Paulo
Novembro de 2016

Resumo

Recuperação de imagens baseada em conteúdo é uma relevante área da visão computacional pois ela possibilita buscar e organizar imagens digitais através de seus conteúdos visuais. Neste trabalho, este processo é realizado por meio de descritores de formas e medidas de similaridade. Algumas aplicações destes descritores são, por exemplo, correspondência de imagens, reconhecimento de caracteres e classificação de forma. Em 2015, um novo método descritor de formas foi desenvolvido: o *Complex Network Spectrum Signature* (CNSS). Este método é baseado em redes complexas e teoria espectral de grafos e apresentou um alto grau de precisão nas bases em que foi testado. O CNSS utiliza o contorno da forma de um objeto para criar uma rede de pixels interconectados e então calcula um vetor de características para essa rede, o qual resume os principais atributos da forma do objeto. Através da comparação entre os vetores de características é possível medir a similaridade entre imagens. Neste trabalho, uma implementação do CNSS é feita e testada para duas bases de imagens. Apesar de o método ter sido implementado exatamente como descrito no artigo, os resultados obtidos apresentaram uma precisão abaixo da relatada pelos autores.

Palavras-chave: recuperação de imagens, descritor de forma, rede complexa, teoria espectral de grafos.

Abstract

Content-based image retrieval is a relevant area of computer vision because it enables the search and organization of digital images through their visual content. In this work, this process is carried out through shape descriptors and similarity measures. Some applications of these descriptors are, for example, image matching, character recognition, and shape classification. In 2015 a new shape descriptor method was developed: the Complex Network Spectrum Signature (CNSS). This method is based on complex networks and spectral graph theory and presented a high degree of precision in the databases tested. The CNSS uses the shape contour of an object to create a network of interconnected pixels and then calculates a feature vector for that network, which summarizes the main attributes of the shape of the object. By comparing feature vectors it is possible to measure the similarity between images. In this work, a CNSS implementation is made and tested for two image databases. Although the method was implemented exactly as described in the article, the results obtained presented a precision below that reported by the authors.

Keywords: image retrieval, shape descriptor, complex network, spectral graph theory.

Sumário

Capítulo 1

Introdução

A interação entre luz e objetos permite que seres humanos e animais usem o sentido da visão para obter e analisar informações visuais do ambiente externo e, assim, aprender sobre o mundo que os rodeia (da F. Costa e Jr., 2009).

Com o surgimento dos computadores, inúmeras atividades, antes somente realizadas por seres humanos, puderam ser executadas automaticamente, tais como a identificação de pessoas a partir de características biométricas através de imagens digitais, a análise de sequências de DNA, a busca de imagens baseada em seus conteúdos, dentre outras (Souza, 2013). No entanto, devido à necessidade de obter resultados cada vez mais rapidamente e com maior precisão, torna-se imprescindível o desenvolvimento contínuo de novos sistemas de Visão Computacional que simulem esse sentido natural (de Jesus Gomes Leandro, 2014).

Particularmente, a assimilação e representação das formas de objetos destacam-se entre as competências mais significativas da visão humana e computacional (Garrigan e Kellman, 2011). Neste trabalho, o foco será em um sistema de visão computacional na área de recuperação de imagens baseada em conteúdo, conhecida como CBIR (*Content-Based Image Retrieval*). Ela pode ser entendida como qualquer tecnologia que ajude a buscar e organizar uma coleção de imagens digitais através de seus conteúdos visuais (Datta *et al.*, 2008). O fato do sistema ser baseado em conteúdo significa que ele utiliza exclusivamente informações contidas na própria imagem, descartando o uso de qualquer outro dado associado à mesma, como *tags* ou descrições.

Diversas características da imagem podem ser utilizadas para a busca, como cor, textura e forma. Utilizar a forma de um objeto para reconhecimento de objetos e recuperação de imagens é um importante tópico na área de visão computacional e ainda uma tarefa bastante complexa (Shu e Wu, 2010). Neste contexto, o desafio está em desenvolver descritores de formas e medidas de similaridades com alta acurácia. Algumas aplicações destes descrito-

res são, por exemplo, correspondência de imagens, classificação de forma e reconhecimento de caracteres (Amanatiadis *et al.*, 2011).

Em 2015, três pesquisadores do Rio Grande do Sul propuseram um novo método descritor de formas (de Oliveira *et al.*, 2015). O nome dado a este método foi *Complex Network Spectrum Signature* (CNSS).

Como o próprio nome sugere, ele é baseado em redes complexas e teoria espectral de grafos. O CNSS cria, para cada imagem, uma rede de pixels interconectados para o qual um vetor de características é calculado. A comparação entre os vetores de características, os quais resumem as principais características da forma do objeto, é feita por meio de uma função de distância. Quanto menor a distância entre dois vetores característicos, mais as imagens são similares.

O CNSS, além de apresentar um alto grau de precisão nas bases testadas, é bastante rápido quando comparado com outros métodos: sua função de distância tem complexidade linear. O método *Height Functions*, por exemplo, é utilizado para reconhecimento de contornos e tem função de distância com complexidade cúbica, o que inviabiliza sua aplicação em bases maiores (Wang *et al.*, 2012). Percebe-se assim a importância do CNSS, sendo possível compará-lo ao estado da arte. Contudo, seu potencial ainda não foi inteiramente explorado, uma vez que ele é bastante recente e não foi ainda extensivamente testado para diferentes tipos de bases.

1.1 Objetivos

Os objetivos principais deste trabalho são fazer uma revisão bibliográfica sobre alguns conceitos teóricos importantes tais como descritores de forma, redes complexas e teoria espectral de grafos, implementar o *Complex Network Spectrum Signature* e repetir os testes para algumas bases presentes no artigo. Espera-se, desse modo, obter resultados próximos àqueles obtidos pelos autores.

Considerando que os objetivos acima tenham sido alcançados, o trabalho passa a ser direcionado em busca de objetivos secundários: melhorar a performance do método e aplicá-lo a outros tipos de bases de imagens. Assim, como um segundo objetivo, pretende-se realizar alterações no algoritmo de modo a torná-lo ainda mais preciso na tarefa de identificar e classificar imagens semelhantes de uma mesma classe. Além disso, como um terceiro objetivo, sua performance será avaliada quando aplicado a diferentes bases de imagens não testadas anteriormente.

1.2 Organização da monografia

A monografia está organizada como relatado abaixo. No capítulo 2 são apresentados os conceitos teóricos utilizados no trabalho. No capítulo 3 o algoritmo do descritor de formas é explicado passo a passo e no capítulo 4 são expostos os detalhes da implementação. No capítulo 5 são apresentados os experimentos para validar o algoritmo implementado. No capítulo 6 encontram-se os resultados do descritor de formas aplicado a algumas bases de imagens. No capítulo 7 são feitas as conclusões e no capítulo 8, dedicado à parte subjetiva, são discutidos os desafios e frustrações presentes ao longo do processo de realização deste trabalho e quais disciplinas foram relevantes para o mesmo. Finalmente, no capítulo 9 são feitos os agradecimentos.

Capítulo 2

Conceitos Teóricos

Neste capítulo serão apresentados os conceitos teóricos relevantes para a produção deste trabalho. Na seção 2.1 serão discutidos os diferentes tipos de descritores de formas para imagens bidimensionais. Na seção 2.2 a ideia de redes complexas será introduzida e na seção 2.3 a teoria de grafos será abordada, com ênfase na teoria espectral de grafos.

2.1 Descritores de Formas

Dada a representação de um objeto através de uma imagem, é imprescindível valer-se de algum método que seja capaz de caracterizar numericamente a forma desse objeto. Métodos que realizam tal tarefa são chamados descritores de formas.

Eles são responsáveis por colher toda informação significativa relativa à forma do objeto e transformá-la em dados numéricos, a partir dos quais é possível calcular medidas de similaridade e realizar comparação entre formas. Em geral o produto dessa transformação é um vetor numérico, o qual deve resumir as principais características da forma.

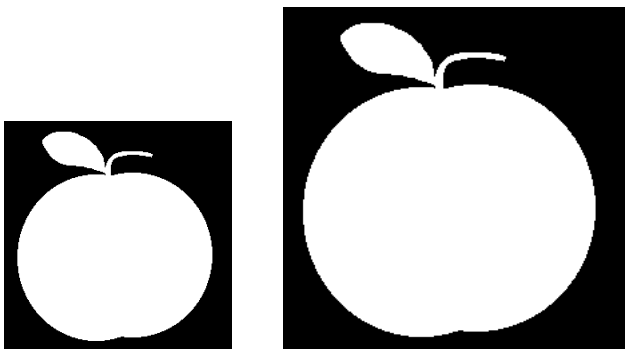
A eficiência de um descritor de formas é função de sua capacidade de gerar descritores semelhantes para formas semelhantes. Objetos com formas semelhantes podem ser agrupados em conjuntos chamados classes. Para uma melhor compreensão, tome um conjunto de imagens de maçãs. Duas maçãs nunca apresentam formas exatamente iguais, mas elas, certamente, guardam muitas semelhanças: ambas lembram uma forma esférica e podem conter um curto cabo no topo, no qual uma ou mais folhas estão atadas.

O olho humano é capaz de identificar as formas dos objetos da figura 2.1 e classificá-las como maçãs. Um descritor de formas eficiente deve conseguir realizar o mesmo trabalho, gerando assim descritores semelhantes para objetos de uma mesma classe. Analogamente,

quanto maior a diferença entre as formas de dois objetos, maior deve ser a discrepância entre seus descritores.

Figura 2.1: *Algumas maçãs (amostras da base MPEG7)*

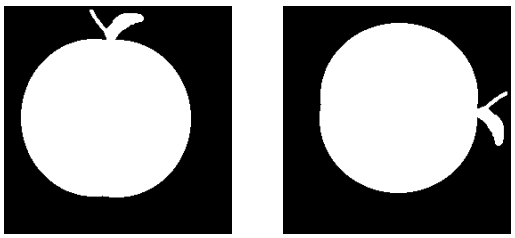
Os descritores de forma também devem ser invariantes a efeitos de escala e rotação dos objetos. Em outras palavras, é necessário que o descritor possa identificar a semelhança entre duas formas ainda que os seus tamanhos sejam diferentes.



(a) *Maçã* **(b)** *Maçã 50% maior*

Figura 2.2: *Maçãs em diferentes escalas*

As duas formas da figura 2.2 são idênticas, exceto pela escala, e o descritor deve ser capaz de identificar que representam o mesmo objeto. Do mesmo modo, se uma forma é rotacionada (figura 2.3), espera-se que o descritor reconheça que são o mesmo objeto, independentemente do ângulo de rotação da imagem.



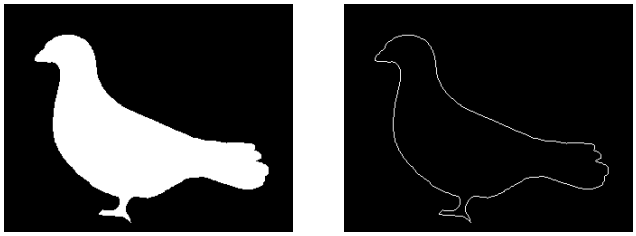
(a) *Maçã* **(b)** *Maçã após rotação*

Figura 2.3: *Maçãs com diferentes ângulos de rotação*

Descritores de formas podem ser divididos em duas classes de métodos: métodos baseados no contorno e métodos baseados na região. Se as características da forma são

extraídas somente de sua borda, o método é classificado como de contorno. Por outro lado, se toda a região da forma é considerada na obtenção das características da forma, então o método é classificado como de região.

A figura 2.4 mostra um exemplo da região e do contorno da figura de um pássaro.



(a) *Região da forma* **(b)** *Contorno da forma*

Figura 2.4: *Exemplo dos tipos de formas*

2.1.1 Descritores baseados no contorno

Descritores baseados no contorno utilizam apenas a borda da forma no processo de extração de suas características. Eles podem ser classificados em dois tipos: abordagem contínua e abordagem discreta.

Métodos de abordagem contínua, também chamados de globais, calculam um vetor característico com as informações da borda do objeto. Estes vetores, juntamente com alguma métrica para calcular a distância entre eles, são utilizados para fazer comparações de similaridade entre formas. O CNSS, método tema deste trabalho, é um exemplo de método baseado no contorno de abordagem contínua.

Já os métodos de abordagem discreta, também chamados de estruturais, dividem o contorno em segmentos de acordo com algum critério em particular e, em seguida, calculam uma *string* ou um grafo como representação da forma do objeto. As medidas de similaridade são feitas através da comparação entre essas *strings* ou grafos (Zhang e Lu, 2004).

2.1.2 Descritores baseados na região

Descritores baseados na região extraem informação de toda a extensão do forma do objeto, ou seja, levam em conta não somente a borda da forma mas também todos os pixels delimitados por ela.

Uma vantagem dos métodos baseados na região é que eles podem ser aplicados a formas não conectadas e disjuntas. Entretanto, representações baseadas na região não res-

saltam as características do contorno, as quais são primordiais para a percepção de formas pela visão humana (Zhang e Lu, 2003).

Os descritores baseados na região também podem ser divididos em duas classes: globais, que computam um vetor considerando a totalidade dos pixels, e estruturais, que extraem informações de segmentos da região a fim de representar a forma em uma *string* ou um grafo.

2.2 Redes Complexas

Redes estão em toda parte e diversos tipos podem ser encontrados no cotidiano: redes de energia elétrica, de telecomunicações, de transporte público, de negócios de uma empresa. Com o avanço da tecnologia, outras redes surgiram, como redes de computadores, a Internet, redes neurais e as populares redes sociais. Os seres humanos também podem ser considerados, como indivíduos, partes de uma rede de relacionamentos interpessoais. Além disso, o próprio funcionamento do corpo humano depende de uma complexa rede de reações bioquímicas (Boccaletti *et al.*, 2006).

Desde o início dos anos 2000 houve um aumento substancial de pesquisa nessa área com o intuito de compreender propriedades de redes cada vez maiores. Isto só foi possível devido à disponibilidade de computadores e redes de comunicação, que permitiram coletar e analisar dados em larga escala (Newman, 2003).

Um exemplo evidente de rede complexa é a Internet. No início do ano de 2016 foi apresentado um novo método para estimar o número de páginas indexadas por sites de busca, como Google, Yahoo e Bing (van de Bosch *et al.*, 2016). A estimação é feita através de um método que combina frequências de palavras em coleções de texto offline e o número de páginas contendo aquela palavra que são retornadas pelos motores de busca. Os resultados mostram que a World Wide Web deve conter pelo menos 4.77 bilhões de páginas atualmente (de Kunder, 2016).

Estudos mostram que muitas redes do mundo real compartilham duas propriedades fundamentais: o efeito de mundo pequeno (*small-world*), indicando que as distâncias nessas redes são muito pequenas, e o efeito livre de escala (*scale-free*), relacionado com a grande variabilidade dos graus (número de ligações) dos vértices dessas redes (van der Hofstad, 2009).

Uma rede apresenta o efeito *small-world* se é provável que dois pontos da rede possam ser conectados através de um curto caminho, isto é, passando por poucos vértices. Um exemplo desse fenômeno pode ser observado na rede em forma de anel da figura 2.5.

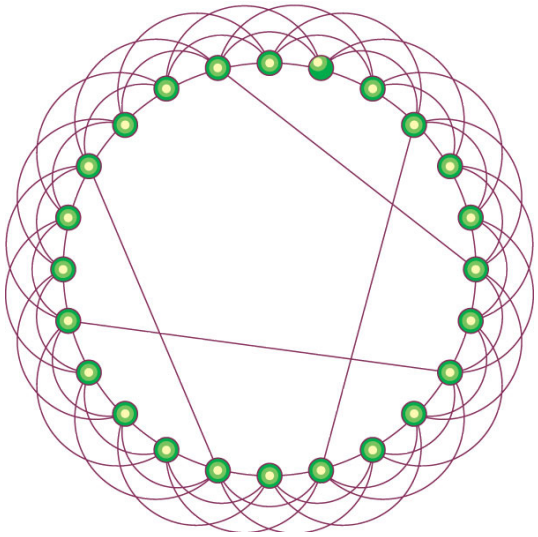
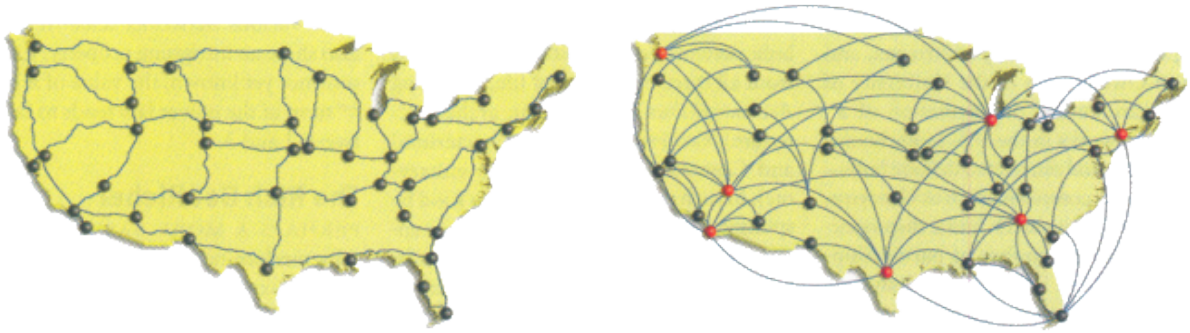


Figura 2.5: *Exemplo de rede de mundo pequeno.*
Figura retirada de (Strogatz, 2001).

Já o efeito *scale-free* se manifesta quando a maioria dos vértices da rede fazem poucas ligações, enquanto que alguns poucos vértices, denominados *hubs*, contêm muitas ligações. Isso ocorre quando a distribuição de graus dos vértices de uma rede segue a lei de potência.

A primeira rede da figura 2.6 representa o sistema simplificado das rodovias nos Estados Unidos. A maioria dos vértices têm mais ou menos o mesmo número de ligações, característica de redes aleatórias. Já a segunda rede representa o sistema simplificado do sistema de transporte aéreo dos Estados Unidos, o qual apresenta atributos de redes de livre escala, com os *hubs* sendo representados em vermelho.



(a) *Rede aleatória* **(b)** *Rede livre de escala*

Figura 2.6: *Exemplo das diferenças entre uma rede aleatória e uma rede livre de escala.*
Figura retirada de (Barabási e Bonabeau, 2003).

O conceito de rede complexa será essencial para o método descritor de formas implementado neste trabalho, dado que os pixels do contorno das formas serão modelados como redes.

2.3 Teoria de Grafos

Em ciência da computação, a teoria de grafos pode ser utilizada para o estudo e representação de uma rede. Um grafo $G = (V, E)$ consiste de um conjunto finito não vazio V (o conjunto de vértices de G) e um conjunto E (o conjunto de arestas de G, que são subconjuntos de dois elementos (i, j) de V). Dois vértices $u, v \in V$ são adjacentes se existe a aresta $(u, v) \in E$ que conecta esses vértices.

A matriz de adjacência A de um grafo G é usada para representar as relações de adjacência entre os vértices, isto é, para indicar quais pares de vértices estão conectados por uma aresta. O elemento a_{ij} da matriz de adjacência A é 1 se os vértices i e j são adjacentes. Caso contrário, o elemento é igual a zero. Matematicamente,

$$a_{ij} = \begin{cases} 1, & \text{se } (i,j) \in E \\ 0, & \text{caso contrário} \end{cases}$$

Como exemplo, a rede ferroviária de um país pode ser representada por um grafo, no qual os vértices do grafo são as cidades atendidas pelos trens e as arestas do grafo correspondem aos trechos de ferrovia que conectam uma cidade à outra.

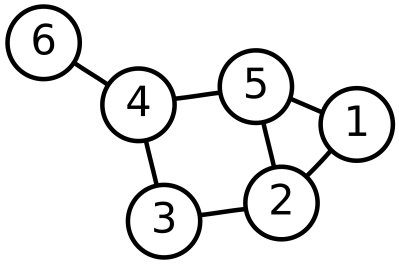


Figura 2.7: *Exemplo de um grafo com 6 vértices e 7 arestas.*
Figura retirada de Wikipedia (https://en.wikipedia.org/wiki/Graph_theory).

Se o grafo da figura 2.7 for a representação da rede ferroviária do exemplo anterior, é possível concluir que há 6 cidades e 7 ferrovias que interligam pares de cidades. Assim, estando na cidade 4, é possível viajar de trem para as cidades 3, 5 e 6. Entretanto, para ir até as cidades 1 e 2 será necessário passar por outras cidades, uma vez que não há ferrovias (arestas) ligando as cidades (vértices) diretamente.

A matriz de adjacência A do grafo da figura 2.7 é da seguinte forma:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

É importante mencionar que este grafo é denominado não direcionado, pois o seu conjunto de arestas traduz relações de adjacência simétrica, isto é, as arestas (1, 2) e (2, 1), por exemplo, são consideradas como uma única aresta. Isso significa que se é possível viajar de uma cidade u para uma cidade v, então também é possível viajar de v para u.

Em grafos não direcionados, as matrizes de adjacência são simétricas com relação à diagonal principal. Isso significa que o elemento a_{ij} é igual ao elemento a_{ji}, o que pode ser observado na matriz de adjacência A do grafo da figura 2.7.

As arestas de um grafo também podem ser associadas a um valor denominado peso. O peso de uma aresta guarda alguma informação numérica entre os dois vértices que ela conecta. Assim, ao grafo da figura 2.7, que representa uma malha ferroviária, podem ser adicionados pesos nas arestas para representar, por exemplo, a distância entre as cidades (figura 2.8).

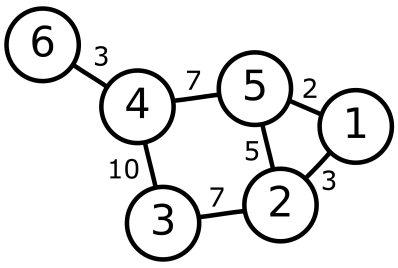


Figura 2.8: *Exemplo de um grafo com pesos nas arestas*

2.3.1 Teoria Espectral de Grafos

Um dos principais objetivos da teoria de grafos é extrair as principais propriedades e a estrutura de um grafo a partir de seu espectro, uma vez que autovalores desempenham um papel fundamental no entendimento de grafos (Chung, 1997). Visando esse objetivo, foi desenvolvida a teoria espectral de grafos, que é o estudo e análise de grafos através de autovalores e autovetores de matrizes derivadas da estrutura desses grafos (Spielman, 2011). Neste trabalho, a matriz adotada para esse estudo será a matriz de adjacência.

Em álgebra linear, os autovalores de uma matriz $A_{n,n}$ são as n raízes de seu polinômio característico, que é definido pela seguinte equação:

$$p(\lambda) = det(\lambda I - A),$$

onde *det* é o operador determinante de uma matriz e I é a matriz identidade de ordem n. O conjunto dessas raízes é chamado espectro e é denotado por $\lambda(A)$. Assim, $\lambda(A) = \{\lambda_1, \lambda_2, ..., \lambda_n\}$ (Golub e Loan, 1996).

No contexto da teoria espectral de grafos o conjunto de autovalores da matriz de adjacência de um grafo G será chamado de espectro do grafo G. Além disso, o espectro do grafo, que será denotado por $\lambda(G)$, é apresentado como o conjunto dos i autovalores distintos associados com suas multiplicidades m (Biggs, 1993):

$$\lambda(G) = \begin{bmatrix} \lambda_1 & \lambda_2 & ... & \lambda_i \\ m(\lambda_1) & m(\lambda_2) & ... & m(\lambda_i) \end{bmatrix}.$$

A primeira linha do espectro lista os autovalores em ordem crescente e a segunda linha contém a multiplicidade de cada autovalor, isto é, o número de vezes que ele foi encontrado.

Embora a utilização do espectro seja uma boa técnica para caracterizar um grafo, é necessário estar atento à questão da coespectralidade. Dois grafos são coespectrais se, apesar de apresentarem propriedades diferentes, possuem o mesmo espectro. Pode-se dizer, então, que esse par coespectral de grafos não pode ser determinado por seus espectros, pois o mesmo espectro caracteriza dois grafos diferentes (de Souza, 2016).

Sabe-se que o número de pares coespectrais aumenta em função do número de vértices do grafo até um valor de $n = 10$ vértices e, então, passa a decrescer. Mais ainda, o número de pares coespectrais tende a zero quando o número de vértices dos grafos tende ao infinito (Haemers e Spence, 2004).

Esse resultado é muito positivo para a realização deste trabalho, pois os grafos que serão utilizados contêm centenas e até milhares de vértices. Logo, supõe-se que a chance de estes grafos não serem determinados por seus espectros deve ser baixa.

Capítulo 3

Algoritmo

Neste capítulo o método *Complex Network Signature Shape* (de Oliveira *et al.*, 2015) será visto em detalhes. Na seção 3.1 é demonstrado como o contorno da forma pode ser modelado em uma rede complexa. Na seção 3.2 é explicado como a teoria espectral de grafos pode ser utilizada para caracterizar os grafos formados pelos pixels do contorno da forma. Na seção 3.3 é apresentada a forma de medir a similaridade entre duas formas através do cálculo da distância entre duas assinaturas de forma.

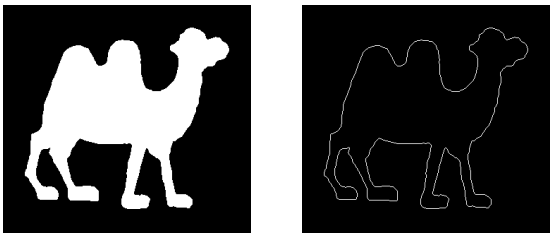
3.1 Modelagem do contorno da forma

Dada a forma de um objeto é simples obter o contorno da mesma. Neste trabalho, os objetos são representados por imagens binárias, isto é, a região da forma do objeto é dada por um conjunto de pixels brancos sobre um fundo preto.

Considerando que cada pixel tem quatro pixels vizinhos (superior, inferior, direita e esquerda), um pixel está na borda se ele faz parte do objeto, ou seja, é um pixel branco, e pelo menos um de seus vizinhos é um pixel preto. Através da varredura da imagem, pixel a pixel, é possível encontrar o contorno da forma, que são todos os pixels que separam a região interior da forma e o fundo preto.

A figura 3.1 mostra um exemplo da região da forma de uma imagem binária retirada da base MPEG-7 (a) e o contorno extraído dessa forma (b).

Define-se o contorno da forma como um conjunto de pixels $S = (p_1, p_2, ..., p_n)$, onde $p_i = (x_i, y_i)$ é o pixel de índice i localizado na coordenada (x_i, y_i) da imagem e n é o número total de pixels do contorno.



(a) *Imagem binária* **(b)** *Contorno da forma*

Figura 3.1: *Exemplos de imagem binária e contorno da forma*

Um grafo $G = (V, E)$ então é formado através da associação de cada pixel do contorno a um vértice do grafo, isto é, $V = S$, e a criação de uma aresta entre todos os pares de vértices, ou seja, para todo par de vértices i e j em V existe uma aresta $e_{ij} \in E$. Esse tipo especial de grafo, onde todo vértice é adjacente a todos os outros vértices, é chamado de grafo completo. Um exemplo de um grafo completo com 7 vértices pode ser visto na figura 3.2.

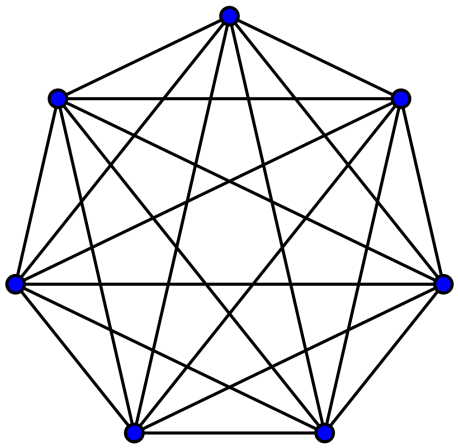


Figura 3.2: *Um grafo completo com 7 vértices.*
Figura retirada de Wikipedia (https://en.wikipedia.org/wiki/Complete_graph)

O próximo passo é associar um peso a cada uma das arestas do grafo. O valor do peso da aresta e_{ij} será dado pela distância euclidiana $d(p_i, p_j)$ entre os vértices p_i e p_j. Sabe-se que a distância euclidiana é invariante a transformações ortogonais, ou seja, seu valor não é afetado se a forma sofre translação ou rotação (Joseph, 2015). Dado dois vértices $p_i = (x_i, y_i)$ e $p_j = (x_j, y_j)$ a distância euclidiana entre eles pode ser calculada usando a seguinte fórmula:

$$d(p_i, p_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

Essas distâncias podem ser organizadas em uma matriz de pesos W, onde cada entrada w_{ij} é a distância entre os vértices p_i e p_j. A fim de evitar efeitos de escala, serão utilizadas as distâncias relativas entre os vértices. Para isso, os pesos da matriz W devem ser normalizados em relação à maior distância entre dois vértices:

$$w_{ij} = \frac{d(i,j)}{max(W)},$$

onde $d(i,j)$ é a distância entre os vértices i e j, max é a função que retorna o máximo valor de seu argumento, W é a matriz de pesos anteriormente calculada e w_{ij} é o novo peso da matriz W normalizada.

A partir da nova matriz de pesos normalizada dez matrizes de adjacência podem ser construídas. Uma vez que todas as distâncias normalizadas encontram-se no intervalo $[0,1]$, cada uma das dez matrizes de adjacência será associada a um décimo desse intervalo.

Desse modo, a primeira matriz de adjacência inclui uma aresta entre dois vértices se as distâncias entre eles está dentro da faixa $[0, 0.1]$. Todas as distâncias fora desse intervalo não são consideradas, ou seja, a matriz de adjacência tem entradas 0 para vértices separados por uma distância superior a 0.1. Analogamente, a segunda matriz inclui uma aresta entre dois vértices se as distâncias entre eles está dentro da faixa $[0.1, 0.2]$ e assim por diante.

Matematicamente, as entradas $m(i,j)_t$ da matriz de adjacência M_t podem ser obtidas com a equação abaixo:

$$m(i,j)_t = \begin{cases} 1, & \text{se } \frac{t-1}{10} \leq w(i,j) \leq \frac{t}{10} \\ 0, & \text{caso contrário} \end{cases} \quad (1 \leq t \leq 10)$$

3.2 Assinatura da forma

Como foi visto na seção 2.3.1, a teoria espectral de grafos pode ser utilizada para extrair importantes informações de um grafo. Sua topologia, por exemplo, pode ser analisada através de autovalores e autovetores associados à matriz de adjacência do grafo. Além disso, uma vez que a matriz de adjacência é simétrica, todos seus autovalores são reais (Kar *et al.*, 2008).

Com as dez matrizes de adjacência $M_1, M_2, ..., M_{10}$ calculadas na seção anterior, os autovalores de cada uma delas podem ser encontrados através da fórmula do polinômio característico:

$$p(\lambda_t) = det(\lambda_t I - M_t) \quad (1 \leq t \leq 10)$$

Em seguida, os autovalores de cada uma das matrizes de adjacência são ordenados e

armazenados em uma outra matriz, juntamente com suas multiplicidades. Essa matriz é o espectro do grafo:

$$\lambda(M_t) = \begin{bmatrix} \lambda_{t,1} & \lambda_{t,2} & ... & \lambda_{t,i} \\ m(\lambda_{t,1}) & m(\lambda_{t,2}) & ... & m(\lambda_{t,i}) \end{bmatrix} \quad (1 \leq t \leq 10)$$

Neste trabalho, somente os autovalores, que estão na primeira linha do espectro, serão utilizados. Na figura 3.3 há um exemplo de forma de avião e a figura 3.4 mostra como os autovalores variam para essa forma. Esse comportamento dos autovalores é típico: nota-se que a maioria dos valores na região central do espectro são zero ou próximos de zero.

Figura 3.3: *Forma de um avião*

Consequentemente, apenas os maiores e os menores autovalores são relevantes para descrever a estrutura de um grafo. O maior autovalor é chamado de raio espectral ou índice do grafo e o menor fornece informações sobre a expansão e as propriedades de aleatoriedade do grafo (Brouwer e Haemers, 2011).

Por essa razão, os autores do artigo decidiram usar apenas os três menores e os três maiores autovalores de cada espectro obtido para a construção da assinatura da forma, outro nome dado aos descritores de forma. Logo, para uma matriz de adjacência M_t com autovalores $\lambda_{t,1}, ..., \lambda_{t,n}$, tomam-se os três menores $(\lambda_{t,1}, \lambda_{t,2}, \lambda_{t,3})$ e os três maiores $(\lambda_{t,n-2}, \lambda_{t,n-1}, \lambda_{t,n})$.

Em seguida, um vetor característico s é formado com os principais autovalores de todas as matrizes de adjacência:

$$\begin{aligned} s = (&\lambda_{1,1}, \lambda_{1,2}, \lambda_{1,3}, \lambda_{1,n-2}, \lambda_{1,n-1}, \lambda_{1,n}, \\ &\lambda_{2,1}, \lambda_{2,2}, \lambda_{2,3}, \lambda_{2,n-2}, \lambda_{2,n-1}, \lambda_{2,n}, ... \\ &\lambda_{10,1}, \lambda_{10,2}, \lambda_{10,3}, \lambda_{10,n-2}, \lambda_{10,n-1}, \lambda_{10,n}) \end{aligned}$$

O operador módulo é então aplicado a todos os elementos do vetor s para que todos

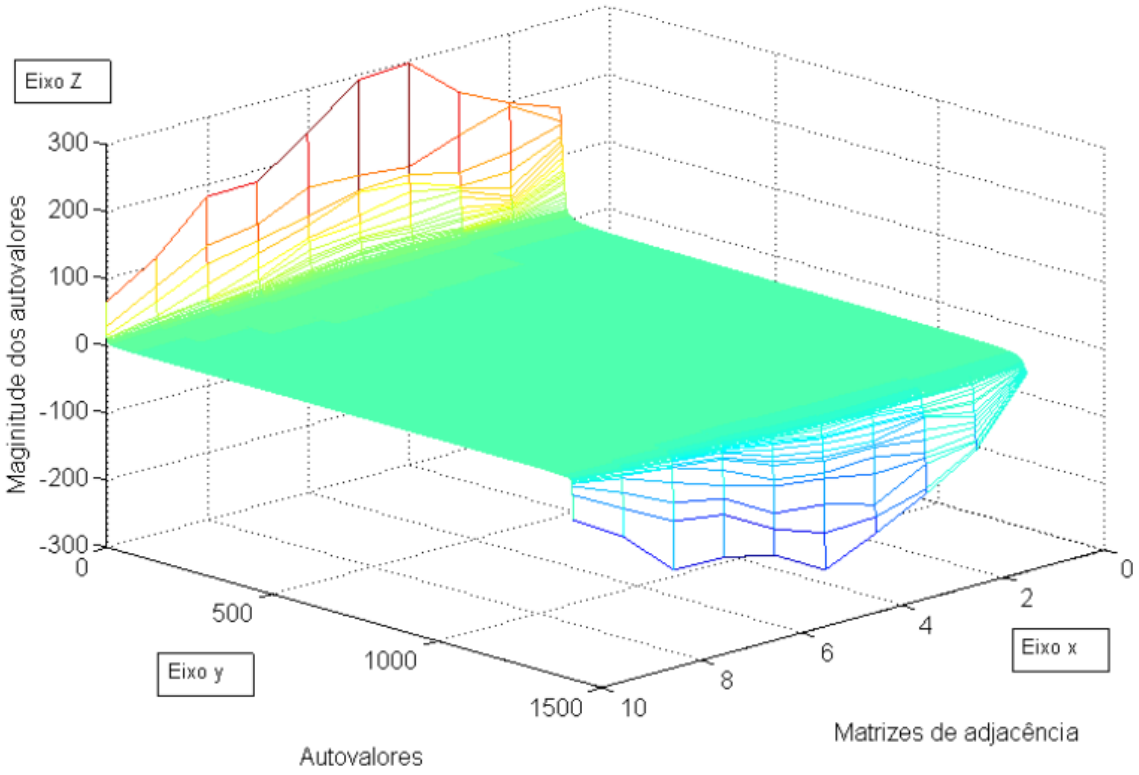


Figura 3.4: *Autovalores para a forma do avião.*
Figura retirada de (de Oliveira et al., 2015)

os valores sejam positivos.

$$\begin{aligned} |s| = (&|\lambda_{1,1}|, |\lambda_{1,2}|, |\lambda_{1,3}|, |\lambda_{1,n-2}|, |\lambda_{1,n-1}|, |\lambda_{1,n}|, \\ &|\lambda_{2,1}|, |\lambda_{2,2}|, |\lambda_{2,3}|, |\lambda_{2,n-2}|, |\lambda_{2,n-1}|, |\lambda_{2,n}|, ... \\ &|\lambda_{10,1}|, |\lambda_{10,2}|, |\lambda_{10,3}|, |\lambda_{10,n-2}|, |\lambda_{10,n-1}|, |\lambda_{10,n}|) \end{aligned}$$

Por fim, o vetor deve ser normalizado para permitir a comparação entre diferentes escalas de autovalores. O vetor resultante sn recebe o nome de assinatura da forma e é obtido através da seguinte equação:

$$sn = \frac{|s|}{\sum_{t=1}^{10}(|\lambda_{t,1}| + |\lambda_{t,2}| + |\lambda_{t,3}| + |\lambda_{t,n-2}| + |\lambda_{t,n-1}| + |\lambda_{t,n}|)}$$

3.3 Similaridade entre formas

Agora que as formas dos objetos estão caracterizadas por suas assinaturas, falta definir um método para realizar a comparação entre duas assinaturas. Isso será feito através da distância de Hellinger, que é função do coeficiente de Bhattacharyya.

O coeficiente de Bhattacharyya $bc(sn_1, sn_2)$ para um par de assinaturas sn_1 e sn_2 é calculado pela equação abaixo:

$$bc(sn_1, sn_2) = \sum_k \sqrt{sn_1(k)sn_2(k)},$$

onde os termos $sn_1(k)$ e $sn_2(k)$ são o k-ésimo valor das assinaturas das formas.

Com o coeficiente de Bhattacharyya determinado é possível calcular a distância de Hellinger $dh(sn_1, sn_2)$ entre as assinaturas das formas:

$$dh(sn_1, sn_2) = \sqrt{1 - bc(sn_1, sn_2)}$$

A distância de Hellinger é um valor entre zero e um e sua função é medir a similaridade entre duas assinaturas de formas. Assim, quanto mais próxima de zero for a distância, mais similares as formas dos objetos devem ser. Por outro lado, quanto maior for a distância, mais distintas devem ser as formas.

Capítulo 4

Implementação

4.1 Primeiros passos

Inicialmente, a ideia era se familiarizar com a manipulação das imagens binárias da base MPEG7 CE Shape-1 Part B[1], que é um dos conjuntos de imagens utilizados para teste no artigo tema deste trabalho. Para isso, foi desenvolvido um programa preliminar que utilizava uma biblioteca de funções C++ disponibilizadas pelo professor Paulo Miranda para realizar as seguintes operações:

- ler uma imagem (recebendo o nome do arquivo como argumento)
- escrever uma imagem (recebendo a imagem e o nome do arquivo como argumento)
- clonar uma imagem (recebendo a imagem como argumento)
- inverter o valor dos pixels da imagem, isto é, os pixels brancos são transformados em preto e vice-versa (recebendo a imagem como argumento)
- extrair a borda da imagem (recebendo a imagem como argumento)

O uso correto das operações acima era essencial e pré-requisito para começar a implementação de fato do *Complex Network Spectrum Signature*.

[1] A base está disponível para download em: http://www.imageprocessingplace.com/root_files_V3/image_databases.htm

4.2 Primeira versão do programa

Assim que as operações básicas da seção 4.1 foram aprendidas através da manipulação de funções, a programação do método pôde ser iniciada. Como as funções de manipulação de imagens estavam escritas em C++, decidiu-se manter essa linguagem de programação. A primeira versão do programa a ser escrito devia receber como entrada duas imagens e devolver a distância entre as suas assinaturas, de modo que a similaridade entre as duas formas pudesse ser verificada.

Primeiramente, o programa lê as duas imagens e faz a extração de seus contornos, que são escritos como duas novas imagens. Para a identificação do contorno, realiza-se a varredura de todos os pixels da imagem e, para cada pixel do objeto (branco) encontrado, verifica-se se algum pixel vizinho faz parte do fundo (preto). Em caso afirmativo, isso significa que aquele pixel do objeto faz parte de seu contorno.

Realiza-se então a varredura das imagens dos contornos, identificando o número de pixels que os compõem e as coordenadas x e y de cada pixel.

Dois grafos G_1 e G_2 são gerados, um para cada imagem, com números de vértices igual ao número de pixels de seus respectivos contornos. Como explicado na seção 3.1, os grafos gerados são completos, ou seja, existe uma aresta entre todo par de vértices. As distâncias entre os vértices são calculadas através da equação de distância euclidiana e esses valores são associados às arestas do grafo. As dez matrizes de adjacência são então geradas para cada um dos grafos.

Para o cálculo dos vetores de características, foi necessário adotar uma biblioteca capaz de calcular autovalores de matrizes. Após uma pesquisa na internet, foi encontrada a ferramenta Eigen, que é uma biblioteca rápida e confiável escrita em C++ utilizada para cálculos de álgebra linear (Guennebaud *et al.*, 2010).

Mais especificamente, a classe utilizada foi a EigenSolver, que computa autovalores e autovetores de matrizes. No Eigen, todas as matrizes são objetos da classe Matrix, então primeiro todas as matrizes de adjacência precisaram ser modeladas como objetos da classe Matrix. Em seguida, foi instanciado um objeto da classe EigenSolver, responsável pela computação dos autovalores.

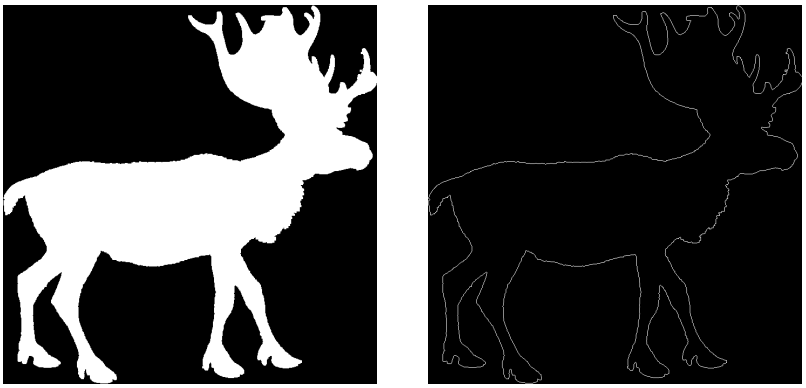
Com os autovalores calculados, bastava selecionar os 3 maiores e os 3 menores associados a cada matriz e armazená-los no vetor de características, chamado de assinatura da forma. A etapa final da implementação - desenvolvimento de funções para o cálculo do coeficiente de Bhattacharyya e da distância entre duas assinaturas - foi relativamente mais fácil do que as etapas anteriores.

4.3 Dificuldades na implementação

Com o método implementado, foi possível fazer alguns testes iniciais. Duas imagens do banco MPEG7 CE Shape-1 Part B eram passadas como argumento para o programa e a distância entre as assinaturas era devolvida.

As primeiras tentativas foram feitas passando como argumentos pares de imagens que pertenciam à mesma classe e as distâncias entre as assinaturas eram próximas de zero, como esperado. Por outro lado, quando as imagens pertenciam a classes diferentes, a distância tendia a ser maior.

Entretanto, notou-se que o programa não terminava de rodar para algumas imagens, como por exemplo *deer-1* (figura 4.1).



(a) *Região da forma* **(b)** *Contorno da forma*

Figura 4.1: *Região e contorno da forma do deer-1*

Mesmo após realizar algumas otimizações no código com o objetivo de torná-lo mais rápido e eficiente, o programa ainda não apresentava a performance desejada e não era capaz de calcular a assinatura para algumas formas. Observou-se que o fator limitante estava na quantidade de cálculos decorrente do alto número de pixels dos contornos dessas formas.

A imagem do contorno do cervo acima, por exemplo, contém 4957 pixels. Isso significa que cada uma das 10 matrizes de adjacência geradas para essa imagem tem 4957 linhas e 4957 colunas, ou seja, mais de 24 milhões de elementos. Para cada matriz, 4957 autovalores tinham que ser calculados e, em seguida, ordenados a fim de se tomar os 3 menores e os 3 maiores.

Consequentemente, foi inevitável buscar algum meio de contornar esse problema.

4.4 Segunda versão do programa

A primeira ideia de lidar com a dificuldade encontrada foi reduzir o número de pontos dos contornos através de um filtro que selecionaria apenas os pontos mais relevantes para a descrição da forma. Entretanto, essa ideia foi logo refutada pois caso fosse adotada haveria perda de informação, isto é, a assinatura da forma não seria tão precisa. Além disso, essa estratégia aumentaria a probabilidade de ocorrer coespectralidade dos grafos.

A segunda tentativa foi utilizar a ferramenta ARPACK++, que é uma coleção de classes escritas em C++ que funcionam como uma interface para ARPACK, um conjunto de funções desenvolvidas em FORTRAN para o cálculo de poucos autovalores e autovetores de matrizes grandes e esparsas (Lehoucq *et al.*, 1996).

Como as matrizes de adjacência são grandes e esparsas, o uso de ARPACK++ poderia ser uma maneira viável de calcular apenas os autovalores relevantes para o método CNSS, dispensando o cômputo de todos os autovalores e sua posterior ordenação.

Todavia, como ARPACK++ simplesmente atua como uma interface para a versão original escrita em FORTRAN, ambos pacotes precisaram ser instalados e a sua configuração e uso mostraram-se ser bastante complicados.

A dificuldade em trabalhar com ARPACK++ levou à busca de uma solução mais simples e, finalmente, a biblioteca Spectra cumpriu esse papel (Qiu, 2015). Spectra é uma abreviação para *Sparse Eigenvalue Computation Toolkit as a Redesigned ARPACK* e consiste em uma coleção de funções em C++ que também têm a função de calcular alguns autovalores de matrizes de grande escala. Spectra foi desenvolvida baseada no ARPACK, mas não depende da biblioteca original em FORTRAN.

Assim, a classe SymEigsSolver, projetada para trabalhar com matrizes reais e simétricas, foi utilizada e permitiu o cálculo de apenas os autovalores relevantes para cada matriz de adjacência.

A segunda versão do programa funcionou para todas as imagens independentemente do tamanho de seus contornos e proporcionou um ganho substancial em tempo.

4.5 Terceira versão do programa

A fim de determinar a similaridade entre todos os pares de imagens das bases a serem testadas, uma terceira versão do programa foi implementada.

Essa versão separou o método em duas partes:

1. Extração da assinatura da forma;
2. Cálculo da distância entre duas assinaturas.

Desse modo, a primeira parte ficou responsável por ler uma só imagem, calcular a assinatura de sua forma e escrevê-la em um arquivo. Este procedimento foi colocado dentro de um *loop*, o que permitiu extrair todas as assinaturas das imagens em um determinado diretório.

Posteriormente, a segunda parte ficou encarregada de ler duas assinaturas e devolver a distância entre elas. Esta tarefa também foi aplicada a todos os pares de assinaturas de um determinado diretório.

Capítulo 5

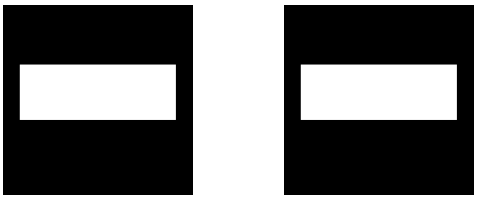
Experimentos

A fim de testar se o cálculo da distância entre duas formas retornava valores consistentes, alguns experimentos foram realizados utilizando a segunda versão do programa e passando como argumento duas imagens simples geradas no Photoshop. O software ImageMagick[1] também foi utilizado para gerar novas imagens fazendo rotações no sentido horário.

5.1 Distâncias entre formas

Os pares de imagens, bem como os índices Bhattacharyya e as distâncias de Hellinger podem ser vistos abaixo.

Experimento 1: rectangle1 e rectangle1



Índice Bhattacharyya	0.9999
Distância de Hellinger	0.0001

Experimento 2: rectangle1 e rectangle1-rotate60

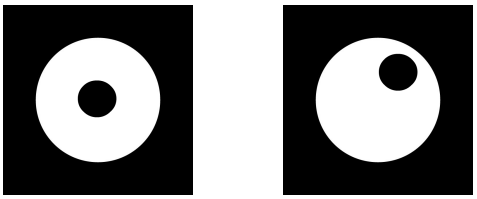
Índice Bhattacharyya	0.9999
Distância de Hellinger	0.0013

[1] https://www.imagemagick.org/

Experimento 3: circle1 e circle2

Índice Bhattacharyya	0.9992
Distância de Hellinger	0.0282

Experimento 4: ring1 e ring2



Índice Bhattacharyya	0.9977
Distância de Hellinger	0.0471

Experimento 5: ring1 e ring3

Índice Bhattacharyya	0.9950
Distância de Hellinger	0.0705

Experimento 6: square1 e square2

Índice Bhattacharyya	0.9924
Distância de Hellinger	0.0869

Experimento 7: square1 e square2-rotate75

Índice Bhattacharyya	0.9914
Distância de Hellinger	0.0924

Experimento 8: rectangle1 e rectangle2

Índice Bhattacharyya	0.9902
Distância de Hellinger	0.0989

Experimento 9: rectangle1 e rectangle2-rotate90

Índice Bhattacharyya	0.9902
Distância de Hellinger	0.0989

Experimento 10: square2 e rectangle2

Índice Bhattacharyya	0.9819
Distância de Hellinger	0.1344

Experimento 11: square2-rotate75 e rectangle1-rotate60

Índice Bhattacharyya	0.9804
Distância de Hellinger	0.1399

Experimento 12: ring1 e square1

Índice Bhattacharyya	0.9659
Distância de Hellinger	0.1844

Experimento 13: square1 e circle1

Índice Bhattacharyya	0.9631
Distância de Hellinger	0.1920

Experimento 14: ring2 e rectangle2

Índice Bhattacharyya	0.9595
Distância de Hellinger	0.2011

Experimento 15: circle2 e rectangle1

Índice Bhattacharyya	0.9575
Distância de Hellinger	0.2059

Experimento 16: circle1 e rectangle2

Índice Bhattacharyya	0.9520
Distância de Hellinger	0.2190

Experimento 17: square2-rotate75 e circle2

Índice Bhattacharyya	0.9511
Distância de Hellinger	0.2209

5.2 Análise dos experimentos

Analisando as distâncias entre as assinaturas das formas, observa-se que as distâncias são menores para pares de imagens da mesma classe (experimentos de 1 a 9) e maiores para pares de imagens de classes diferentes (experimentos de 10 a 17).

Além disso, as distâncias entre as imagens tendem a aumentar à medida que as formas tornam-se mais diferentes. Por exemplo, as distâncias entre um quadrado e um retângulo (experimentos 10 e 11) são menores do que a distância entre um quadrado e um círculo (experimento 13) ou entre um círculo e um retângulo (experimentos 15 e 16).

Ademais, os experimentos mostram que o método parece ser de fato invariante a rotações e mudanças de escala, como esperado. A distância entre os dois retângulos mantém-se pequena mesmo após a rotação (experimentos 1 e 2). O mesmo ocorre para pares de quadrados (experimentos 6 e 7) e pares de retângulos (experimentos 8 e 9). Diferentes escalas da mesma forma também retornam distâncias baixas (experimento 3, 5, 6 e 8).

Capítulo 6

Resultados

6.1 Matrizes de distância

Utilizando a última versão do programa, foram calculadas as distâncias entre todos os pares de imagens de duas bases de dados: MPEG7 CE Shape-1 Part B e *leaves database*[1]. Os resultados foram plotados nas matrizes da figura 6.1.

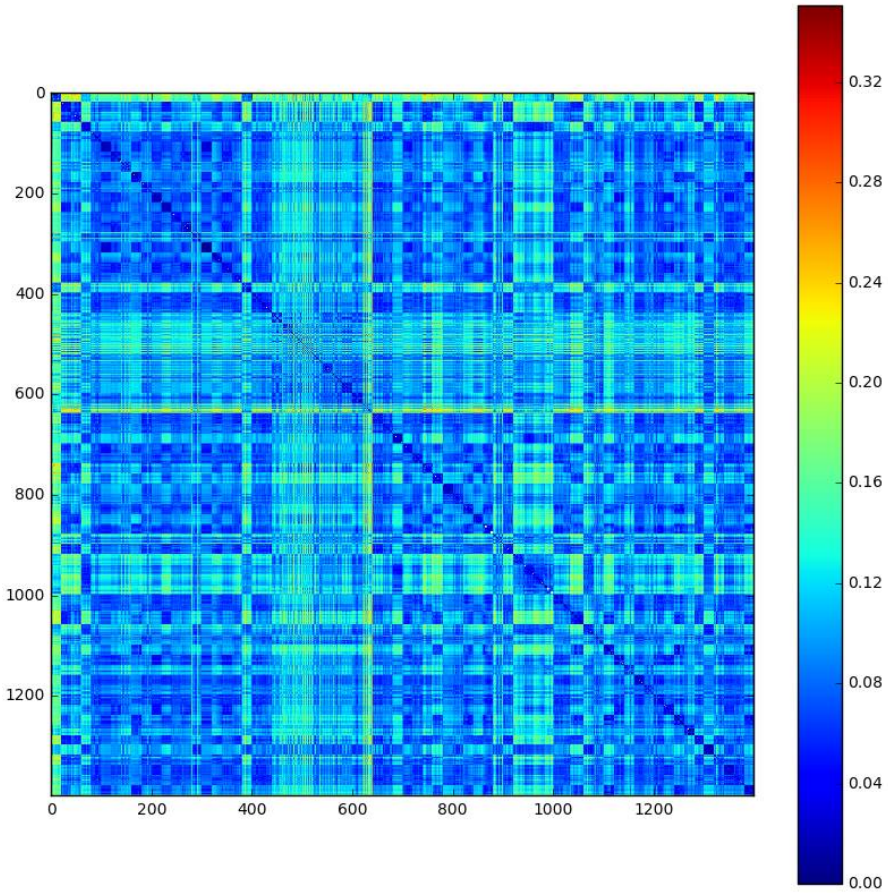
A base MPEG7 CE Shape-1 Part B tem 1400 imagens e a base das folhas tem 600, logo as matrizes são quadradas de dimensões 1400 e 600, respectivamente.

As diagonais das matrizes correspondem às distâncias entre uma forma e ela mesma e, consequentemente, devem ser zero ou muito próximas de zero, efeito que é observado em ambas as matrizes e representado pela cor azul escuro.

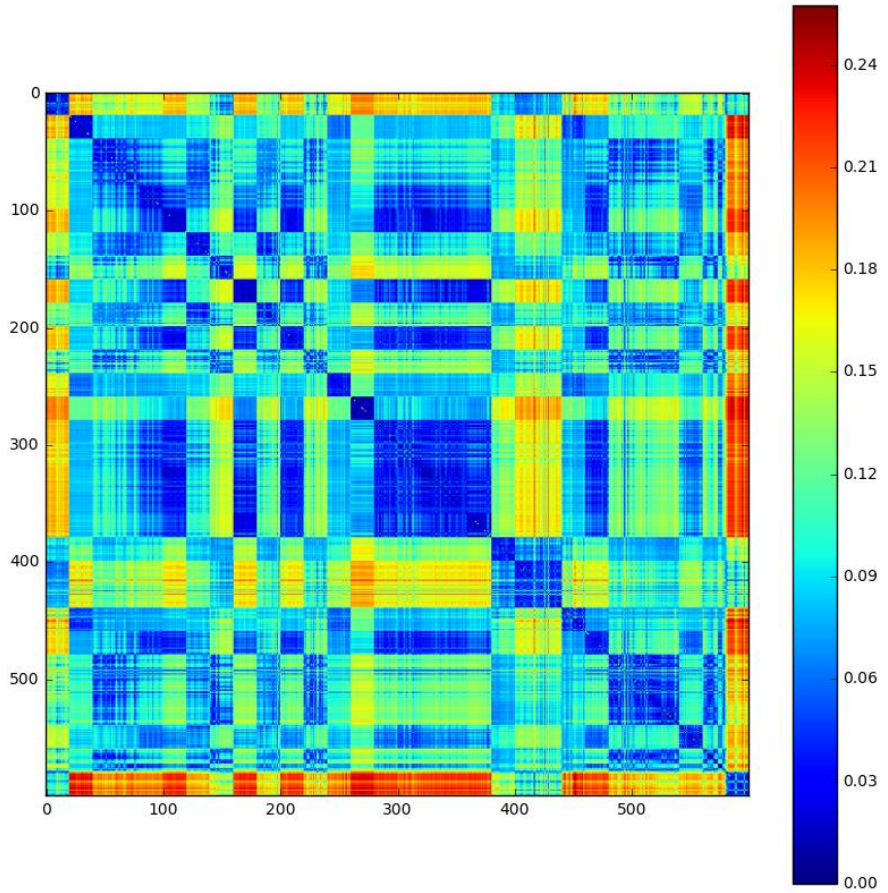
Nota-se também que a primeira matriz apresentou uma baixa variância, representada por diferentes tons de azul, quando comparada com a segunda matriz, que é mais colorida.

Para medir a precisão do método implementado foi usado o índice Bullseye, que é explicado na próxima seção.

[1] A base de folhas está disponível em: http://fractal.ifsc.usp.br/dataset/ShapeCN.php.



(a) *MPEG-7 CE Shape-1 Part B*



(b) *Leaves database*

Figura 6.1: *Distâncias entre pares de imagens para duas bases*

6.2 Índice Bullseye

O artigo tema deste trabalho utiliza o método Bullseye como índice de precisão de recuperação de imagens.

Para o cálculo desse índice, seleciona-se uma imagem s do banco de dados como chave de busca e calculam-se as distâncias entre a assinatura de sua forma e as outras assinaturas dos outros elementos da base.

As imagens da base são então ordenadas de forma crescente em relação à distância à chave de busca, isto é, em ordem decrescente de similaridade.

Em seguida, é escolhido um número $2C$ de imagens semelhantes igual ao dobro do tamanho C das classes desse banco de dados. Na base MPEG-7, por exemplo, cada classe tem tamanho 20. Por isso, para essa base tomam-se as 40 imagens mais similares à chave de busca e conta-se o número de elementos que pertencem à mesma classe da chave.

Por exemplo, se a busca está sendo feita na base MPEG-7 e o objeto chave é a imagem de um camelo, selecionam-se as 40 imagens mais similares à chave e conta-se quantas dessas 40 imagens estão na classe de camelos.

Esse valor é dividido pelo número de elementos da classe da chave e o resultado é o índice de precisão Bullseye:

$$p_{bullseye}(s) = \frac{(\#\text{elementos da mesma classe de } s) \cap (2C \text{ mais similares})}{C}.$$

Um exemplo do procedimento explicado pode ser visto na tabela 6.1, no qual a imagem *apple-1* (figura 6.2) foi utilizada como chave de busca e 19 maçãs foram encontradas dentro do conjunto das 40 imagens mais similares, ou seja, uma precisão Bullseye de 95%. Um segundo exemplo está na tabela 6.2, que usa a imagem *hammer-1* (figura 6.3) como chave e a busca retorna somente 4 elementos da classe, resultando em uma baixíssima precisão de 20%.

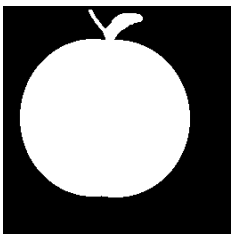


Figura 6.2: *apple-1*

Figura 6.3: *hammer-1*

Tabela 6.1: *Elementos mais similares à chave apple-1*

Posição	Nome	Imagem	Distância	Posição	Nome	Imagem	Distância
1	apple-1		0.0	21	cup-13		0.0874
2	apple-20		0.0265	22	apple-6		0.0887
3	apple-12		0.047	23	cup-6		0.0889
4	apple-7		0.0472	24	Misk-8		0.0895
5	apple-10		0.0491	25	apple-11		0.0905
6	apple-17		0.0505	26	cup-3		0.0911
7	apple-18		0.0597	27	apple-3		0.0917
8	apple-9		0.0701	28	cup-14		0.0924
9	apple-5		0.071	29	apple-19		0.0929
10	apple-8		0.0712	30	jar-16		0.0932
11	apple-16		0.0729	31	jar-17		0.0932
12	apple-13		0.0743	32	jar-18		0.0932
13	apple-15		0.0761	33	jar-19		0.0932
14	cup-1		0.0761	34	apple-14		0.0933
15	cup-5		0.0787	35	cup-9		0.0937
16	cup-15		0.0814	36	device3-6		0.094
17	cup-19		0.0851	37	device4-3		0.0944
18	Misk-3		0.0856	38	Misk-1		0.0944
19	cup-16		0.0863	39	device3-3		0.0948
20	apple-4		0.0869	40	cup-12		0.0955

Tabela 6.2: *Elementos mais similares à chave hammer-1*

Posição	Nome	Imagem	Distância	Posição	Nome	Imagem	Distância
1	hammer-1		0.0	21	Bone-17		0.0397
2	hammer-2		0.0064	22	fish-11		0.04
3	hammer-3		0.0064	23	watch-15		0.04
4	hammer-20		0.0064	24	watch-16		0.04
5	Bone-18		0.0272	25	fork-13		0.0408
6	guitar-16		0.0354	26	watch-20		0.0413
7	spoon-1		0.0354	27	spoon-4		0.0415
8	spoon-2		0.0354	28	spoon-5		0.0415
9	spoon-3		0.0354	29	spoon-6		0.0415
10	spoon-20		0.0354	30	spoon-7		0.0415
11	Bone-20		0.0361	31	watch-8		0.0419
12	spoon-8		0.0362	32	tree-19		0.042
13	spoon-9		0.0362	33	Bone-8		0.0423
14	spoon-10		0.0362	34	pencil-6		0.0423
15	spoon-11		0.0362	35	pencil-12		0.0425
16	fish-10		0.0367	36	pencil-7		0.0427
17	Bone-19		0.0368	37	pencil-8		0.0429
18	pencil-4		0.0387	38	guitar-20		0.043
19	watch-17		0.0388	39	pencil-14		0.043
20	watch-19		0.0389	40	guitar-8		0.0433

6.3 Análise dos resultados

Utilizando a equação do índice Bullseye explicado na seção anterior para todas as imagens de uma base e calculando-se a média obtém-se o índice de precisão Bullseye médio para aquela base. Esse cálculo foi feito para as bases MPEG7 CE Shape-1 Part B e *leaves database*. Os resultados estão na tabela 6.3.

Tabela 6.3: *Bullseyes médios para bases de imagens*

Base	Valor encontrado	Valor dos autores
MPEG7 CE Shape-1 Part B	61.15%	89.47%
Leaves database	62.63%	82.15%

É possível ver que, apesar do método CNSS ter sido implementado exatamente como descrito no artigo, a precisão da versão implementada foi muito menor do que a relatada pelos autores. A razão da diferença entre os valores obtidos é desconhecida, mas uma possibilidade seria algum detalhe de implementação, como um pré-processamento de dados utilizado, que não foi explicado no artigo.

Uma outra versão do CNSS foi implementada em Python por Anderson Meirelles Freitas, que também é orientando do professor Paulo Miranda, e a precisão obtida para o MPEG7 CE Shape-1 Part B foi de 61.02%.

Entrou-se em contato com os autores na tentativa de ter acesso ao código-fonte e poder compará-lo às versões do CNSS implementadas, mas eles não puderam disponibilizar o código no momento por estarem trabalhando na sua otimização.

Capítulo 7

Conclusões

Neste trabalho foi implementado o descritor de formas chamado *Complext Network Spectrum Signature*, que utiliza os conceitos de redes complexas e teoria espectral para medir a similaridade entre imagens.

Como visto no capítulo 3, o método é baseado no contorno da forma, que é modelado como um grafo completo. A partir deste grafo, várias matrizes de adjacência são derivadas considerando as diferentes faixas de distâncias entre os vértices. Através da equação do polinômio característico, os autovalores dessas matrizes são calculados e os mais relevantes são agrupados em um vetor de características chamado de assinatura da forma, que armazena as principais informações sobre a forma de um objeto. Através da equação de Hellinger, pode-se calcular a distância entre duas assinaturas, um valor que representa o grau de similaridade entre as duas formas analisadas.

Alguns experimentos foram feitos para verificar o funcionamento do método e os valores encontrados foram coerentes. Observou-se que as distâncias entre imagens da mesma classe eram menores do que aquelas entre imagens de classes diferentes. Além disso, o método demonstrou ser invariante a mudanças de escala e rotações.

Os resultados obtidos mostraram que a implementação gerou valores aquém das expectativas, pois a precisão Bullseye atingida foi bastante inferior àquela mencionada no artigo dos pesquisadores do Rio Grande do Sul. Entretanto, a realização deste trabalho permitiu disponibilizar uma implementação do método CNSS que pode ser aprimorada com pouco esforço, uma vez que o código está bem organizado e documentado.

A princípio, o objetivo do trabalho era alcançar a precisão relatada pelos autores e então testar mudanças na implementação com o intuito de melhorar a performance do método e testá-lo em outras bases. Em virtude dos resultados obtidos, o objetivo tornou-se descobrir a causa da disparidade entre as precisões encontradas.

Futuramente, se for possível ter acesso ao código-fonte do método, uma comparação entre as implementações poderá ser realizada e a causa da diferença dos resultados identificada.

Capítulo 8

Parte subjetiva

8.1 Desafios e frustrações

O principal desafio enfrentado foi conseguir conciliar a realização desse trabalho com dois empregos e ainda ter algum tempo para a vida pessoal. A parte final de produção da monografia também foi bastante trabalhosa, porém muito interessante pois permitiu apronfundar meus conhecimentos em LaTeX.

Na minha opinião, a maior frustração foi não alcançar a mesma precisão que os pesquisadores criadores do CNSS obtiveram. Além disso, como não foi possível ter acesso ao código-fonte no momento, outra frustração foi não conseguir identificar a razão da diferença entre os graus de precisão.

8.2 Disciplinas Relevantes

As disciplinas da graduação mais relevantes para a produção deste trabalho foram as seguintes:

- MAC0110 (Introdução à Computação), por ser o primeiro contato com a prática de programação.
- MAC0323 (Estruturas de Dados), uma vez que os conceitos de estruturas de dados são utilizados no desenvolvimento de todo tipo de software.
- MAC0328 (Algoritmos em Grafos), porque apresentou a teoria de grafos e diferentes modos de implementá-los, elementos fundamentais para este trabalho.

- FLC0474 (Língua Portuguesa), pois forneceu técnicas importantes para a produção de textos acadêmicos.

Capítulo 9

Agradecimentos

Agradeço imensamente toda a atenção e orientação dadas por meu supervisor. Também agredeço muito a ajuda de Anderson Freitas, orientando do professor Paulo, que me ajudou bastante com os testes do trabalho. Sem vocês, a realização deste projeto não seria possível.

Referências Bibliográficas

Amanatiadis *et al.***(2011)** A. Amanatiadis, V. G. Kaburlasos, A. Gasteratos e S. E. Papadakis. Evaluation of shape descriptors for shape-based image retrieval. *IET Image Processing*, 5:493–499. Citado na pág. 2

Barabási e Bonabeau(2003) Albert-Lászlo Barabási e Eric Bonabeau. Scale-free networks. *Scientific American*, páginas 50–59. Citado na pág. 9

Biggs(1993) Norman Biggs. *Algebraic Graph Theory*. Cambridge University Press. Citado na pág. 12

Boccaletti *et al.***(2006)** S. Boccaletti, V. Latora, Y. Moreno, M. Chavez e D.-U. Hwang. Complex networks: Structure and dynamics. *Physics Reports*, 424:175–308. Citado na pág. 8

Brouwer e Haemers(2011) Andries E. Brouwer e Willem H. Haemers. *Spectra of Graphs*. Springer. Citado na pág. 16

Chung(1997) Fan Chung. Eigenvalues and the laplacian of a graph. Em *Spectral Graph Theory*. AMS Publications. Citado na pág. 11

da F. Costa e Jr.(2009) L. da F. Costa e R. M. Cesar Jr. *Shape Analysis and Classification: Theory and Practice*. CRC Press, 2º edição. Taylor & Francis Group LLC, New York, USA, ISBN 978-0-8493-7929. Citado na pág. 1

Datta *et al.***(2008)** R. Datta, D. Joshi, J. Li e J. Z. Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Comput.*, Surv. 40(2, article 5). doi: 10.1145/1348246.1348248. Citado na pág. 1

de Jesus Gomes Leandro(2014) Jorge de Jesus Gomes Leandro. Análise de formas usando wavelets em grafos. doi: 10.11606/T.45.2014.tde-02072014-150049. Tese (Doutorado em Ciência da Computação) - Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo. Citado na pág. 1

de Kunder(2016) Maurice de Kunder. The size of the world wide web (the internet). http://www.worldwidewebsize.com/, 2016. Último acesso em 15/11/2016. Citado na pág. 8

de Oliveira *et al.***(2015)** A. B. de Oliveira, P. R. da Silva e D. A. C. Barone. A novel 2d shape signature method based on complex network spectrum. *Pattern Recognition Letters*, 63:43–49. Citado na pág. 2, 13, 17

de Souza(2016) Bruna Santos de Souza. Produtos e coespectralidade de grafos. Tese (Pós-graduação em Matemática Aplicada) - Instituto de Matemática, Universidade Federal do Rio Grande do Sul. Citado na pág. 12

Garrigan e Kellman(2011) Patrick Garrigan e Philip J. Kellman. The role of constant curvature in 2d contour shape representations. *Perception*, 4:1290–1308. doi: 10.1068/p6970. Citado na pág. 1

Golub e Loan(1996) Gene H. Golub e Charles F. Van Loan. *Matrix Computations*. Johns Hopkins University Press. Citado na pág. 12

Guennebaud *et al.***(2010)** Gael Guennebaud, Benoit Jacob *et al.* Eigen v3. http://eigen.tuxfamily.org, 2010. Citado na pág. 20

Haemers e Spence(2004) Willem H. Haemers e Edward Spence. Enumeration of cospectral graphs. *European Journal of Combinatorics*, 25(2):199–211. Citado na pág. 12

Joseph(2015) Esdras Joseph. The mahalanobis distance for functional data with applications in statistical problems. Tese (Doutorado em Estatística) - Departamento de Estatística, Universidade Carlos III, Madri. Citado na pág. 14

Kar *et al.***(2008)** Soummya Kar, Saeed Aldosari e José M. F. Moura. Topology for distributed inference on graphs. *IEEE Transaction on Signal Processing*, 56:2609–2613. doi: 10.1109/TSP.2008.923536. Citado na pág. 15

Lehoucq *et al.***(1996)** Rich Lehoucq, Kristi Maschhoff, Danny Sorensen e Chao Yang. Arpack software. http://www.caam.rice.edu/software/ARPACK/, 1996. Último acesso em 15/11/2016. Citado na pág. 22

Newman(2003) M. E. J. Newman. The structure and function of complex networks. *SIAM Review*, 45:167–256. Citado na pág. 8

Qiu(2015) Yixuan Qiu. Spectra, c++ library for large scale eigenvalue problems. http://yixuan.cos.name/spectra/, 2015. Último acesso em 15/11/2016. Citado na pág. 22

Shu e Wu(2010) X. Shu e X-J Wu. A novel contour descriptor for 2d shape matching and its application to image retrieval. *Image and Vision Computing*. doi: 10.1016/j.imavis.2010.11.001. Citado na pág. 1

Souza(2013) Gustavo Botelho Souza. Análise de formas planas em imagens digitais: Uma nova abordagem baseada na transformada de hough. Tese (Pós-graduação em Ciência da Computação) - Universidade Estadual Paulista, São José do Rio Preto. Citado na pág. 1

Spielman(2011) Daniel A. Spielman. Spectral graph theory. Em *Combinatorial Scientific Computing*. Chapman and Hall/CRC Press. Citado na pág. 11

Strogatz(2001) Steven H. Strogatz. Exploring complex networks. *Nature*, 410:268–276. doi: 10.1038/35065725. Citado na pág. 9

van de Bosch *et al.***(2016)** A. van de Bosch, T. Bogers e M. de Kunder. Estimating search engine index size variability: a 9-year longitudinal study. *Scientometrics*, 107:839–856. doi: 10.1007/s11192-016-1863-z. Citado na pág. 8

van der Hofstad(2009) Remco van der Hofstad. *Random graphs and complex networks*. Citado na pág. 8

Wang *et al.***(2012)** J. Wang, X. Bai, X. You, W. Liu e L. J. Latecki. Shape matching and classification using height functions. *Pattern Recognition Letters*, 33:134–143. Citado na pág. 2

Zhang e Lu(2003) D. Zhang e G. Lu. Evaluation of mpeg-7 shape descriptors against other shape descriptors. *Multimedia Systems*, 9:15–30. doi: 10.1007/s00530-002-0075-y. Citado na pág. 8

Zhang e Lu(2004) D. Zhang e G. Lu. Review of shape representation and description techniques. *Pattern Recognition Letters*, 37(1):1–19. Citado na pág. 7