

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE CIÊNCIA DA COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO



**SIMILARIDADE DE FORMAS VIA IDENTIFICAÇÃO E
CARACTERIZAÇÃO DE SALIÊNCIAS**

GLAUCO VITOR PEDROSA

Uberlândia - Minas Gerais

2011

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE CIÊNCIA DA COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO



GLAUCO VITOR PEDROSA

SIMILARIDADE DE FORMAS VIA IDENTIFICAÇÃO E CARACTERIZAÇÃO DE SALIÊNCIAS

Dissertação de Mestrado apresentada à Faculdade de Ciência da Computação da Universidade Federal de Uberlândia, Minas Gerais, como parte dos requisitos exigidos para obtenção do título de Mestre em Ciência da Computação.

Área de concentração: Banco de Dados.

Orientadora:

Prof^a. Dr^a. Celia Aparecida Zorzo Barcelos

Uberlândia, Minas Gerais

2011

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE CIÊNCIA DA COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Os abaixo assinados, por meio deste, certificam que leram e recomendam para a Faculdade de Ciência da Computação a aceitação da dissertação intitulada “**Similaridade de Formas via Identificação e Caracterização de Saliências**” por **Glauco Vitor Pedrosa** como parte dos requisitos exigidos para a obtenção do título de **Mestre em Ciência da Computação**.

Uberlândia, 12 de Agosto de 2011

Orientadora:

Prof^a. Dr^a. Celia Aparecida Zorzo Barcelos
Universidade Federal de Uberlândia

Banca Examinadora:

Prof. Dr. Bruno Augusto Nassif Travençolo
Universidade Federal de Uberlândia

Prof^a. Dr^a. Agma Juci Machado Traina
Universidade de São Paulo

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE CIÊNCIA DA COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Data: Agosto de 2011

Autor: **Glauco Vitor Pedrosa**
Título: **Similaridade de Formas via Identificação e Caracterização de Saliências**
Faculdade: **Faculdade de Ciência da Computação**
Grau: **Mestrado**

Fica garantido à Universidade Federal de Uberlândia o direito de circulação e impressão de cópias deste documento para propósitos exclusivamente acadêmicos, desde que o autor seja devidamente informado.

Autor

O AUTOR RESERVA PARA SI QUALQUER OUTRO DIREITO DE PUBLICAÇÃO DESTE DOCUMENTO, NÃO PODENDO O MESMO SER IMPRESSO OU REPRODUZIDO, SEJA NA TOTALIDADE OU EM PARTES, SEM A PERMISSÃO ESCRITA DO AUTOR.

Agradecimentos

Agradeço...

A Deus, por minha vida.

A minha família pela dedicação, confiança, apoio, carinho e amor incondicional em todos os momentos.

À minha orientadora professora Célia Aparecida Zorzo Barcelos, por me ensinar, incentivar, direcionar e por ser um exemplo a quem sempre seguirei.

A meus amigos da UFU que se mostraram companheiros e, diretamente ou indiretamente, contribuíram para a realização deste trabalho.

Ao professor Marcos Aurélio Batista pelos conselhos e atenção que me auxiliaram no desenvolvimento desse trabalho.

Ao professor Alexandre Xavier Falcão pela disponibilização dos códigos fontes dos descritores.

A CAPES e CNPq pelo apoio financeiro.

Resumo

O número de imagens disponíveis tem aumentado consideravelmente e, como consequência, há um interesse crescente na busca por imagens em grandes bases de dados. Para isso, as características intrínsecas da imagem devem ser representadas de tal maneira que se possa realmente diferenciar perceptualmente duas imagens. Em geral, as características mais analisadas em uma imagem são: cor, forma e/ou textura. Em reconhecimento de padrões e áreas afins, a forma é uma das características mais amplamente utilizada e explorada em sistemas de recuperação de imagens baseado em conteúdo.

O interesse deste trabalho é descrever formas usando seus pontos de saliência. Esses pontos são definidos como aqueles que possuem altos valores de curvatura ao longo do contorno da forma e eles são bastante úteis para a caracterização de formas, devido à sua capacidade de representação compacta, invariante a rotação e translação. A principal contribuição desse trabalho é a proposta de um novo descritor que analisa a similaridade de formas via identificação e caracterização dos pontos de saliências. Esse descritor utiliza três abordagens propostas neste trabalho: um detector de pontos de saliência robusto a ruídos, uma proposta de representação das saliência através de sua posição relativa angular e seu valor de curvatura, e uma medida para analisar a similaridade entre formas. Experimentos foram realizados a fim de ilustrar o desempenho do descritor proposto quando comparado com outros descritores de formas encontrados na literatura. A partir dos experimentos observa-se que o descritor proposto consegue recuperar imagens visualmente semelhantes, além de exigir pouco espaço para armazenamento das características extraídas.

Palavras chave: processamento de imagens digitais, recuperação de informação, recuperação de imagens baseada em conteúdo, descrição de formas, pontos de saliência.

Abstract

The number of images available has grown considerably and, as a consequence there is a growing interest in retrieving images in large databases. For this purpose, the intrinsic image features should be represented in such a way that two images can be perceptually differentiated. In general, the image features analyzed are: color, shape and/or texture. In pattern recognition and related areas, shape is one of the most widely image features exploited in content based image retrieval systems.

In this work, we are interested in describing shapes using salience points. Saliences are defined as the points of high curvature along shape contour. These points are very useful for shape description, because they have the ability to represent a shape in a compact manner, invariant to rotation and translation. The main contribution of this work is a new shape descriptor proposed to analyze the similarity between shapes represented by its salience points. This descriptor utilizes three techniques proposed in this work: a salience point detector robust to noise, a salience representation using angular relative position and curvature value, and a distance function to analyze the similarity between two shapes. Experiments were made in order to illustrate the performance of the proposed descriptor while comparing it with other shape-based descriptors in the literature. From the experiments we can note that the proposed descriptor can retrieve images visually similar, requiring low space for storing the extracted features.

Keywords: digital image processing, information retrieval, content based image retrieval, shape description, salience points.

Sumário

Lista de Figuras	xi
Lista de Tabelas	xiii
Lista de Abreviaturas e Siglas	xiv
1 Introdução	15
I Fundamentação Teórica e Trabalhos Correlatos	19
2 Recuperação de Imagens Baseada em Conteúdo	20
2.1 Introdução	20
2.2 Exemplos de CBIR	21
2.3 Arquitetura de um Sistema CBIR	21
2.3.1 Estruturas de Indexação	22
2.3.2 <i>Gap</i> Semântico e Realimentação de Relevância	23
2.4 Descritores de Imagens	24
2.5 Análise de Similaridade	25
2.5.1 Tipos de Consultas por Similaridade	26
2.5.2 Propriedades Métricas	26
2.5.3 Funções de Distância	27
2.6 Avaliação da Performance de um CBIR	29
2.6.1 Curva Precisão-Revocação (<i>Precision-Recall</i>)	29
2.6.2 Precisão- <i>R</i>	31
2.7 Considerações Finais	31
3 Caracterização de Formas	32
3.1 Introdução	32
3.2 Classificação dos Descritores	32
3.3 Características Geométricas Simples	33
3.3.1 Área	34

3.3.2	Centróide	34
3.3.3	Excentricidade	34
3.3.4	Fecho Convexo	35
3.3.5	Relação de Aspecto	35
3.3.6	Solidez	36
3.4	Descritores de Formas	36
3.4.1	Curvature Scale Space (CSS)	36
3.4.2	Tensor Scale Descriptor (TSD)	37
3.4.3	Descritores de Fourier	38
3.4.4	Invariantes de Momento	40
3.4.5	Beam Angle Statistics (BAS)	41
3.4.6	Multiscale Fractal Dimension (MS Fractal)	42
3.4.7	Contour Saliences (CS)	42
3.5	Considerações Finais	45
II	Método Proposto	47
4	Deteção de Pontos de Saliência em Contornos	48
4.1	Introdução	48
4.2	Cálculo da Curvatura	50
4.2.1	Discretização usando K-Curvatura	50
4.3	Difusão Anisotrópica Não-Linear	52
4.4	Localização das Saliências	54
4.5	Considerações Finais	55
5	Representação dos Pontos de Saliência e Similaridade	56
5.1	Introdução	56
5.2	Caracterização dos Pontos de Saliência	57
5.2.1	Posição Relativa Angular	58
5.2.2	Relevância de uma Saliência	59
5.3	Análise da Similaridade	60
5.3.1	Similaridade entre Saliências	62
5.4	Considerações Finais	64
6	Resultados Experimentais	65
6.1	Introdução	65
6.2	Deteção de Saliências	65
6.2.1	Base de Teste	66
6.2.2	Detectores Avaliados	66

6.2.3	Medida de Avaliação	67
6.2.4	Resultados Obtidos	68
6.3	Recuperação de Formas	69
6.3.1	Base de Imagens	69
6.3.2	Descritores Avaliados	70
6.3.3	Medida de Avaliação	71
6.3.4	Resultados Obtidos	71
7	Conclusão	76
7.1	Trabalhos Futuros	77
7.2	Publicações Realizadas	77
7.2.1	Artigos completos publicados em periódicos	77
7.2.2	Artigos publicados em anais de congressos	77
7.2.3	Resumos publicados em anais de congressos	77
	Referências Bibliográficas	78

Lista de Figuras

1.1	Uma imagem com um objeto (a) e sua forma (silhueta) em 2D (b).	16
1.2	Os pontos de saliências de algumas formas.	17
1.3	Metodologia usada para analisar a similaridade de formas baseado nos pontos de saliência.	18
2.1	<i>Framework</i> de um sistema de recuperação de imagem por conteúdo.	22
2.2	Arquitetura de um sistema de recuperação de imagem por conteúdo.	22
2.3	Conjuntos de documentos na Recuperação de Informação.	30
3.1	Classificação das técnicas de descrição e representação de formas.	33
3.2	Centróide de uma forma.	34
3.3	Fecho convexo de uma forma.	35
3.4	Mínimo retângulo delimitador de uma forma.	35
3.5	Obtenção do descritor CSS: (a) larguras $\sigma = 1,4,7,10,12,14$ do filtro gaussiano; (b) árvore de intervalo	37
3.6	Fatores do <i>Tensor Scale</i> .	38
3.7	Exemplo de curva de orientação obtidas com o descritor TSD: (a) uma forma; (b) curva de orientação	39
3.8	Beams que conectam o ponto p_i .	41
3.9	(a) Imagem original (b) contorno rotulado (c) imagem rotulada (d) imagem diferença (e) esqueleto obtido a partir do <i>threshold</i> (f) pontos de saliências.	43
3.10	Esqueleto de uma forma usando diferentes <i>thresholds</i> .	44
4.1	Curvatura da silhueta de uma estrela sem ruído e com ruído.	49
4.2	Os 5-vizinhos p_{i+5} e p_{i-5} do ponto p_i .	51
4.3	Gráficos mostrando a representação do contorno da figura (a) para diferentes valores de K no cálculo da curvatura. A figura (a) tem 937 pontos de contorno.	52
4.4	Uma curva e seus pontos de máximo e mínimo locais.	54
5.1	Saliência sobre um segmento de contorno: (a) mais fechado (b) mais suave	56

5.2	Duas formas com o mesmo número de saliências, porém visualmente diferentes.	57
5.3	Posição angular da saliência S em relação ao centróide C e um outro ponto C' qualquer escolhido arbitrariamente.	58
5.4	Dois segmentos de contorno semelhantes, sem e com ruído. Os pontos representam a vizinhança considerada no cálculo da curvatura.	59
5.5	Contorno de uma forma sendo suavizado.	60
5.6	Representação gráfica dos valores das saliências das formas da figuras (a) e (b) usando: (c) a representação proposta neste trabalho e (d) a representação proposta por [Torres e Falcão 2007].	60
6.1	Algumas imagens que compõe a base usada nos experimentos.	66
6.2	Pontos de saliência de uma forma detectados pelo método proposto usando diferentes valores de iteração para a função de suavização.	67
6.3	Pontos de saliência detectados: (a) pelo método proposto por [Torres et al. 2004], (b) pelo método proposto por [Chetverikov 2003] e (c) pela técnica proposta neste trabalho.	69
6.4	Banco de dados Kimia-99.	70
6.5	Dois exemplos de cada classe do banco Kimia-216.	70
6.6	Performance dos descritores usando o banco de dados Kimia-99.	72
6.7	Performance dos descritores usando o banco de dados Kimia-216.	72
6.8	Performance dos descritores usando o banco de dados MPEG-7.	73
6.9	Exemplo de uma consulta de uma forma do banco de dados MPEG-7. . .	74
6.10	Imagens do banco MPEG-7 de classes diferentes mas visualmente bastante similares.	74
6.11	Imagens do banco MPEG-7 da mesma classe mas visualmente bastante diferentes.	74
6.12	(a) Algumas consultas realizadas pelo descritor proposto; (b) gráfico mostrando a curva de PR dessas consultas.	75

Lista de Tabelas

6.1	Resultados comparativos entre os detectores de saliência avaliados.	68
6.2	Dimensão dos vetor de característica e espaço de armazenamento (total) utilizado pelos descritores considerando o banco MPEG-7. <i>*n é o número de saliências da forma.</i>	75

Lista de Abreviaturas e Siglas

CBIR	Content-Based Image Retrieval
QBIC	Query by Image Content
IBM	International Business Machines
BAS	Beam Angle Statistics
MS Fractal	Multiscale Fractal Dimension
MI	Moment Invariants
TSDIZ	Tensor Scale Descriptor with Influence Zones
CS	Contour Saliences
MPEG	Moving Picture Experts Group
PR	Precisão vs. Revocação
FD	Fourier Descriptor
EDP	Equação Diferencial Parcial
DFT	Discrete Fourier-Transform
SSD	Shape Saliences Descriptor
SSD+GF	Shape Saliences Descriptor plus Global Features
MAM	Metric Access Methods

Capítulo 1

Introdução

Diversas áreas do conhecimento humano fazem uso de imagens digitais. Essas imagens são usadas para diversos fins: na comunicação as imagens dão dinamismo à linguagem; na medicina são de fundamental importância para a apresentação de diversos tipos de exames médicos; na segurança auxiliam o processo de monitoramento e garantem o uso de senhas confiáveis, etc. Como consequência, essas imagens precisam ser adquiridas, armazenadas, indexadas e consultadas eficientemente.

Com o avanço rápido das tecnologias que auxiliam na aquisição e no armazenamento de imagens, está havendo um aumento de imagens geradas e disponibilizadas, por isso uma das preocupações nos dias atuais está na necessidade de criação de sistemas de busca por informações.

Em um primeiro momento, a busca por imagens era feita por indexação de palavras-chave associadas a cada imagem [Veltkamp e Tanase 2000]. Porém, com o crescimento da disponibilidade de imagens, essa tarefa ficou inviável de ser realizada, pois o processo de anotação era dependente do auxílio humano. Além disso, a tarefa de indexação sofria o efeito da subjetividade da percepção humana, pois diferentes usuários podem utilizar diferentes termos para descrever uma determinada imagem, ou mesmo o próprio usuário, em momentos distintos, pode anotar diferentemente uma mesma imagem. Devido a essas dificuldades, deu-se início à pesquisas em recuperação de imagens baseado em conteúdo, cuja principal vantagem é a automatização do processo de descrição e recuperação de imagens. Assim, surgiram os CBIRs (CBIR - *Content Based Image Retrieval*) que são sistemas projetados para fazer a recuperação de imagens baseado em conteúdo [Smeulders et al. 2000]. Basicamente, um sistema CBIR possui a função de, dada uma imagem de consulta, retornar as imagens de uma base de dados mais similares a ela.

Para fazer a recuperação de uma imagem baseada em seu conteúdo, primeiro deve-se definir quais características intrínsecas serão descritas nessa imagem. Em geral, essas características são: cor, forma ou textura. A forma é a silhueta do objeto contido na imagem, por exemplo, a figura 1.1(b) ilustra a forma do objeto contido na figura 1.1(a). O uso de formas para caracterizar imagens têm utilidade em diversas aplicações, por

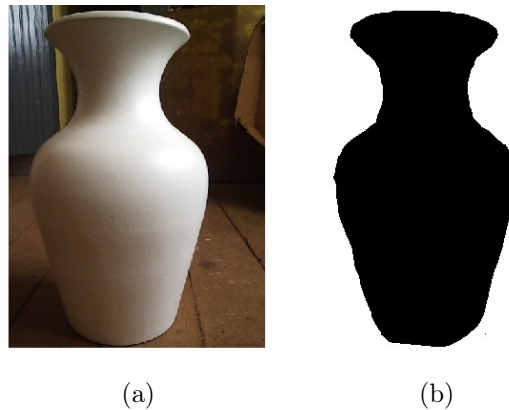


Figura 1.1: Uma imagem com um objeto (a) e sua forma (silhueta) em 2D (b).

exemplo, no reconhecimento de ações humanas [Wang et al. 2007], detecção de impressão digital [Yuan et al. 2009], reconhecimento de caracteres [Dubey e Sinthupinyo 2010], etc.

Um sistema de recuperação de imagens necessita de um descritor, cujo objetivo é agrupar imagens com características similares em uma dada região do espaço de medida, e grupos de imagens com características distintas em regiões diferentes deste espaço. Diferentes descritores podem gerar diferentes resultados, levando à um agrupamento mais próximo das expectativas do usuário, sendo que o descritor ideal é aquele capaz de capturar informações perceptivamente similares nas imagens.

Segundo [Torres e Falcão 2006], um descritor é caracterizado por: (i) um algoritmo de extração, que tem o objetivo de criar um vetor com as características relevantes da imagem; (ii) uma medida de similaridade, para efetuar a comparação entre dois vetores de características.

Um descritor de formas deve ser invariante a transformações geométricas, tais como rotação, translação e escala. Outras propriedades desejáveis são: insensibilidade a ruído, geração de vetores de características compactos (que requerem pouco espaço de armazenamento) e função de extração computacionalmente eficiente [Costa e Cesar 2001].

A análise de formas pode ser feita usando diferentes abordagens, e na literatura encontram-se diversos descritores para tal fim [Zhang e Lu 2004]. Alguns descritores clássicos, como os Descritores de Fourier [Gonzalez e Woods 2001], analisam apenas o contorno da forma. Essa maneira de analisar formas é bastante popular na literatura, porém ela pode sofrer efeitos negativos quando o contorno apresenta muito ruído. Outros descritores, como o “Invariantes de Momento” [Liao e Pawlak 1996], analisam todo o objeto, ou seja, o contorno juntamente com a região interna à forma.

O objetivo deste trabalho é explorar o uso de pontos de saliência para descrever e analisar a similaridade de formas. Saliências são caracterizadas por serem os pontos de maior curvatura ao longo do contorno da forma. Elas têm a capacidade de capturar a essência de um objeto de maneira compacta, uma vez que, quaisquer pontos sobressalentes ou pontos reentrantes são saliências, e são essas as características que diferenciam dois

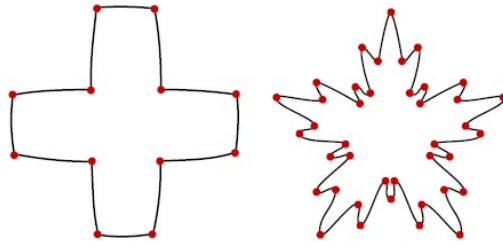


Figura 1.2: Os pontos de saliências de algumas formas.

objetos. A figura 1.2 mostra os pontos de saliência de algumas silhuetas de objetos.

Apesar dos pontos salientes serem invariantes à rotação e translação, outros aspectos contribuem para a qualidade dessa abordagem na análise de formas, tais como: a sensibilidade do algoritmo de detecção de saliências em formas com contorno ruidoso, o método de representar os pontos de saliências e a medida de similaridade usada para calcular a distância entre duas formas descritas por seus pontos de saliências.

Neste trabalho é proposto um novo descritor de formas via identificação e caracterização de saliências, referenciado como “Descritor de Saliências de Formas” (*Shape Saliences Descriptor* - SSD), que é inspirado no descritor “Saliências do Contorno” (*Contour Saliences* - CS), proposto por [Torres e Falcão 2007]. Na tentativa de caracterizar e reconhecer formas com uma abordagem menos sensível a ruídos presentes no contorno, o SSD utiliza outros métodos, diferentes do descritor CS, para caracterizar e analisar a similaridade de formas.

A metodologia adotada pelo descritor proposto está esquematizada na figura 1.3 e consiste em: detectar os pontos de saliência da forma; representar os pontos extraídos, gerando um vetor de características e usar uma medida de similaridade, que permita comparar formas com número de saliências diferentes. As principais contribuições estão relacionadas com essas três etapas. Neste trabalho são propostos:

1. uma técnica eficiente de detectar os pontos de saliência, que consiste em usar como referência o valor de curvatura dos pontos do contorno da forma e uma função de suavização anisotrópica, via equações diferenciais parciais, para eliminar pontos ruidosos;
2. um método de representar os pontos de saliências por meio da posição relativa angular e o valor de curvatura do ponto em relação ao contorno;
3. uma medida de similaridade, que permite comparar vetores de tamanho diferentes e com uma abordagem invariante à rotação e espelhamento.

A combinação dessas três propostas constituem o descritor SSD e elas foram aplicadas na descrição e reconhecimento de formas. Testes experimentais foram realizados utilizando diferentes base de dados, comparando a abordagem de descrição de formas via análise de saliências com outros diferentes descritores encontrados na literatura.

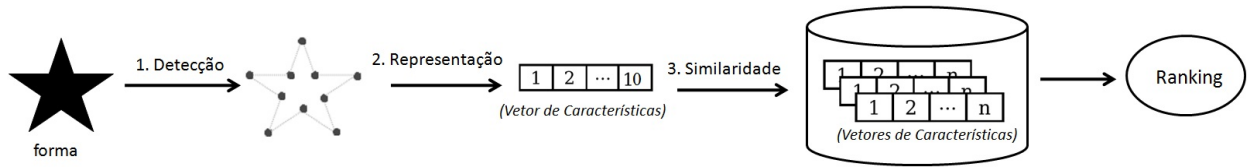


Figura 1.3: Metodologia usada para analisar a similaridade de formas baseado nos pontos de saliência.

Essa dissertação está organizada em duas partes, sendo uma delas referente à fundamentação teórica e trabalhos correlatos e uma parte detalhando o método proposto para descrição de formas. No capítulo 2 são apresentados os conceitos básicos sobre recuperação de imagens; no capítulo 3 é feita uma revisão bibliográfica de alguns descritores de formas encontrados na literatura; no capítulo 4 é apresentado a técnica de detecção de pontos de saliência em contornos; no capítulo 5 são propostos os métodos de representação dos pontos salientes e a função de similaridade usada para medir a distância entre duas formas; no capítulo 6 são descritos os resultados dos experimentos realizados para demonstrar a eficácia das técnicas propostas; e, finalmente, no capítulo 7 serão apresentadas as conclusões, as perspectivas para trabalhos futuros e as publicações realizadas durante o desenvolvimento do trabalho.

Parte I

Fundamentação Teórica e Trabalhos Correlatos

Capítulo 2

Recuperação de Imagens Baseada em Conteúdo

2.1 Introdução

A *World Wide Web*, juntamente com as inúmeras áreas do conhecimento humano, vem se apoiando cada vez mais no uso de imagens. Essas imagens são usadas para diversos fins: na comunicação as imagens dão dinamismo à linguagem, na medicina são de fundamental importância para os exames médicos [Müller et al. 2004], na segurança garantem a proteção de propriedades intelectuais [Lopes et al. 2006], além de outras aplicações, tais como sensoriamento remoto [Santos et al. 2009], controle de pragas [Weeks et al. 1999], etc. Por isso, a quantidade de informação visual sendo gerada é gigantesca, o que torna necessário o uso de maneiras eficientes de armazenar, manter e recuperar imagens.

Uma das alternativas é que a recuperação pode ser feita a partir de informações textuais associadas à cada imagem, descrevendo sua semântica. Contudo, diante do tamanho das coleções e da dificuldade de descrever os objetos da base de dados, é necessário um processo totalmente automático para inferir semântica ao documento. Para isto, utilizam-se técnicas de análise do conteúdo das imagens, o que originou um novo ramo da recuperação de informação.

A recuperação de imagens baseada em conteúdo é uma alternativa à busca baseada em descrições textuais. Os sistemas projetados para fazer a recuperação de imagens baseado em conteúdo são denominados CBIR (CBIR - *Content Based Image Retrieval*), e são eles o principal interesse dentro deste trabalho.

2.2 Exemplos de CBIR

Os primeiros CBIR começaram a ser desenvolvidos na década de 80, apresentado por [Chang e Fu 1980]. Porém, foi na década de 90, que começaram a surgir CBIR mais relevantes como o QBIC (*Query By Image Content*), da IBM e desenvolvido por [Flickner et al. 1995], que é um dos mais avançados e conhecidos sistemas de recuperação baseado em conteúdo de sua época. O QBIC utiliza os principais aspectos da imagem: cor, forma e textura para efetuar a recuperação. Histograma de cores é utilizado na recuperação baseada em cor e descritores simples e momentos são empregados na recuperação baseada em forma e textura. Além disso, uma estrutura de indexação é implementada dando maior rapidez à busca.

Outro CBIR que merece destaque é o sistema desenvolvido por [Jacobs et al. 1995]. Nele, informação espacial é extraída da imagem e sua representação é feita através de wavelets. A informação espacial neste contexto é a disposição física das regiões.

O VisualSeek é um sistema desenvolvido por [Smith e Chang 1996] que permite busca por cor, região e layout espacial. Assim como o QBIC, o VisualSeek implementa uma estrutura de indexação.

Uma revisão mais detalhada desses e de outros sistemas CBIR pode ser encontrada em [Veltkamp e Tanase 2000].

2.3 Arquitetura de um Sistema CBIR

A busca de imagens através de sistemas CBIR são usualmente executadas utilizando uma imagem consulta com a finalidade de recuperar as imagens mais semelhantes à imagem referência. Esse método é chamado de Consulta-por-Exemplo (*Query by Example*) ou QBE. A recuperação começa quando o sistema apresenta uma seleção inicial de imagens de referência através de uma interface. O sistema pode apresentar alguma imagem representativa ou o usuário pode escolher alguma aleatoriamente. Essa imagem selecionada é usada para recuperar as imagens mais semelhantes a ela dentro de um banco de dados.

O funcionamento de um sistema CBIR simples pode ser ilustrado pelo *framework* presente na figura 2.1: dada uma imagem de consulta, aplica-se um algoritmo de extração de características sobre essa imagem e sobre todas as imagens da base de dados; então é utilizado uma “função de distância” para medir a similaridade entre as características extraídas da imagem consulta e as extraídas das imagens da base de dados; por fim o sistema retorna as imagens mais próximas (mais similares) à imagem consulta e envia para um módulo de interface onde serão visualizadas pelo usuário.

Geralmente, o processo de extração dos vetores de características é realizado uma única vez para cada imagem da base de dados, sendo utilizado de maneira *offline*. Os vetores armazenados são usados posteriormente no processamento de consultas.

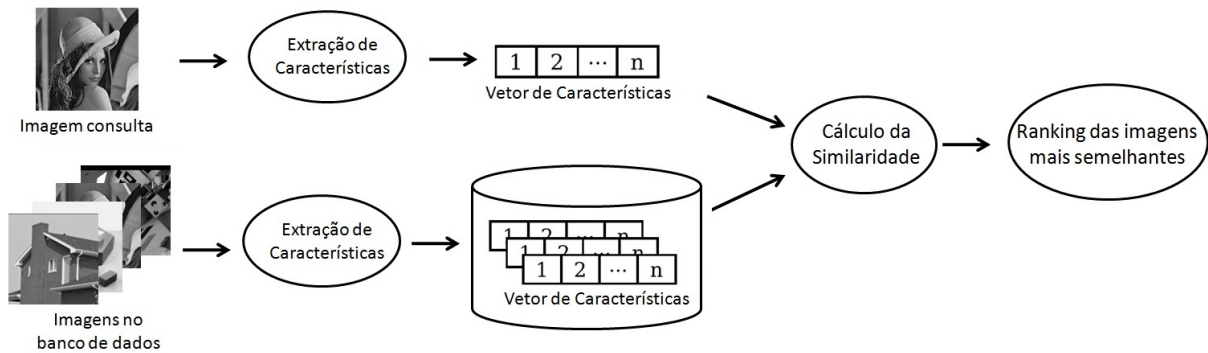


Figura 2.1: *Framework* de um sistema de recuperação de imagem por conteúdo.

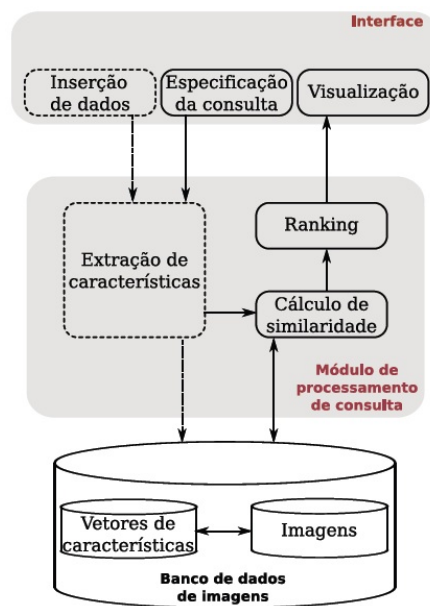


Figura 2.2: Arquitetura de um sistema de recuperação de imagem por conteúdo.

O *framework* da figura 2.1 também pode ser entendido como dois subsistemas, como esquematizado pela figura 2.2: um módulo de interface responsável pela inserção e visualização dos dados e um módulo de processamento de consultas que tem o objetivo de extrair características, analisar a similaridade entre as imagens, ordenar as imagens do banco de dados em ordem decrescente de similaridade em relação à imagem de consulta e enviar para o módulo de interface para exibir ao usuário.

2.3.1 Estruturas de Indexação

Um aspecto importante na implementação de um sistema de recuperação de imagens por conteúdo está relacionado à utilização de estruturas de indexação apropriadas que possam agilizar a realização de consultas por similaridade, ou seja, minimizar o número de cálculos de distância necessários para executar uma consulta. As estruturas de indexação são ferramentas fundamentais que permitem os sistemas de recuperação de imagens armazenar e recuperar eficientemente um grande volume de informação.

Os Métodos de Acesso Métricos (*Metric Access Methods* - MAM) são as estruturas mais adequadas em sistemas de consulta por similaridade em espaços métricos genéricos, ou seja, espaços que englobam tanto dados espaciais com dimensão definida quanto dados adimensionais. Os MAM são estrutura de indexação baseadas em distância que utilizam exclusivamente funções de distância para organizar os objetos na base de dados. A idéia básica consiste na escolha de um objeto arbitrário central e na aplicação de uma função de distância para dividir os demais objetos em vários subconjuntos. Dessa maneira, a estrutura de indexação é construída executando-se esse mesmo procedimento, recursivamente, para cada subconjunto não vazio.

Na maioria dos sistemas de recuperação, as imagens no banco de dados são indexadas, de acordo com seus vetores de características, pelo uso de estruturas como *M-tree* [Ciaccia et al. 1997] ou *Slim-tree* [Traina et al. 2002].

2.3.2 *Gap* Semântico e Realimentação de Relevância

Ao tentar recuperar uma imagem em um sistema de recuperação de imagens, o usuário depara-se com o problema da quantidade excessiva de dados irrelevantes, seja por dificuldade da escolha de uma imagem adequada para consulta ou pela falta de descritores apropriados que consiga incorporar a subjetividade da percepção visual. Tal problema está relacionado com o *gap* semântico, que é a diferença entre características de baixo nível extraídas da imagem, e o alto nível da informação que o usuário necessita. O objetivo da redução do *gap* semântico é estabelecer uma ligação entre a expectativa do usuário e as características extraídas da imagem.

A técnica de realimentação de relevância (*relevance feedback*, [Liua et al. 2007]) tem sido utilizada com o intuito de incorporar a subjetividade da percepção visual de usuários à recuperação de imagens por conteúdo. Essa técnica possibilita ao usuário informar quais as imagens ele considera relevantes dentro de um conjunto de imagens retornado pelo sistema. A idéia da técnica de realimentação de relevância é “aprender” a vontade do usuário durante um determinado número de iterações. Dessa maneira, o sistema retorna imagens cada vez mais similares à vontade do usuário, aprendendo o conceito estabelecido por ele.

Um aspecto importante para a definição de um mecanismo de realimentação de relevância é o algoritmo de aprendizado. Em alguns trabalhos, o aprendizado consiste em estimar o vetor de característica que melhor representa o padrão de consulta. Outros, atribuem pesos para cada posição do vetor de características e para cada descritor utilizado. Assim, o aprendizado consiste em estimar esses pesos que melhor representem a percepção visual do usuário.

Realimentação de relevância endereça duas questões referentes ao processo de recuperação de imagens por conteúdo [Silva et al. 2011]. A primeira delas reside na lacuna

semântica (*semantic gap*) entre as propriedades visuais de alto nível, através dos quais o usuário tem a percepção da informação visual, e a descrição de baixo nível utilizada para a representação das imagens. A outra diz respeito ao caráter subjetivo da percepção da imagem pelo usuário: diferentes pessoas, ou a mesma pessoa em diferentes momentos, podem ter percepções visuais distintas de uma mesma imagem. Com realimentação de relevância essas duas questões são contornadas de maneira transparente para o usuário e são geralmente utilizadas na implementação de um CBIR.

2.4 Descritores de Imagens

Para que um sistema de recuperação de imagem por conteúdo seja viável, é necessário que as imagens sejam descritas por suas propriedades (forma, textura e/ou cor), e que sejam representadas por um vetor de características. Descritores de imagens são necessários para extrair e comparar os vetores de características entre as imagens, viabilizando a indexação e busca de imagens.

Segundo [Torres e Falcão 2006], um *descriptor simples* D (ou simplesmente, descriptor) é definido por um par (ϵ_D, δ_D) , onde:

- ϵ_D : é uma função que extrai um vetor de característica $\vec{v}$ de uma imagem;
- δ_D : é uma função de similaridade $\mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$ que computa a similaridade entre duas imagens a partir da distância entre seus vetores de características correspondentes.

Já um *descriptor composto* $\hat{D}$ é um par $(\mathcal{D}, \delta_D)$, onde:

- $\mathcal{D} = \{D_1, D_2, \dots, D_k\}$ é um conjunto de k descritores simples pré-definidos;
- δ_D é uma função de similaridade que combina os valores de similaridade obtidos de cada descriptor $D_i \in \mathcal{D}$, $i = 1, 2, \dots, k$.

Um vetor de características $\vec{v}$ de uma imagem é um ponto no espaço $\mathbb{R}^n$: $\vec{v} = (\vec{v}_1, \vec{v}_2, \dots, \vec{v}_n)$, onde n é a dimensão do vetor. Basicamente um descriptor tem o objetivo de caracterizar informações presentes nas imagens, de tal maneira que seja compatíveis com a intuição humana. Esse vetor sintetiza as propriedades da imagem de acordo com os elementos escolhidos e pode ter qualquer dimensão. Diferentes tipos de vetores de características podem necessitar de funções de similaridade distintas.

Geralmente, as características de cor são facilmente obtidas diretamente a partir da intensidade dos pixels, por exemplo histograma de cores sobre a imagem inteira ou sobre uma região segmentada da imagem. Vários descritores de cor têm sido propostos [Huang et al. 1997, Manjunath et al. 2001, Pass et al. 1996, Stricker e Orengo 1995, Swain e Ballard 1991]. Em geral, eles podem ser agrupados em duas classes levando-se em conta

se codificam informação relacionada à distribuição espacial de cor ou não. O descritor de cor mais comum é o histograma, que descreve o conteúdo global de uma imagem pelo percentual de pixels de cada cor [Swain e Ballard 1991] e pode-se utilizar a distância Euclidiana, por exemplo, para a comparação entre diferentes histogramas.

Por outro lado, uma definição precisa de textura ainda não existe. A noção de textura geralmente se refere à presença de um padrão espacial, que tem algumas propriedades de homogeneidade. Em particular, a homogeneidade não pode resultar da presença de apenas uma cor única para as regiões, mas exige interação de várias cores. Matriz de co-ocorrência [Haralick et al. 1973] é uma das técnicas mais tradicionais para codificar informação de textura. Ela descreve o relacionamento espacial entre níveis de cinza em uma imagem. Exemplos de descritores de textura baseados na frequência incluem, por exemplo, os coeficientes de Garbor [Manjunath e Ma 1996]. Mais recentemente, merece destaque a proposta de descritores baseados na Decomposição Piramidal Steerable, apresentado por [Zegarra et al. 2008].

As formas dos objetos em uma imagem podem ser obtidas por detectores de bordas, a partir do qual uma representação geométrica pode ser derivada. No capítulo 3, serão apresentadas e detalhadas algumas das técnicas encontradas na literatura para descrever e representar características de formas em imagens.

2.5 Análise de Similaridade

Para conseguir uma recuperação de imagens eficaz, que satisfaça às expectativas do usuário, as características extraídas da imagem devem ser combinadas com uma função de distância. Essa função é definida de forma a calcular o grau de similaridade entre dois objetos, retornando um valor que quantifica o quão similar eles são. Esse valor é maior que zero e é inversamente proporcional ao nível de similaridade das imagens, ou seja, para objetos menos similares esse valor é grande e para objetos muito similares ele é próximo de zero.

A análise de similaridade de um descritor é normalmente a responsável pela maior parte do tempo gasto pelo sistema de busca de imagens por conteúdo ao processar uma consulta. Durante um processo de busca, a imagem de consulta deve ser comparada a uma quantidade muito grande de imagens, pois, apenas após todas as comparações serem realizadas é que se pode ordenar os resultados. Desta maneira, pode-se notar a importância de que a função de distância seja bastante rápida. Diversas são as métricas encontradas na literatura, e cada tipo de características extraídas em uma imagem requer uma função de distância específica.

2.5.1 Tipos de Consultas por Similaridade

Os sistemas de recuperação tradicionais realizam suas consultas por igualdade e ordem. Porém, as aplicações que lidam com dados complexos requerem a realização de consultas que realizem busca por objetos da base que sejam similares a um objeto de consulta de acordo com uma certa medida de similaridade e baseada em uma função de distância.

Existem basicamente dois tipos de consultas por similaridade: as consultas por abrangência (*Range query* - R_q), cujo foco está centrado em todos os objetos da base de dados que sejam similares ao objeto de consulta até no máximo um certo limiar; e as consultas aos k -Vizinhos mais Próximos (*k-Nearest neighbor query* - $kNNq$), que está relacionado à recuperação dos k objetos mais similares ao objeto de consulta.

Seja um objeto de consulta $o_q \in \mathbb{O}$, um conjunto de objetos $\mathcal{O} \subseteq \mathbb{O}$ e uma função de distância $d()$, esses dois tipos de consultas podem ser definidos formalmente como:

- Consulta por Abrangência: dado uma distância máxima de busca ξ , a consulta por abrangência $R_q(o_q, \xi)$ recupera todos os objetos de $\mathcal{O}$ que estejam a uma distância de o_q de no máximo ξ . O subconjunto resposta resultante $\mathcal{O}' \subseteq \mathcal{O}$ é $\{o_i \in \mathcal{O} \mid d(o_q, o_i) \leq \xi\}$.
- Consulta aos k -Vizinhos mais Próximos: dado um número inteiro k , a consulta $kNNq(o_q, k)$ recupera os k objetos mais próximos de o_q em $\mathcal{O}$. O subconjunto resposta resultante $\mathcal{O}' \subseteq \mathcal{O}$ é $\{o_i \in \mathcal{O} \mid |\mathcal{O}'| = k \text{ e } \forall o_r \in \mathcal{O}', \forall o_j \in [\mathcal{O} - \mathcal{O}'], d(o_q, o_r) \leq d(o_q, o_j)\}$.

2.5.2 Propriedades Métricas

As funções de distância, empregadas pelas técnicas de recuperação por conteúdo, são definidas dentro de um espaço métrico e podem ser formalmente descritas da seguinte maneira: dado um conjunto $\mathcal{O}$, uma função de distância métrica em $\mathcal{O}$ é uma função $d : \mathcal{O} \times \mathcal{O} \rightarrow \mathbb{R}$ que possui as seguintes propriedades:

1. Simetria: $d(o_q, o_r) = d(o_r, o_q)$
2. Não negatividade: $d(o_q, o_r) > 0$ se $o_q \neq o_r$ e $d(o_q, o_r) = 0$ se $o_q = o_r$
3. Desigualdade triangular: $d(o_q, o_p) \leq d(o_q, o_r) + d(o_r, o_p)$

onde o_p, o_q e $o_r \in \mathcal{O}$.

Se uma função de distância não satisfaz apenas a propriedade 3, então ela é chamada de semi-métrica. Se ela não satisfaz a propriedade de simetria, então ela é denominada quase-métrica. Se existe valores distintos o_q e o_r tal que $d(o_q, o_r) = 0$ e $d(o_q, o_r) \geq 0$, então ela é uma pseudo-métrica.

É desejável que uma medida de similaridade possa satisfazer todas as três propriedades. Se essas propriedades são requeridas ou não, irá depender da aplicação.

2.5.3 Funções de Distância

Funções de distância são um dos componentes de descritores de imagens responsáveis pela comparação de vetores de características. Os valores calculados pelas funções de distância são responsáveis pela ordenação das imagens retornadas numa determinada consulta.

Existem muitas funções de distância que podem ser aplicadas para medir a similaridade entre dois objetos. Nesta seção são mostrada algumas das funções de distância mais comuns usadas por descritores de imagens.

Distância Euclidiana

A distância Euclidiana é a métrica mais utilizada na literatura. Ela produz a menor distância real para chegar de um ponto a outro. Dados dois pontos o e o' , pertencentes um espaço vetorial n -dimensional, ou seja, $o = \{o_1, \dots, o_n\}$ e $o' = \{o'_1, \dots, o'_n\}$, a distância Euclidiana d_{euc} é dada por:

$$d_{euc}(o, o') = \sqrt{\sum_{i=1}^n (o_i - o'_i)^2} \quad (2.1)$$

Embora seja de grande importância, não se pode negar que o custo computacional da mesma para grandes quantidades de dados é bastante elevado.

Distância City-Block

A métrica City-Block, também conhecida como distância de *Manhattan*, é baseada em vizinhança-de-4 e, como consequência, só permite caminhar em quatro direções para ir de um ponto a outro. Com isso, a distância City-Block nem sempre fornece a menor distância em linha reta entre dois pontos assim como a distância Euclidiana, mas é bastante utilizada por seu cálculo ser menos custoso computacionalmente do que o da distância Euclidiana.

Seja os pontos $o = \{o_1, \dots, o_n\}$ e $o' = \{o'_1, \dots, o'_n\}$, a distância City-Block d_{cb} é definida como:

$$d_{cb}(o, o') = \sum_{i=1}^n |o_i - o'_i| \quad (2.2)$$

Dado um ponto, seus pontos adjacentes na vizinhança-de-4 têm distância 1 e seus pontos adjacentes na vizinhança diagonal têm distância 2.

Distância Chessboard

A métrica Chessboard, também conhecida como distância de *Chebyshev*, possui esse nome porque saindo de um ponto qualquer em um plano cartesiano, é possível se mover em

qualquer direção, assim como os movimentos de um rei no tabuleiro de xadrez. Portanto, esta métrica é baseada em uma vizinhança-de-8.

Assim como na distância City-block, a maior motivação para o uso da métrica Chessboard é o fato desta poder substituir a distância Euclidiana com maior eficácia, uma vez que seu cálculo é computacionalmente mais rápido.

Dados dois pontos $o = \{o_1, \dots, o_n\}$ e $o' = \{o'_1, \dots, o'_n\}$, a distância Chessboard d_{chess} é definida como:

$$d_{chess}(o, o') = \max_i (|o_i - o'_i|) \quad (2.3)$$

Essa distância é uma métrica induzida pela norma do supremo ou L_∞ .

Distância de Minkowski

Considerando dois pontos o e o' , pertencentes um espaço vetorial n -dimensional, ou seja, $o = \{o_1, \dots, o_n\}$ e $o' = \{o'_1, \dots, o'_n\}$, a distância de Minkowski, L_p , é dada por:

$$L_p(o, o') = \sqrt[p]{\sum_{i=1}^n |o_i - o'_i|^p} \quad (2.4)$$

onde p é um parâmetro que modifica a distância de Minkowski resultando em outras distâncias conhecidas. Se $p = 2$ tem-se a distância Euclidiana, se $p = 1$ tem-se distância City-block.

Caso $p \geq 1$ tem-se uma distância métrica, e caso $p < 1$ a desigualdade triangular não é obedecida o que resulta na inexistência de distância métrica.

Uma variante da família de distância de Minkowski é a distância de Minkowski ponderada L'_p , a qual atribui pesos distintos no cálculo da distância, levando-se em consideração que determinados atributos podem exercer maior ou menor relevância em relação ao contexto que está sendo analisado. Essa distância ponderada é formalmente definida como:

$$L'_p(o, o') = \sqrt[p]{\sum_{i=1}^n w_i |o_i - o'_i|^p} \quad (2.5)$$

onde w_i é o vetor de ponderação $w = (w_1, \dots, w_n)$

Portanto, para diferentes contextos podem ser estipulados diferentes vetores de ponderação.

Distância de Hausdorff

Em muitas aplicações, nem todos os pontos de um conjunto $\mathcal{O}$ precisa ter um ponto correspondente em um outro conjunto $\mathcal{O}'$. Tipicamente, esses dois conjuntos $\mathcal{O}$ e $\mathcal{O}'$ são de tamanho diferentes, sendo assim não existe uma correspondência 1-1 entre todos

os pontos. Nesse caso, a medida de similaridade frequentemente usada é a distância de Hausdorff.

Dados dois conjuntos de pontos $\mathcal{O}$ e $\mathcal{O}'$, a distância de Hausdorff entre esses conjuntos é dado por:

$$H(\mathcal{O}, \mathcal{O}') = \max\{\max_{o \in \mathcal{O}} \min_{o' \in \mathcal{O}'} d(o, o'), \max_{o' \in \mathcal{O}'} \min_{o \in \mathcal{O}} d(o, o')\} \quad (2.6)$$

onde $d(o, o')$ é a distância entre os elementos de $\mathcal{O}$ e $\mathcal{O}'$ e normalmente se utiliza distância Euclidiana.

Devido a sensibilidade ao ruído, modificações na distância de Hausdorff foram feitas e podem ser encontradas na literatura. [Zhang e Lu 2004] abordam algumas destas melhorias.

Embora a distância de Hausdorff permita o casamento parcial entre dois conjuntos, ela não possui invariância à mudança de escala, rotação ou translação, fato que restringe bastante o seu uso.

2.6 Avaliação da Performance de um CBIR

Quando um usuário realiza uma consulta via CBIR, ele espera que o sistema recupere apenas documentos que sejam do seu interesse (documentos relevantes). É possível verificar se um CBIR gera satisfação subjetiva, se o mesmo está recuperando uma grande quantidade de documentos relevantes e também se possui precisão na recuperação, ou seja, se não precisa fornecer muitos documentos irrelevantes para recuperar uma quantidade satisfatória de documentos relevantes.

O que se deseja na realidade é estimar se o CBIR está obtendo os documentos pertinentes à consulta. Sendo assim, é preciso saber de antemão quais são os documentos considerados relevantes pelo usuário. Considerando uma coleção à qual se submete uma consulta, vários conjuntos de documentos estão envolvidos. Dentre eles os mais importantes são: o conjunto de documentos pertinentes à consulta (conjunto de documentos relevantes R) e o conjunto de documentos recuperados (conjunto resposta A). Dentro desse último tem se documentos relevantes e não relevantes, sendo que o conjunto de maior interesse é o de documentos relevantes recuperados Ra . A figura 2.3 ilustra a relação entre estes conjuntos.

2.6.1 Curva Precisão-Revocação (*Precision-Recall*)

Dentre as diversas medidas para avaliação da performance de um CBIR, precisão e revocação são as principais. Dada uma consulta q , a precisão é definida como a fração entre o número de imagens relevantes recuperadas, $Ra(q)$, e o número de imagens recuperadas,

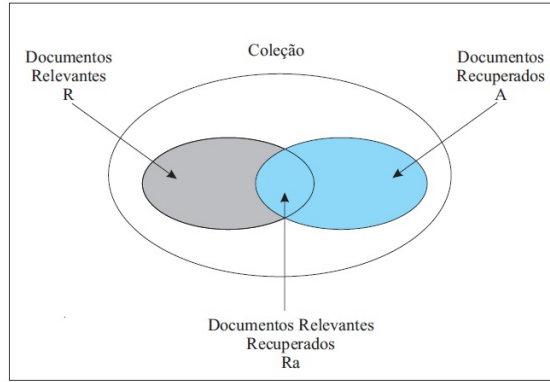


Figura 2.3: Conjuntos de documentos na Recuperação de Informação.

A:

$$Pr(q) = \frac{Ra(q)}{A} \quad (2.7)$$

Por outro lado, a revocação é definida como a fração entre o número de imagens relevantes recuperadas, $Ra(q)$, e o número total de imagens relevantes na coleção, R :

$$Re(q) = \frac{Ra(q)}{R} \quad (2.8)$$

Essas duas medidas estão intimamente ligadas, pois dependem da quantidade de documentos relevantes recuperados, Ra . Em geral, aumentando a revocação a precisão diminui. Isso ocorre porque à medida que documentos são inseridos na resposta a uma dada consulta aumenta-se a quantidade de documentos relevantes recuperados, contudo, juntamente com os documentos relevantes, documentos não relevantes são inseridos em maior proporção que os relevantes, produzindo o efeito de reduzir a precisão uma vez que a resposta perde em qualidade. Assim, é importante buscar um equilíbrio entre as duas medidas, uma maneira de facilitar a análise entre as mesmas é traçar o gráfico da Precisão *versus* Revocação. Como a revocação é aumentada à medida que acrescentamos documentos ao resultado da busca, utilizam-se níveis de revocação para análise da performance.

Nas equações 2.7 e 2.8, a precisão e revocação são estimadas para uma dada consulta individual q . Usualmente os sistemas de recuperação de informação são avaliados para várias consultas distintas. Para avaliar a performance de recuperação sobre todas as consultas de teste, a taxa média de precisão em cada nível de revocação é adotada:

$$\overline{Pr}(r) = \sum_{i=1}^{N_q} \frac{Pr_i(r)}{N_q} \quad (2.9)$$

onde $\overline{Pr}(r)$ é a precisão média no nível de revocação r , N_q é o número de consultas de teste e $Pr_i(r)$ é a precisão no nível de revocação r para a i -ésima consulta.

2.6.2 Precisão- R

A medida de Precisão- R é dada pela precisão entre os R primeiros documentos do *ranking*, onde R é o número total de documentos relevantes para a consulta em análise. A medida de precisão é definida pela equação 2.7.

Uma precisão de 100% seria alcançada quando apenas R documentos são recuperados sem que qualquer outro não relevante o seja, assim a precisão- R mostra o quão perto de uma precisão 100% se tem.

2.7 Considerações Finais

Neste capítulo foi apresentado uma visão geral do funcionamento de um sistema de recuperação imagem baseado em conteúdo, em inglês CBIR (*Content Based Image Retrieval*). Essa abordagem de recuperar informação é de grande interesse para diversas aplicações, uma vez que um grande conjunto de imagens digitais vem sendo gerado e armazenado em bancos de dados.

Foi visto que um CBIR ideal é aquele capaz de encontrar de maneira eficiente imagens perceptivamente similares com base em uma coleção de dados. Por isso, é necessário que ele faça uso de um descritor que seja capaz caracterizar as informações presentes nas imagens, de tal maneira que elas sejam compatíveis com a intuição humana.

O primeiro passo para recuperar uma imagem é extrair as características contidas nessa imagem, que frequentemente são cor, forma ou textura. Depois, é necessário representar, através de valores, essas informações de tal maneira a se construir um vetor de características. Por fim, a similaridade é feita através de uma função de distância entre esses vetores. Essa função calcula a distância entre dois objetos e retorna um valor real positivo que indica o grau de semelhança entre eles. Em geral, quanto menor a distância entre os vetores mais similares eles são. Foram apresentadas várias funções de distância que podem ser aplicadas para medir a similaridade entre dois objetos, como também algumas das funções mais comuns usadas por sistemas de recuperação por conteúdo, bem como as propriedades que essa função deve satisfazer para ser considerada uma função de distância métrica e os tipos de consulta por similaridade.

O interesse desse trabalho é a recuperação de imagens baseado em formas. Diversos descritores para formas têm sido propostos na literatura, e no próximo capítulo algumas dessas técnicas serão apresentadas.

Capítulo 3

Caracterização de Formas

3.1 Introdução

Forma é um conceito que é amplamente entendido, mas bem difícil de se definir formalmente. A percepção humana de formas é um conceito de alto-nível e as definições matemáticas tendem a descrever formas com características de baixo nível.

A identificação, reconhecimento e recuperação de formas em imagens têm utilidade em diversas aplicações, por exemplo, no reconhecimento de caracteres alfanuméricos, na detecção e reconhecimento de pessoas em sistemas de segurança, no rastreamento de objetos em vídeos, etc.

Para uma máquina, a forma é um dos aspectos mais difícil de se detectar, devido à dificuldade de segmentar os objetos de interesse presentes na imagem. Entende-se por segmentação o processo de extração de objetos relevantes em uma imagem, o que não é uma tarefa trivial, pois diferentes objetos requerem diferentes algoritmos ou parâmetros de segmentação, o que torna complexa a tarefa de segmentação. Os descritores de formas exigem, muitas vezes, que as imagens sejam previamente segmentadas.

Após o processo de segmentação, os objetos de interesse na imagem se encontram separados do fundo e de outros objetos irrelevantes, podendo ser utilizados diretamente pelos descritores de formas. Por este motivo, geralmente as propostas de descritores já supõem que as imagens foram segmentadas em um processo anterior. Esta estratégia também é utilizada neste trabalho.

Nas próximas seções, serão apresentadas algumas abordagens encontradas na literatura para descrever e caracterizar formas.

3.2 Classificação dos Descritores

Vários métodos para reconhecer e descrever formas foram e continuam sendo desenvolvidos. A análise de formas de uma imagem pode ser feita de maneiras diferentes. De

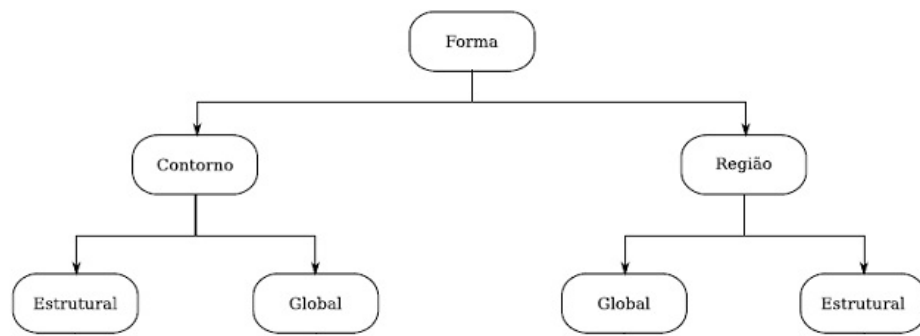


Figura 3.1: Classificação das técnicas de descrição e representação de formas.

acordo com [Zhang e Lu 2004], esses métodos podem ser classificados em dois grandes grupos: métodos baseados em contorno e métodos baseados em região. Os métodos baseados em contornos apenas analisam o contorno da forma, enquanto os métodos baseados em região analisa a forma (o objeto) como um todo. A figura 3.1 ilustra esta classificação.

A abordagem baseada em contorno é mais popular na literatura, entretanto ela pode sofrer efeitos negativos na presença de ruído. Além disso, não é sempre que o contorno de uma forma está presente na imagem. Por esse motivo, a abordagem baseada em região pode ser mais robusta na recuperação de formas por considerar a informação de toda a forma ao invés de considerar apenas a informação do contorno.

Cada um dos métodos (contorno e região) são ainda subdivididos em global e estrutural, como visto na figura 3.1. A diferença entre essas duas abordagens está no fato de que na abordagem global não se divide a forma em sub-partes, enquanto que na abordagem estrutural a forma é dividida em segmentos e cada método determina um tipo diferente de primitiva. Entretanto, a abordagem estrutural é mais complexa computacionalmente, especialmente para o cálculo de distância, além disso são mais sensíveis a ruído.

Exemplos de descritores baseado em contorno com abordagem global são os Descritores de Fourier [Gonzalez e Woods 2001], e com abordagem estrutural tem-se o *Curvature Scale Space* (CSS), proposto por [Mokhtarian e Mackworth 1986]. Os descritores baseados em Momentos Invariantes [Liao e Pawlak 1996] são exemplos de métodos regionais com abordagem global, e analisar uma forma através do seu fecho-convexo é uma maneira de descrever uma forma em regiões com uma abordagem estrutural.

3.3 Características Geométricas Simples

Frequentemente, o uso de características geométricas podem representar eficientemente formas simples e que sejam bem diferentes entre si. Porém, o uso isolado dessas características não são adequadas, neste caso, essas características são combinadas com outros descritores para aumentar o poder de discriminação de formas.

Uma forma pode ser descrita por diferentes aspectos geométricos, tais como: excentricidade, perímetro, convexidade, taxa de circularidade, etc. Nas próximas subseções algu-

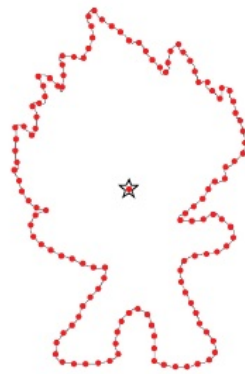


Figura 3.2: Centróide de uma forma.

mas dessas características serão detalhadas, apenas aquelas que serão citadas no decorrer do trabalho.

3.3.1 Área

Para uma imagem binária, a área A de um objeto pode ser facilmente calculada através da simples contagem do número de pontos do interior do objeto. Esta característica é naturalmente dependente da escala e do tamanho do objeto, mas é invariante à rotação e translação.

3.3.2 Centróide

Na geometria, o centróide é o ponto no interior da forma geométrica que define o seu centro geométrico. O centróide (g_x, g_y) pode ser facilmente obtido através da equação:

$$\begin{cases} g_x = \frac{1}{n_c} \sum_{i=1}^{n_c} x_i \\ g_y = \frac{1}{n_c} \sum_{i=1}^{n_c} y_i \end{cases}$$

onde n_c é o número de pontos do contorno da forma, $(x_i, y_i)_{i=1 \dots n_c}$ são as coordenadas dos pontos do contorno do objeto.

A figura 3.2 mostra o centróide de uma forma. Esse ponto é invariante às transformações geométricas de escala, rotação e translação.

3.3.3 Excentricidade

A excentricidade é uma característica geométrica medida pelos eixos principais da forma. Esses eixos são definidos como os dois eixos da maior elipse que contém a forma. A excentricidade é dada pela razão entre o menor eixo e o maior eixo dessa elipse. O valor é dado entre 0 e 1: uma silhueta cujo valor de excentricidade é igual a 1 é um círculo, enquanto que uma forma com excentricidade igual a 0 é uma linha.



Figura 3.3: Fecho convexo de uma forma.

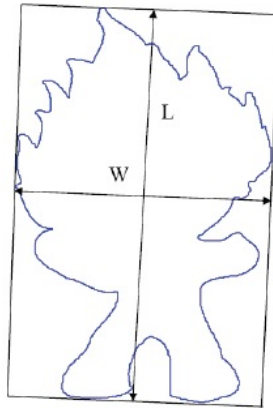


Figura 3.4: Mínimo retângulo delimitador de uma forma.

3.3.4 Fecho Convexo

Por definição [Peura e Iivarinen 1997], uma região R é convexa se, e somente se, para quaisquer dois pontos $x_1, x_2 \in R$, todo o segmento traçado entre os pontos x_1 e x_2 estiver contido em R . Se houver alguma concavidade, parte do segmento ou todo ele passa por fora da região. O fecho convexo de R é na verdade a menor região convexa H que satisfaz a condição $R \subset H$, caso não haja nenhuma concavidade o fecho convexo da região é igual a ela própria. Na figura 3.3 a região em cinza é o fecho convexo da forma.

3.3.5 Relação de Aspecto

O mínimo retângulo delimitador, também chamado de caixa delimitadora mínima, é o menor retângulo que contém todos os pontos da forma. Essa informação é usada para medir a taxa de aspecto de uma forma, que é definida como razão entre comprimento L e a largura W do mínimo retângulo delimitador. A figura 3.4 mostra esses parâmetros.

3.3.6 Solidez

Solidez é uma medida que especifica a razão entre o número de pontos do fecho convexo H e a área A da forma:

$$\text{Solidez} = \frac{A}{H} \quad (3.1)$$

A solidez de uma região convexa é sempre igual a 1.

3.4 Descritores de Formas

Nesta seção uma revisão bibliográfica de alguns dos descritores mais recentes e mais citados na literatura são apresentados e detalhados.

3.4.1 Curvature Scale Space (CSS)

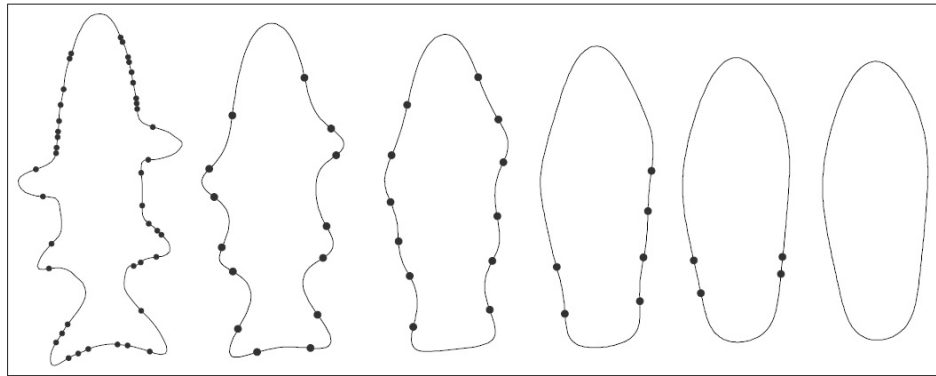
O *Curvature Scale Space* (CSS), proposto por [Mokhtarian e Mackworth 1986], é um descritor que visa caracterizar formas através de seu contorno. Esse método consiste em encontrar os pontos de inflexão do contorno quando o mesmo é suavizado por um filtro gaussiano de largura σ crescente.

O CSS é considerado um método confiável e bastante rápido para a análise de formas em grandes bases de dados. Este método é bastante robusto em relação a ruídos, escala e mudanças de orientação.

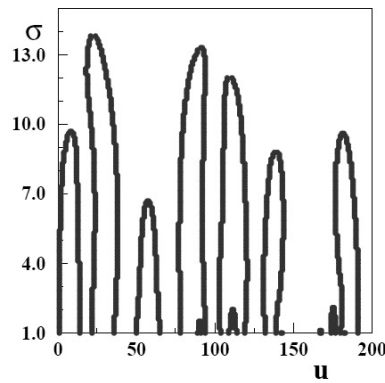
Para calcular a representação *scale space* de uma forma, inicialmente é obtido o contorno da mesma, e então os pontos do contorno são ordenados percorrendo o contorno no sentido anti-horário. As coordenadas $x(i)$ e $y(i)$ representam as coordenadas paramétricas do contorno Γ , com $\Gamma(i) = (x(i), y(i))$, para $i = 1 \dots n_c$, onde n_c é o número de pontos do contorno. A versão evoluída (suavizada) Γ_σ do contorno é definida como: $\Gamma_\sigma(i, \sigma) = (X(i, \sigma), Y(i, \sigma))$, onde $X(i, \sigma) = x(i) * g(i, \sigma)$, $Y(i, \sigma) = y(i) * g(i, \sigma)$, “*” é o operador de convolução e $g(i, \sigma)$ é uma função gaussiana de largura σ . A representação CSS de Γ consiste na identificação dos pontos de inflexão de Γ_σ para valores crescente de σ .

Como pode ser observado na figura 3.5, espera-se que os pontos de inflexão do contorno que permanecerem na representação sejam as características mais significativas da forma. Em 3.5(a) a aplicação do filtro gaussiano com várias larguras pode ser visualizada, sendo que os pontos marcados são pontos de inflexão do contorno. Em 3.5(b) tem-se a árvore de intervalo, que é o produto final do processo de suavização e localização dos pontos que são cruzamentos por zero. Essa árvore guarda informações sobre os pontos de inflexão.

Assim, o vetor do CSS representa uma organização multiescala dos pontos de curvatura zero no contorno. Neste sentido, a dimensão do vetor pode variar para formas diferentes,



(a)



(b)

Figura 3.5: Obtenção do descritor CSS: (a) larguras $\sigma = 1, 4, 7, 10, 12, 14$ do filtro gaussiano; (b) árvore de intervalo

requisitando um algoritmo de casamento especial para o cálculo de dois vetores CSS. Uma publicação mais recente e detalhada sobre CSS pode ser encontrada em [Mokhtarian e Bober 2003].

3.4.2 Tensor Scale Descriptor (TSD)

Tensor Scale, inicialmente apresentado por [Saha 2005], é um parâmetro morfométrico que unifica a representação de orientação, espessura e anisotropia de estruturas locais na imagem, que pode ser utilizado em várias aplicações de visão computacional e processamento de imagem.

O *Tensor Scale*, em qualquer ponto p da imagem, é a representação paramétrica da maior elipse centrada em p que está contida em uma mesma região homogênea, de acordo com um critério predeterminado (por exemplo: uma textura, uma diferença entre os brilhos dos pixels, etc). A elipse centrada na origem p é unicamente definida por três fatores:

- Orientação(p) = θ
- Anisotropia(p) = $\sqrt{1 - \frac{|t_2(p)|^2}{|t_1(p)|^2}}$

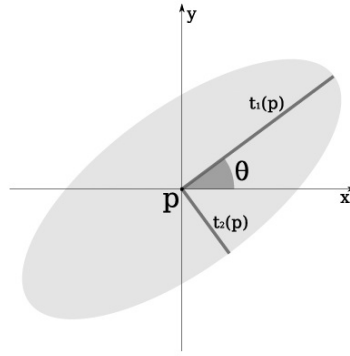


Figura 3.6: Fatores do *Tensor Scale*.

- Espessura(p) = $|t_2(p)|$

onde, $|t_1(p)|$ e $|t_2(p)|$ denotam o comprimento dos dois semi-eixos da elipse centrada em p : $|t_1(p)|$ é o maior eixo e $|t_2(p)|$ o menor eixo; e θ é o ângulo entre $t_1(p)$ e o eixo horizontal.

Na Figura 3.6, pode-se visualizar um exemplo dos três fatores do *Tensor Scale*: a elipse centrada em p possui a orientação representada pelo ângulo θ entre o eixo- x e o maior eixo da elipse $t_1(p)$; a anisotropia, calculada a partir dos comprimentos dos semi-eixos $t_1(p)$ e $t_2(p)$; e a espessura, representada pelo semi-eixo menor $t_2(p)$.

Para representar formas, o descritor TSDIZ (*Tensor Scale Descriptor with Influence Zones*), proposto por [Andaló et al. 2010], baseia-se no fato de que objetos distintos geralmente apresentam diferentes distribuições das orientações do *Tensor Scale*. A partir desta suposição, a proposta do descritor é comparar formas analisando a distribuição de orientações. Para isso, o contorno da forma é dividido em n_k segmentos e então, para cada segmento, a média angular é calculada sobre os pontos da imagem que estão sob influência desse segmento. A Figura 3.7 mostra o histograma das orientações do *Tensor Scale* da forma da figura 3.7(a) para $n_k = 50$.

O vetor de características do descritor TSDIZ é formada por n_k posições correspondentes às orientações médias das zonas de influência mapeadas para os segmentos.

3.4.3 Descritores de Fourier

Descritores de Fourier (*Fourier Descriptors* - FD) consistem na obtenção de um conjunto de coeficientes que descrevem um dado contorno. Esse descritor é amplamente utilizado, pois é um descritor simples, de implementação direta.

Seja $B = \{p_0, \dots, p_{n_c-1}\}$ o conjunto formado pelos pontos do contorno de uma forma, ordenados percorrendo o contorno no sentido horário (ou anti-horário), $p_i = (x_i, y_i)$, para $i = 0, \dots, n_c - 1$, onde n_c é o número de pontos do contorno e x_i e y_i representam as coordenadas do ponto p_i na imagem. Cada ponto p_i é representado por número complexo s_i , obtendo-se assim uma representação unidimensional. Tal número é representado por: $s_i = x_i + jy_i$. Assim, o eixo x é o eixo real e o eixo y é o eixo imaginário. A partir desses

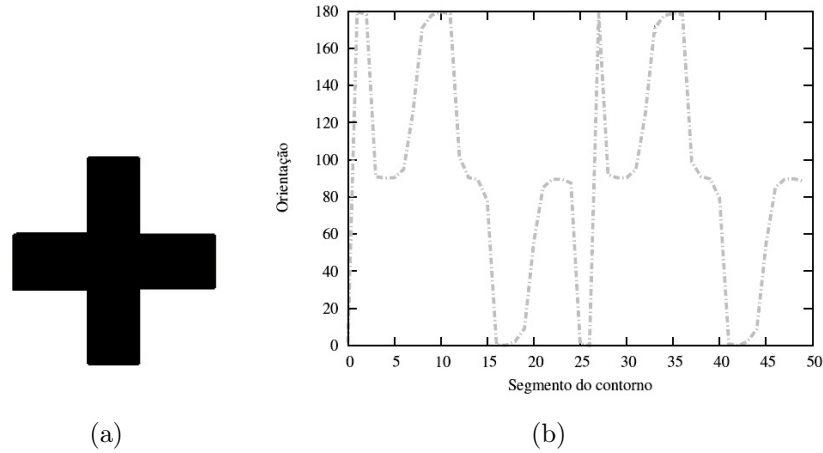


Figura 3.7: Exemplo de curva de orientação obtidas com o descritor TSD: (a) uma forma; (b) curva de orientação

dados a transformada discreta de Fourier (*Discrete Fourier Transform* - DFT) é definida:

$$a(p_i) = \frac{1}{n_c} \sum_{k=0}^{n_c-1} s_i \exp[-j2\pi(k/n_c)] \quad (3.2)$$

para $i = 0, 1, \dots, n_c - 1$.

Os coeficientes complexos $a(p_i)$ são chamados de descritores de Fourier. Para se reconstruir o contorno utiliza-se a transformada inversa de Fourier:

$$s(p_i) = \frac{1}{n_c} \sum_{u=0}^{n-1} a(p_i) \exp[-j2\pi(k/n_c)] \quad (3.3)$$

para $i = 0, 1, \dots, n_c - 1$.

Muitas vezes, torna-se desnecessário utilizar os n_c pontos do contorno, já que o contorno pode possuir um grande número de pontos. Utilizam-se então apenas M pontos igualmente espaçados, onde M geralmente é uma potência de 2.

A distância Euclidiana pode ser usada para medir a similaridade entre dois vetores do descritor de Fourier.

Descritores de Fourier são sensíveis a translação, rotação e mudança de escala. Contudo, através de simples mudanças pode-se obter invariância quanto a estas transformações. Além disso, os descritores de Fourier são sensíveis à ruído, o que faz com que ele não apresente bons resultados em um banco de dados que tenha imagens ruidosas. Mais detalhes sobre esse descritor pode ser encontrado em [Gonzalez e Woods 2001].

Além de descrever formas, os descritores de Fourier também podem descrever texturas da imagem, por isso, são bastante utilizados e difundidos, tornando alvo de constantes adaptações, o que tem trazido melhorias importantes em contextos bem definidos.

3.4.4 Invariantes de Momento

A estatística sempre contribuiu para o processamento de imagens. Dentre essas contribuições uma das mais importantes é a aplicação de momentos invariantes no reconhecimento de padrões. Momento é uma característica numérica de uma distribuição de frequência. O momento de uma variável aleatória x centrada em um momento t é dada por:

$$\mu_t(x) = \sum_{i=1}^n \frac{(x_i - \bar{x})^t f_i}{n} \quad (3.4)$$

onde n é a quantidade de elementos da distribuição, x_i é um valor associado à uma variável aleatória, f_i é a frequência com que x_i aparece na amostra e $\bar{x}$ é a média do conjunto de valores do espaço amostral.

Normalmente, apenas sete momentos invariantes são calculados, sendo que os mesmos baseiam-se nos momentos centrais de segunda e terceira ordem. O fundamento que leva a calcular apenas estes sete momentos é que exatamente essa quantidade representa detalhes distinguíveis pelo ser humano. Um descritor baseado em momento pode usar todos os sete ou apenas alguns deles. O cálculo desses 7 momentos pode ser feito da seguinte forma:

1. Cálculo dos momentos centrais de segunda e terceira ordem $\mu_{00}, \mu_{11}, \mu_{20}, \mu_{02}, \mu_{21}, \mu_{12}, \mu_{30}, \mu_{03}$:

$$\mu_{pq} = \sum_{i=1}^n ((x(i) - \bar{x})^p \cdot (y(i) - \bar{y})^q) \quad (3.5)$$

onde n é o número de pontos (x, y) do contorno fechado, e $\bar{x}$ e $\bar{y}$ são as médias das componentes x e y dos pontos do contorno, respectivamente.

2. Normalização dos momentos centrais μ_{pq} , gerando os momentos centrais normalizados m_{pq} , através das expressões

$$m_{pq} = \frac{\mu_{pq}}{(\mu_{00})^\gamma} \quad (3.6)$$

onde $\gamma = 0.5 * (p + q) + 1$

3. Cálculo dos momentos invariantes através das expressões:

- $M1 = m_{20} + m_{02}$
- $M2 = (m_{20} - m_{02})^2 + 4(m_{11})^2$
- $M3 = (m_{30} - 3m_{12})^2 + (m_{03} - 3m_{21})^2$
- $M4 = (m_{30} + 3m_{12})^2 + (m_{03} + 3m_{21})^2$

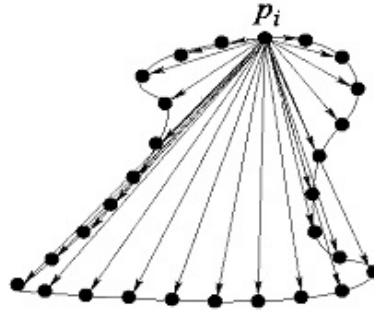


Figura 3.8: Beams que conectam o ponto p_i .

- $M5 = (m_{30} - 3m_{12})(m_{03} + 3m_{21})[(m_{30} + 3m_{12})^2 - 3(m_{03} + 3m_{21})^2] + (m_{30} - 3m_{12})(m_{03} + 3m_{21})[(m_{30} + 3m_{12})^2 - 3(m_{03} + 3m_{21})^2]$
- $M6 = (m_{20} - m_{02})(m_{30} + 3m_{12})(m_{03} + 3m_{21})^2 + 4m_{11}(m_{30} - m_{12})(m_{30} + 3m_{21})$
- $M7 = (3m_{12} - m_{30})(m_{30} + m_{12})[(m_{30} + m_{12})^2 - 3(m_{03} + m_{21})^2] + (m_{03} - 3m_{21})(m_{03} + 3m_{12})[(m_{03} + m_{21})^2 - 3(m_{30} + 3m_{12})^2]$

A vantagem de se descrever formas usando *Moment Invariants* é que momentos caracterizam a regularidade de um objeto, pois eles refletem o espalhamento do mesmo em relação ao seu ponto médio. Essa característica faz o descritor ser invariante à transformações geométricas.

Neste descritor, também pode-se utilizar a distância Euclidiana para o cálculo da similaridade, uma vez que os vetores tem dimensões iguais.

3.4.5 Beam Angle Statistics (BAS)

O BAS, proposto por [Arica e Vural 2003], é um descritor baseado em *beams* que se originam em pontos do contorno. *Beams* são conjuntos de linhas que conectam um determinado ponto de referência com todos os outros pontos do contorno. Em cada ponto do contorno, o ângulo entre um par de *beams* é calculado e o vetor de características é definido usando-se estatísticas de primeira, segunda e terceira ordem aplicadas aos ângulos com base em um conjunto de sistemas de vizinhança.

Seja $B = \{p_1, \dots, p_{n_c}\}$ o conjunto formado pelos pontos do contorno de uma forma, ordenados percorrendo o contorno no sentido horário (ou anti-horário), onde $p_i = (x_i, y_i)$, para $i = 1, \dots, n_c$, n_c é o número de pontos do contorno e x_i e y_i representam as coordenadas do ponto p_i na imagem.

Para cada ponto p_i , os *beams* de p_i são definidos pelo vetor: $L(p_i) = \{p_{i+k}, p_{i-k}\}$, para $k = 1 \dots n_c/2$. A figura 3.8 mostra os *beams* do ponto p_i . Para o ponto p_i o ângulo do *beam* entre seu k -vizinho é dado por:

$$C_k(p_i) = \{\theta_{p_{i+k}} - \theta_{p_{i-k}}\} \quad (3.7)$$

onde, $\theta_{p_{i+k}} = \tan^{-1}\left(\frac{x_{i+k} - x_i}{y_{i+k} - y_i}\right)$ e $\theta_{p_{i-k}} = \tan^{-1}\left(\frac{x_{i-k} - x_i}{y_{i-k} - y_i}\right)$, para $k = 1 \dots n/2$.

A estatística do ângulo do *beam* do ponto p_i é dado pelo seu m -ésimo momento definido por:

$$E[C^m(p_i)] = \sum_k C_k^m P_k(C_k(p_i)) \quad (3.8)$$

$m = 0, 1, 2, 3, \dots$ e $P_k(C_k(p_i))$ é a densidade probabilidade da função $C_k(p_i)$.

Segundo os autores, até a terceira ordem é suficiente para representar um contorno. Assim, o vetor de característica é formado pela estatísticas de primeira, segunda e terceira ordem de cada ponto p_i do contorno:

$$p_i = \{E(C^1(p_i)), E(C^2(p_i)), E(C^3(p_i))\} \quad (3.9)$$

Além disso, para os autores torna-se desnecessário utilizar todos os pontos do contorno, já que o contorno pode possuir um grande número de pontos. A partir de testes experimentais é sugerido utilizar uma amostra de 60 pontos do contorno igualmente espaçados.

3.4.6 Multiscale Fractal Dimension (MS Fractal)

Enquanto que a dimensão topológica é restrita à valores inteiros, a dimensão fractal permite valores fracionários. Disseminado por [Mandelbrot 1982], a dimensão fractal provê um meio de caracterizar auto-similaridade (ou auto-afinidade) de objetos reais ou abstratos.

Dada $\mathcal{S}$ uma forma representada por suas coordenadas Cartesianas (x, y) e $\mathcal{S}_r$ sua dilatação Euclidiana por um raio r , a dimensão fractal de Minkowski-Bouligand [Tricot 1995] é definida pela função:

$$F = 2 - \lim_{r \rightarrow 0} \frac{\log(A(r))}{\log(r)} \quad (3.10)$$

onde $A(r)$ é a área da forma $\mathcal{S}_r$.

A equação 3.10 pode ser estimada pela interpolação da curva logarítmica $A(r)$ em termos do raio de dilatação r , computando o coeficiente angular $A'(r)$ desta linha e considerando F como $F(r) = 2 - A'(r)$, como descrito por [Torres et al. 2004].

O vetor de características é formado por 50 amostras deste polinômio e a distância entre dois vetores é calculada por meio da distância Euclidiana.

3.4.7 Contour Saliences (CS)

O descritor Saliências do Contorno (*Contour Saliences* - CS), proposto por [Torres e Falcão 2007], utiliza os pontos salientes do contorno para representação de formas.

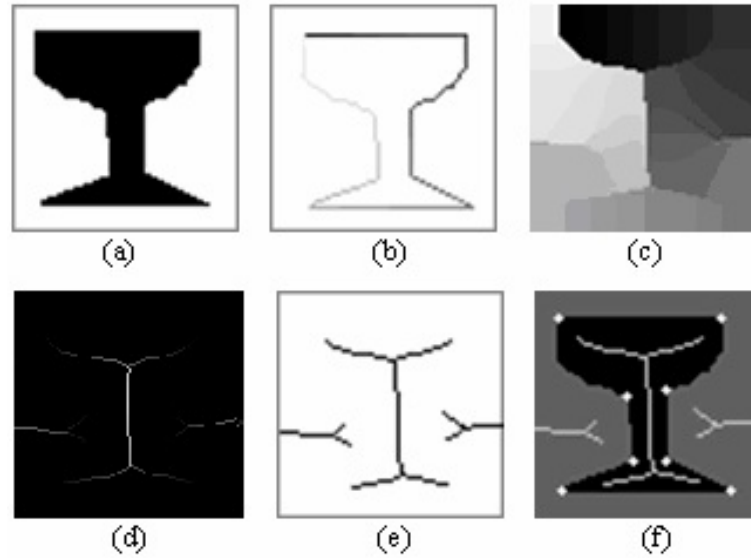


Figura 3.9: (a) Imagem original (b) contorno rotulado (c) imagem rotulada (d) imagem diferença (e) esqueleto obtido a partir do *threshold* (f) pontos de saliências.

Detecção dos Pontos de Saliência via Esqueletização

Para encontrar os pontos de saliência, [Torres e Falcão 2007] utilizam o esqueleto da forma como um referencial. Cada extremidade do esqueleto corresponde a um ponto de saliência: uma extremidade do esqueleto interno corresponde a uma saliência côncava e uma extremidade do esqueleto externo corresponde a uma saliência convexa.

Para encontrar o esqueleto, os autores utilizam a técnica de dilatação dos pontos do contorno. Esse procedimento pode ser visualizado pela figura 3.9 e é definido pelos seguintes passos:

- 1º passo: percorrer todos os pontos do contorno da forma no sentido horário atribuindo um rótulo incremental à cada um (figura 3.9-b). Portanto, os valores desses rótulos variam de 1 a n_c , onde n_c é o numero de pontos do contorno;
- 2º passo: para cada ponto p da imagem, encontrar o ponto do contorno q mais próximo e atribuir o rótulo de q à p (figura 3.9-c);
- 3º passo: encontrar a imagem diferença D para cada ponto p da imagem, dada pela equação:

$$D(p) = \max_{\forall q \in A_4(p)} \{\min(p - q, n_c - (p - q))\} \quad (3.11)$$

onde $A_4(p)$ é o conjunto dos pontos da vizinhança-de-4 do ponto p (figura 3.9-d).

- 4º passo: aplicar um *threshold* para eliminar os pontos menos relevantes (mais escuros). Os autores definiram esse threshold como 5% de n_c , onde n_c é o número de pontos do contorno. Dessa maneira, os pontos da imagem com valores menores que 5% de n_c não farão parte do esqueleto da forma (figura 3.9-e).

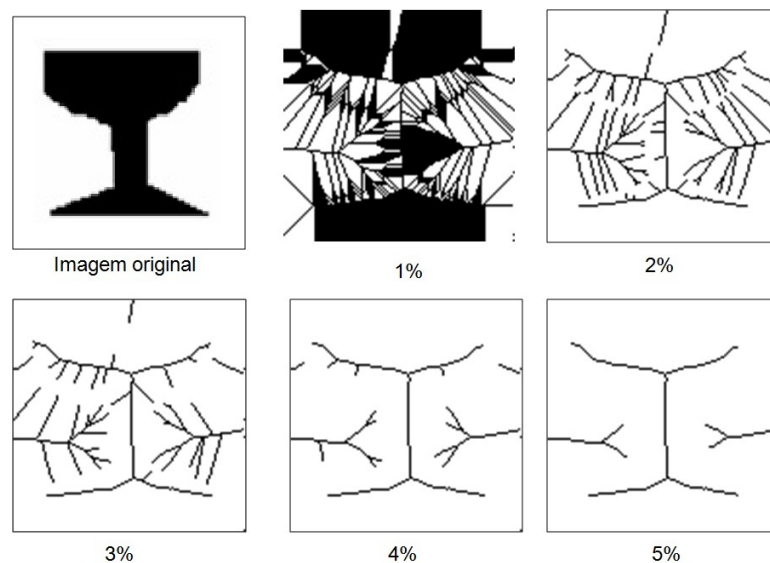


Figura 3.10: Esqueleto de uma forma usando diferentes *thresholds*.

Essa maneira de detecção dos pontos de saliência é dependente de um *threshold* para determinar os pontos que irão fazer parte do esqueleto da imagem. Diferentes valores para o *threshold* irão resultar em um esqueleto com mais ou menos ramificações e, conseqüentemente, mais ou menos pontos de saliência serão detectados. A figura 3.10 ilustra o esqueleto de uma forma obtido usando diferentes *thresholds*.

Representação dos Pontos de Saliência

Após localizar os pontos de saliências, o vetor de características para uma forma com n saliências é dado por: $\mathcal{S} = \{s_1, \dots, s_n\}$, onde cada ponto s_i é uma dupla $(s_{1,i}, s_{2,i})$, $s_{1,i}$ é a posição relativa e $s_{2,i}$ o valor que representa a área de influência da saliência na imagem.

A posição relativa $s_{1,i}$, adotada pelo descritor CS, é dada pela posição da saliência em relação à todos os pontos do contorno. Para isso o contorno é rotulado percorrendo-o em sentido horário, dessa maneira os valores desses rótulos irão variar de 1 até n_c , onde n_c é o número de pontos do contorno. O valor da posição relativa $s_{1,i}$ é dada pelo rótulo da saliência dividido por n_c .

A área de influência $s_{2,i}$ é obtida determinando-se um raio ao redor do ponto de saliência e contando a quantidade de pontos dentro desse raio que possui o mesmo rótulo que a saliência, tal como obtido no 2º passo do algoritmo para detecção do esqueleto.

Função de Similaridade

Para medir a similaridade entre duas formas caracterizadas pelos seus pontos de saliência, o descritor CS utiliza um algoritmo heurístico de casamento. Dados dois contorno distintos representados pelos conjuntos de saliência $A = \{a_1, \dots, a_n\}$ e $B = \{b_1, \dots, b_m\}$, onde n e m representam o número de saliências da forma A e B respectivamente, o algoritmo de casamento é dado por:

1. Crie $A' = \{a'_1, \dots, a'_n\}$ e $B' = \{b'_1, \dots, b'_m\}$, ordenando A e B por ordem decrescente dos valores de saliência $a_{2,i}$ e $b_{2,i}$, respectivamente.
2. Crie a lista L contendo pares de pontos candidatos a casamento de A' e B' . Um par (a'_i, b'_j) está em L se $|a'_{2,i} - b'_{2,j}| \leq 0.2a'_{2,1}$. E um par (b'_j, a'_i) está em L se $|b'_{2,j} - a'_{2,i}| \leq 0.2b'_{2,1}$;
3. Para cada par de candidatos da lista L , encontre o parâmetro de deslocamento α , onde $\alpha = a'_{1,i} - b'_{1,j}$. Desloque os pontos de saliência $a'_{1,i}$ de A' por α , obtendo $A'' = \{a''_1, \dots, a''_n\}$;
4. A distância d entre A'' e B é dada por:

$$d = \sum_{k=1}^{\min\{m,n\}} d_k \quad (3.12)$$

onde,

$$d_k = \begin{cases} \sqrt{(a''_{1,k} - b_{1,k})^2 + (a''_{2,k} - b_{2,k})^2}, & \text{se } |a''_{1,k} - b_{1,k}| \leq 0.2, \\ a''_{2,k} + b_{2,k}, & \text{caso contrário.} \end{cases}$$

Finalmente, se $n \neq m$, a posição relativa dos pontos não casados são adicionados na distância;

5. Repita os passos 3 e 4 considerando outros pares candidatos em L ;
6. Selecione a menor distância d como sendo a distância final entre A e B .

Esse algoritmo de casamento entre as saliências não é invariante à espelhamento entre os vetores de características, ou seja, duas formas idênticas porém invertidas terão valores diferentes de similaridade.

3.5 Considerações Finais

Existem diversas maneiras de descrever formas, sendo encontrada uma vasta gama de métodos para tal fim na literatura. Neste capítulo, foram discutidos alguns desses métodos, com o objetivo de fornecer uma visão geral sobre a caracterização de formas.

Frequentemente, o uso de características geométricas podem reconhecer eficientemente formas simples. Porém, o ideal é que essas características sejam combinadas com outros descritores para aumentar o poder de discriminação de formas.

Neste capítulo foi apresentado e detalhado alguns descritores encontrados na literatura. Os descritores de Fourier e os “Invariantes de Momento” são descritores simples, de implementação direta e continuam sendo citados e aprimorados até hoje. Os descritores de Fourier são um dos principais descritores encontrados na literatura devido sua grande

quantidade de aplicações. O descritor CSS é bastante citado e comparado por outros trabalhos devido a sua robustez à ruídos, já que utiliza uma função gaussiana para suavizar o contorno da forma. O TSDIZ é um descritor de formas recente baseado em um padrão morfométrico, denominado *Tensor Scale*.

O descritor “Saliências do Contorno” (*Contour Saliences* - CS), utiliza as saliências do contorno para descrição de formas. Para detectar os pontos de saliência, esse descritor utiliza o esqueleto da forma como um referencial. Para representar os pontos duas informações são utilizadas: posição relativa e área de influência. E para medir a similaridade um algoritmo de casamento heurístico é apresentado.

Motivado pelo descritor CS, neste trabalho é proposto um novo descritor para as saliências das formas, referenciado como *Shape Saliences Descriptor* (SSD), que utiliza outros métodos, mais robusto à ruídos, do que as técnicas utilizadas pelo descritor CS. Nos próximos capítulos, os métodos de detecção, representação e similaridade usados pelo descritor SSD serão detalhados e discutidos.

Parte II

Método Proposto

Capítulo 4

Detecção de Pontos de Saliência em Contornos

4.1 Introdução

Saliências são caracterizadas por serem os pontos de maior curvatura ao longo do contorno de uma forma, porém em contornos ruidosos a sua detecção não é uma tarefa fácil, uma vez que pontos ruidosos também podem apresentar alto valor de curvatura. A figura 4.1 ilustra essa situação: na silhueta da estrela sem ruído é fácil identificar os 10 picos que correspondem às 10 saliências da estrela, porém, na silhueta ruidosa é difícil localizar os 10 picos mais relevantes da curvatura. Um detector de saliências deve ser capaz de localizar esses pontos mesmo em imagens com alto nível de ruído. Além disso, de acordo com [Mokhtarian e Suomela 1998], esse detector deve satisfazer outros importantes critérios, tais como: todos os pontos de saliência devem ser detectados; nenhuma falsa saliência deve ser detectada e o detector deve ser computacionalmente eficiente.

A detecção de saliências pode ser feita por meio de diferentes abordagens, por esse motivo na literatura pode-se encontrar diversas técnicas. Alguns métodos clássicos e simples, como os detectores apresentadas por [Rosenfeld e Johnston 1973, Rosenfeld e Weszka 1975], utilizam K-cosseno para calcular o ângulo de cada ponto do contorno, onde K é um parâmetro que se refere à vizinhança dos pontos considerados no cálculo do ângulo. O detector apresentado por [Freeman e Davis 1977] detecta os pontos de saliência analisando a variação da curvatura através das diferenças angulares entre o contorno e um segmento de linha reta ao longo do contorno. O trabalho de [Asada e Brady 1986] usa suavização Gaussiana e B-splines cúbicas para calcular a curvatura dos pontos do contorno. Outros trabalhos baseados no trabalho de [Asada e Brady 1986] foram propostos por [Langridge 1982], [Medioni e Yasumoto 1995] e [Fischler e Bolles 1986]. Alguns trabalhos mais atuais, como a técnica apresentada por [Chetverikov 2003], consiste em classificar como saliência o ponto do contorno que tem um ângulo de abertura específico

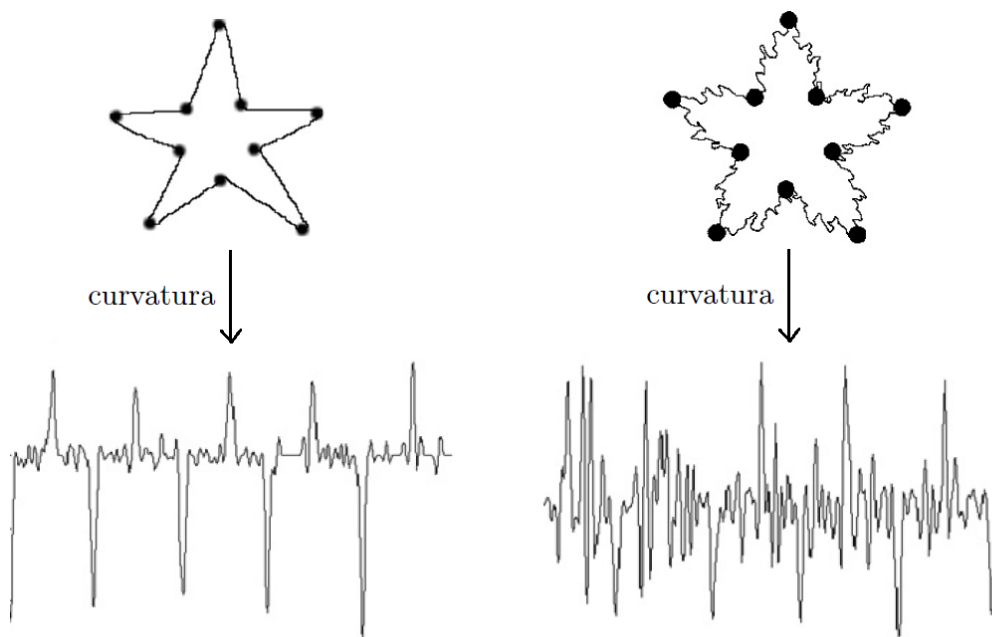


Figura 4.1: Curvatura da silhueta de uma estrela sem ruído e com ruído.

calculado em relação aos seus K -ésimos vizinhos e obedecendo à um espaçamento mínimo entre saliências adjacentes. Existem outros trabalhos que usam metodologias diferentes para o auxílio na detecção de saliências, tais como redes neurais [Tsai 1997] e lógica fuzzy [Lee e Bien 1996]. A técnica proposta por [Torres e Falcão 2007] utiliza o esqueleto da forma como um referencial: os pontos de saliências são dados pelas extremidades do esqueleto interno e externo da forma.

Apesar do grande número de técnicas publicadas nas últimas décadas, a tarefa de detecção de saliências é ainda um campo fértil para o avanço de pesquisas. Quase todas as técnicas encontradas na literatura necessitam de parâmetros para: limitar o valor mínimo de curvatura de um ponto para este ser considerado uma saliência; e/ou limitar o espaçamento mínimo entre pontos de saliência adjacentes. Encontrar um valor padrão para esses parâmetros, válido para todas as imagens, não é uma tarefa fácil e sempre vai depender do auxílio da percepção humana.

Nesse trabalho, um novo método para detectar saliências é apresentado. A proposta é utilizar um filtro anisotrópico para eliminar os pontos ruidosos do contorno, que não são saliências, mas também apresentam alto valor de curvatura. Essa técnica de suavização é não-linear, uma vez que ela efetua uma suavização mais fraca em determinados pontos e mais forte em outros. Isso permite que um ponto de saliência verdadeiro não seja eliminado. A principal vantagem dessa nova abordagem de detecção é que ela é robusta a ruídos e evita que o usuário tenha que definir vários parâmetros.

A metodologia do detector proposto é a seguinte: seja $B = \{p_1, \dots, p_{n_c}\}$ o conjunto formado pelos pontos do contorno de uma forma, ordenados percorrendo o contorno no sentido horário (ou anti-horário), onde $p_i = (x_i, y_i)$, para $i = 1, \dots, n_c$, n_c é o número de

pontos do contorno e x_i e y_i representam as coordenadas do ponto p_i na imagem. Para detectar os pontos de saliências, utilizando a técnica proposta, três fases são necessárias:

1. Primeira fase consiste em calcular o conjunto $\tilde{B} = \{\Psi(p_1), \dots, \Psi(p_{n_c})\}$, onde $\Psi(p_i)_{i=1\dots n_c}$ representa o valor da curvatura do ponto p_i .
2. No segundo passo, com o objetivo de reduzir os altos valores de curvatura dos pontos ruidosos, é aplicada uma suavização no conjunto $\tilde{B}$ usando um filtro de suavização anisotrópica;
3. Depois de suavizado, a terceira fase consiste em localizar em $\tilde{B}$ os pontos de máximo e mínimo locais que serão os pontos de saliência.

Nas próximas seções, essas três fases serão discutidas e os detalhes de implementação serão descritos.

4.2 Cálculo da Curvatura

O conceito de curvatura tem uma forte motivação biológica, pois é um importante aspecto analisado pelo sistema visual humano [Attneave 1954]. O cálculo da curvatura tem sido amplamente usado em diferentes aplicações na área de processamento de imagens, por exemplo na representação de formas [Park e Han 1998], extração de características [Lim et al. 1995] e reconhecimento de objetos [Bandera et al. 1999].

Seja $\Gamma(s) = (x(s), y(s))$ uma curva, a curvatura é definida como o valor da mudança de inclinação da curva em relação ao seu comprimento s . O ângulo de inclinação de uma curva é dado pela equação:

$$\theta(s) = \arctan\left(\frac{dy(s)}{dx(s)}\right) \quad (4.1)$$

A curvatura é calculada sobre a taxa de variação do ângulo de inclinação $\theta(s)$ pelo comprimento da curva:

$$\Psi(s) = \frac{d\theta(s)}{ds} = \frac{\dot{x}(s)\ddot{y}(s) - \dot{y}(s)\ddot{x}(s)}{(\dot{x}(s)^2 + \dot{y}(s)^2)^{3/2}} \quad (4.2)$$

onde $\dot{x}$ e $\dot{y}$ representam a primeira derivada em relação a x e y respectivamente; e $\ddot{x}$ e $\ddot{y}$ representam a segunda derivada em relação a x e y respectivamente.

4.2.1 Discretização usando K-Curvatura

As derivadas da equação de curvatura 4.2 podem ser aproximadas por diferenças finitas. Dado um ponto $p_i = (x_i, y_i)$ e seus vizinhos $p_{i+1} = (x_{i+1}, y_{i+1})$ e $p_{i-1} = (x_{i-1}, y_{i-1})$,

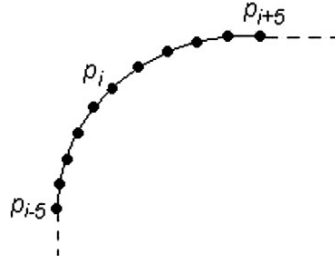


Figura 4.2: Os 5-vizinhos p_{i+5} e p_{i-5} do ponto p_i .

a discretização da equação é dada por:

$$\Psi(p_i) = \frac{(x_{i+1} - x_{i-1})(y_{i+1} - 2y_i + y_{i-1}) - (y_{i+1} - y_{i-1})(x_{i+1} - 2x_i + x_{i-1})}{((x_{i+1} - x_{i-1})^2 + (y_{i+1} - y_{i-1})^2)^{3/2}} \quad (4.3)$$

Alguns trabalhos, como as técnicas de [Rosenfeld e Johnston 1973, Rosenfeld e Weszka 1975, Arica e Vural 2003], introduzem uma “suavização” no cálculo da curvatura. Esse procedimento é obtido pelo método conhecido como K-curvatura, que é definido como a taxa de inclinação do ângulo entre um ponto e seus K-ésimos vizinhos. Como exemplo ilustrativo, a figura 4.2 mostra os 5-vizinhos de um ponto p_i .

Porém, a escolha do valor do parâmetro K interfere na resolução da representação da curvatura de um contorno. A figura 4.3-(b,c,d,e) apresenta os gráficos da K-curvatura, para diferentes valores de K, referente à silhueta do peixe dado na figura 4.3-a. Os picos e vales dos gráficos correspondem, respectivamente, a convexidades e concavidades do contorno. Como claramente pode-se ver, quando K aumenta mais informações sobre o contorno da forma são perdidos. Por outro lado, quando K diminui detalhes finos são capturados, inclusive algumas convexidades e concavidades redundantes do contorno. Um valor “ideal” para K é aquele capaz de capturar somente as informações necessárias para a representação do contorno da forma, excluindo as informações redundantes como os ruídos. Entretanto, encontrar esse valor “ideal” para K, válido para todos os tipos de silhuetas, não é uma tarefa fácil. Diferentes trabalhos usam diferentes valores, por exemplo, [Arica e Vural 2003] indicam que um valor apropriado para K seria 10% do valor total de pontos do contorno, porém, a figura 4.3-b é um bom exemplo que ilustra que, a escolha desse valor, não consegue capturar importantes convexidades e concavidades do contorno. Outras técnicas, como o detector apresentado por [Rosenfeld e Johnston 1973], atribui ao usuário a tarefa de escolha do parâmetro K.

Para evitar a escolha de um valor para K, a idéia do detector proposto neste trabalho é usar um valor suficientemente pequeno para K, por exemplo $K = 1$, desse modo todas as convexidades e concavidades do contorno da forma serão capturadas, inclusive informações indesejadas como os ruídos. Então, uma função de suavização adaptativa seria aplicada para eliminar os picos e vales redundantes da curvatura, preservando os picos e vales significativos. Esse procedimento torna a detecção dos pontos de saliência mais eficiente.

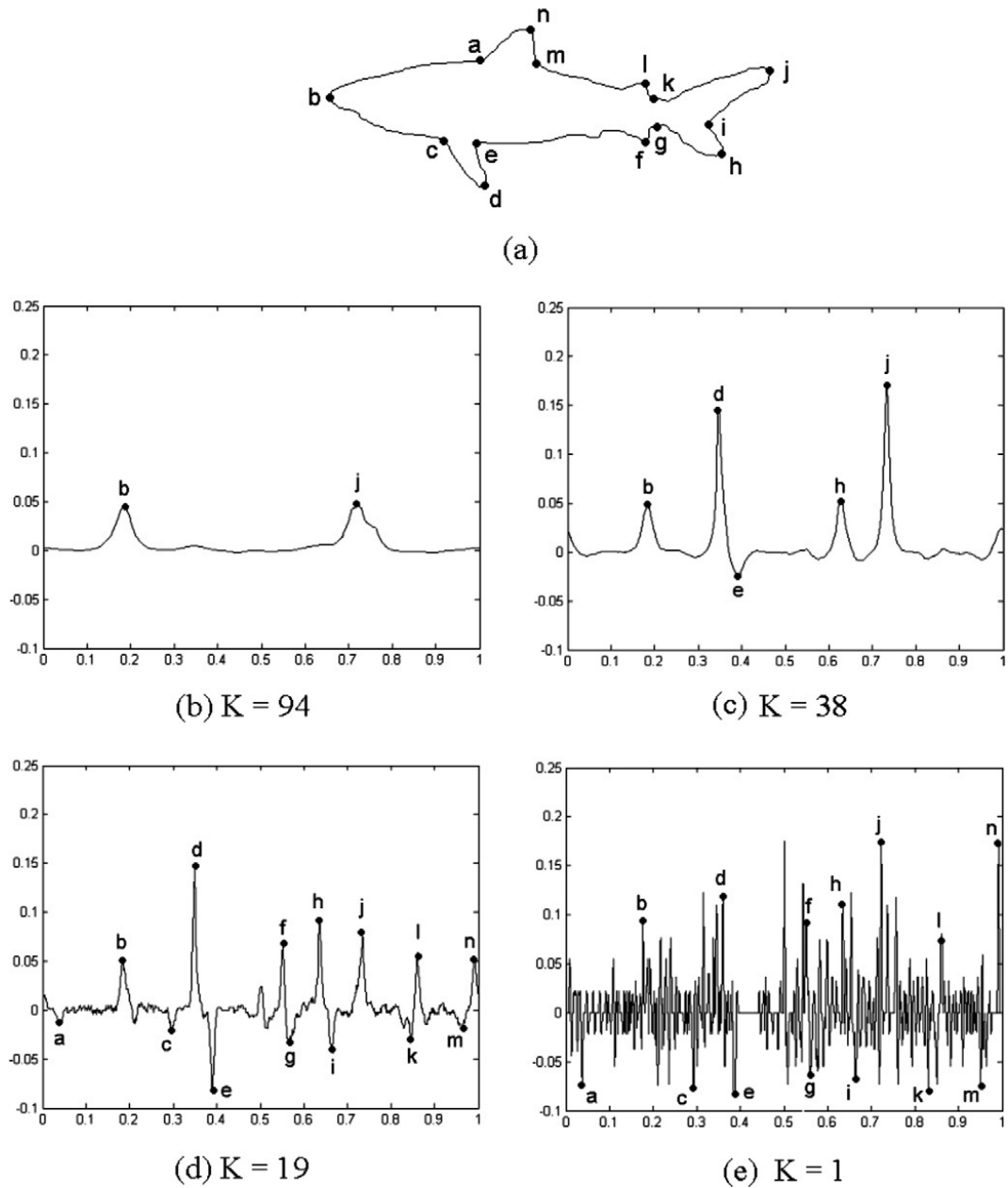


Figura 4.3: Gráficos mostrando a representação do contorno da figura (a) para diferentes valores de K no cálculo da curvatura. A figura (a) tem 937 pontos de contorno.

4.3 Difusão Anisotrópica Não-Linear

O uso de equações diferenciais parciais (EDPs) no processamento de imagens tem crescido significativamente nos últimos anos. A idéia básica é deformar uma imagem, curva ou superfície com uma equação diferencial parcial e obter os resultados esperados como uma solução para essa equação. Uma das principais vantagens do uso de EDPs é que elas transformam a análise de uma imagem em um domínio contínuo, o que simplifica o formalismo.

As EDPs tem sido bastante utilizadas como filtros. Um importante filtro baseado em uma EDP com preservação de bordas foi introduzido por [Perona e Malik 1990].

Eles formularam um filtro anisotrópico que é um processo de difusão que preserva as regiões que tem mudança brusca de valores (bordas). Matematicamente, dada uma função $I \subset \mathbb{R}^p \rightarrow \mathbb{R}$, $p = 1, 2$ ou 3 , o filtro anisotrópico é definido como:

$$\begin{aligned} u_t &= \text{div}(g(|\nabla u|)\nabla u) \\ u_0 &= I \end{aligned} \quad (4.4)$$

onde I é a imagem ruidosa, g é uma função tal que $g(0) = 1$, $g(s) \geq 0$, e $g(s) \rightarrow 0$ quando $s \rightarrow \infty$. A idéia é que se $|\nabla u|$ é grande, então a difusão será pequena e, portanto, a localização de um ponto onde há mudança brusca de valor será preservada. Portanto, a função g permite uma difusão localizada adaptativa.

Qualquer função contínua e decrescente bastaria como uma função de difusão adaptativa. [Perona e Malik 1990] sugerem duas opções para a função g :

$$g(s) = \exp(-(\frac{s}{\rho})^2) \quad (4.5)$$

$$g(s) = \frac{1}{1 + (\frac{s}{\rho})^2} \quad (4.6)$$

onde $s = |\nabla u|$. O parâmetro ρ refere-se à uma constante da difusão. Teoricamente ele é usado para tornar $g \sim 0$ em pontos de mudança brusca de valor e $g \sim 1$ nos demais. Ele depende das características da imagem e seu objetivo é acelerar ou reduzir a velocidade de suavização de uma maneira anisotrópica. Neste trabalho, o valor de ρ foi fixado em 1.

A equação anisotrópica 4.4 será aplicada na curva discreta dada pelo conjunto $\tilde{B} = \{\Psi(p_1), \dots, \Psi(p_{n_c})\}$, onde $\Psi(p_i)_{i=1\dots n_c}$ representa o valor da curvatura do ponto p_i calculado na fase anterior. Usando aproximação por Euler e diferenças finitas, a discretização da equação 4.4 é dada por:

$$\frac{u_{t+1} - u_t}{\Delta t} = \text{div}(g(|\nabla u_t|)\nabla u_t) \quad (4.7)$$

Aplicando essa equação em cada ponto de $\tilde{B}$, tem-se:

$$\frac{\beta_{t+1} - \beta_t}{\Delta t} = (g(\beta_{i,t}^+), g(\beta_{i,t}^-)) \cdot (\beta_{i,t}^+, \beta_{i,t}^-) \quad (4.8)$$

ou

$$\begin{aligned} \beta_{i,t+1} &= \beta_{i,t} + \Delta t (g(\beta_{i,t}^+) \beta_{i,t}^+ + g(\beta_{i,t}^-) \beta_{i,t}^-) \\ \beta_{i,0} &= \Psi(p_i) \end{aligned} \quad (4.9)$$

onde t é o passo de iteração, $i = 1\dots n_c$, n_c é o número de pontos do contorno, Δt é a variação temporal, g é a função de difusão adaptativa, $\beta_{i,t}^+ = (\beta_{i+1,t} - \beta_{i,t})$ e $\beta_{i,t}^- = (\beta_{i-1,t} - \beta_{i,t})$.

Como resultado, tem-se um novo conjunto $\tilde{B}^* = \{\Psi^*(p_1), \dots, \Psi^*(p_{n_c})\}$, onde $\Psi^*(p_i) =$

$\beta_{i,m}$ e m é o numero de iterações.

4.4 Localização das Saliências

A próxima etapa é detectar os pontos salientes remanescentes depois da fase de suavização. Esses pontos serão os pontos de máximo e mínimo locais. Um ponto de máximo local satisfaz as seguintes condições: $(\Psi^*(p_i) - \Psi^*(p_{i+1})) > 0$ e $(\Psi^*(p_i) - \Psi^*(p_{i-1})) > 0$. E um ponto de mínimo local as condições: $(\Psi^*(p_i) - \Psi^*(p_{i+1})) < 0$ e $(\Psi^*(p_i) - \Psi^*(p_{i-1})) < 0$.

Entretanto, nem todo os pontos de máximo e mínimo locais são, visualmente, pontos de saliência no contorno da forma. Nota-se que sempre entre dois pontos de máximo local existe um ponto de mínimo. Da mesma maneira que entre dois pontos de mínimo existe um ponto de máximo. Sendo assim, para evitar a detecção de um ponto que não é visualmente uma saliência, algumas outras restrições devem ser adicionadas: um ponto de máximo local somente será considerado uma saliência se o seu valor de curvatura for positivo e um ponto de mínimo local será considerado saliência se seu valor de curvatura for negativa. Teoricamente, essas condições se baseiam no fato de que saliências convexas tem valor de curvatura positiva, enquanto que saliências côncavas tem curvatura negativa. Para ilustrar esse fato, os pontos b, d e f da figura 4.4-a são pontos de máximo locais na curvatura suavizada mostrada na figura 4.4-b, porém eles tem valores de curvatura negativos. Com essa duas novas restrições impostas, esses pontos não serão considerados saliências.

Em resumo, um ponto p_i será uma saliência convexa se satisfazer as seguintes condições:

$$1a) \Psi^*(p_i) > 0$$

$$2a) (\Psi^*(p_i) - \Psi^*(p_{i-1})) > 0$$

$$3a) (\Psi^*(p_i) - \Psi^*(p_{i+1})) > 0$$

De modo semelhante, para o ponto p_i ser uma saliência côncava ele deve satisfazer as condições:

$$1b) \Psi^*(p_i) < 0$$

$$2b) (\Psi^*(p_i) - \Psi^*(p_{i-1})) < 0$$

$$3b) (\Psi^*(p_i) - \Psi^*(p_{i+1})) < 0$$

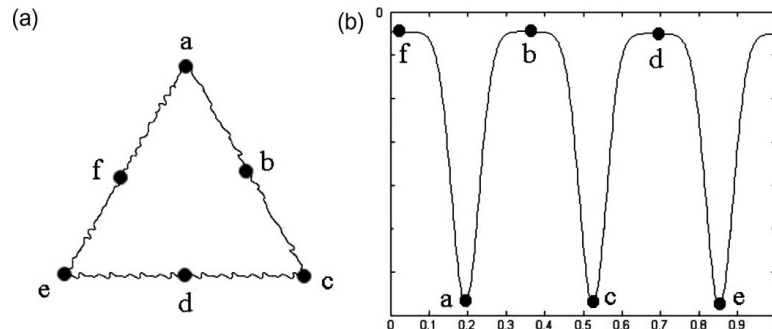


Figura 4.4: Uma curva e seus pontos de máximo e mínimo locais.

4.5 Considerações Finais

Neste capítulo foi apresentado um novo método de detectar pontos de saliência em formas. Esse método é baseado na suavização dos valores de curvatura dos pontos do contorno através de um filtro anisotrópico não-linear.

A nova abordagem de detecção de saliências, proposta neste capítulo, permite localizar esses pontos mesmo em contorno com alto nível de ruído. O método é simples e o único parâmetro a ser definido é o número de iteração para a função de suavização. Essa suavização é usada para eliminar a influência de ruídos e remover pequenos detalhes presentes no contorno da forma. Aumentando o número de iterações da função de suavização, saliências menos relevantes serão suprimidas.

A performance desse novo detector proposto será investigado no capítulo 6. Comparações serão feitas com outros detectores, inclusive com o método utilizado pelo descritor CS.

No próximo capítulo será detalhado o método utilizado para descrever e analisar a similaridade de formas com base nos pontos de saliência detectados usando o modelo proposto neste capítulo.

Capítulo 5

Representação dos Pontos de Saliência e Similaridade

5.1 Introdução

Após a fase de detecção da localização dos pontos de saliência, precisa-se determinar o valor representativo da saliência nestes pontos. O desafio é encontrar um modo de representação que seja invariante a transformações geométricas e robusto a possíveis ruídos presentes no contorno.

A abordagem adotada nesse trabalho para caracterizar uma saliência é analisar o segmento do contorno em que ela se localiza. Sendo assim, de maneira intuitiva, uma saliência será “mais relevante” quanto maior for a concavidade (ou convexidade) do segmento de contorno em que ela se encontra. A figura 5.1 exemplifica esse fato: o segmento de contorno da figura 5.1(a) é visualmente mais “chamativo” que o segmento da figura 5.1(b), portanto, a representação da saliência em um segmento de contorno mais fechado deve ter uma caracterização mais relevante que um segmento mais aberto.

Além disso, outro aspecto importante que deve ser considerado na representação de uma saliência é a sua localização espacial no contorno da forma. Essa informação é visualmente relevante quando se têm duas formas com o mesmo número de saliências. Para exemplificar, a figura 5.2 mostra duas silhuetas (uma mão e uma estrela) com o mesmo número de saliências, porém na estrela as saliências estão “igualmente” espaçadas,

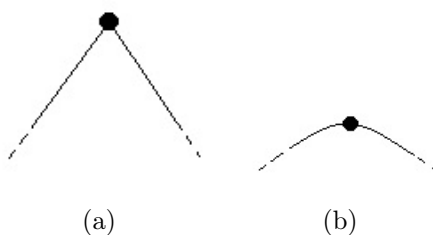


Figura 5.1: Saliência sobre um segmento de contorno: (a) mais fechado (b) mais suave

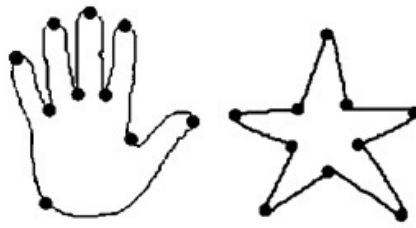


Figura 5.2: Duas formas com o mesmo número de saliências, porém visualmente diferentes.

enquanto que na mão as saliências se concentram em uma região do contorno (dedos). Nesse caso, a relação espacial das saliências fornecem uma informação importante para diferenciar essas duas formas.

Depois de representar cada saliência, a análise de similaridade entre as formas é dada através de uma função de distância, a qual irá mensurar a semelhança entre pares de formas. No entanto, as funções de distância tradicionais, tal como a distância Euclidiana, não é viável de ser utilizada, uma vez que diferentes silhuetas podem ter números de saliências diferentes, portanto é necessário que a função de distância seja capaz de comparar vetores com tamanho diferentes.

Neste capítulo, são apresentados os métodos usados para representar as saliências e medir a similaridade entre duas formas. Essas duas novas propostas, que serão detalhadas nas próximas seções, permitem caracterizar saliências e analisar a similaridade entre duas formas com uma abordagem robusta a ruídos, invariante à transformações geométricas e independente do número de saliências que formas distintas possam ter.

5.2 Caracterização dos Pontos de Saliência

O descritor proposto por [Torres e Falcão 2007] representa cada ponto de saliência s_i por um par $(s_{1,i}, s_{2,i})$, onde $s_{1,i}$ representa a posição relativa da saliência no contorno e $s_{2,i}$ quantifica a relevância da saliência através de sua área de influência na forma. Seguindo essa idéia, neste trabalho cada saliência s_i também será representada por um par $(s_{1,i}, s_{2,i})$, porém a proposta é utilizar outra abordagem para caracterizar os elementos desse par, de tal modo que a representação seja menos sensível a ruídos do que a proposta original.

Para representar a posição relativa $s_{1,i}$, neste trabalho adota-se a relação angular entre o ponto de saliência e o centróide da forma, e para caracterizar a relevância visual $s_{2,i}$ será utilizada a informação sobre a curvatura do segmento de contorno onde a saliência se localiza. Essas duas novas propostas de representação, que serão discutidas nas próximas subseções, conseguem caracterizar uma saliência com uma abordagem robusta a ruídos presentes no contorno e invariante a transformações de escala.

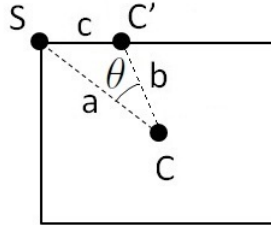


Figura 5.3: Posição angular da saliência S em relação ao centróide C e um outro ponto C' qualquer escolhido arbitrariamente.

5.2.1 Posição Relativa Angular

Uma abordagem, invariante a escala e ruídos presentes no contorno, é usar o centróide da forma como um ponto de referência. A proposta é usar o ângulo entre a saliência, o centróide da forma e um outro ponto fixo qualquer do contorno para representar a posição relativa de uma saliência.

Seja S a saliência, C o centróide da forma e C' um ponto fixo qualquer, o cálculo do ângulo entre esses três pontos é baseado na lei dos cossenos para triângulos quaisquer, nesse caso o triângulo é dado por SCC' . Assumindo que a é a medida do segmento $\overline{SC}$, b a medida do segmento $\overline{C'C}$ e c a medida do segmento $\overline{SC'}$ (ver figura 5.3), a lei dos cossenos garante que:

$$c^2 = a^2 + b^2 - 2ab \cos \theta \quad (5.1)$$

Assim, tem-se que o ângulo θ é dado por:

$$\theta = \arccos\left(\frac{a^2 + b^2 - c^2}{2ab}\right) \quad (5.2)$$

Esse método de representar a posição relativa das saliências é invariante a escala e ruídos no contorno, porém sofre com o problema de rotação, uma vez que a escolha do ponto arbitrário C' pode ser diferente em cada situação.

Para normalizar os valores, as saliências são ordenadas percorrendo o contorno no sentido horário e os valores são divididos pelo ângulo da última saliência. Essa normalização sofre com o problema de espelhamento, ou seja, duas formas idênticas porém invertidas terão a ordenação de suas saliências diferentes, já que a ordenação sempre ocorre no sentido horário.

A medida de similaridade deve levar em consideração as propriedades de rotação e espelhamento no momento de comparar as saliências, a fim de reconhecer duas formas invariantes a essas duas propriedades.

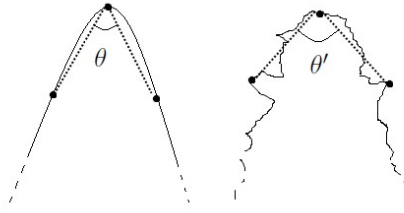


Figura 5.4: Dois segmentos de contorno semelhantes, sem e com ruído. Os pontos representam a vizinhança considerada no cálculo da curvatura.

5.2.2 Relevância de uma Saliência

O conceito de relevância adotada neste trabalho para caracterizar uma saliência consiste em analisar o segmento de contorno em que ela se localiza. Para quantificar esse valor de relevância, calcula-se a concavidade (ou convexidade) do segmento através da função curvatura (Eq. 4.2).

Como explicado e discutido na seção 4.2.1, a função curvatura (Eq. 4.2) pode ser discretizada usando diferentes valores para K , onde K representa a vizinhança considerada no cálculo da curvatura. Para valores pequenos de K , por exemplo $K = 1$, a curvatura não consegue capturar informações globais relevantes sobre o segmento de contorno onde se localiza a saliência, e para K muito grande as informações locais importantes são perdidas.

Na tentativa de capturar aspectos locais e globais, a proposta deste trabalho é usar cinco valores diferentes para K no cálculo da curvatura. Os valores de K utilizados são proporcionais ao número de pontos do contorno e múltiplos de 5. Em outras palavras, se n_c for o número de pontos do contorno, os valores usados são $K = \{\frac{n_c}{5}, \frac{n_c}{10}, \frac{n_c}{15}, \frac{n_c}{20}, \frac{n_c}{25}\}$. Para representar a saliência, a média da curvatura para esses cinco valores de K é adotada.

Porém, deve-se levar em consideração de que ruídos no contorno podem prejudicar a análise da informação da concavidade (convexidade) do segmento de contorno onde está localizada a saliência. A figura 5.4 ilustra essa situação: a concavidade do contorno com ruído é maior que a concavidade do contorno sem ruído ($\theta' > \theta$), sendo assim as duas saliências sobre esses segmentos terão valor de curvatura muito diferente, o que não é o ideal.

Para evitar que o ruído penalize o cálculo da curvatura, é proposto um processo de suavização simples no contorno da forma, via um filtro gaussiano, antes de se calcular a curvatura de cada saliência. A figura 5.5 mostra o contorno de uma forma sendo suavizado: essa suavização permite que, a representação das saliências de uma forma com contorno ruidoso, seja semelhante à uma forma com um contorno sem ruídos.

A figura 5.6 ilustra os vetores de características de duas formas visualmente semelhantes, porém uma forma com contorno ruidoso e outra sem ruído. As figuras 5.6(a) e 5.6(b) mostram estes objetos e seus pontos de saliências, enquanto que os gráficos das figuras 5.6(c) e 5.6(d) mostram os valores das saliências usando a representação proposta neste

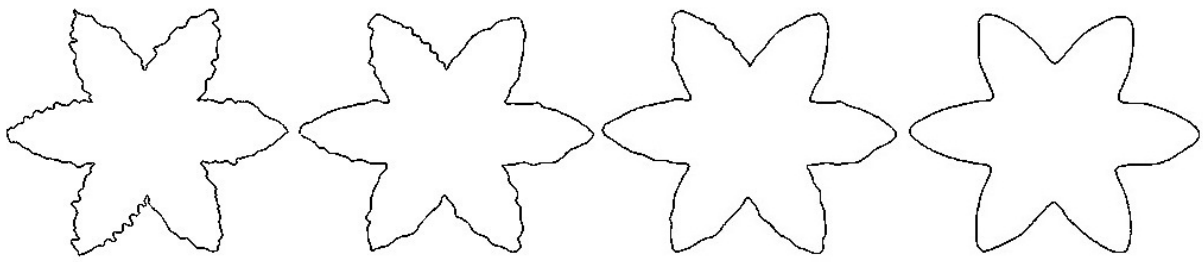


Figura 5.5: Contorno de uma forma sendo suavizado.

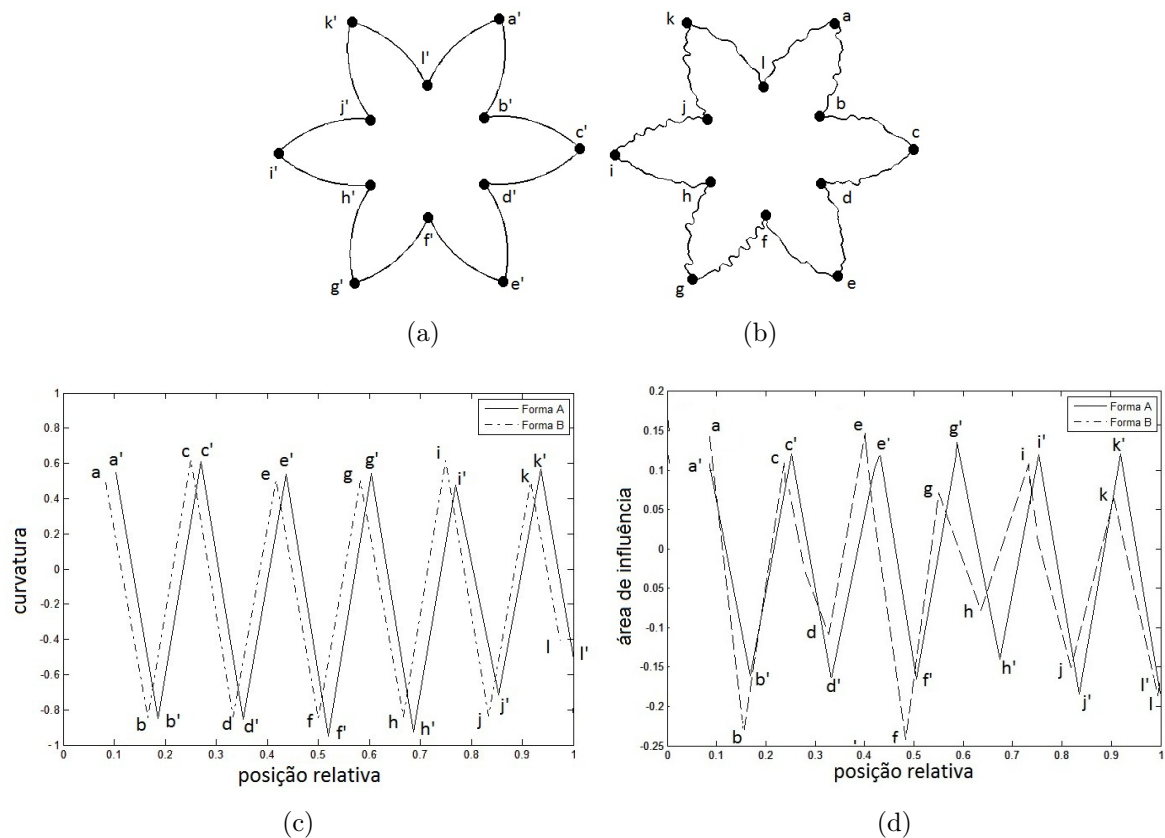


Figura 5.6: Representação gráfica dos valores das saliências das formas da figuras (a) e (b) usando: (c) a representação proposta neste trabalho e (d) a representação proposta por [Torres e Falcão 2007].

trabalho e a representação proposta por [Torres e Falcão 2007]. Nota-se que os valores das saliências, usando a representação proposta, são praticamente semelhantes, apenas deslocados. Já para a representação proposta por [Torres e Falcão 2007], os vetores de características para essas duas formas são bem diferentes, o que não é o ideal.

5.3 Análise da Similaridade

Uma vez que as saliências de uma forma foram localizadas e representadas em vetores, a análise de similaridade entre duas formas A e B consiste na comparação entre seus pares de vetores de características através de uma função de distância. Na literatura pode-se

encontrar diversas métricas para tal fim, como explorado na seção 2.5.3, porém para o domínio em questão deve-se levar em conta que diferentes silhuetas podem ter vetores de características com dimensões diferentes. Nesse caso uma distância elástica deve ser utilizada para analisar a similaridade entre formas caracterizadas por suas saliências. No contexto desse trabalho, o termo “elástica” refere-se à uma medida capaz de comparar vetores com dimensões diferentes.

A medida de similaridade elástica, proposta por [Torres e Falcão 2007] para o descritor *Contour Saliences* (CS), é baseada em um algoritmo que registra os vetores em relação a um ponto de referência e computa a similaridade. Esta métrica é baseada no algoritmo proposto por [Mokhtarian e Mackworth 1986] para o descritor *Curvature Scale Space* (CSS) e foi descrito na seção 3.4.7 do Capítulo 3. Essa medida de similaridade é baseada em um método heurístico de casamento, ou seja, uma aproximação da solução ideal, e que nem sempre é a melhor maneira de medir a semelhança entre formas caracterizadas por suas saliências.

Neste trabalho é apresentada uma função de distância $d_s(A, B)$, que analisa a similaridade entre as saliências das formas A e B através da minimização de um funcional que busca a melhor correspondência entre as saliências de ambas as formas. Essa métrica é invariante a rotação e espelhamento entre os vetores de características e pode ser computacionalmente implementada usando programação dinâmica.

Para aumentar o poder discriminativo da função de distância $d_s(A, B)$, ela é combinada com uma função $d_g(A, B)$ que mede a distância entre as características geométricas globais das formas A e B . A inclusão dessa informação consegue aprimorar, ainda mais, a eficiência da métrica em reconhecer formas em banco de dados complexos, tal como explorado por [Jain e Vailaya 1998] e [Abbasi et al. 1999]. As características geométricas utilizadas são: relação de aspecto (RA), excentricidade (E) e solidez (S). Segundo [Adamek e Connor 2004] essas três características incluem informações suficientes sobre as propriedades globais de uma forma. Portanto, a função $d_g(A, B)$, proposta por [Adamek e Connor 2004], é dada por:

$$d_g(A, B) = |RA_A - RA_B| + |E_A - E_B| + |S_A - S_B| \quad (5.3)$$

onde RA_S , E_S , e S_S são relação de aspecto, excentricidade e solidez da forma S .

Formalmente, a função de distância proposta para analisar a similaridade entre duas formas A e B é dada por:

$$D_f(A, B) = d_g(A, B) + \frac{d_s(A, B)}{c_f(A, B)} \quad (5.4)$$

onde $d_g(A, B)$ é a distância entre as características globais das formas A e B , $d_s(A, B)$ é a distância entre as saliências das formas A e B e $c_f(A, B)$ é uma função que mede a complexidade entre as formas A e B . Essa função é usada para penalizar a distância

$d_s(A, B)$ de tal maneira que para formas muito complexas $c_f(A, B) \rightarrow \infty$, então $\frac{d_s(A, B)}{c_f(A, B)} \rightarrow 0$.

Neste trabalho, a complexidade de uma forma $\mathcal{S}$ é considerada como a diferença absoluta entre o valor máximo e mínimo da curvatura medida em todas as saliências da forma, isto é:

$$CF(\mathcal{S}) = |\mathcal{S}_{ma} - \mathcal{S}_{mi}| \quad (5.5)$$

onde $\mathcal{S} = \{(s_{1,1}, s_{2,1}), \dots, (s_{1,n}, s_{2,n})\}$, n é o número de saliências, $\mathcal{S}_{ma} = \max_{1 \leq i \leq n} (s_{2,i})$ e $\mathcal{S}_{mi} = \min_{1 \leq i \leq n} (s_{2,i})$.

Assim, a função $c_f(A, B)$ que mede a complexidade entre duas formas A e B é dada por:

$$c_f(A, B) = \kappa + CF(A) + CF(B) \quad (5.6)$$

onde $CF(A)$ e $CF(B)$ são as complexidades das formas A e B e κ é uma constante adicionada para evitar a divisão por número muito pequenos. Nos experimentos, foi adotado o valor $\kappa = 1$.

5.3.1 Similaridade entre Saliências

O objetivo da função de similaridade $d_s(A, B)$ é medir a distância entre duas formas A e B representadas por seus pontos de saliências: $A = \{a_1, \dots, a_n\}$ e $B = \{b_1, \dots, b_m\}$, onde n e m representam o número de saliências da forma A e B respectivamente. Cada ponto a_i é uma dupla $(a_{1,i}, a_{2,i})$, onde $a_{1,i} \in [0, 1]$ representa a posição angular e $a_{2,i} \in [-1, 1]$ representa o valor de curvatura da i -ésima saliência da forma A (mesmo caso para a forma B).

Considerando que ambos os vetores A e B estão ordenados pela posição relativa das saliências e supondo, sem perda de generalidade, que $m \leq n$, a idéia da similaridade é encontrar a melhor correspondência entre as saliências de A e B . Formalmente, essa correspondência é definida por uma função $\varphi : \{1, \dots, n\} \rightarrow \{1, \dots, m\}$, tal que $\varphi(i) \leq \varphi(i+1)$, para $i = 1 \dots n-1$, e a_i é mapeada em $b_{\varphi(i)}$ para $i \in \{1, \dots, n\}$. A distância entre A e B é calculada por:

$$D^{A,B}(\varphi) = \sum_{i=1}^n \|a_i - b_{\varphi(i)}\| \quad (5.7)$$

onde $\|a_i - b_{\varphi(i)}\| = |a_{1,i} - b_{1,\varphi(i)}| + |a_{2,i} - b_{2,\varphi(i)}|$.

Entretanto, para tornar a função de distância 5.7 invariante à rotação, ela deve ser calculada sobre todos os possíveis deslocamentos entre os vetores A e B . O valor do deslocamento é definido como a diferença entre o primeiro ponto do vetor B sobre todos

os pontos do vetor A , ou seja, $t_k = (b_{1,1} - a_{1,k})$ para $k = 1, \dots, n$. O deslocamento consiste em somar t_k nas posições relativas $a_{1,i}$ do vetor A . Sendo assim, para cada deslocamento um novo vetor $A^k = \{a_1^k, \dots, a_n^k\}$ é gerado, onde $a_i^k = (a_{1,i}^k, a_{2,i}^k)$, $a_{1,i}^k = ((a_{1,i} + t_k) \bmod 1)$ e $a_{2,i}^k = a_{2,i}$. Formalmente, a função de distância para o vetor deslocado A^k é dada por:

$$D^{A^k, B}(\varphi) = \sum_{i=1}^n \|a_i^k - b_{\varphi(i)}\| \quad (5.8)$$

Portanto, para a similaridade ser invariante a rotação, o objetivo é encontrar o melhor deslocamento t_k tal que a distância $D^{A^k, B}(\varphi)$ seja mínima.

Para a função de distância ser invariante a espelhamento, basta inverter a posição dos elementos $a_{2,i}$ do vetor A , ou seja, obter um novo vetor $A_I = \{a_{I_1}, \dots, a_{I_n}\}$, onde $a_{I_i} = (a_{1,i}, a_{2,(n+1)-i})$.

A distância final entre A e B é dada pela menor distância obtida usando o vetor original para representar a forma A e seu vetor invertido, considerando todos os possíveis deslocamentos. Em outras palavras, a distância final é dada por:

$$d_s(A, B) = \min\{\min_k(D^{A^k, B}(\varphi)), \min_k(D^{A_I^k, B}(\varphi))\} \quad (5.9)$$

Implementação Computacional

O problema de encontrar a função de correspondência φ , na equação 5.7, pode ser feito através de um algoritmo baseado em programação dinâmica. Para isso, primeiro é definido uma matriz de diferenças $r : n \times m$ usando todos os elementos dos vetores A e B :

$$r = (r_{ij}) = \|a_i - b_j\| \quad (5.10)$$

Assim, o problema de minimização pode ser visto como um problema de encontrar o caminho de custo mínimo sobre essa matriz de diferenças. Para obter a solução, a matriz r é tratada como um grafo acíclico direcionado, onde um elemento $r_{i,j}$ é mapeado em $r_{k,l}$, se e somente se, $i \leq k \leq i+1$ e $j \leq l \leq j+1$. O caminho de custo mínimo por ser obtido através da seguinte função recursiva:

$$M(i, j) = \begin{cases} r_{ij}, & \text{se } i=1 \text{ e } j=1; \\ r_{ij} + M(i, j-1), & \text{se } i=1 \text{ e } j>1; \\ r_{ij} + M(i-1, j), & \text{se } i>1 \text{ e } j=1; \\ r_{ij} + \min(M(i-1, j), M(i-1, j-1), \\ M(i, j-1)), & \text{se } 1<i \leq n \text{ e } 1<j \leq m; \\ \infty, & \text{caso contrário.} \end{cases} \quad (5.11)$$

A distância é obtida em $M(n, m)$.

Para a medida de similaridade ser invariante a rotação, a distância deve ser recalculada sobre todos os possíveis valores de deslocamentos t_k , onde $k = 1, \dots, n$. Para isso, a posição relativa de todos os elementos do vetor A são deslocadas por t_k de maneira circular e re-ordenados. A matriz de diferenças 5.10 é novamente construída usando esse vetor deslocado, e um novo caminho de custo mínimo é calculado pela equação 5.11. A distância final considerada é a menor distância sobre os n possíveis deslocamentos

Além disso, para tornar a medida de similaridade invariante à espelhamento, o valor de curvatura dos elementos do vetor A são invertidos de posição, ou seja, o valor de curvatura do último elemento se torna o primeiro, o penúltimo o segundo, e assim por diante. A distância é novamente calculada usando o vetor A invertido, e a menor distância entre o vetor invertido e o vetor original é considerada como a distância final entre as formas A e B .

5.4 Considerações Finais

Neste capítulo foi apresentado um novo método para representar as saliências e medir a similaridade entre duas formas. Essas novas propostas fornecem um meio eficiente de caracterizar e reconhecer formas de maneira invariante à escala, translação, rotação, espelhamento e robutez à ruídos presentes no contorno.

Para representar uma saliência, foram utilizadas duas informações: a sua posição angular em relação ao centróide e seu valor de curvatura. Essas duas informações permitem caracterizar a saliência de uma maneira intuitiva à percepção humana.

A função de similaridade é uma medida elástica que permite comparar dois vetores de tamanho diferentes. A idéia é encontrar a melhor correspondência entre as saliências das duas formas. Uma vez encontrada essa correspondência, a distância é obtida através da diferença entre os valores de cada ponto. Para tornar a distância invariante à rotação e espelhamento, é realizado um deslocamento e inversão de um dos vetores que estão sendo comparados.

Para aumentar o poder de discriminação da função de similaridade, duas normalizações são realizadas: a primeira é dividir a distância pela complexidade das formas e a segunda é usar um conjunto de características geométricas que capturem as informações globais de cada forma, a fim de descartar formas bem dissimilares.

No próximo capítulo, essas novas propostas são investigadas na análise de similaridade de formas, usando um banco de dados heterogêneo e comparadas com outros descritores presentes na literatura.

Capítulo 6

Resultados Experimentais

6.1 Introdução

Neste capítulo, a performance das técnicas propostas de detecção, representação e similaridade de saliências são avaliadas e comparadas com outros métodos existentes na literatura. Os experimentos foram divididos em duas partes: detecção de saliências e recuperação de formas.

O primeiro experimento consiste em analisar a robustez do detector de saliências proposto no capítulo 4, explorando o desempenho da técnica na detecção de saliências em contornos com e sem ruídos enquanto comparado com outros métodos.

O segundo experimento investiga a combinação das técnicas de detecção, representação e similaridade, propostas nos capítulos 4 e 5, no reconhecimento de formas caracterizadas por seus pontos de saliência. A combinação dessas propostas constituem o descritor SSD. Para comparação, três diferentes bancos de imagens foram utilizados e comparados com seis outras diferentes abordagens de descrição de formas encontradas na literatura.

6.2 Detecção de Saliências

O objetivo dessa seção é analisar a performance da técnica proposta no capítulo 4 em detectar saliências em formas com contorno ruidoso e sem ruídos e compará-la com outros detectores encontrados na literatura. O experimento consiste em contar o número de pontos detectados e medir a proporção de acerto e erros.

A tarefa de comparar resultados não é fácil, uma vez que classificar se um dado ponto de uma forma é uma saliência, depende da aplicação ou do ponto de vista do usuário, pois o que é ponto saliente para um usuário pode não ser para o outro. Sendo assim, para realizar os experimentos uma base de imagens foi criada com os pontos de saliências selecionados manualmente por um humano.

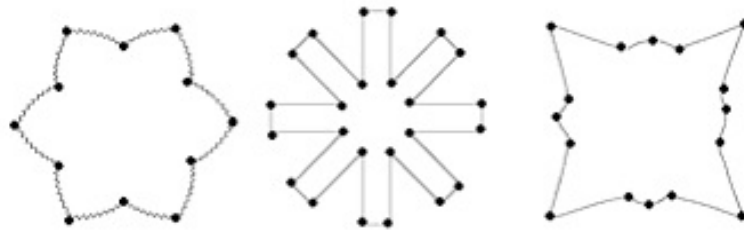


Figura 6.1: Algumas imagens que compõe a base usada nos experimentos.

6.2.1 Base de Teste

Para uma efetiva comparação, foi construído um banco de imagens composto por 120 diferentes silhuetas de objetos com e sem ruídos. Essas imagens foram retiradas da classe *device* do banco de dados MPEG-7 Part B [Bober 2001].

Manualmente, as posições dos pontos de saliência foram marcadas, resultando em um total de 1853 saliências (considerando todas as imagens). A figura 6.1 mostra algumas formas que compõe o banco de dados com suas respectivas saliências selecionadas manualmente.

6.2.2 Detectores Avaliados

O método proposto foi comparado com duas diferentes abordagens de detecção de saliências:

- a técnica baseada em esqueleto proposta por [Torres et al. 2004];
- e a técnica baseada em K-vizinhos proposta por [Chetverikov 2003].

Essas duas técnicas fazem uso de diferentes metodologias na detecção de pontos de saliência em contornos. A técnica baseada em esqueleto, discutida na seção 3.4.7 do capítulo 3, utiliza uma abordagem de detecção global uma vez que ela usa como referência o esqueleto interno e externo de toda a forma. Já o detector baseado em K-vizinhos é uma técnica computada localmente, considerando todos os pontos do contorno: primeiramente é calculado, para cada ponto do contorno, o tamanho do ângulo de abertura desse ponto em relação ao seu K-ésimo ponto vizinho; depois, aqueles pontos que tem um ângulo maior que o tamanho especificado pelo usuário são selecionados como possíveis candidatos à saliência; o último passo consiste em remover, dessa lista de candidatos, aqueles pontos que não obedecem a um espaçamento mínimo de saliências adjacentes.

Nos testes, os parâmetros utilizados nessas duas técnicas foram os valores padrões, tal como proposto pelos autores em seus artigos. Para o método baseado em esqueleto, o *threshold* utilizado foi 0.5; para a técnica baseada em K-vizinhos, a distância usada entre os pontos foi 7 e o limite do ângulo foi 150°.

Para o detector proposto no capítulo 4, o único parâmetro a ser definido é o número de iterações para a função de suavização. A função de suavização é usada para eliminar

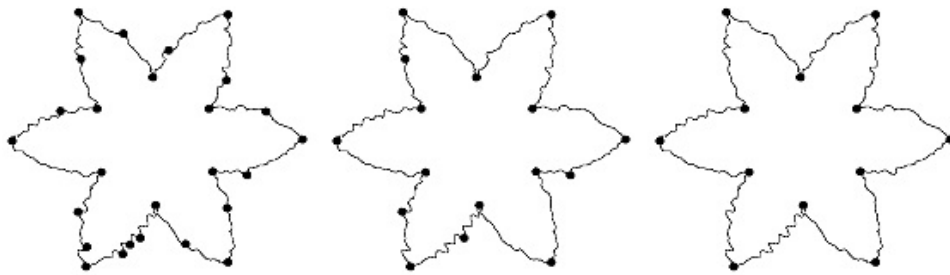


Figura 6.2: Pontos de saliência de uma forma detectados pelo método proposto usando diferentes valores de iteração para a função de suavização.

a influência de ruídos e remover pequenos detalhes, assim, o número de iterações para a função de suavização depende da força de ruídos presente na forma ou da quantidade de pequenos detalhes que o usuário deseja eliminar. Aumentando o número de iterações alguns pontos de saliência menos relevantes serão suprimidos. A figura 6.2 apresenta as saliências detectadas pelo método proposto utilizando diferentes valores de iteração para a função de suavização. Através de testes experimentais realizados, indica-se que o ideal é um número de iterações superior a 500. Todos os testes apresentados nesse trabalho foram feitos usando 500 iterações.

6.2.3 Medida de Avaliação

Para analisar a acurácia dos detectores, a medida de avaliação adotada foi a mesma usada por [Mokhtarian e Mohanna 2006], que é calculada como:

$$\text{Acurácia} = \frac{\text{Precisão} + \text{Revocação}}{2} \quad (6.1)$$

A Acurácia é uma medida da correlação entre o resultado obtido e o resultado esperado, ou seja, ela mede o quanto o resultado obtido está relacionado com o “valor real” esperado. Ela informa o quanto o resultado obtido é “bom”, ou seja, quanto o valor estimado é próximo do valor real e fornece a confiabilidade do valor obtido. O valor máximo de Acurácia é 1.0, isso significa que todos os pontos detectados são saliências e nenhum ponto falso foi detectado.

A Precisão mede a razão de pontos que são saliência dentre todos os pontos detectados. Ela é calculada pela fórmula:

$$\text{Precisão} = \frac{V}{V + F} \quad (6.2)$$

onde V é a quantidade de pontos detectados que são saliências e F a quantidade de pontos detectados que não são saliências. O maior valor de Precisão é 1.0, isso significa que todos os pontos detectados são saliências, mas não diz se todas as saliências manualmente selecionadas na base de dados foram detectadas.

A Revocação mostra a proporção de saliências que foram detectadas em razão da

	Precisão	Revocação	Acurácia
[Chetverikov 2003]	0.21	0.96	0.59
[Torres et al. 2004]	0.95	0.75	0.85
Detector Proposto	0.89	0.92	★ 0.91

Tabela 6.1: Resultados comparativos entre os detectores de saliência avaliados.

quantidade de saliências esperadas. Ela é calculada como:

$$\text{Revocação} = \frac{V}{N} \quad (6.3)$$

onde V é a quantidade de pontos detectados que são saliências e N é o total de saliências manualmente selecionadas na base de teste, no caso $N = 1853$. Assim como Precisão, o maior valor para a Revocação é 1.0, o que significa que todas as 1853 saliências selecionadas manualmente na base de dados foram detectadas, porém o uso isolado dessa medida não diz nada sobre quantas falsas saliências foram detectadas.

6.2.4 Resultados Obtidos

Os resultados obtidos estão apresentados na Tabela 6.2.4. A técnica baseada em esqueleto teve o melhor valor de Precisão, uma vez que ela foi a que menos detectou falsos pontos, entretanto ela deixou de detectar muitas saliências, isso justifica o baixo valor de Revocação. Por outro lado, a técnica de [Chetverikov 2003] teve o melhor valor de Revocação e o pior valor de Precisão, uma vez que ela foi capaz de detectar quase todas as saliências, porém ela falhou detectando grande quantidade de falsos pontos: ela detectou 6768 saliências em todas as imagens da base de teste, enquanto que apenas 1853 saliências verdadeiras deveriam ser detectadas. Por outro lado, o detector proposto equilibrar esses resultados: ele detectou o máximo de saliências com o mínimo de pontos falsos, obtendo o melhor valor de Acurácia.

A figura 6.3 apresenta um exemplo de uma forma com suas saliências detectadas pelos detectores avaliados. Como pode ser visto, a técnica proposta detectou todas as saliências relevantes, enquanto que a técnica de [Torres et al. 2004] deixou de detectar importantes saliências e o método de [Chetverikov 2003] detectou muitas saliências irrelevantes que não são importantes para representar a forma.

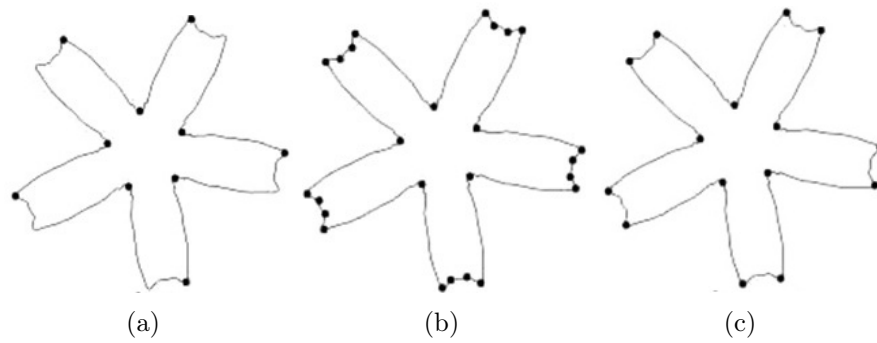


Figura 6.3: Pontos de saliência detectados: (a) pelo método proposto por [Torres et al. 2004], (b) pelo método proposto por [Chetverikov 2003] e (c) pela técnica proposta neste trabalho.

6.3 Recuperação de Formas

O problema de recuperação consiste em classificar quais imagens, de uma base de dados, são consideradas mais similares à uma dada imagem de consulta. Essa classificação é feita por uma medida de similaridade e um *ranking* é gerado. As imagens mais similares, ou seja, aquelas com menor distância, aparecem nas primeiras posições do *ranking*.

O objetivo dos experimentos dessa seção é avaliar a eficiência do descritor proposto na recuperação de formas, comparando-o com outros diferentes descritores encontrados na literatura. A avaliação dessa eficiência é uma tarefa muito complexa, que envolve questões relacionadas com a definição de uma base de imagens, o conjunto de imagens consulta que serão usadas, a determinação do conjunto de imagens relevantes à imagem consulta e a medida para avaliar a performance de recuperação dos descritores. Nas subseções seguintes essas etapas são detalhadas.

6.3.1 Base de Imagens

Para os experimentos, três bases de imagens foram utilizadas:

- MPEG-7 CE-shape-1 part B: essa base é formada por 1400 imagens classificadas em 70 classes (20 formas em cada classe). Ela é composta por silhuetas de diversos objetos, como frutas, animais, objetos, etc. Essa base de imagens é a mais usada para avaliar a performance de descritores de formas. As figuras 6.9, 6.10 e 6.11 ilustram alguns exemplos de imagens desse banco.
- Kimia-99: consiste em 99 formas, agrupadas em 9 classes com 11 formas em cada classe, conforme mostrado na figura 6.4. As imagens de uma mesma classe são diferentes entre si por terem algumas transformações como articulação, oclusão e deformação das partes.
- Kimia-216: é um banco composto por 216 formas, agrupadas em 18 classes com 12 formas em cada classe. Esse banco é um subconjunto da base MPEG-7, porém

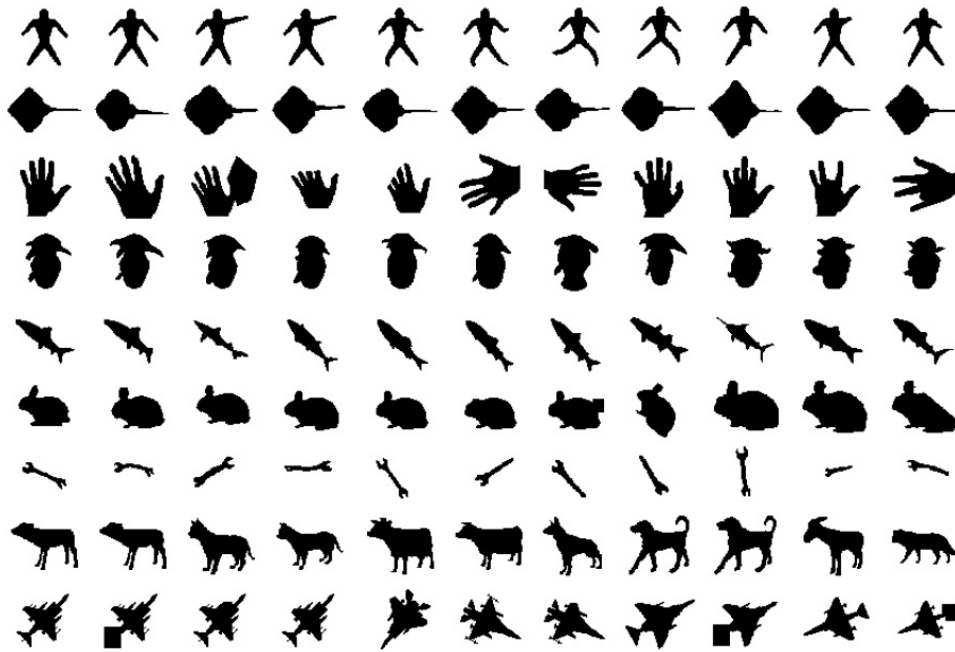


Figura 6.4: Banco de dados Kimia-99.



Figura 6.5: Dois exemplos de cada classe do banco Kimia-216.

comparado com o banco MPEG-7, imagens de classes distintas são visualmente bem diferentes. A figura 6.5 ilustra dois exemplos de cada classe desse banco.

6.3.2 Descritores Avaliados

Seis outros diferentes descritores encontrados na literatura serão usados para avaliar a performance do descritor proposto neste trabalho, são eles:

- BAS (*Beam Angle Statistics*) [Arica e Vural 2003];
- MS Fractal (*Multiscale Fractal Dimension*) [Torres et al. 2004]
- MI (*Moment Invariants*) [Liao e Pawlak 1996]
- Fourier (*Fourier Descriptor*) [Gonzalez e Woods 2001]
- TSDIZ (*Tensor Scale Descriptor with Influence Zones*) [Andaló et al. 2010]
- CS (*Contour Saliences*) [Torres e Falcão 2007]

Esses seis descritores serão comparados entre si e com duas versões do descritor SSD: uma versão que utiliza somente a medida de similaridade d_s (eq. 5.9) baseada nas saliências das formas, e uma outra versão que utiliza a distância D_f (eq. 5.4), que faz uso de três características globais na medida de similaridade (solidez, relação de aspecto e excentricidade). Essa última versão será referenciada como SSD+GF (*Shape Saliences Descriptor plus Global Features*).

6.3.3 Medida de Avaliação

A performance dos descritores foi avaliada usando a medida de Precisão vs. Revocação (PR). A curva PR indica o compromisso entre essas duas medidas e a curva mais alto no gráfico indica o descritor com a melhor eficácia.

Para os experimentos foi adotado a técnica de consulta-por-exemplo (*query-by-example*). Esse paradigma é baseado em: dado uma imagem consulta, suas características são extraídas, uma medida de similaridade é usada para calcular a distância entre essas características e as características das imagens do banco de dados (previamente extraídas) e, finalmente, um *ranking* é gerado com as imagens com a menor distância nas primeiras posições.

6.3.4 Resultados Obtidos

Analisando a performance dos descritores no banco de dados Kimia-99, o descritor SSD+GF foi o que obteve a melhor curva de PR, como pode ser visto no gráfico da figura 6.6 que mostra as curvas de PR para todos os descritores avaliados. O descritor SSD teve melhor desempenho que os descritores Fourier, MI e CS.

Para o banco de dados Kimia-216, novamente o descritor SSD+GF obteve a melhor curva para todos os valores de revocação, tal como reportado no gráfico da figura 6.7. Esse resultado mostra a habilidade do descritor em reconhecer formas parcialmente oclusas, já que grande parte das imagens desse banco de dados possuem alguma transformação visual como articulação e/ou oclusão. O descritor BAS obteve a segunda melhor curva de PR e o descritor SSD obteve performance superior que os descritores MS Fractal, Fourier, MI e CS.

Considerando o banco MPEG-7, o descritor SSD+GF obteve a segunda melhor curva de PR, o descritor BAS apresentou a melhor curva, como mostrado pela figura 6.8. Um importante fato que se deve levar em consideração é que o reconhecimento de formas, usando o banco MPEG-7, não é uma tarefa trivial, já que existem imagens da mesma classe que são visualmente bastante diferentes, como pode ser visto comparando as imagens da figura 6.11. Além disso, existem imagens de classes diferentes que são visualmente bastante similares, por exemplo as formas da figura 6.10. O descritor SSD+GF conseguiu

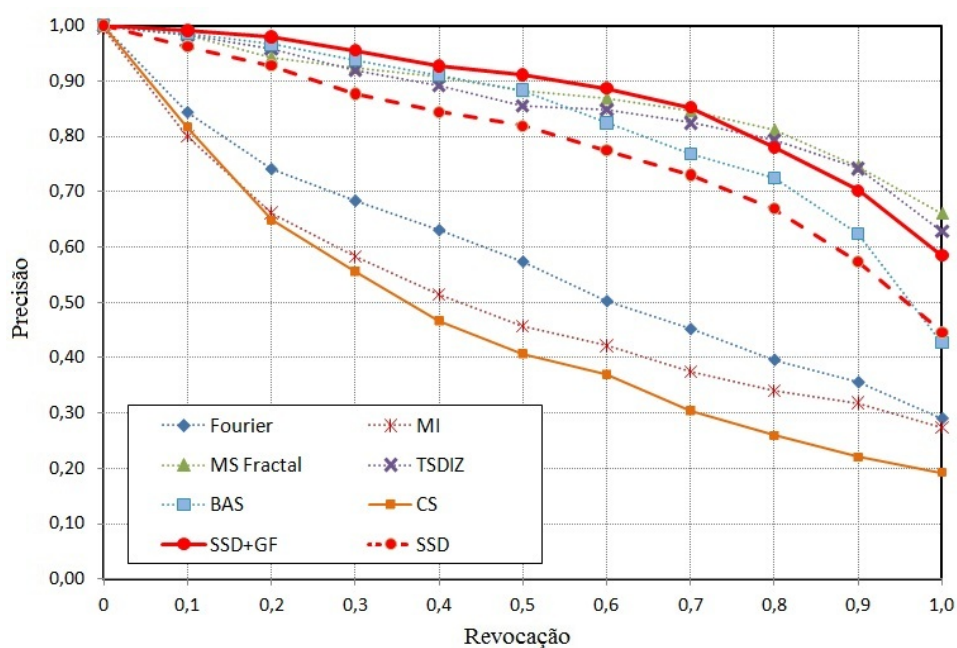


Figura 6.6: Performance dos descritores usando o banco de dados Kimia-99.

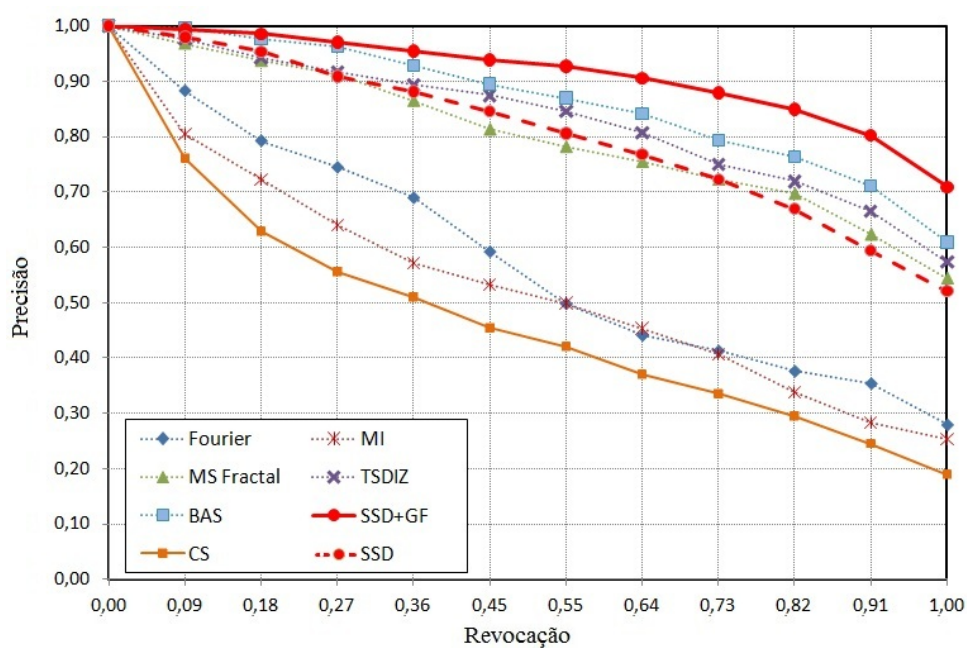


Figura 6.7: Performance dos descritores usando o banco de dados Kimia-216.

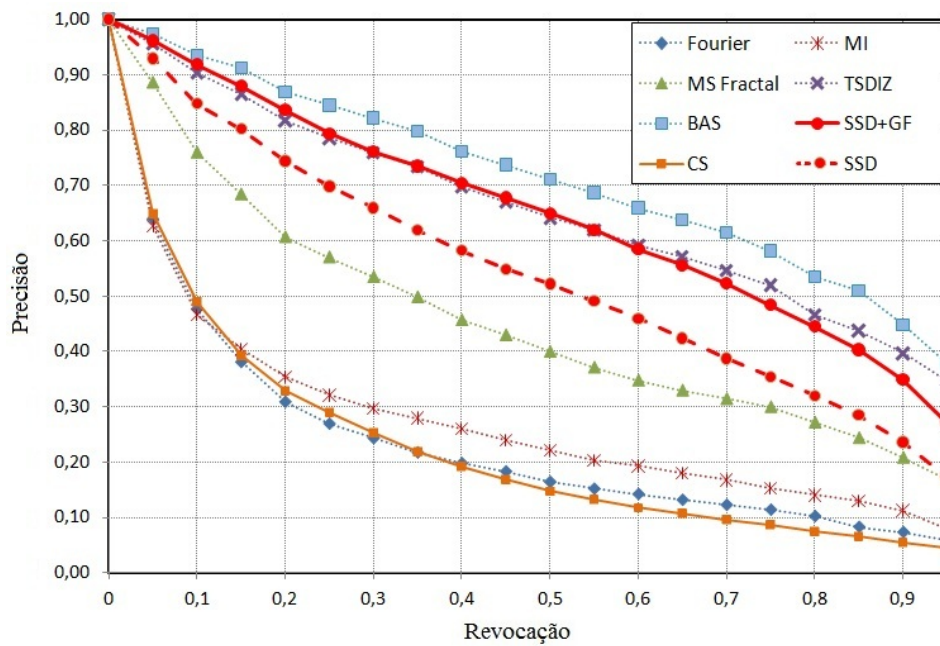


Figura 6.8: Performance dos descritores usando o banco de dados MPEG-7.

recuperar muitas imagens semelhantes de outras classes, e esse fato penalizou a curva média de PR.

A figura 6.9 mostra o resultado de uma consulta realizada usando o descritor SSD+GF e os outros descritores avaliados para o banco MPEG-7. As imagens com uma borda ao redor não pertencem à mesma classe que a imagem consulta. Neste exemplo, o descritor SSD+GF retornou apenas 1 imagem irrelevante nas 10 primeiras posições. Os descritores BAS e TSDIZ retornaram 4 imagens visualmente bem diferentes nas 10 primeiras posições. Esse exemplo mostra que o descritor SSD+GF conseguiu capturar informações que são visualmente similares entre as imagens. A figura 6.12-a exibe algumas outras consultas que ilustram essa situação, as imagens com uma borda ao redor não pertencem à classe da imagem consulta porém são visualmente bastante semelhantes. O gráfico de PR é construído considerando relevantes apenas as imagens da mesma classe da imagem consulta e os três exemplos da figura 6.12-a apresentaram uma curva de PR ruim, como pode ser visto na figura 6.12-b, e esse fato contribuiu negativamente na curva média de PR. A recuperação de formas usando o banco de dados Kimia-216 mostra a habilidade do descritor SSD+GF em reconhecer silhuetas visualmente similares: nessa base as classes são visualmente bastante diferentes e o descritor SSD+GF apresentou a melhor performance dentre os descritores analisados.

Outro aspecto positivo, à favor do descritor proposto, é a sua capacidade de representar formas de uma maneira compacta. O descritor proposto utiliza um vetor de característica com dimensão variável, que depende da característica da forma. Esse fato é positivo, pois exige pouco espaço para armazenamento das informações. A tabela 6.2 mostra o espaço que cada descritor utilizou, para armazenar os vetores de características extraídos das

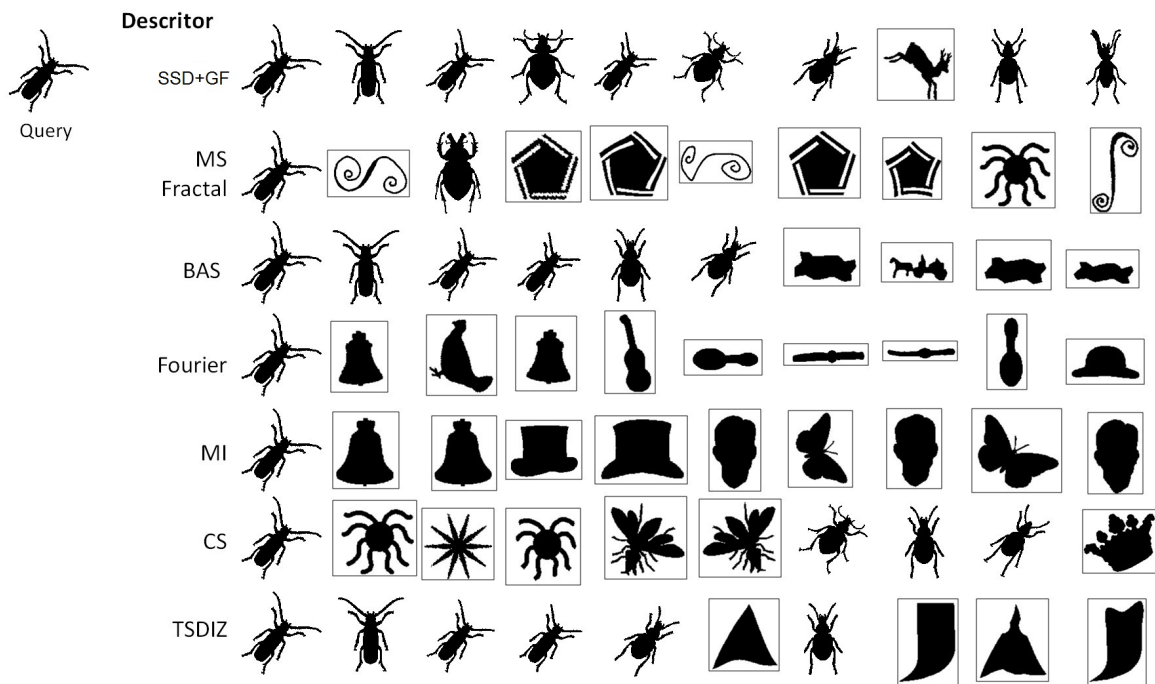


Figura 6.9: Exemplo de uma consulta de uma forma do banco de dados MPEG-7.

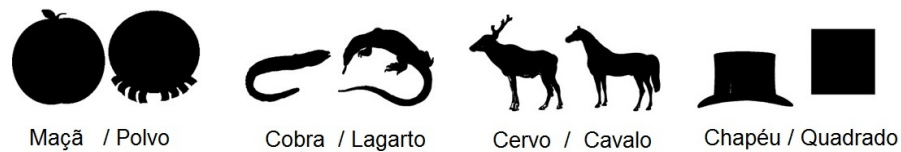


Figura 6.10: Imagens do banco MPEG-7 de classes diferentes mas visualmente bastante similares.

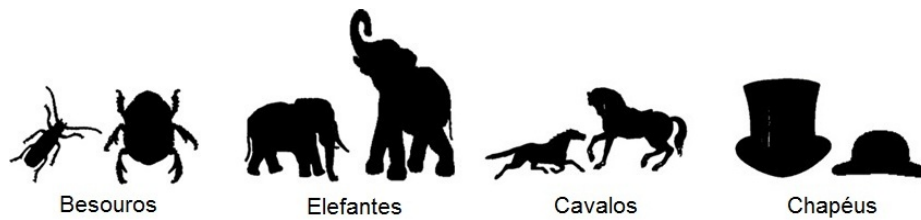
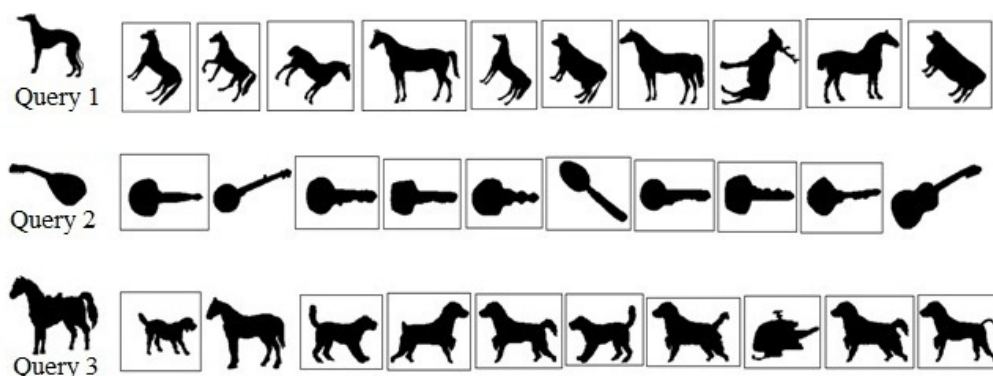
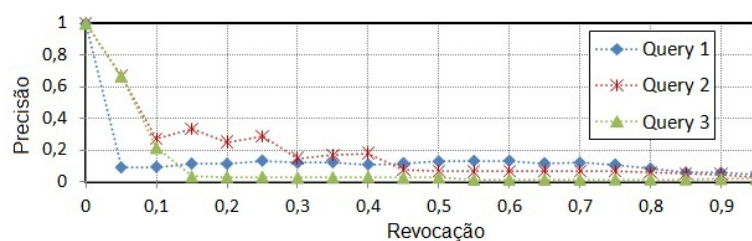


Figura 6.11: Imagens do banco MPEG-7 da mesma classe mas visualmente bastante diferentes.

imagens do banco de dados MPEG-7. O descritor SSD+GF exigiu o 3º menor espaço de armazenamento, perdendo apenas pelos descritores MI e CS, porém esses dois últimos descritores não apresentaram boa performance na recuperação de formas, como pode ser visto nos gráficos de PR reportados anteriormente. O descritor BAS foi o que exigiu maior espaço para armazenar seus vetores de características. O descritor proposto teve um ganho de 73% de espaço para armazenamento comparado com o BAS.



(a)



(b)

Figura 6.12: (a) Algumas consultas realizadas pelo descritor proposto; (b) gráfico mostrando a curva de PR dessas consultas.

Descritor	Dimensão do vetor	Espaço de armazenamento
MS Fractal	100	1,30 MB
MI	14	0,20 MB
Fourier	126	1,68 MB
TSDIZ	60	0,90 MB
BAS	180	2,67 MB
CS	$2 \times n$	0,47 MB
SSD+GF	$3 + 2 \times n$	0,73 MB

Tabela 6.2: Dimensão dos vetor de característica e espaço de armazenamento (total) utilizado pelos descritores considerando o banco MPEG-7. **n é o número de saliências da forma.*

Capítulo 7

Conclusão

A área de recuperação e caracterização de imagens tem avançado muito nos últimos anos, porém esse campo de pesquisa ainda possui grandes desafios a serem alcançados. O problema de analisar a similaridade requer um método que capture a essência da imagem. Essa situação deve ser levada em consideração para obter uma performance próxima à percepção humana. Além disso, outra necessidade é a extração de um número suficiente de características para representar uma imagem, de tal maneira que reduza a dimensionalidade dos dados originais e que possa realmente diferenciar perceptualmente duas imagens. Baseando-se nessas necessidades, diversas pesquisas estão sendo realizadas com o objetivo de obter extratores de características eficientes para analisar a similaridade entre imagens.

Neste trabalho foi apresentado uma abordagem de analisar a similaridade entre formas através do uso de pontos de saliência, referenciado como *Shape Saliences Descriptor* (SSD). Essa maneira de análise tem uma forte motivação biológica, uma vez que, visualmente, duas formas são diferenciadas por esses pontos. Em comparação com outros descritores de formas, o principal ganho do método de descrição proposto nesse trabalho é a qualidade no reconhecimento de formas associado com uma representação compacta. Além disso, a técnica proposta é invariante à algumas transformações geométricas, tais como translação, rotação, espelhamento e insensível à ruídos.

O descritor SSD utiliza: (i) um detector de saliências robusto ao ruído, (ii) uma representação baseada na posição angular relativa da saliência e o seu valor de curvatura, e (iii) uma medida de similaridade, que permite comparar as formas com um número diferente de pontos de saliência, invariante à rotação e espelhamento. Características globais (relação de aspecto, excentricidade e solidez) das formas foram incorporadas na distância de similaridade para aumentar ainda mais a capacidade de discriminação das imagens, originando uma nova versão do descritor SSD referenciado como SSD+GF (*Shape Saliences Descriptor plus Global Features*).

Resultados comparativos foram realizados utilizando seis diferentes descritores de formas encontrados na literatura. O descritor SSD+GF apresentou a melhor curva de Pre-

cisão *versus* Revocação para dois banco de dados (Kimia-99 e Kimia-216) e a segunda melhor curva para o banco MPEG-7. A principal vantagem do descritor proposto é a habilidade de capturar informações que são visualmente semelhantes entre as formas.

7.1 Trabalhos Futuros

Para trabalhos futuros, sugerem-se alguns tópicos relacionados às possíveis extensões e validação do método apresentado, tais como:

- explorar o método em imagens genéricas;
- testar o método proposto em bases específicas, como imagens médicas;
- estender o método proposto para o reconhecimento de formas tridimensionais;
- utilizar uma abordagem inteligente, via algoritmo genéticos, para identificar um conjunto de parâmetros que, associados às características originais re-caracterizassem as formas, adaptando o descritor proposto à bases específicas;
- usar mecanismos de inclusão da percepção visual do usuário no processo de recuperação, por exemplo a partir de técnicas de realimentação de relevância.

7.2 Publicações Realizadas

7.2.1 Artigos completos publicados em periódicos

1. Pedrosa, G.V., Barcelos, C.A.Z. (2010). Anisotropic diffusion for effective shape corner point detection. *Pattern Recognition Letters*, 31(12):1658-1664.

7.2.2 Artigos publicados em anais de congressos

1. Pedrosa, G.V., Barcelos, C.A.Z., Batista, M.A. (2011). An image retrieval system using shape salience points. In: *IEEE International Symposium on Circuits and System (ISCAS)*, Rio de Janeiro.
2. Pedrosa, G.V., Barcelos, C.A.Z., Batista, M.A. (2011). A shape contour descriptor based on salience points. In: *IEEE International Conference on Image Processing (ICIP)*, Bruxelas.

7.2.3 Resumos publicados em anais de congressos

1. Pedrosa, G.V., Barcelos, C.A.Z., Batista, M.A. (2011). Caracterização de formas através de pontos salientes. In: *Congresso de Matemática Aplicada e Computacional (CMAC)*, Uberlândia.

Referências Bibliográficas

- [Abbasi et al. 1999] Abbasi, S., Mokhtarian, F., e Kittler, J. (1999). Curvature scale space image in shape similarity retrieval. *MultiMedia Systems*, 7(6):467–476.
- [Adamek e Connor 2004] Adamek, T. e Connor, N. E. O. (2004). A multiscale representation method for nonrigid shapes with a single closed contour. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(5):742–753.
- [Andaló et al. 2010] Andaló, F. A., Miranda, P. A. V., Torres, R. S., e Falcão, A. X. (2010). Shape Feature Extraction and Description based on Tensor Scale. *Pattern Recognition*, 43(1):26–36.
- [Arica e Vural 2003] Arica, N. e Vural, F. T. Y. (2003). Abstract BAS: A Perceptual Shape Descriptor Based on the Beam Angle Statistics. *Pattern Recognition Letters*, 24(9-10):1627–1639.
- [Asada e Brady 1986] Asada, H. e Brady, M. (1986). The curvature primal sketch. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8(1):2–14.
- [Attneave 1954] Attneave, F. (1954). Some informational aspects of visual perception. *Psychological Review*, 61(3):183–193.
- [Bandera et al. 1999] Bandera, A., Urdiales, C., Arrebola, F., e Sandoval, F. (1999). 2D object recognition based on curvature functions obtained from local histograms of the contour chain code. *Pattern Recognition Letters*, 20(1):49–55.
- [Bober 2001] Bober, M. (2001). MPEG-7 Visual Shape Descriptors. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(9):716–719.
- [Chang e Fu 1980] Chang, N. S. e Fu, K. S. (1980). Query-by-pictorial-example. *IEEE Transactions on Software Engineering*, 6(6):519–524.
- [Chetverikov 2003] Chetverikov, D. (2003). A Simple and Efficient Algorithm for Detection of High Curvature Points in Planar Curves. In *Computer Analysis of Images and Patterns*, pp. 746–753.
- [Ciaccia et al. 1997] Ciaccia, P., Patella, M., e Zezula, P. (1997). M-Tree: an efficient access method for similarity search in metric spaces. In *Proceedings of 23rd International Conference on Very Large Data Bases*, pp. 426–435.
- [Costa e Cesar 2001] Costa, L. F. e Cesar, J. R. M. (2001). *Shape analysis and classification: theory and practice*. CRC Press, Florida, USA.

- [Dubey e Sinthupinyo 2010] Dubey, P. e Sinthupinyo, W. (2010). New approach on structural feature extraction for character recognition. In *Communications and Information Technologies (ISCIT), 2010 International Symposium on*, pp. 946–949.
- [Fischler e Bolles 1986] Fischler, M. A. e Bolles, R. C. (1986). Perceptual organization and curve partitioning. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8(1):100–105.
- [Flickner et al. 1995] Flickner, M., Sawhney, H., Niblack, W., Huang, Q., Ashley, J., Dom, B., Gorkani, M., Hafner, J., Lee, D., Petkovic, D., Steele, D., e Yanker, P. (1995). Query by image and video content: the QBIC system. *IEEE Computer*, 28(9):23–32.
- [Freeman e Davis 1977] Freeman, H. e Davis, L. S. (1977). A Corner-Finding Algorithm for Chain-Coded Curves. *IEEE Trans. Comput.*, 26(3):297–303.
- [Gonzalez e Woods 2001] Gonzalez, R. C. e Woods, R. E. (2001). *Digital image processing*. Longman Publishing Co. Inc.
- [Haralick et al. 1973] Haralick, R. M., Shanmugam, K., e Dinstein, I. (1973). Textural Features for Image Classification. *IEEE Transactions on Systems, Man and Cybernetics*, 3(6):610–621.
- [Huang et al. 1997] Huang, J., Kumar, S., Mitra, M., Zhu, W., e Zabih, R. (1997). Image Indexing Using Color Correlograms. *IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 762–768.
- [Jacobs et al. 1995] Jacobs, C. E., Finkelstein, A., e Salesin, D. H. (1995). Fast Multiresolution Image Query.
- [Jain e Vailaya 1998] Jain, A. K. e Vailaya, A. (1998). Shape-based retrieval: A case study with trademark image databases. *Pattern Recognition*, 31(9):1367–1390.
- [Langridge 1982] Langridge, D. J. (1982). Curve Encoding and the Detection of Discontinuities. *Computer Graphics and Image Processing*, 20(1):58–71.
- [Lee e Bien 1996] Lee, K. e Bien, Z. (1996). Grey-Level Corner Detector Using Fuzzy Logic. *Pattern Recognition Letters*, 17(9):939–950.
- [Liao e Pawlak 1996] Liao, S. e Pawlak, M. (1996). On image analysis by moments. *IEEE Trans. Pattern Anal. Mach. Intell.*, 18(3):254–266.
- [Lim et al. 1995] Lim, K. B., Xin, K., e Hong, G. S. (1995). Detection and estimation of circular arc segments. *Pattern Recogn. Lett.*, 16(6):627–636.
- [Liua et al. 2007] Liua, Y., Zhanga, D., Lua, G., e Ma, W.-Y. (2007). A survey of content-based image retrieval with high-level semantics. *Pattern Recognition*, 40.
- [Lopes et al. 2006] Lopes, I. O., Barcelos, C. A. Z., Batista, M. A., e Silva, A. M. (2006). Enhanced Watermarking Scheme Based on Texture Analysis. In *Advanced Concepts for Intelligent Vision Systems (ACIVS)*, pp. 746–756.
- [Mandelbrot 1982] Mandelbrot, B. (1982). The Fractal Geometry of Nature. *W.H. Freeman and Co.*

- [Manjunath e Ma 1996] Manjunath, B. S. e Ma, W. Y. (1996). Texture Features for Browsing and Retrieval of Image Data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(8):837–842.
- [Manjunath et al. 2001] Manjunath, B. S., Ohm, J. R., Vasudevan, V. V., e Yamada, A. (2001). Color and Texture Descriptors. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(6):703–715.
- [Medioni e Yasumoto 1995] Medioni, G. e Yasumoto, Y. (1995). *Corner detection and curve representation using cubic B-splines*. IEEE Computer Society Press, Los Alamitos, CA, USA.
- [Müller et al. 2004] Müller, H., Michoux, N., Bandon, D., e Geissbuhler, A. (2004). A review of content-based image retrieval systems in medical applications - clinical benefits and future directions. *International Journal of Medical Informatics*, 73(1).
- [Mokhtarian e Bober 2003] Mokhtarian, F. e Bober, M. (2003). *Curvature Scale Space Representation: Theory, Applications, and MPEG-7 Standardization*. Kluwer Academic Publishers.
- [Mokhtarian e Mackworth 1986] Mokhtarian, F. e Mackworth, A. K. (1986). Scale-Based Description and Recognition of Planar Curves and Two-Dimensional Shapes. *IEEE Pattern Anal. Mach. Intell.*, 8(1):34–43.
- [Mokhtarian e Mohanna 2006] Mokhtarian, F. e Mohanna, F. (2006). Performance evaluation of corner detectors using consistency and accuracy measures. *Computer Vision and Image Understanding*, 102(1):81–94.
- [Mokhtarian e Suomela 1998] Mokhtarian, F. e Suomela, R. (1998). Robust Image Corner Detection Through Curvature Scale Space. *IEEE Trans. Pattern Anal. Mach. Intell.*, 20(12):1376–1381.
- [Park e Han 1998] Park, J. S. e Han, J. H. (1998). Contour Matching: A curvature-based approach. *Image and Vision Computing*, 16.
- [Pass et al. 1996] Pass, G., Zabih, R., e Miller, J. (1996). Comparing Images Using Color Coherence Vectors. *In Proceedings of the fourth ACM international conference on Multimedia*, pp. 65–73.
- [Perona e Malik 1990] Perona, P. e Malik, J. (1990). Scale-Space and Edge Detection Using Anisotropic Diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7):629–639.
- [Peura e Iivarinen 1997] Peura, M. e Iivarinen, J. (1997). Efficiency of simple shape descriptors. *Proc. 3rd International Workshop on Visual Form (IWVF3)*.
- [Rosenfeld e Johnston 1973] Rosenfeld, A. e Johnston, E. (1973). Angle Detection on Digital Curves. *IEEE Trans. Comput.*, 22:875–878.
- [Rosenfeld e Weszka 1975] Rosenfeld, A. e Weszka, J. S. (1975). An Improved Method of Angle Detection on Digital Curves. *IEEE Trans. Comput.*, 24(9):940–941.

- [Saha 2005] Saha, P. (2005). Tensor Scale: a local morphometric parameter with applications to computer vision and image processing. *Computer Vision and Image Understanding*, 99(1):384–413.
- [Santos et al. 2009] Santos, J. A., Lamparelli, R., e da S. Torres, R. (2009). Using Relevance Feedback for Classifying Remote Sensing Images. *In: Brazilian Remote Sensing Symposium*, pp. 7909–7916.
- [Silva et al. 2011] Silva, A. T., Magalhães, L. P., e Falcão, A. X. (2011). Uma Abordagem CBIR Baseada em Realimentação de Relevância e Classificação por Floresta de Caminhos Ótimos. *IV Encontro dos Alunos e Docentes do Departamento de Engenharia de Computação e Automação Industrial - EADCA*, pp. 1–4.
- [Smeulders et al. 2000] Smeulders, A. M., Worring, M., Santini, S., Gupta, A., e Jain, R. (2000). Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1349–1380.
- [Smith e Chang 1996] Smith, J. R. e Chang, S. F. (1996). Visual SEEK: A Fully Automatic Content-Based Image Query System. *ACM Multimedia*, (96):87–98.
- [Stricker e Orengo 1995] Stricker, M. A. e Orengo, M. (1995). Similarity of Color Images. *In Storage and Retrieval for Image and Video Databases (SPIE)*, pp. 381–392.
- [Swain e Ballard 1991] Swain, M. e Ballard, D. (1991). Color Indexing. *International Journal of Computer Vision*, 7(1):11–32.
- [Torres e Falcão 2006] Torres, R. D. S. e Falcão, A. X. (2006). Content-Based Image Retrieval: Theory and Applications. *Revista de Informática Teórica e Aplicada*, 13:161–185.
- [Torres e Falcão 2007] Torres, R. S. e Falcão, A. X. (2007). Contour Saliency Descriptors for Effective Image Retrieval and Analysis. *Image and Vision Computing*, 25(1):3–13.
- [Torres et al. 2004] Torres, R. S., Falcão, A. X., e Costa, L. F. (2004). A graph-based approach for multiscale shape analysis. *Pattern Recognition*, 37(6):1163–1174.
- [Traina et al. 2002] Traina, J. C., Traina, A. J. M., Faloutsos, C., e Seeger, B. (2002). Fast indexing and visualization of metric datasets using slim-trees. *IEEE Transactions on Knowledge and Data Engineering*, 14(2):244–260.
- [Tricot 1995] Tricot, C. (1995). Curves and Fractal Dimension. *Springer-Verlag*.
- [Tsai 1997] Tsai, D. (1997). Boundary Based Corner Detection Using Neural Networks. *Pattern Recognition*, 30(1):85–97.
- [Veltkamp e Tanase 2000] Veltkamp, R. C. e Tanase, M. (2000). Content-based image retrieval systems: a survey. Technical Report 34, Department of Computing Science, Utrecht University.
- [Wang et al. 2007] Wang, L., Suter, D., Monash., U., e Melbourne, V. (2007). Learning and Matching of Dynamic Shape Manifolds for Human Action Recognition. *IEEE Transactions on Image Processing*, pp. 1646–1661.

- [Weeks et al. 1999] Weeks, P. J. D., O'Neill, M. A., Gaston, K. J., e Gauld, I. D. (1999). Automating insect identification: exploring the limitations of a prototype system. *Journal of Applied Entomology*, 123(1).
- [Yuan et al. 2009] Yuan, Z., Wu, J., ichiro Kamata, S., Ahrary, A., e Peimin, Y. (2009). Fingerprint image enhancement by super resolution with early stopping. *IEEE International Conference on Intelligent Computing and Intelligent Systems (ICIS)*, 4:20–22.
- [Zegarra et al. 2008] Zegarra, J. A. M., Papa, J. P., Leite, N. J., da S. Torres, R., e Falcão, A. X. (2008). Learning How to Extract Rotation-Invariant and Scale-Invariant Features from Texture Images. *Eurasip Journal on Advances in Signal Processing*.
- [Zhang e Lu 2004] Zhang, D. e Lu, G. (2004). Review of shape representation and description. *Pattern Recognition*, 37(1):1–19.