José Gilvan Rodrigues Maia

Detecção e Reconhecimento de Objetos usando Descritores Locais

Fortaleza – CE

Maio / 2010

José Gilvan Rodrigues Maia

Detecção e Reconhecimento de Objetos usando Descritores Locais

Trabalho apresentado ao Programa de Doutorado em Ciência da Computação da Universidade Federal do Ceará como requisito parcial para obtenção do título de Doutor em Ciência da Computação.

Orientador:
Prof. Dr. Fernando Antônio de Carvalho Gomes

Co-orientador:
Prof. Dr. Creto Augusto Vidal

Mestrado e Doutorado em Ciência da Computação
Departamento de Computação
Centro de Ciências
Universidade Federal do Ceará

Fortaleza – CE

Maio / 2010

Tese apresentada ao Programa de Pós-Graduação do Departamento de Computação da Universidade Federal do Ceará como parte dos requisitos para obtenção do título de Doutor em Ciência da Computação. Aprovada pela Banca Examinadora abaixo assinada.

Prof. Dr. Fernando Antônio de Carvalho Gomes
Departamento de Computação – UFC
Orientador

Prof. Dr. Creto Augusto Vidal
Departamento de Computação – UFC
Co-Orientador

Prof. Dr. José Antonio Fernandes de Macêdo
Departamento de Computação – UFC

Prof. Dr. Ruy Luiz Milidiú
Departamento de Informática – PUC/Rio

Prof. Dr. Francisco Nivando Bezerra
Departamento de Informática – IFCE/Maracanaú

Dedico esta dissertação a meus pais, mestres,
esposa, filho, amigos e família,
pelo apoio e incentivo nos momentos difíceis
que não foram poucos nem muitos,
simplesmente o suficiente para entender quão duro é o
trabalho necessário para realizar nossos projetos de vida.
Não estive sozinho nessa jornada.
Obrigado.

Meus sinceros agradecimentos para:

– O professor doutor Fernando Carvalho, pela formação, orientação e incentivo;

– Os professores doutores Creto Vidal e Joaquim Bento, pela formação e orientação;

– A professora doutora Maria Andréia Formico Rodrigues, pela formação e orientação;

– O professor mestre Osvaldo de Souza, pela incansável coação benevolente;

– O professor doutor Tavares, pela colaboração e pelas discussões pertinentes;

– O colegiado do Departamento de Computação, pela formação e pela oportunidade;

– Todos os participantes BiMo, pela ajuda em diversos momentos;

– Minha família. Vocês são tudo que tenho;

– Minha esposa Adriana e nosso bebê. Vocês iluminam minha vida;

– Todos os meus amigos;

– O pessoal do lendário time Sony Ericsson, hoje UFC Virtual;

– O pessoal do CRAb;

– Todos os colegas da Computação da UFC.

"O tempo é o melhor autor – sempre encontra um final perfeito."

Charles Chaplin

Resumo

A Visão Computacional tem por objetivo investigar teorias para viabilizar a interpretação de imagens por parte de sistemas artificiais implementados em computadores, conferindo-lhes a capacidade de tomar decisões em função do ambiente que observam através de mecanismos de hardware e software. Sob esta perspectiva, a Teoria dos Espaços de Escala fornece um formidável arcabouço matemático para a realização de diversas tarefas de visão em baixo-nível através de algoritmos. Essa teoria viabiliza o desenvolvimento de técnicas virtualmente invariantes aos aspectos de localização, orientação e escala das estruturas presentes nas imagens analisadas, dispensando o conhecimento *a priori* sobre a configuração geométrica dessas estruturas. As abordagens de visão baseadas em Descritores Locais são construídas sobre esse arcabouço matemático com o propósito de derivar modelos computacionais confiáveis para a detecção e o reconhecimento de objetos. Nesse tipo de abordagem, um objeto é representado por um conjunto de *descritores*, que nada mais são do que vetores de características construídos especialmente para assimilar propriedades das estruturas locais em *pontos-chaves*. Os pontos-chaves, por sua vez, correspondem às partes "notáveis"do objeto que são detectadas automaticamente em múltiplas escalas através de mecanismos tipicamente não-supervisionados. As abordagens baseadas em Descritores Locais apresentam diversas vantagens sobre as demais existentes. Detecção e reconhecimento são tarefas realizadas simultaneamente, dispensando mecanismos externos propensos a erros – como a normalização geométrica, por exemplo. É também possível extrair a representação de objetos a partir de uma única imagem, algo impraticável em várias outras abordagens. Além disso, a correspondência entre duas imagens pode ser obtida diretamente, sem pré ou pós-processamentos envolvendo a construção de estruturas de dados complexas ou a resolução de problemas de otimização. Por produzir uma representação espontaneamente redundante na qual vários pontos-chaves são considerados, os objetos podem ser reconhecidos de forma confiável mesmo sob a ocorrência de oclusões. Nossa contribuição neste trabalho é a apresentação de uma metodologia para a utilização de classificadores supervisionados não-lineares na detecção de pontos-chaves. Em particular, novas possibilidades são investigadas usando Máquinas de Vetores de Suporte (SVMs, *Support Vector Machines*). Teoricamente, esta abordagem possibilita a obtenção de um comportamento similar ao apresentado por detectores não-supervisionados – inclusive combinações deles. Além disso, a capacidade de generalização de SVMs permite romper com paradigmas existentes: conhecimento externo sobre a noção de pontos de interesse pode ser usado na derivação automática de novos detectores mais adequados em aplicações específicas. Experimentos computacionais foram realizados com o objetivo de comprovar a eficácia da metodologia proposta, com especial ênfase em casos reais – localização de olhos e reconhecimento de faces.

[**Palavras-chaves:** Visão Computacional, Detecção de Objetos, Reconhecimento de Objetos, Descritores Locais, Detecção de Pontos-Chaves, Aprendizagem de Máquina, Métodos Não-Lineares]

Abstract

Computer Vision aims to investigate theories and to develop technologies to facilitate the interpretation of images by artificial systems implemented on a computer, thereby giving it the ability to make decisions depending on the environment it perceives using hardware and software mechanisms. From this perspective, the Scale-space Theory provides a solid mathematical framework for carrying out a myriad of low-level vision tasks through algorithms that operate on digital images. This theory enables the development of techniques virtually invariant with respect to the location, orientation and scale of structures shown in the analyzed images, thus dispensing *a priori* knowledge of the geometric configuration of these structures. Vision approaches based on Local Descriptors are built on top of this mathematical framework in order to derive computational models for reliable object detection and recognition. In this type of approach, an object is represented by a set of *descriptors*, which are essentially vectors built especially to assimilate the properties of local structures in *keypoints*. These keypoints, in their turn, correspond to the "outstanding parts"of the object are automatically detected across scales using typically unsupervised mechanisms. This kind of approaches have several advantages over other methodologies. Detection and recognition tasks are performed simultaneously, thus eliminating error-prone, external mechanisms like geometric normalization. It is also possible to extract a representation for an object based on a single image. This is something impractical in many other approaches. Moreover, the correspondence between two images can be obtained directly, without the need for pre- or post-processing involving the construction of complex data structures or the resolution of optimization problems. Also, this approach spontaneously produce a redundant representation that considers several keypoints: hence objects can be reliably recognized even under the occurrence of occlusions. Our main contribution in this thesis is the development of a methodology for applying supervised nonlinear classifiers for keypoint detection. In particular, new possibilities have been investigated using Support Vector Machines (SVM). From the theoretical standpoint, this approach can produce a behavior virtually identical to that presented by non-supervised detectors - including combinations of non-supervised detectors. Moreover, the generalization capabilities of SVMs can also break through existing paradigms: the external knowledge about the concept of interest points can be used for the automatic derivation of new detectors, more suitable for specific applications. Computational experiments were carried out in order to prove the efficacy of the proposed methodology, with special emphasis on important real-world applications - eye location and face recognition.

[**keywords:** Computer Vision, Object Detection, Object Recognition, Local Feature Descriptors, Keypoint Detection, Machine Learning, Nonlinear Methods]

Conteúdo

Lista de Figuras

Lista de Tabelas

Lista de Algoritmos

1 Introdução

1.1 Motivação

A Visão Computacional é o ramo da Ciência da Computação que reúne todas as teorias e tecnologias desenvolvidas com a finalidade de possibilitar que imagens sejam interpretadas por sistemas artificiais implementados em computadores. Ou seja, as técnicas de Visão Computacional devem ser realizadas através de elementos de hardware e software. Esse é um ramo que possui inúmeras aplicações que englobam desde a interação natural entre humanos e máquinas, o controle de processos, a navegação de veículos autônomos, até os sistemas de segurança, dentre outras importantes aplicações.

Na prática, são desenvolvidos algoritmos e técnicas específicas para lidar com os diversos tipos de problemas que surgem nesse ramo da Ciência, pois não existe uma teoria padronizada ou suficientemente genérica para modelar todos os aspectos da percepção visual. Assim, um problema específico deve ser tratado através de uma abordagem específica desenvolvida especialmente para o problema em questão. Segmentação de imagens em partes, melhoramento do contraste, reconstrução de objetos e detectores de bordas são exemplos dos problemas encontrados nesse contexto.

A detecção e o reconhecimento de objetos é, de modo geral, um dos problemas básicos mais comuns em Visão Computacional. Esse problema consiste em determinar se um dado tipo de objetos está presente em uma imagem (detecção) e se algum dos objetos detectados corresponde a um dado exemplar daquele tipo de objetos (reconhecimento). Considere faces humanas como um tipo de objetos. Detectar faces significa associar uma posição as faces humanas quando estas estão presentes na imagem. Reconhecer uma face representa a ação de associar alguma das faces detectadas as faces de pessoas que a máquina conhece – os usuários de um sistema, por exemplo.

Contudo, o que um hipotético sistema genérico de visão computacional pode assumir sobre uma imagem que lhe é fornecida como entrada? Considerando que o contexto da

imagem é livre, a resposta mais sensata a essa pergunta é "nada". Não se podem fazer tais pressuposições *a priori* sem que a capacidade de compreensão das imagens por parte do sistema de visão seja comprometida:

- Idealmente, não há como se fixar uma categoria de objetos, muito menos assumir que algum tipo de cenário ou iluminação será adotado. Dependendo da aplicação, esse tipo de restrição pode ser assumida até certo ponto;
- Não se podem fixar poses e tamanhos preferenciais para que um objeto de interesse apareça na imagem, uma vez que o sistema de visão geralmente não possui informações *a priori*. Assim os sistemas de visão devem ser capazes de analisar a imagem em diferentes posições, orientações e escalas para realizar tarefas em *baixo nível*, reservando a decisão sobre quais configurações devem ser ignoradas ou consideradas para módulos mais inteligentes do sistema de visão computacional.

Os métodos baseados em Descritores Locais construídos usando a representação baseada na Teoria do Espaço de Escala desempenham um papel importante nesse contexto. Isto porque eles viabilizam a concepção de sistemas de visão capazes de detectar e reconhecer objetos sem que se imponham restrições quanto a pose e ao tamanho dos objetos que aparecem na imagem.

Entretanto, esses métodos geralmente são baseados na supressão de pontos não-máximos em relação ao valor de operadores diferenciais (TUYTELAARS; MIKOLAJCZYK, 2008). Assim o processo de detecção caracteriza-se como não supervisionado, visto que nenhum conceito externo influencia diretamente nos tipos de pontos reportados como sendo de interesse.

Muitos esforços podem ser encontrados na literatura no sentido de desenvolver detectores supervisionados (DIAS *et al.*, 1995; AMIT *et al.*, 1996; CHEN; ROCKETT, 1997; TSAI, 1997; KADIR *et al.*, 2004; ROSTEN; T.DRUMMOND, 2005; KIENZLE *et al.*, 2005; ROSTEN; T.DRUMMOND, 2006; SLOT; KIM, 2006; OZUYSAL *et al.*, 2006; LEPETIT; FUA, 2006; OZUYSAL *et al.*, 2007). Todavia, as abordagens existentes não consideram a representação baseada na Teoria do Espaço de Escalas (BABAUD *et al.*, 1986; LINDEBERG, 1991), ou ainda foram desenvolvidas para modelar outras situações (KIENZLE *et al.*, 2005) cuja finalidade difere bastante da detecção de pontos-chaves utilizáveis em subsequentes etapas de detecção e reconhecimento de objetos.

1.2 Objetivos

O presente trabalho tem por objetivo principal investigar a utilização de métodos supervisionados, em particular aqueles baseados em classificadores não-lineares (VAPNIK, 1995) para conduzir ou auxiliar a etapa de detecão de pontos-chaves. Mais especificamente:

- Propor uma metologia para a incorporação de mecanismos supervisionados para aprendizagem de máquina na etapa de detecção de pontos-chaves;
- Desenvolver um estudo de caso através da combinação de um método existente com a metodologia proposta;
- Realizar uma análise crítica dos resultados sob a luz de experimentos práticos que caracterizem a utilização da abordagem proposta no contexto de aplicações do mundo real;
- Comparar os resultados obtidos com aqueles obtidos usando as abordagens que compõem o atual estado-da-arte;
- Propor aplicações e possíveis trabalhos futuros que sigam a mesma linha de pesquisa.

1.3 Organização

O presente trabalho encontra-se organizado da seguinte maneira. O Capítulo 2 apresenta uma visão geral dos conceitos básicos utilizados no decorrer do presente trabalho. Em particular é realizada uma discussão sobre os aspectos mais importantes que devem ser considerados por uma representação suficientemente expressiva de imagens com a finalidade de se realizarem tarefas de Visão Computacional. Esta discussão tem por finalidade motivar a introdução do conceito de espaços de escala, que serão o tema do próximo Capítulo.

O Capítulo 3 é dedicado a exposição dos principais aspectos referentes a Teoria do Espaço de Escalas. Os seguintes temas são abordados nesse ponto: propriedades do núcleo Gaussiano sob os pontos de vista analítico e computacional; definição e construção do Espaço de Escala Linear; derivação dos Espaços de Escalas não-Lineares. A apresentação da abordagem baseada em descritores locais para detecção e o reconhecimento de objetos usando a Teoria do Espaço de Escala encerram, portanto, a apresentação da Teoria do Espaço de Escala como representação em múltiplas escalas.

O Capítulo 4 destina-se a apresentação e análise crítica dos principais trabalhos desenvolvidos no campo da detecção de pontos-chaves. No Capítulo 5 é detalhada a metodologia proposta, concebida especialmente para que métodos não-lineares de aprendizagem de máquina sejam usados no contexto da detecção de pontos-chaves. O caso particular das Máquinas de Vetores de Suporte (VAPNIK, 1995) é discutido no âmbito desta metodologia.

Os Capítulos 6, 7 e 8 detalham a experimentação prática envolvendo o método desenvolvido tanto sob a perspectiva de situações gerais e quanto de aplicações específicas. Esses capítulos apresentam ainda uma reflexão à face dos resultados experimentais obtidos, concentrando-se principalmente na discussão das implicações teóricas pressupostas. Por fim, o Capítulo 9 destina-se a apresentação considerações finais sobre este trabalho e a proposição de sugestões para trabalhos futuros nessa linha de pesquisa.

2 Fundamentos Conceituais

O presente Capítulo destina-se à introdução dos fundamentos conceituais usados ao longo deste trabalho. Primeiramente, é apresentada uma breve introdução sobre a formação de imagens de acordo com o funcionamento do olho humano. Em seguida, são abordados os aspectos de formação e representação de imagens para que estas entidades possam ser manipuladas através de algoritmos em programas de computador.

2.1 Formação e Representação de Imagens

Cada olho humano é formado por várias partes especializadas (WANDELL, 1995), das quais se destacam:

- Córnea, que refrata os raios de luz que entram no olho e exerce o papel de proteção à estrutura interna do olho;
- Íris, porção visível e colorida do olho logo atrás da córnea, que regula a quantidade de luz que entra no olho;
- Pupila, abertura central da íris através da qual a luz passa;
- Cristalino, lente biconvexa natural que auxilia na focalização da imagem sobre a retina;
- Retina, membrana fina que preenche a parede interna posterior do olho e que recebe a luz focalizada pelo cristalino. A retina é recoberta por células fotorreceptoras que transformam a luz em impulsos elétricos, de forma que o cérebro os interprete como imagens;
- Nervo ótico, feixe de nervos que transporta os impulsos elétricos do olho para o cérebro;

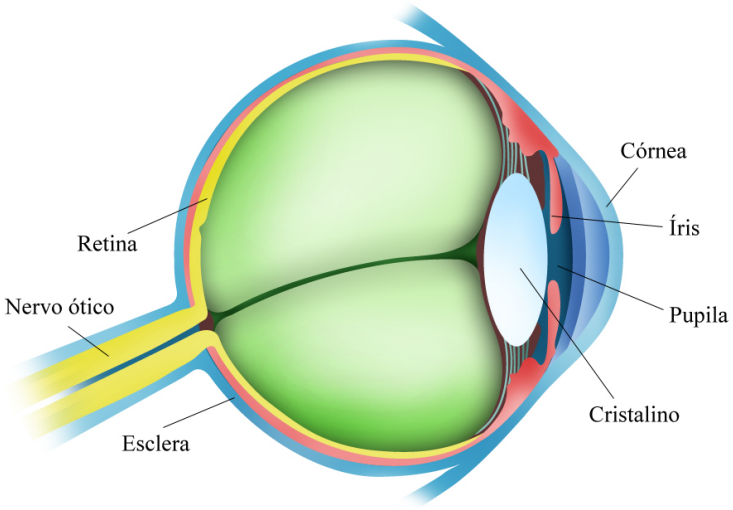

Figura 1: Visão esquemática da estrutura do olho humano, cujas principais partes são a córnea, a íris, a pupila, o cristalino, a retina, o nervo ótico e a esclera.

- Esclera, capa fibrosa e rígida que recobre a região externa do olho, e que, juntamente com a córnea, dá forma ao globo ocular.

Quando os raios de luz provenientes do ambiente atravessam a pupila (Figura 1), uma imagem real e invertida é formada. Essa imagem é projetada sobre a retina com o auxílio do aparato ocular, que atua como uma lente ajustável para que se obtenha uma imagem nítida dos objetos focalizados pelo observador. A retina contém cerca 130 milhões de células fotorreceptoras denominadas cones e bastonetes, responsáveis por captar as informações de crominância (cores) e luminância (luminosidade).

Existem aproximadamente 6 milhões de cones em cada olho humano, sendo que estas células estão concentrados na região fóvea da retina. Por outro lado, os bastonetes aparecem em número muito maior, totalizando aproximadamente 124 milhões de células em cada olho. Apesar de detectarem apenas tons de cinza, os bastonetes são também cerca de 100 vezes mais sensíveis à luz do que os cones. Como resultado, menos de 5% da informação captada pela retina corresponde à percepção das cores. É razoável assumir que a luminância é suficiente para que o cérebro detecte e reconheça objetos em imagens.

As câmeras digitais funcionam de maneira análoga ao olho humano. Nesse caso, uma grade de minúsculos sensores é utilizada para converter a luz em cargas elétricas. Nas câmeras CCD *(Charge-Coupled Device)* introduzidas por (TOMPSETT *et al.*, 1971), por exemplo, uma grade composta por "sítios"de sensores para a recepção de elétrons

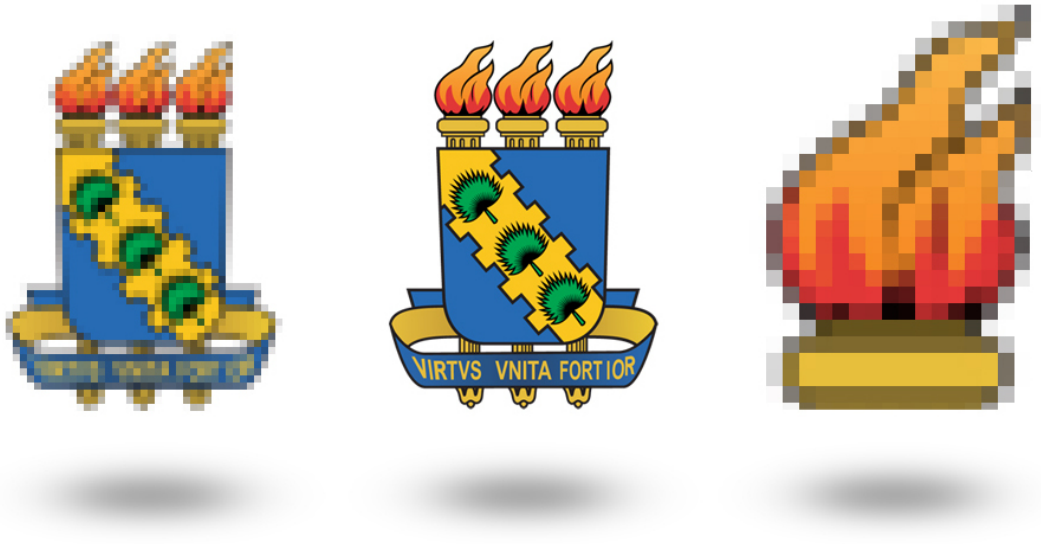

Figura 2: Efeito causado pela variação do número de pixels usado para representar um objeto. À esquerda, vê-se um objeto representado de forma grosseira usando uma pequena quantidade de pixels. No centro, percebem-se maiores detalhes quando um maior número de pixels é usado para representar o mesmo objeto. Perceba que o tamanho dos pixels foi manipulado nesta figura para efeito de apresentação, como à direita.

é sobreposta a uma fina lâmina de silício para medir a quantidade de energia luminosa chegando a cada um desses sítios. Cada sítio é formado por uma camada de dióxido de silício: quando os fótons atingem o silício, elétrons são gerados. Estes elétrons são então capturados usando um potencial elétrico no sítio. Os elétrons capturados são coletados na grade durante um determinado intervalo de tempo a fim de produzir uma imagem.

Dependendo da aplicação, esse intervalo de tempo pode ser extremamente pequeno ou muito grande. As câmeras usadas na fiscalização eletrônica de velocidade veicular, por exemplo, devem ser capazes de capturar imagens contendo placas legíveis mesmo com os veículos a alta velocidade. Nesse caso, o intervalo de tempo é da ordem de milionésimos de segundos. Por outro lado, aplicações como astronomia podem demandar horas para capturar uma única imagem.

As câmeras coloridas são baseadas neste mesmo princípio. Nelas, linhas ou colunas consecutivas de sensores são tornadas sensíveis à luz vermelha, verde ou azul, o que é tipicamente feito usando um revestimento que funciona como um filtro, bloqueando a luz complementar. Como a resolução espacial da câmera é fisicamente limitada, uma melhor qualidade de imagem pode ser obtida usando um divisor para levar a luz aos três sensores simultaneamente.

2.1.1 Imagens *Raster*

Sob o ponto de vista computacional, as imagens capturadas usando câmeras digitais são descritas através de uma grade regular contendo elementos básicos denominados *pixels* que descrevem as propriedades luminosas captadas naquela região da imagem. Esse tipo de imagem é conhecido na literatura como imagem *raster*.

Quanto maior o número de pixels em uma imagem *raster*, melhor o nível de detalhe com que os objetos são representados (Figura 2). Note-se que dois objetos idênticos **a** e **b**, situados próximo e longe da câmera, respectivamente, serão capturados por um número diferente de *pixels* porque a representação *raster* não guarda informações ou evidências sobre as dimensões dos objetos.

O termo **resolução** será utilizado ao longo deste trabalho para denominar o número de *pixels* em uma imagem, correspondendo portanto ao produto de sua altura por sua largura em *pixels*. Tendo em vista que representações multidimensionais de imagens são possíveis (vídeos e imagens 3D, por exemplo), este trabalho considera apenas imagens representadas usando uma grade bidimensional. A menos que o contrário seja claramente afirmado, as imagens consideradas neste trabalho são definidas como um campo escalar, associando portanto um valor de luminância a cada ponto $(x, y) \in \mathbb{Z}^2$. Dada a natureza discreta das imagens raster, suas dimensões horizontal $w \in \mathbb{N}$ e vertical $h \in \mathbb{N}$ são usadas para limitar o domínio de $I(x, y)$.

$$\begin{aligned} &I : [1, w] \times [1, h] \in \mathbb{N}^2 \rightarrow \mathbb{R} \\ &I(x, y) = z \end{aligned} \tag{2.1}$$

2.1.2 Estendendo a representação

As imagens *raster* são representadas usando uma grade regular cujas dimensões em pixels não trazem informação alguma sobre os objetos presentes nesta imagem. Consequentemente, a detecção de objetos e padrões através de algoritmos de Visão Computacional teoricamente deve considerar algum tipo de iteração sobre múltiplas posições, orientações, perspectivas e escalas.

Portanto, uma representação ideal permitiria lidar com uma imagem independentemente do ponto a partir do qual a câmera capta uma determinada cena. Uma representação ideal deveria permitir essa liberdade de ponto de visão sem que a capacidade de expressar os objetos presentes na cena seja perdida. Nesse sentido, (OLIVEIRA *et al.*,

2000) propuseram um método para representar o relevo de uma superfície através de imagens, permitindo reconstruir a aparência dos objetos sob pontos de vista arbitrários para fins de *rendering* (geração de imagens por computador). No entanto, essa técnica requer um considerável esforço de pré-processamento, o que pode torná-la inviável para uso em determinadas aplicações de visão computacional que demandem uma resposta rápida, principalmente em sistemas interativos e sistemas embarcados.

Considerando apenas as imagens bidimensionais, essa representação é *invariante à translação*. Em outras palavras, não há posição preferencial na imagem para que se representem objetos, pois esses são representados da mesma forma. Uma simples iteração sobre cada posição (x, y) é suficiente para lidar com translações.

Entretanto, maiores cuidados devem ser tomados quando se deseja uma representação capaz de expressar objetos em diferentes escalas e orientações. Quanto à orientação, é possível tratá-la de duas formas: (a) através da iteração exaustiva sobre um universo de ângulos, ou (b) através da computação de orientações preferenciais, desde que hajam evidências sobre a posição ou escala do objeto em questão.

Geralmente é possível lidar com a orientação de forma mais eficiente ao se adotar uma representação que expresse naturalmente a posição e a escala para buscar evidências dos objetos. A escala tem um papel fundamental na complexidade dos algoritmos, pois quanto maior um objeto aparece em uma imagem, maior o número de pixels usado para representá-lo. Consequentemente, um maior número de pixels deverá ser visitado para detectar objetos em grandes escalas, comprometendo assim o desempenho de abordagens ingênuas sob o ponto de representação em múltiplas escalas.

Dessa maneira, são necessários teorias, estruturas de dados e algoritmos eficientes capazes de lidar com a representação multiescala no mundo inerentemente discreto dos programas de computador.

2.2 Representações Multiescala

2.2.1 Estruturas de Subdivisão Espacial

Proposta por Klinger (KLINGER, 1971), a *Quadtree* é uma árvore especial utilizada originalmente para codificar imagens através de subdivisões sucessivas do espaço em células adaptativas que representam quadrantes da região a ser representada. Esse tipo de árvore contém apenas dois tipos de nós: folhas e nós internos. Os nós internos contém

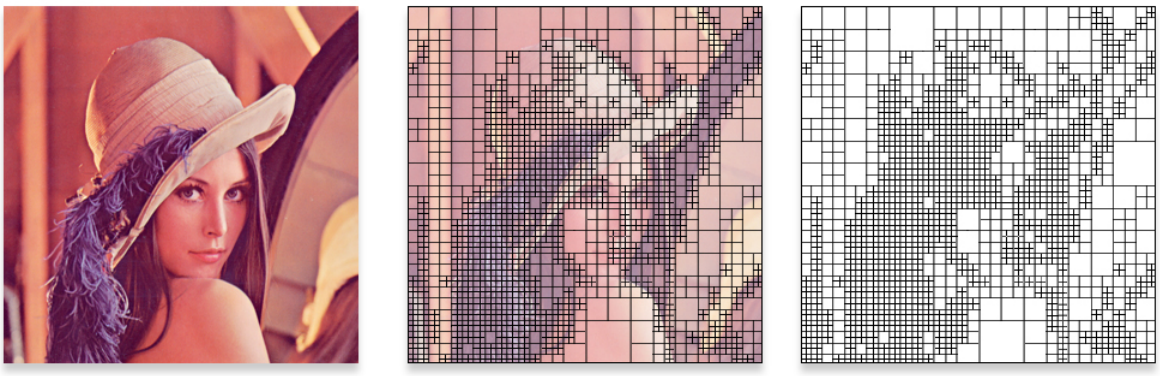

Figura 3: *Quadtree* representando uma foto clássica. Os quadrantes são denotados por quadrados cujas áreas diminuem à medida que a imagem é subdividida. Perceba que um maior nível de subdivisão é encontrado em regiões com maior nível de detalhe, ou seja, regiões nas quais a variação dos níveis de cinza é maior.

exatamente quatro filhos, cada um representando quadrantes de mesma área.

O nó raiz corresponde à imagem original. A área de cada nó é subdividida recursivamente em quadrantes menores, de forma que a área dos nós diminui à medida que a profundidade dos nós aumenta. A subdivisão pára quando encontra-se um nó que corresponde a um único pixel (atingindo, portanto, o limite de subdivisão), ou quando não há variação nas cores dos pixels contidos na região correspondente. Como resultado, um maior número de nós é associado às regiões contendo maior variação nas intensidades dos pixels. A Figura 3 ilustra uma *Quadtree* construída em função de uma imagem.

O processo de subdivisão usado pela *Quadtree* pode ser formalizado da seguinte maneira. Considere uma imagem f em tons de cinza contendo w por h pixels, tal que $w, h \in \mathbb{N}$. Defina uma métrica V sobre a variação das intensidades dos pixels em uma região retangular de f - o desvio padrão das intensidades dos pixels, por exemplo. Seja $f^0 \equiv f$. A imagem f^0 é subdividida em quatro sub-imagens $f_1^1, f_2^1, f_3^1, f_4^1$ quando $V(f^0)$ é superior a um limite pré-estabelecido α. Este procedimento é aplicado recursivamente sobre todas as sub-imagens até que a convergência seja obtida, de forma que cada folha f_j^i represente um bloco suficientemente homogêneo, ou seja, $V(f_j^i) \leq \alpha$.

Entretanto, a abordagem clássica de subdivisão do espaço é pouco eficiente no caso geral. Isso porque a ocorrência de estruturas arredondadas implicam em um maior nível de subdivisão devido à natureza discreta da *Quadtree*. A subdivisão em células de mesmo tamanho gera um grande número de nós intermediários, o que tipicamente pode ser reduzido usando outros tipos de árvores similares, como a Kd-tree (BENTLEY, 1975).

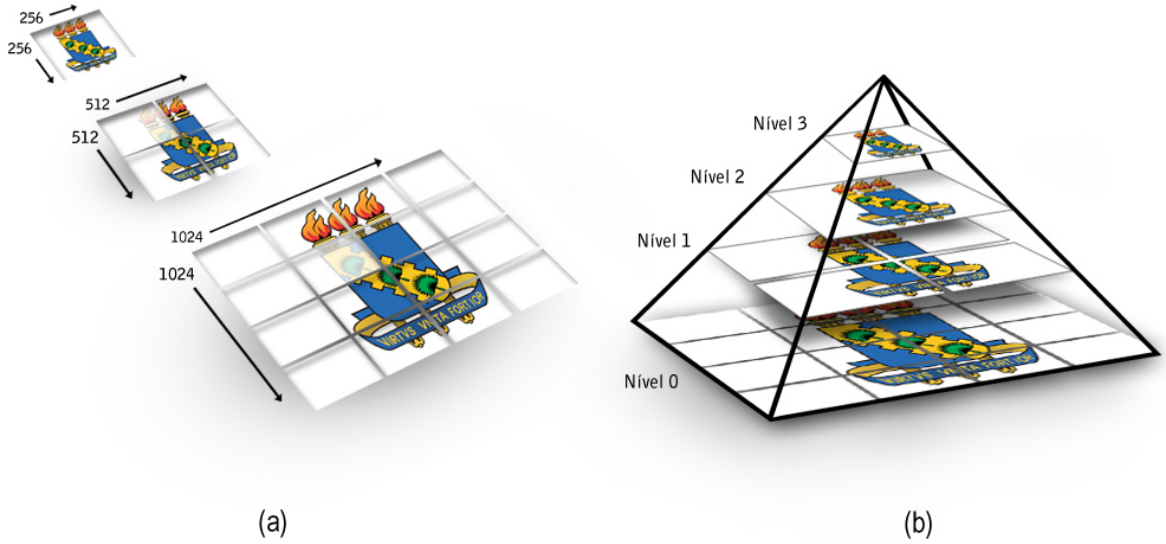

Figura 4: Exemplos de representação de imagens através de pirâmides regulares. A imagem é sucessivamente reduzida a um quarto do seu tamanho, tipicamente combinando um filtro passa-baixas com subamostragem ou apenas por simples subamostragem (a). Cada redução resultante desse processo corresponde a um nível da pirâmide, a imagem original ocupa a base dessa, como ilustrado à direita (b).

A subdivisão recursiva do espaço fornece uma representação multirresolução da imagem original que é extremamente útil para diversos problemas, principalmente aqueles relacionados à Computação Gráfica. Entretanto, a natureza hierárquica da estrutura de dados resultante dificulta operar sobre regiões vizinhas da imagem original através de algoritmos, pois essas podem estar representadas em diferentes níveis da árvore. Uma inconveniência similar ocorre quando é preciso relacionar quadrantes espacialmente vizinhos, mas que se encontram representadas em ramos distintos da árvore.

2.2.2 Pirâmides de Imagens

A idéia de representar imagens usando uma pilha de imagens com níveis decrescentes de resolução espacial, como uma pirâmide, foi proposta quase que simultaneamente por (BURT, 1981) e por (CROWLEY, 1981). Pirâmides de imagens são amplamente usadas em computação gráfica para representar a textura de objetos em diferentes níveis de detalhe (BOARD *et al.*, 2007).

Uma pirâmide de imagens é definida da seguinte maneira. Dada a imagem original f, é derivada uma família de imagens $f_0, f_1, f_2, ..., f_n$, na qual a imagem f_0 representa o nível da base da pirâmide, que corresponde à imagem original e por isso possui a melhor resolução espacial. A imagem f_{i+1} é criada a partir de f_i da seguinte maneira:

- Aplica-se um filtro passa-baixas a f_i, suavizando o sinal da imagem;
- A imagem f_{i+1} é então obtida a partir da subamostragem da imagem suavizada.

A forma com que essas operações de suavização e sub-amostragem são realizadas determina o fator de redução da resolução da imagem original ao longo dos níveis da pirâmide.

Tipicamente, as dimensões da imagem original são reduzidas pela metade à medida que o nível de detalhe diminui, resultando em pirâmides regulares (LINDEBERG, 1994). Pirâmides irregulares podem ser obtidas através de variações nas etapas de suavização e de subamostragem. O resultado desse processo é ilustrado pela Figura 4.

A resolução espacial das imagens em uma pirâmide decresce exponencialmente à medida que o nível de detalhe diminui, o que reduz drasticamente o esforço computacional necessário ao armazenamento e ao processamento dos níveis de detalhe que possuem estruturas de maior escala (CROWLEY, 1981).

Contudo, uma mera computação de uma descrição em múltiplas escalas não atende às necessidades de uma representação adequada à visão computacional. A representação de imagens através de pirâmides corresponde a uma quantização grosseira ao longo da escala, o que dificulta relacionar estruturas ao longo das escalas através de algoritmos (LINDEBERG, 1991). Além disso, dependendo de como os processos de suavização e subamostragem são realizados, as propriedades analíticas necessárias à computação de direções preferenciais podem ser comprometidas.

2.2.3 Integral Image

Também conhecida como *Summed Area Table*, *Integral Image* é uma técnica para o cálculo rápido e eficiente da soma dos valores numa grade regular. Tal técnica foi introduzida primeiramente no universo da Computação Gráfica (CROW, 1984) para reduzir o tempo necessário à geração de pirâmides de imagens usadas para acelerar os cálculos envolvidos no mapeamento de texturas em múltiplos níveis de detalhe. A Integral Image, $II(x, y)$, construída a partir de uma imagem $I(x, y)$ contendo w por h pixels é definida da seguinte maneira.

$$II(x, y) = \sum_{x' \leq x, y' \leq y} I(x', y') \tag{2.2}$$

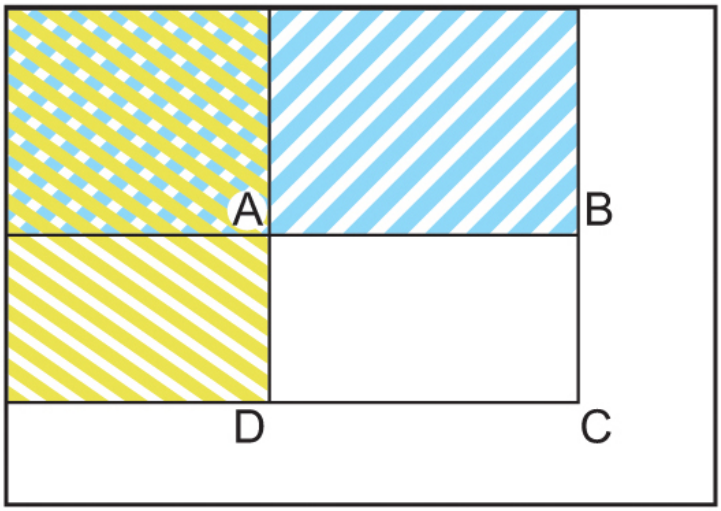

Figura 5: Cálculo da soma dos pixels no retângulo denotado pelos vértices A, B, C e D. Como $II(C)$ contém $II(B)$ e $II(D)$, estes devem ser subtraídos de $II(C)$ para que se obtenha a soma na região $ABCD$. Nesse processo, $II(A)$ foi subtraído duas vezes por estar contido tanto em $II(B)$ quanto em $II(D)$, devendo ser acrescido ao resultado. Tem-se então $II(A) + II(C) - II(B) - II(D)$.

$II(x, y)$ também pode ser definida de forma recursiva.

$$II(1, 1) = I(1, 1) \tag{2.3}$$

$$II(x, y) = I(x, y) + II(x - 1, y) + II(x, y - 1) - II(x - 1, y - 1) \tag{2.4}$$

Esta definição é mais conveniente, pois permite calcular o valor de $II(x, y)$ para todo $x, y \in [1, w] \times [1, h]$ em tempo linear sobre o número de pixels da imagem, acelerando consideravelmente o processo.

Uma vez construída $II(x, y)$, é possível calcular a soma dos valores dos pixels numa região retangular denotada pelos vértices A, B, C e D de forma extremamente eficiente (vide Figura 5). Computacionalmente, esse cálculo requer apenas quatro acessos à tabela, uma adição e duas subtrações, indiferentemente da área que o retângulo ocupa na imagem original $I(x, y)$. Assim, a estrutura de dados pode ser usada para calcular a média dos valores em uma região retangular de dimensões arbitrárias, produzindo assim uma representação capaz de descrever a imagem em múltiplas escalas.

$$\sum_{A(x)<x'\leq C(x), A(y)<y'\leq C(y)} I(x', y') = II(A) + II(C) - II(B) - II(D) \tag{2.5}$$

Observe que é possível construir uma representação semelhante que contenha a soma do quadrado dos valores de cada pixel da imagem original. Tal representação pode ser usada para acelerar cálculos estatísticos.

A técnica de *Integral Image* foi utilizada por Viola & Jones (2004) para acelerar sensivelmente os cálculos de Wavelets de Haar (HAAR, 1910) em escalas arbitrárias no seu *framework* para a detecção de objetos, uma vez que a operação básica para tanto corresponde justamente ao cálculo de um somatório dentro de uma região retangular. Como resultado, os autores foram capazes de produzir o primeiro método reconhecidamente robusto e simultaneamente capaz de detectar faces em tempo-real.

Entretanto, $II(x, y)$ é claramente incapaz de representar variações sobre a orientação, por utilizar uma região retangular de amostragem, que, inevitavelmente, não pode captar as variações de intensidades de pixel em diferentes ângulos de forma homogênea. Para essa finalidade, uma região circular seria mais adequada. Para retângulos orientados em 45º, Lienhart and Maydt (2002) propuseram uma adaptação da representação por *integral image* denominada $rsat(x, y)$ – *Rotated Summed Area Table*. Esta estrutura gera a soma dos pixels no retângulo rotacionado em 45º em relação à imagem original, e que se estende até os limites da imagem tal que o canto mais à direita está localizado em (x, y),

$$rsat(x, y) = \sum_{x' \leq x, x' \leq x - |y - y'|} I(x', y') \tag{2.6}$$

Essa estrutura pode ser construída eficientemente usando duas iterações sobre a imagem original. Analogamente a $II(x, y)$, a soma dos pixels na região retangular denotada pelos vértices L_1, L_2, L_3 e L_4 pode ser computada usando apenas quatro acessos através de $rsat(L_4) + rsat(L_1) - rsat(L_2) - rsat(L_3)$.

Contudo, essa aproximação ainda é incapaz de lidar com outros ângulos. Geralmente essa deficiência é contornada usando algum mecanismo de aprendizagem capaz de absorver um conjunto de exemplos contendo o mesmo objeto em diferentes orientações. Em outras palavras, essa deficiência deve ser tratada pelas aplicações desse método.

2.2.4 Teoria do Espaço de Escala

Dentro do contexto de visão computacional, a Teoria do Espaço de Escala destaca-se entre os demais métodos de representação multiescala porque provê meios para relacionar diferentes escalas de maneira natural e organizada através de uma descrição qualitativa do sinal que representa a imagem original (WITKIN, 1983). Além disso, essa representação

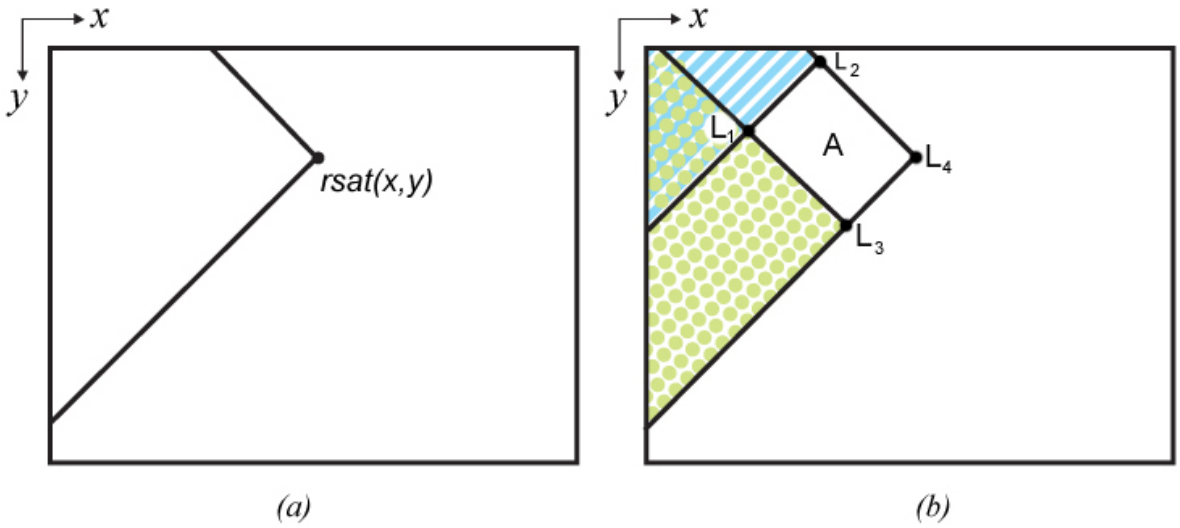

Figura 6: (a) Definição de $rsat(x, y)$, *Rotated Integral Image* (LIENHART; MAYDT, 2002). (b) A soma dos pixels na região retangular denotada pelos vértices L_1, L_2, L_3 e L_4 pode ser computada como $rsat(L_4) + rsat(L_1) - rsat(L_2) - rsat(L_3)$.

permite lidar efetivamente com a ambiguidade da descrição de estruturas em diferentes escalas (LINDEBERG, 1994; LINDEBERG, 1998; LOWE, 1999). Do ponto de vista teórico, assume-se que as coordenadas $(x, y) \in \mathbb{R}^2$ para que se utilizem equações definidas no domínio contínuo.

Historicamente, a teoria do espaço de escala foi elaborada primeiro para os sinais unidimensionais e depois estendida para as imagens. Intuitivamente, o sinal original da imagem $I(x, y)$ é embutido em uma família de sinais derivados $L_I(x, y, t)$, tal que $L_I(x, y, 0) \equiv I(x, y)$ e t representa o nível de simplificação da imagem. Assim, as estruturas presentes na imagem em escalas mais grosseiras (t grande) são simplificações das estruturas correspondentes em escalas mais finas (t pequeno). Quando define-se $L_I(x, y, t)$ como $G(x, y, t) * I(x, y)$, onde $G(x, y, t)$ denota o núcleo de convolução Gaussiano de variância $\sqrt{t}$, $*$ denota a operação de convolução, a família de funções resultante é denominada *Espaço de Escala Linear*.

Dadas duas funções f e g, $f * g$ é definido de forma que seja invariante à posição u na qual deseja-se "observar" f usando g. Geralmente g é uma função normalizada, ou seja, $\int_{-\infty}^{\infty} g(x)\, dx = 1$.

$$f * g = \int_{-\infty}^{\infty} f(u)g(x - u)\, du \tag{2.7}$$

2.2.4.1 Propriedades

É importante observar que, naturalmente, a simplificação progressiva da imagem favorece sua análise ao remover detalhes desnecessários em escalas maiores, que, caso não fossem removidos, poderiam dificultar seu processamento através de algoritmos. Além disso, a ocorrência de um evento caracterizado em certa escala t fornece evidências sobre o tamanho da estrutura correspondente na imagem original. Essa propriedade de *seleção automática de escala* (LINDEBERG, 1994) é útil, pois permite detectar estruturas em uma imagem e, ao mesmo tempo, estimar com boa precisão as escalas destas estruturas.

Uma vez determinada a escala t' e a posição (x', y') de uma estrutura, é possível associar-lhe uma orientação com base em operadores diferencias aplicados a $L_I(x', y', t')$. Assim, a análise baseada na teoria do espaço de escala permite obter a posição, a escala e a orientação de estruturas importantes presentes na imagem.

Claramente esse tipo de método demanda considerável esforço para a construção do espaço de escala em si. Por outro lado, a teoria do espaço de escala fornece um mecanismo elegante e natural para a separação das estruturas presentes em diferentes escalas. Além disso, as propriedades analíticas necessárias à computação consistente de orientações e de operadores diferenciais são preservadas ao longo das escalas.

2.2.4.2 Inspiração Biológica

Há diversos dados neurofisiológicos e psicofísicos que evidenciam a análise multiescala pelo sistema visual primário dos mamíferos (HUBEL; WIESEL, 1962).

Os cones e bastonetes na retina formam campos receptivos de forma aproximadamente circular. Esses campos receptivos possuem uma formidável faixa de variação em relação a seus tamanhos (HUBEL, 1988). Além disso, o perfil de sensibilidade que varia do centro para a periferia da região circular é muito similar ao Laplaciano da função Gaussiana (YOUNG, 1986).

Os campos receptivos na retina projetam-se pelos axônios do nervo óptico para o Núcleo Geniculado Lateral (NGL) situado no tálamo, de onde são transmitidos ao córtex visual no lobo occipital. Os perfis dos campos receptivos do córtex visual guardam forte semelhança com as derivadas do núcleo Gaussiano, ainda que tenham sido propostos para fornecer uma taxonomia adequada aos muitos tipos de células ali encontradas (KOENDERINK; van Doorn., 1988).

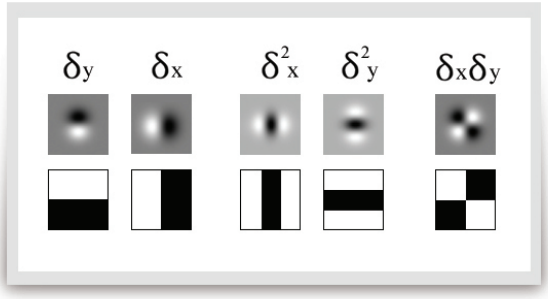

Figura 7: Semelhança entre as wavelets de Haar usadas por (VIOLA; JONES, 2004) e as derivadas parciais do núcleo Gaussiano bidimensional. Essa semelhança pode explicar por que o método desenvolvido por esses autores é capaz de expressar objetos complexos, a exemplo das faces humanas.

Essas evidências sugerem que o uso de uma representação baseada em espaços de escala lineares é capaz de reproduzir, com um bom nível de fidelidade, as respostas usadas pelo cérebro dos primatas para interpretar imagens de forma independente de escalas.

De fato, mesmo que as wavelets de Haar possam ser interpretadas como uma aproximação grosseira dos operadores diferenciais aplicados ao núcleo Gaussiano (Figura 7), o trabalho desenvolvido por (VIOLA; JONES, 2004) corrobora para demonstrar o potencial desse tipo de representação de imagens para efetuar tarefas de Visão Computacional. Em particular, esses autores desenvolveram um método para a detecção de faces humanas, que são objetos complexos.

Nesse contexto, o próximo Capítulo destina-se à apresentação dos principais aspectos acerca da Teoria do Espaço de Escala, assim como da utilização dessa teoria na construção de descrições robustas que expressem os objetos a serem reconhecidos através de algoritmos.

3 Espaços de Escala e Descritores Locais

A ideia intuitiva que motiva o uso de um espaço de escala como representação de imagens é prover a separação entre estruturas com base nos níveis de detalhe aos quais tais estruturas correspondem. Essa heurística é usada para detectar estruturas presentes na imagem, bem como suas respectivas posições, orientações e escalas. Tais informações permitem construir uma representação de cada estrutura, e, considerando sua vizinhança na imagem e a geometria local observada em e. Como resultado, obtém-se uma descrição *local*, que, portanto, independe da posição, da orientação e da escala de e. Tais descrições associadas às estruturas detectadas na imagem são denominadas descritores locais.

O restante deste capítulo está organizado da seguinte forma. Primeiramente, são apresentados os principais conceitos envolvendo os aspectos teóricos sobre o núcleo Gaussiano e a construção de espaços de escala. Em seguida, é feito um apanhado geral sobre descritores locais.

3.1 Espaços de Escala

3.1.1 O Núcleo Gaussiano e suas Propriedades

No Capítulo 2, foi visto que o Espaço de Escala Linear é construído através da convolução da imagem original I com um núcleo Gaussiano de variância t arbitrária. Antes de justificar a adoção desta função neste processo e mostrar que a mesma é única sob certas condições, define-se o núcleo Gaussiano e discutem-se suas principais propriedades analíticas do ponto de vista computacional.

A equação da função Gaussiana de variância t é definida em uma dimensão como

$$G(x,t) = \frac{1}{\sqrt{2\pi t}} e^{-\frac{x^2}{2t}} \tag{3.1}$$

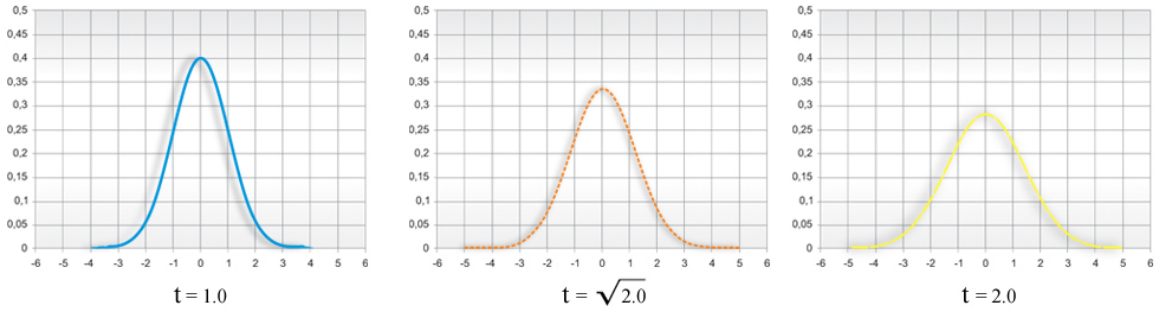

Figura 8: Gráficos para o núcleo Gaussiano unidimensional. Estão inclusos $G_1(x)$, $G_{\sqrt{(2)}}(x)$ e $G_2(x)$. Percebe-se que $G_t(x)$ é uma função ímpar cujo máximo ocorre quando $x = 0$. Observe que à medida que t cresce, mais pontos distantes do centro passam a ser considerados por este operador.

Por conveniência e simplicidade de notação, denota-se a função Gaussiana como G_t quando não estiver atrelada a um domínio. Note-se que a variância de um núcleo Gaussiano é de fato denotada por t. Tomando a transformada bilateral de Laplace de $G_t \equiv G(x,t)$,

$$\mathcal{L}_{G_t} = \int_{-\infty}^{\infty} e^{-st} G_t \, dt = e^{\frac{1}{2}tx^2} \tag{3.2}$$

pode-se derivar a variância como $\mathcal{L}''_G(0) = t$. Claramente, essa função é par. A média de G_t é zero, pois assume valores positivos em um pequeno intervalo centralizado na origem. Isto pode ser visto na Figura 9. Além disso, $G(x,t)$ também é uma função normalizada para qualquer constante t.

G_t também possui outra propriedade bastante interessante do ponto de vista da análise de imagens. Ele é capaz de atenuar rapidamente os sinais referentes ao ruído à medida que t cresce. A convolução com G_t fornece um filtro passa-baixas capaz de eliminar estruturas de tamanho inferior ao seu desvio padrão. Tal fato pode ser demonstrado através do Teorema de Green ou usando variáveis complexas (TEIXEIRA, 2001).

$$\int_{-\infty}^{\infty} G(x,t) \, dx = 1 \tag{3.3}$$

Por sua vez, o núcleo Gaussiano bidimensional de variância t é expresso como

$$G(x,y,t) = G(x,t)G(y,t) \tag{3.4}$$

$$G(x,y,t) = \frac{1}{2\pi t} e^{-(x^2+y^2)/2t} \tag{3.5}$$

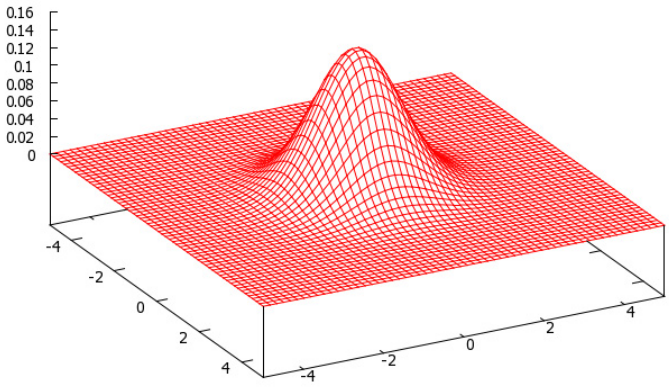

Figura 9: Gráfico para o núcleo Gaussiano bidimensional $G_1(x, y)$.

Desta definição, segue-se que o núcleo Gaussiano bidimensional é claramente uma função separável e rotacionalmente simétrica. A separabilidade é uma propriedade extremamente interessante do ponto de vista computacional: quando definido em $\mathbb{R}^n$ tal que $n > 1$, o núcleo pode ser separado (decomposto) na forma de uma série de convoluções elementares. Considerando o caso 2D, $G_t * I$ pode ser computado através de uma convolução 1D na direção x encadeada com o resultado de outra convolução 1D na direção y.

$$G(x, y, t) * I(x, y) = G(x, t) * (G(y, t) * I(x, y)) \tag{3.6}$$

De fato, o núcleo Gaussiano é o único operador circularmente simétrico que pode ser decomposto desta forma. Além disso, os núcleos elementares usados nessa decomposição são idênticos, exceto por sua orientação. Os programas de computador tiram proveito dessa propriedade, minimizando tanto o número de operações matemáticas envolvidas quanto o espaço necessário em memória.

A convolução de dois núcleos gaussianos de variâncias t_1 e t_2 produz um núcleo gaussiano de variância igual a $t_1 + t_2$. Ou seja, $G_{t_1} * G_{t_2} = G_{t_1+t_2}$. Ou seja, a propriedade de semigrupo é válida. Essa propriedade pode ser explorada para acelerar o cálculo de L_I de várias formas:

- Basta computar uma série de convoluções usando núcleos gaussianos G_{t_i} tais que $\sum t_i = t$, para que se obtenha o resultado da convolução com o núcleo correspondente a uma escala t qualquer. Como na prática isso envolve a computação de amostras do núcleo Gaussiano cujo número aumenta a medida que t cresce, esse processo passa a exigir menos memória e menos processamento;

- Os núcleos correspondendo às escalas t_i podem ser pré-amostrados e subsequentemente armazenados para simplificar a computação de convoluções. Consequentemente, a construção de uma sequência de imagens representando a iteração sobre diferentes escalas pode ser realizada de forma mais eficiente.

3.1.2 Espaço de Escala Linear

Há várias opções de filtros passa-baixas que podem ser usados para simplificar uma imagem. Então, por que escolher justamente o núcleo Gaussiano como função geradora do espaço de escalas? Justamente porque o núcleo Gaussiano é único para a construção de um espaço de escala que **não estabelece premissas** quanto à finalidade de representação dos objetos, ou seja, toda a informação é tratada da mesma forma. Pode-se constatar isso ao se estabelecer um conjunto de condições básicas desejáveis por uma representação capaz de lidar com imagens sem que sejam fornecidas informações *a priori*:

- Invariância a translação. Como não há posição preferencial, a representação deve ser considerada homogênea no domínio de I;
- Invariância a orientação. A representação deve ser *isotrópica*, ou seja, não deve haver uma orientação preferencial para que se observem quaisquer propriedades.

Foi constatado por Koenderink (KOENDERIK; DOORN, 1984) que a equação geradora do espaço de escala linear corresponde à equação de difusão linear. Essa equação diferencial parcial descreve a variação da temperatura com o tempo t em uma região *isotrópica* e *homogênea*, $I(x, y)$, a partir de uma distribuição inicial do calor sobre a superfície. A equação de difusão linear impõe as mesmas restrições usadas para construir um espaço de escala linear. Assim, temos as seguintes restrições assumindo que t representa o tempo decorrido desde que a difusão do calor iniciou-se a partir de uma distribuição inicial sobre a superfície

$$\frac{\partial L_t}{\partial t} = \frac{\partial^2 L_t}{\partial x^2} + \frac{\partial^2 L_t}{\partial y^2} \tag{3.7}$$

$$L(x, y, 0) = I(x, y) \tag{3.8}$$

que são prontamente satisfeitas por $G_t(x, y) * I(x, y)$.

Por sua definição, percebe-se que a equação de calor impõe a adoção de uma solução capaz de satisfazer simultaneamente as condições básicas enumeradas anteriormente. Além disso, a definição dessa solução como um núcleo de convolução é satisfeita apenas pelo

núcleo Gaussiano. Observe que o conceito de escala é abstraído ao passo que podemos produzir L_t para qualquer valor arbitrário de t.

Há outras formas de se obter esta derivação do núcleo Gaussiano como função geradora do espaço de escala: por exemplo, através de uma abordagem axiomática sobre as propriedades de tal núcleo (WITKIN, 1983). Por outro lado, outros tipos de espaços de escala podem ser obtidos quando se relaxa a restrição (3.7), o que implica na construção, através de uma família de funções, a partir de um processo de difusão possivelmente *anisotrópico* ou *heterogêneo*. Esse tipo de espaço de escalas é denominado não-linear, e será visto mais adiante.

O espaço de escala gerado através da convolução com o núcleo Gaussiano é conhecido como Espaço de Escala Linear ou Espaço de Escala Gaussiano. Este possui as seguintes propriedades básicas (VELHO *et al.*, 2000):

- **Invariância a Translação**, pois L_t é gerada através do operador de convolução, que é sabidamente invariante a translação;
- **Linearidade**, ou seja, o mapeamento L_t do sinal original f no espaço de escala é uma transformação linear $L_{f+\lambda g}(x, y, t) = L_f(x, y, t) + \lambda L_g(x, y, t)$
- **Causalidade**. O sinal é simplificado à medida que t (a escala) cresce. Em outras palavras, o número de pontos críticos que correspondem aos picos e às depressões característicos da imagem em uma dada escala não aumentam à medida que t cresce. Entretanto, essa propriedade é válida apenas para o caso unidimensional (VELHO *et al.*, 2000). Apesar disso, é razoável admitir que a simplificação com o crescimento de t ocorre na maioria dos casos.

3.1.3 Espaços de Escala não-Lineares

Todas as regiões da imagem original são progressivamente suavizadas de forma homogênea durante a construção dos Espaços de Escala Lineares. Esse processo inclui regiões de possível interesse como as arestas. Com isso, as arestas podem se mover devido a suavização de sua periferia, o que pode reduzir a precisão na sua localização. Isso acontece porque o paradigma adotado visa o não comprometimento da porção "primária"do sistema de visão computacional, pois toda a informação é processada da mesma maneira.

Contudo, é possível relaxar essa restrição reduzindo a apenas dois os principais aspectos que caracterizam uma representação multiescala como sendo um espaço de escala.

A imagem deve ser simplificada à medida que a escala cresce *(1)*. Deve ser possível estabelecer relações sobre diferentes escalas *(2)*.

Seguindo essa linha de pensamento, Perona e Malik (PERONA; MALIK, 1992) foram os primeiros a propor uma definição alternativa do conceito de espaços de escala no qual o processo de difusão ocorre de forma anisotrópica. Os autores mostraram que, controlando o fator de difusão, é possível favorecer a simplificação de áreas homogêneas ao mesmo tempo em que as informações pertinentes às arestas são preservadas. Essa abordagem mostrou-se particularmente efetiva para efetuar a segmentação progressiva de imagens, e também para produzir imagens simplificadas cujas arestas são preservadas. Uma aplicação direta é a análise e o processamento de imagens médicas. Contudo, esse tipo de espaço de escala é mais lento de se construir do que a tradicional abordagem linear. Além disso, esse processo é geralmente orientado à preservação de arestas.

É possível, também, desenvolver o conceito de espaço de escalas usando morfologia matemática. Nesse caso o espaço é caracterizado por uma pilha de imagens erodidas ou dilatadas usando um elemento estruturante de tamanho proporcional à escala. O leitor interessado em maiores detalhes do relacionamento entre os espaços de escala e a teoria dos reticulados é convidado a ler os seguintes trabalhos nessa área: (BROCKETT; MARAGOS, 1992), (JACKWAY, 1992) e (BOOMGAARD; SMEULDERS, 1994).

Apesar desses conceitos serem extremamente úteis para a área biomédica, por exemplo, o escopo desta tese limita-se ao uso do espaço de escala linear, visto que esse não compromete a percepção da pose e do tamanho dos objetos.

3.2 Detecção e Reconhecimento de Objetos usando a Teoria do Espaço de Escala

Métodos clássicos como PCA (TURK; PENTLAND, 1991), LDA (ETEMAD; CHELLAPPA, 1997) e ICA (BARTLETT *et al.*, 2002) tentam assistir a separação de imagens representando objetos (faces, por exemplo) ao propiciar transformações que são capazes de, respectivamente: maximizar as diferenças entre os elementos amostrados de acordo com sua distribuição; minimizar as diferenças inter-classes e maximizar as semelhanças intraclasse; e prover independência a estatística.

Contudo, tais métodos são fortemente influenciados pela iluminação do ambiente e pelo cenário de fundo. Portanto é extremamente difícil determinar quais aspectos foram realmente considerados para construir uma dada transformação dos dados. Outra

deficiência é que a noção de escala não pode ser aplicada aqui: qualquer objeto a ser reconhecido deve ser primeiro *detectado*, extraído e normalizado geometricamente antes que possa ser submetido ao processo de reconhecimento. Por esse motivo, os resultados obtidos com esse tipo de método não são muito animadores.

De uma forma geral, a grande maioria dos métodos concebidos para o reconhecimento de objetos estão fortemente atrelados a uma etapa de detecção e outra etapa de normalização. Consequentemente, a precisão com que cada uma dessas etapas é realizada influencia sensivelmente a qualidade final dos resultados obtidos através desse tipo de abordagem.

E se fosse possível obter correspondências entre duas imagens de forma direta, sem que fosse preciso utilizar um grande conjunto de amostras para treinamento? Os métodos baseados em descritores locais proveem meios para viabilizar esse tipo de abordagem.

3.2.1 Descritores Locais

Ao se adotar a abordagem de descritores locais, a tarefa de obter correspondências entre imagens é dividida em três etapas:

- Seleção de pontos-chaves, em que se detectam estruturas que caracterizam a imagem. A posição, a orientação e a escala de cada estrutura também são obtidas nessa etapa;
- Associação de descritor local, na qual cada cada ponto-chave tem sua vizinhança representada por um vetor de "características";
- Determinação de correspondências entre pontos-chaves, etapa na qual determina-se a correspondência entre duas imagens com base na correspondências entre os descritores locais que representam as partes importantes dessas imagens.

Dessa maneira, um conjunto $K_I = \bigcup p_j$ de pontos *notáveis* da imagem I é obtido durante a etapa de seleção de pontos-chaves. Geralmente, tem-se três tipos de estruturas como resultado: bolhas, cantos e junções. Os métodos geralmente observam a variação de alguma propriedade ao longo da escala para detectar tais estruturas, determinando, portanto, a posição e a escala das mesmas. Após isso, uma orientação é tipicamente associada ao ponto-chave com base no comportamento do gradiente na vizinhança desse ponto (LOWE, 2004; BAY *et al.*, 2008; MIKOLAJCZYK; SCHMID, 2004).

A propriedade mais importante de um detector de pontos é a sua repetibilidade, ou seja, um detector de pontos-chaves ideal deve selecionar exatamente o mesmo conjunto de pontos sob diferentes condições de visualização da mesma cena ou do mesmo objeto (BAY *et al.*, 2008). Essa propriedade claramente favorece a obtenção de bons resultados durante a etapa de determinação de correspondências entre pontos-chaves, uma vez que os mesmos pontos-chaves estão disponíveis em duas imagens representando o mesmo objeto ou cena.

Sob este aspecto surge um dilema fundamental em vários contextos. Um detector deve reportar um grande número de pontos-chaves pouco estáveis ou um menor número de pontos-chaves muito estáveis? Obviamente, responder a esta questão é uma tarefa que requer o escrutínio sobre um problema particular. Do ponto de vista prático, geralmente é melhor obter o menor conjunto possível de pontos desde que todos os seus elementos sejam considerados estáveis, pois a computação ocorre de forma eficiente sem comprometer o resultado final. Entretanto, quanto maior o número de pontos-chaves, maior a redundância na representação e consequentemente espera-se que maior resistência as oclusões seja obtida nesse caso.

Durante a segunda etapa, um vetor de *características* é computado para cada ponto-chave p_j com base em medidas extraídas a partir da vizinhança desse ponto no espaço de escala. Esse vetor é denominado descritor local. De um modo geral, os descritores locais tendem a ser naturalmente resistentes ao ruído, uma vez que o próprio conceito de espaço de escala está ligado à suavização progressiva da imagem através de um filtro passa-baixas. Há duas importantes propriedades básicas (LOWE, 2004) que um descritor local ideal deve possuir:

- Invariância a rotação e a escala. A orientação e a escala associadas a um ponto-chave geralmente são consideradas no cálculo do vetor de características, obtendo assim o resultado esperado;
- Capacidade de discernimento entre os demais descritores computados a partir de outros pontos-chaves, colaborando assim para reduzir o número de falsas correspondências.

Além disso, também é desejável que o descritor local seja robusto com relação a erros de detecção, a distorções geométricas devidas a diferentes pontos de vista tridimensionais, e a deformações fotométricas causadas por diferentes condições de iluminação (MIKOLAJCZYK; SCHMID, 2004).

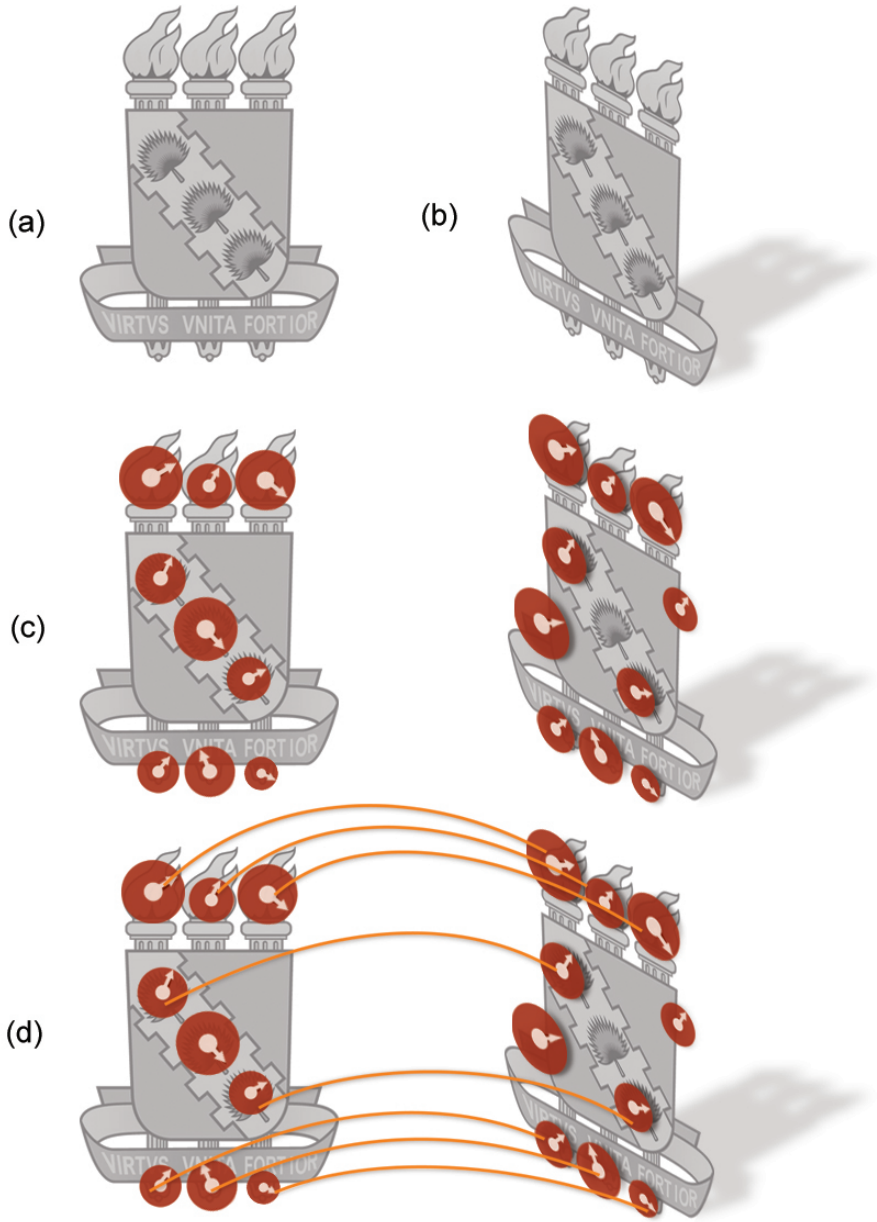

Figura 10: Ilustração do processo de detecção e reconhecimento de objetos usando o método de descritores locais. Múltiplos pontos-chaves (c) são detectados a partir de uma imagem de entrada (a) em diferentes configurações de posição, orientação e tamanho. Em seguida, um descritor é computado para cada ponto-chave em seu sistema local de coordenadas. Esses descritores são usados para determinar associações entre os pontos detectados em duas imagens distintas (c), efetivamente detectando correspondências entre objetos presentes nas duas imagens.

Na última etapa, os pontos-chaves pré-computados K_O representando um objeto são comparados com os pontos-chaves K_I computados sob demanda a partir de uma imagem de teste I. Isso é realizado com base nos descritores locais associados a cada ponto-chave. Na prática, uma métrica de similaridade é usada para essa comparação. Geralmente a Distância Euclidiana entre os descritores (LOWE, 2004; MIKOLAJCZYK; SCHMID, 2004; BAY *et al.*, 2008) é adotada para que se obtenham os pares de pontos correspondentes p_o, p_i tais que $p_o \in K_O, p_i \in K_I$. A devida computação de quaisquer métricas de similaridade depende da dimensão do descritor, que pode variar de acordo com o método adotado. Por causa disso, é importante que aplicações interativas reduzam a dimensionalidade dos descritores ao mínimo possível, de tal forma que a capacidade de discernimento entre os descritores não seja prejudicada.

3.2.1.1 Vantagens

A abordagem baseada em descritores locais apresenta várias vantagens sobre os demais métodos para a detecção e o reconhecimento de objetos, das quais as seguintes se destacam:

- Detectar e reconhecer um objeto são tarefas realizadas simultaneamente, logo essa abordagem não carece de mecanismos externos sujeitos a erros, como por exemplo normalização geométrica;
- É possível realizar a correspondência direta entre duas imagens. Essa tarefa dispensa etapas de pré ou pós-processamento, como a construção de estruturas de dados complexas ou resolver problemas de otimização global. Em consequência disso este método é amplamente utilizado para construir panoramas a partir de conjuntos de imagens de um mesmo local tomadas sob diferentes ângulos;
- É possível representar a estrutura de um objeto a partir de uma única imagem. Isso significa maior capacidade de generalização, que é obtida por outros métodos através da utilização de sofisticados mecanismos de aprendizagem de máquina;
- Os descritores locais são construídos para prover invariância à pose e ao tamanho dos objetos aparecendo nas imagens. Consequentemente, esse aspecto que é uma grande limitação para a maioria dos outros métodos é superado de forma natural;
- Os descritores locais são, em geral, suficientemente robustos às variações na iluminação, para que sejam usados em aplicações do mundo real;

- Por utilizar uma representação espontaneamente redundante que considera vários pontos-chaves, é possível reconhecer objetos de forma confiável mesmo sob a ocorrência de oclusões.

3.2.1.2 Desvantagens

A principal desvantagem desse tipo de método está no esforço computacional necessário à construção do espaço de escala, etapa que geralmente consome a maior parte do processamento. Note-se que é possível implementar tais métodos usando o hardware gráfico programável ou ainda usando técnicas aproximativas baseadas em números inteiros para que limitações de desempenho sejam contornadas.

Uma desvantagem comum a muitos descritores locais é que eles consideram uma geometria aproximadamente plana para as estruturas encontrados nas imagens. Ou seja, a caracterização de cada ponto-chave comumente limita-se a uma região plana que é observada de um certo ângulo. Contudo, há esforços no sentido de se evidenciar o formato de estruturas (HARRIS; STEPHENS, 1988; MIKOLAJCZYK *et al.*, 2005), particularmente as bolhas. Isso possibilita que a distorção local das estruturas causada pela perspectiva seja revertida, até certo limite, antes que o cálculo dos descritores locais seja efetuado. Consequentemente, esse tipo de abordagem tipicamente utiliza elipses orientadas para descrever a vizinhança dos pontos-chaves que é considerada para o cálculo de descritores, tornando os métodos mais robustos quanto às distorções de perspectiva. Contudo, a capacidade de discernimento entre os pontos-chaves é afetada, principalmente porque, dependendo da perspectiva, duas estruturas podem ser confundidas localmente: por exemplo uma elipse e um círculo legítimos.

Os métodos baseados em descritores locais possuem outra desvantagem, mais sutil. O usuário não influencia no conceito usado para definir quais são as *partes importantes dos objetos* que serão associadas a pontos-chaves e consequentemente, descritores locais. Portanto, esse método tende a funcionar bem no caso geral. Em suma, o processo de seleção de pontos-chaves *não é supervisionado*. Toda a representação produzida por um método de descritores locais depende puramente de alguns poucos parâmetros usados nesse processo e que carregam pouco significado sobre quais são as estruturas locais são preferíveis para se representar um determinado objeto.

Dessa forma, os métodos existentes não consideram o uso de mecanismos supervisionados para que se aprenda um conceito de pontos-chaves importantes para uma certa aplicação (vide Capítulo 4). É possível, por exemplo, que o detector de pontos-chaves

não reporte a boca ou as narinas de uma face devido a sua definição do conceito de "estabilidade". Por outro lado, o mesmo detector pode também reportar pontos-chaves indesejáveis justamente porque não incorpora o conceito de "parte importante"para a aplicação.

4 Trabalhos Relacionados

No capítulo 4, observou-se que a Teoria do Espaço de Escala é uma poderosa ferramenta para derivar uma família de funções, as quais preservam as características de homogeneidade e isotropia da representação, usando um processo de difusão do calor cuja configuração inicial é uma imagem. Consequentemente não há favorecimento de posições ou de orientações ao longo das escalas (Espaço de Escala Gaussiano ou Linear). Essas restrições podem ser relaxadas para que se produzam Espaços de Escala não-Lineares capazes de viabilizar a realização de operações complexas como suavizar arbitrariamente uma imagem preservando a nitidez de suas arestas. Em particular, o espaço de escala *linear* fornece um arcabouço matemático sobre o qual estruturas especiais das imagens podem ser detectadas e representadas usando descritores locais, consistindo portanto em uma abordagem prática e elegante para a detecção e o reconhecimento de objetos em imagens.

A literatura de Visão Computacional é rica em métodos que usam abordagens semelhantes, porém moldadas sob a perspectiva de outras teorias. Por conseguinte, o presente capítulo é dedicado exclusivamente à apresentação e à discussão de trabalhos relacionados desenvolvidos no contexto dos métodos de descritores locais baseados na Teoria do Espaço de Escala. Note-se que há outros detectores na literatura (TUYTELAARS *et al.*, 1999; TUYTELAARS; Van Gool, 2000; TUYTELAARS; GOOL, 2004; KADIR; BRADY, 2001; KADIR *et al.*, 2004; TUYTELAARS; MIKOLAJCZYK, 2008). Porém, como seus resultados quanto à repetibilidade são relativamente pobres (MIKOLAJCZYK; SCHMID, 2005; MIKOLAJCZYK *et al.*, 2005; BAY *et al.*, 2008), os mesmos serão desconsiderados no restante do presente capítulo.

Deve ser observado que os detectores de pontos-chaves encontrados na literatura não utilizam informação alguma *a priori* sobre os pontos que serão detectados, caracterizando esse processo como *não-supervisionado*. De fato, nesse contexto não há como estabelecer o conceito de pontos *falsamente detectados* nem de pontos *não-detectados*. Em (TUYTELAARS; MIKOLAJCZYK, 2008), os autores inclusive defendem que, por isso, o termo mais

correto a se utilizar seria *extrator* de pontos-chaves, de forma que o termo *detector* é empregado no presente trabalho tese porque já foi amplamente difundido na literatura especializada.

4.1 Detector de Cantos de Harris

A abordagem clássica proposta por Harris e Stephens (HARRIS; STEPHENS, 1988) é um aprimoramento da técnica proposta por Moravec – que cunhou o termo *pontos de interesse* durante sua pesquisa sobre navegação automática de veículos. Os autores propuseram usar os autovalores λ_1 e λ_2 da matriz Hessiana,

$$\mathbf{H}_I = \begin{pmatrix} \frac{\partial^2 I}{\partial x^2} & \frac{\partial^2 I}{\partial x \partial y} \\ \frac{\partial^2 I}{\partial y \partial x} & \frac{\partial^2 I}{\partial y^2} \end{pmatrix} \tag{4.1}$$

para a classificação de pontos da imagem. Essa matriz representa um tensor contendo as derivadas de segunda ordem em cada pixel, de forma que os autovalores de $\mathbf{H}_I$ correspondem às curvaturas principais locais naquele ponto da imagem. A curvatura local contém informação suficiente para que cantos sejam detectados de forma confiável.

- Quando λ_1 e $\lambda 2$ são grandes, e $\lambda_1 \sim \lambda 2$, o ponto da imagem representa uma junção;
- Quando $\lambda_1 \gg \lambda 2$ ou $\lambda_2 \gg \lambda 1$, o ponto da imagem pertence a uma aresta;
- Finalmente, quando λ_1 e $\lambda 2$ tendem a zero, a região é considerada plana. Logo, não há pontos de interesse.

Como a computação dos autovalores é computacionalmente cara, os autores propuseram a adoção de uma métrica alternativa para que a detecção de cantos, na qual uma constante k é determinada empiricamente dependendo da aplicação

$$M_c = \lambda_1 \lambda_2 - k\left(\lambda_1 + \lambda_2\right)^2 = \det\left(\mathbf{H}_I\right) - kTr^2\left(\mathbf{H}_I\right) \tag{4.2}$$

Esse operador é invariante a rotação e a deformações lineares na *luminância* na imagem. Apesar disso, essa métrica não é invariante a escala, o que dificulta sua pronta utilização em diversos contextos.

Figura 11: Imagem de um campo de girassóis, usada como entrada para a reprodução do experimento realizado por Lindeberg.

4.2 Detecção de Estruturas com Seleção Automática de Escala

Do ponto de vista de representação, Witkin (WITKIN, 1983) apoiou que uma abordagem na qual as propriedades de uma imagem fossem descritas em termos de operadores diferenciais geométricos. De acordo com o autor, combinações lineares ou não-lineares de derivadas extraídas a partir do espaço de escala seriam adequadas e suficientes para muitas operações de baixo nível. A principal motivação disso é que esse "linguajar" é naturalmente adequado para expressar, simultaneamente, propriedades físicas e geométricas (LINDEBERG, 1994).

Partindo desse pressuposto, Lindeberg observou que as singularidades de tais expressões desempenham um papel fundamental na descrição de características geométricas: mais que isso, tais eventos são absolutamente invariantes ao redimensionamento das coordenadas espaciais. Assim, sugeriu que o estudo da evolução de operadores diferenciais

ao longo da escala *normalizados* seria útil para a detecção de estruturas, de tal forma que a escala dessas estruturas é selecionada *automaticamente*:

- Uma propriedade conhecida dos espaços de escala é que a amplitude das derivadas espaciais geralmente decresce com o *crescimento da escala*. Intuitivamente, espera-se que o valor numérico de tais derivadas diminua à medida que a imagem torna-se mais suavizada – o que é consequência direta da propriedade de *causalidade* do espaço de escala linear;

- De uma forma geral, o resultado numérico dos operadores diferenciais deve ser corrigido usando um fator multiplicativo proporcional à escala t e à ordem do operador. Por exemplo, $\nabla^2 L^2$ deve ser normalizado usando t como fator enquanto o fator t^2 deve ser usado para normalizar o determinante da matriz Hessiana;

- Assim Lindeberg propôs seu princípio para a seleção de escalas, "Na ausência de outra evidência, assumir que a ocorrência de um máximo de uma combinação de derivadas normalizadas sobre as escalas pode ser interpretada como um tamanho característico de uma estrutura correspondente nos dados"(LINDEBERG, 1998). A ocorrência de máximos locais já havia sido usada por outros autores para a detecção de arestas e bolhas, sendo denominada *supressão de não-máximos* (WITKIN, 1983).

A Figura 11 contém um campo de girassóis semelhante àquele usado por Lindeberg em seus experimentos para demonstrar o funcionamento de seu princípio de detecção de seleção automática de escalas durante a detecção de estruturas. Esse experimento foi reproduzido neste trabalho com a finalidade de entender o funcionamento dos operadores Laplaciano e Hessiano para a detecção de estruturas. O resultado desse experimento pode ser visto na Figura 12. Eis algumas observações importantes sobre o uso desses operadores:

- Uma forte resposta é obtida quando o Laplaciano é usado para a detecção de máximos, pois esse operador corresponde ao divergente do gradiente. Por outro lado, o operador Hessiano apresenta respostas mais fortes nos cantos e junções. Justamente por isso tende a reportar uma menor quantidade de pontos-chaves;

- Geralmente a limiarização usando uma constante é usada para que pequenas alterações (como ruído) possam ser desconsideradas durante o processo de detecção de estruturas. Isso também é exemplificado na Figura 12;

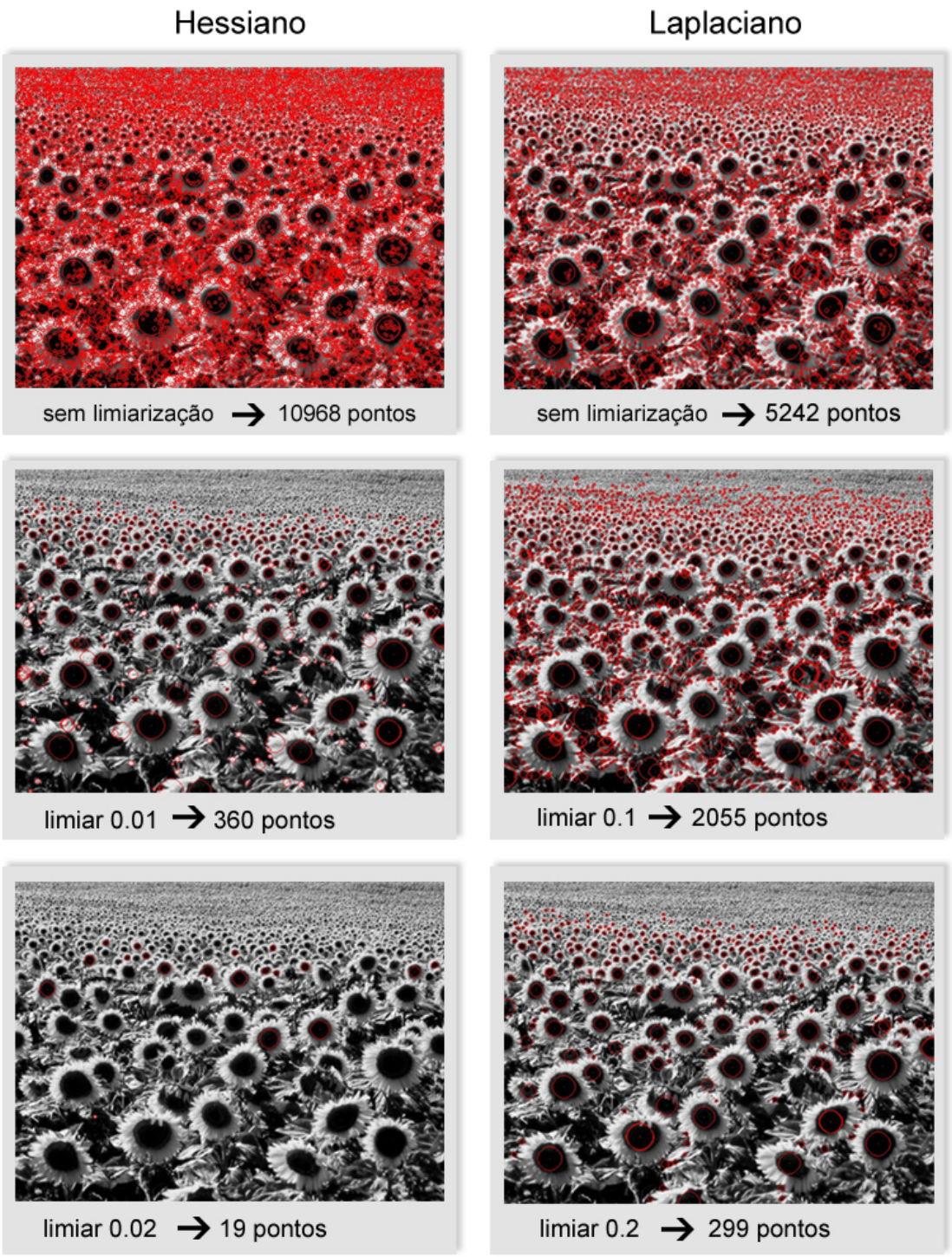

Figura 12: Resultados da detecção de pontos-chaves usando os operadores Hessiano e Laplaciano normalizados na escala para um campo de girassóis. Cada ponto e sua respectiva escala é denotado por um círculo vermelho. Na primeira coluna, são apresentados os pontos-chaves correspondentes aos limiares 0, 0.01 e 0.02, respectivamente, sobre o valor do Hessiano. A segunda coluna contém os resultados para os limiares 0, 0.1 e 0.2, sobre o valor do Laplaciano. O Hessiano é um operador bastante sensível: o número de pontos diminui rapidamente à medida que o limiar cresce.

- Enquanto o Laplaciano é um operador relativamente estável com relação ao limiar escolhido, o Hessiano é bem mais sensível à escolha desse limiar. Isso pode ser resolvido utilizando a raiz quadrada do Hessiano;

- O sinal do Laplaciano fornece informações sobre que tipo de evento caracterizou a estrutura detectada. Valores negativos denotam saliências, enquanto valores negativos correspondem a depressões. Como o determinante do Hessiano é um operador inerentemente quadrático, seu sinal não pode ser usado para determinar se a estrutura detectada corresponde a uma saliência ou a uma depressão;

- De fato, o estudo da evolução desses operadores diferenciais fornece meios para que sejam detectadas estruturas com seleção automática de escala.

Esse princípio de detecção de estruturas foi adotado por outros pesquisadores, os quais propuseram sua utilização no contexto de métodos baseados em descritores locais, como visto a seguir.

4.2.1 SIFT

O método SIFT (*Scale Invariant Feature Transform*) (LOWE, 1999) foi proposto por Lowe há mais de uma década, e proporcionou grande fôlego às abordagens de representação de objetos através de suas partes. De fato, esse trabalho significou um grande avanço em diversas áreas, como, por exemplo: reconhecimento de objetos, navegação automática de robôs, rastreamento e criação de imagens panorâmicas.

Esta técnica utiliza uma representação que combina a computação de uma pilha de imagens contendo níveis de detalhe do espaço de escala linear com pirâmides de imagens. Essa pilha de imagens em vários níveis de detalhe é geralmente denominada *oitavas de Gaussianas*. Cada nível f_i da pirâmide contém uma oitava obtida através da subamostragem sobre oitava f_{i-1}, localizada no nível imediatamente inferior. Esse processo é repetido recursivamente até que se obtenha o nível de representação desejado, geralmente contendo blocos de 8×8 pixels, associado as maiores escalas observáveis. Esse processo é ilustrado pela Figura 13.

Há alguns detalhes importantes sobre a construção dessa representação. Como uma etapa de pré-processamento, a imagem é ampliada por um fator linear de escala 2, efetivamente quadruplicando o número de pixels ao mesmo tempo que a imagem resultante é borrada. No caso, um filtro bilinear é aplicado para que os máximos locais sejam realçados.

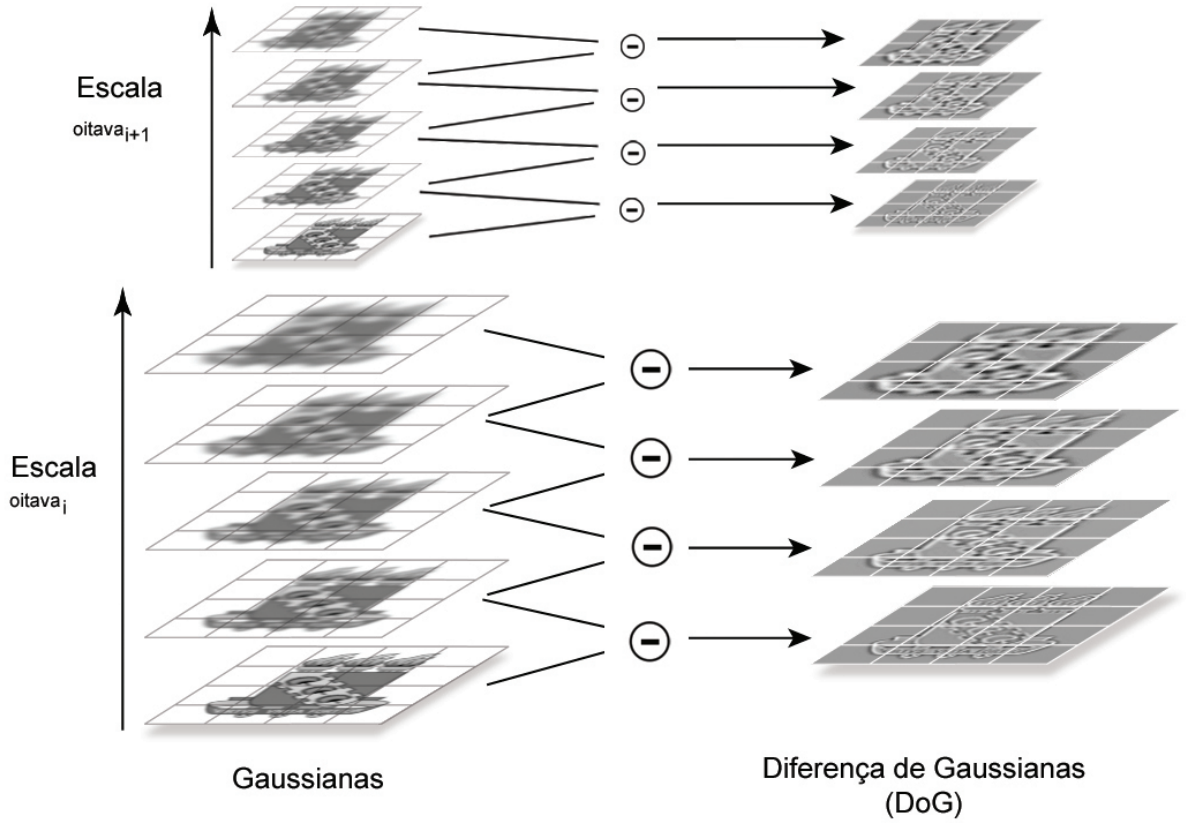

Figura 13: Representação multiescala usada pelo método SIFT (LOWE, 1999). Cada nível da pirâmide contém uma pilha de imagens (uma *oitava*) $f_i = L_{t_i}$ extraídas a partir do espaço de escala linear. Observe que nesse método a complexidade necessária para a detecção de estruturas diminui exponencialmente com respeito à escala em consequência da adoção de uma pirâmide de imagens.

Isso é importante para favorecer a posterior detecção de pontos-chaves estáveis. Contudo, pontos extras passam a ser detectados devido a este tipo de filtragem. Consequentemente, esse método carece de mecanismos inteligentes para que esses pontos extras não influenciem o resultado final.

Além disso, Lowe propôs que a detecção de pontos-chaves fosse feita considerando a Diferença de Gaussianas presentes em uma oitava (*DoG*, *Differences of Gaussian*) como uma aproximação do Laplaciano da Gaussiana (*LoG*, *Laplacian of Gaussian*) normalizado com respeito a escala. Ele mostrou que tal aproximação é válida porque o espaço de escala satisfaz as restrições da equação de propagação do calor. Dessa forma, $\nabla^2 G$ pode ser calculado como uma aproximação por diferenças finitas

$$t\nabla^2 G = \frac{\partial G}{\partial t} \approx \frac{G(x, y, kt) - G(x, y, t)}{kt - t} \tag{4.3}$$

portanto

$$G(x, y, kt) - G(x, y, t) \approx (k - 1)\, t^2 \nabla^2 G \tag{4.4}$$

Ou seja, quando tomadas as diferenças entre Gaussianas que diferem por um fator constante, o termo t^2 necessário para a obtenção de um Laplaciano invariante à escala já está incorporado nesta aproximação. Também observe que, nesse caso, $k-1$ é uma constante. Assim, a detecção de pontos-chaves é feita considerando três níveis consecutivos do *DoG*, como ilustrado na Figura 13. Obviamente, isso significa que o método requer que pelo menos quatro níveis do espaço de escala Gaussiano estejam presentes em cada oitava.

Entretanto, pontos-chaves espúrios podem ser encontrados em dois casos:

- Pontos de baixo contraste que geralmente correspondem a uma amplificação do ruído em decorrência do pré-redimensionamento usando o filtro bilinear;
- Ao longo das arestas, pois o operador *DoG* apresenta fortes respostas nas arestas.

Os pontos de baixo contraste podem ser eliminados considerando apenas 80% do valor original do *DoG* para a detecção de mínimos ou máximos. A resposta nas arestas pode ser controlada usando a razão $Tr\left(\mathbf{H}_L\right)^2 / Det\left(\mathbf{H}_L\right)$ para analisar a curvatura local da região (LOWE, 2004), o que ocorre de forma semelhante ao operador de Harris para a detecção de cantos.

Essa expressão é proporcional a razão entre os autovalores λ_1 e λ_2 da matriz Hessiana que descrevem a curvatura local, mais exatamente $(r+1)^2/r$. O valor $r = 10$ é geralmente escolhido como um limite superior. Além disso, o valor da razão $Tr\left(\mathbf{H}_L\right)^2 / Det\left(\mathbf{H}_L\right)$ deve

ser estritamente positivo, caso contrário considera-se que um ponto degenerado foi obtido.

Brown e Lowe (BROWN; LOWE, 2002) desenvolveram um método para ajustar uma função quadrática $3D$ de forma a determinar a localização interpolada do máximo local correspondente a um ponto-chave detectado usando o operador DoG. O método em questão baseia-se no uso do gradiente descendente para deslocar o ponto detectado em direção ao máximo ou ao mínimo local, melhorando portanto a localização do ponto-chave em relação ao espaço e à escala. Os experimentos apresentados por esses autores demonstraram que esse método é capaz de melhorar substancialmente tanto a estabilidade dos pontos-chaves quanto o resultado final do processo de detecção e reconhecimento.

Apesar de não ser um método novo, SIFT ainda é bastante utilizado na prática em várias aplicações por dois motivos. A detecção de pontos-chaves é realizada de forma eficiente, explorando tanto as propriedades de separabilidade e semigrupo do núcleo Gaussiano quanto a rápida simplificação das oitavas através de uma pirâmide de imagens. O descritor é uma espécie de histograma tridimensional do gradiente em torno do ponto-chave, que tem-se mostrado bastante robusto para várias aplicações (MIKOLAJCZYK; SCHMID, 2004; MIKOLAJCZYK; SCHMID, 2005; OZUYSAL *et al.*, 2007).

Contudo, deve-se observar que a detecção de pontos-chaves é um processo *não-supervisionado*. Ou seja, a despeito de alguns parâmetros (número de oitavas, número de imagens por oitava, limiares, etc) esse método não permite aos usuários prover uma descrição daquilo que considera importante se ter como pontos-chaves. Além disso, os pontos-chaves podem ser detectados na vizinhança de contornos ou em arestas, o que os torna menos estáveis porque sua localização é sensível ao ruido e a pequenas variações de textura (MIKOLAJCZYK; SCHMID, 2004)

Considere por exemplo, uma aplicação de processamento de faces humanas na qual os olhos desempenham um papel fundamental: nesse caso o detector de pontos-chaves deve primar em reportar os olhos em detrimento de outras características, de forma que os olhos devem ser pontos-chaves extremamente estáveis. Claramente, o detector de pontos-chaves do SIFT não está preparado para lidar com esse tipo de situação. Esse exemplo é melhor explorado no Capítulo 6.

4.2.2 Métodos de Mikolajczyk *et al.*

Mikolajczyk and Schmid (MIKOLAJCZYK; SCHMID, 2002) adaptaram os métodos baseados na métrica de Harris e no determinante da matriz Hessiana para que esses detectores

fossem utilizados no contexto do espaço de escalas, de forma a obter métodos de detecção invariante a escala que os autores denominaram *Harris-Laplace* e *Hessiano-Laplace*, respectivamente.

Estendendo esses detectores, Mikolajczyk e outros autores também propuseram o uso de um algoritmo iterativo para a autoadaptação de uma elipse ao formato local das estruturas detectadas (MIKOLAJCZYK *et al.*, 2005). Essa adaptação tem por objetivo computar uma transformação afim local para cada ponto de interesse: os detectores resultantes foram denominados de *Harris-Afim* e *Hessiano-Afim*. Note-se que, em todos esses casos, a detecção de pontos-chaves ***não*** *é supervisionada*.

4.2.2.1 Harris-Laplace

O detector Harris-Laplace é uma evolução natural do detector bidimensional de cantos de Harris para o espaço de escala – aliás, a escala é uma limitação bem conhecida desse método. Basicamente, a matriz Hessiana é derivada para o caso da representação via L_t, obtendo-se uma métrica M_t adaptada à escala local considerando-se $\mathbf{H}_{L_t}$

$$M_t = \det\left(\mathbf{H}_{L_t}\right) - \alpha\, Tr^2\left(\mathbf{H}_{L_t}\right) \tag{4.5}$$

Observe que a detecção de cantos ocorre da mesma maneira que no caso 2D, só que iterando sobre várias escalas L_{t_i} para efetuar a seleção automática de escala. Basicamente o valor obtido de M_t é comparado com seus 8 vizinhos para efetuar a supressão de não-máximos.

4.2.2.2 Hessiano-Laplace

O detector de pontos Hessiano-Laplace funciona de forma semelhante ao Harris-Laplace. De fato, o algoritmo de detecção é praticamente o mesmo: a única diferença está na métrica utilizada. No caso, em uma dada escala t, os extremos locais dos valores do Laplaciano e do Hessiano são detectados simultaneamente. Essa abordagem difere daquela usada por (LINDEBERG, 1998) e (LOWE, 1999) simplesmente por considerar valores extremos do determinante da matriz Hessiana. Ao maximizar o valor do Hessiano, essa abordagem penaliza estruturas muito longas nas quais $\delta^2 x$, $\delta^2 y$ e $\delta x \delta y$ assumem valores muito pequenos, caracterizando mudanças de sinal. Em termos de estabilidade, o detector de Hessiano-Laplace apresenta melhores resultados do que o Harris-Laplace (MIKOLAJCZYK; SCHMID, 2005). Do ponto de vista intuitivo, esse resultado é esperado, pois as bolhas detectadas via Harris-Laplace fornecem um maior volume de informações

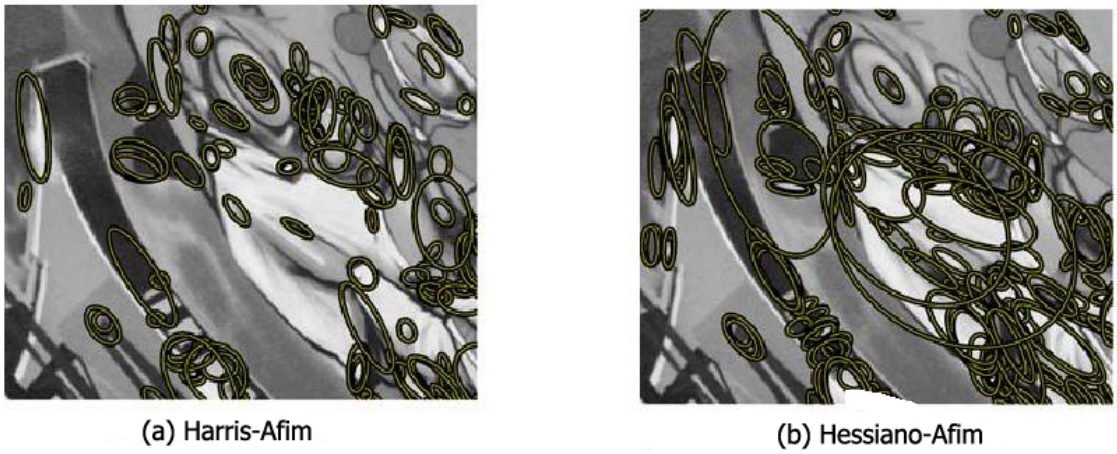

Figura 14: Resultado para a aplicação do detector Harris-Afim (a) e Hessiano-Afim (b) sobre uma mesma imagem de uma ilustração pintada sobre uma superfície plana. Perceba como o detector Harris-Afim reporta cantos obtidos em várias escalas ao passo que o detector Hessiano-Afim geralmente identifica estruturas do tipo bolha.

visuais do que os cantos, já que as bolhas geralmente são caracterizadas por um maior número de pixels.

4.2.2.3 Harris-Afim e Hessiano-Afim

O detector de pontos-chaves Harris-Afim é basicamente uma extensão do método Harris-Laplace. Quando um ponto-chave é detectado, um algoritmo iterativo determina a matriz de derivadas de segunda ordem que transforma a região anisotrópica detectada em uma região normalizada. Como efeito, obtém-se uma região cuja medida de *isotropia* é suficientemente próxima de 1. Para tanto, o algoritmo opera da seguinte maneira:

- Uma *matriz de adaptação de forma* **U** é usada para transformar a imagem para um sistema de coordenadas de referência;
- Os parâmetros de localização (posição e escala) dos pontos-chaves são refinados neste sistema de referência. Na prática, em toda iteração, o algoritmo deve descobrir quais são os vários parâmetros que definem cada região de interesse;
- A matriz de derivadas de segunda ordem é computada nesse sistema de referência e deve possuir uma medida de isotropia cujo valor seja aproximadamente 1 ao final da iteração.

Analogamente, o detector Hessiano-Afim é praticamente idêntico ao detector Harris-Afim, salvo que a métrica de Harris-Laplace é substituída pela métrica usada no detector

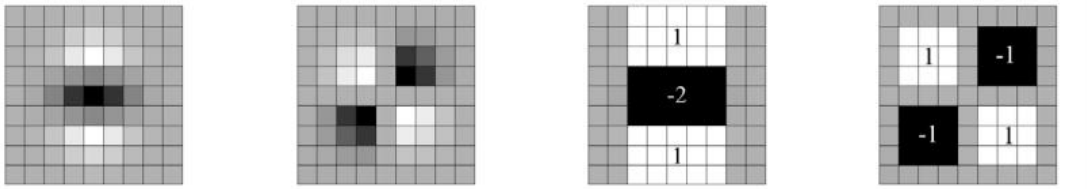

Figura 15: A esquerda, derivadas Gaussianas de segunda ordem nas direções y e xy, após discretização e recorte. A direita, a aproximação usando *box filters* usada por (BAY *et al.*, 2008). As regiões em preto possuem sinal negativo e as regiões em branco possuem sinal positivo. As regiões em cinza possuem valor igual a zero.

Hessiano-Laplace.

4.2.3 SURF

Bay e seus coautores (BAY *et al.*, 2008) propuseram uma versão relaxada do operador HoG na qual wavelets de Haar são usadas para computar uma aproximação das derivadas de segunda ordem do núcleo Gaussiano. Essa aproximação, ilustrada na Figura 15, foi usada por estes autores para a construção do método SURF (*Speedep-Up Robust Features*). De fato, a forma dessas derivadas é muito similar às usadas no trabalho de Viola e Jones (2004).

A detecção de pontos-chaves do método SURF explora o uso de imagens integrais para computar eficientemente uma aproximação do operador HoG em diferentes escalas, o que lhe confere um desempenho de 3 a 7 vezes melhor do que o apresentado pelo método SIFT. As posições detectadas são também refinadas usando interpolação (BROWN; LOWE, 2002), que é realizada com respeito ao valor do determinante da matriz Hessiana. Como o operador HoG apresenta fortes respostas nos cantos e junções, o número de pontos-chaves detectados pelo método SURF geralmente é bem menor do que o número de pontos reportados quando os operadores LoG ou DoG são usados. Apesar disso, em (BAY *et al.*, 2008), os autores afirmam que o seu método reporta pontos-chaves tão estáveis quanto aqueles decorrentes do uso de SIFT.

Os métodos SURF e SIFT apresentam as seguintes diferenças:

- O método SURF usa um modelo *aproximativo* do espaço de escala, que é baseado em imagens integrais;
- Ao contrário, o detector de pontos usado pelo método SURF não necessita que o tamanho original da imagem seja alterado;

- A detecção é baseada na supressão de não-máximos do determinante da matriz Hessiana enquanto o SIFT utiliza uma aproximação do traço dessa matriz. Em consequência disto, SURF tende a detectar cantos e regiões com textura ao passo que SIFT geralmente tende a detectar bolhas e arestas.

4.3 Métodos Baseados em Aprendizagem de Máquina

Métodos de aprendizagem foram utilizados no contexto de detecção de estruturas em imagens. Chen e Rockett (1997) propuseram um modelo estatístico para rotular cantos em imagens. Redes Neurais Artificiais (ROSENBLATT, 1958), por sua vez, foram utilizadas para a detecção de cantos (DIAS *et al.*, 1995). Ambas as abordagens são aplicadas após a detecção de arestas, usando algum método tradicional dependente da escala (CHEN; ROCKETT, 1997). Tsai (TSAI, 1997) explorou uma idéia semelhante àquela utilizada em (DIAS *et al.*, 1995) para aprimorar a estabilidade de medidas de curvatura. Entretanto, esses trabalhos não consideram a detecção de estruturas em múltiplas escalas.

A idéia de usar as diferenças entre as intensidades dos pontos centrais e sua periferia foi adotada por Rosten e Drummond (ROSTEN; T.DRUMMOND, 2005; ROSTEN; T.DRUMMOND, 2006) para treinar uma Árvore de Decisão (QUINLAN, 1986; BREIMAN *et al.*, 1984). O método de detecção de cantos resultante dessa abordagem é extremamente veloz devido à eficiência desse tipo de estrutura de decisão. No caso, o classificador é treinado a partir de porções de pixels obtidas diretamente de um conjunto de imagens usado como referência. Como consequência disso, o classificador resultante não é invariante à escala. Quanto a este ponto, Rosten e Drummond sugerem que uma árvore de decisão seja treinada para cada escala, o que tende a ser custoso e propenso a erros.

Slot e Kim (2006) propuseram um método de derivação de subconjuntos dos *descritores* SIFT com o objetivo de detectar uma classe específica de objetos. Para tanto, os autores desenvolveram um método para a extração de um "dicionário" de descritores característicos da classe – logo esse método não atua diretamente na detecção de pontos-chaves (SLOT; KIM, 2006).

Ozuysal e seus coautores (OZUYSAL *et al.*, 2006; OZUYSAL *et al.*, 2007) propuseram o uso de Árvores Aleatórias (*Randomized Trees*) para a representação de pontos-chaves. Usando o mesmo princípio proposto por Lindeberg (1998), os valores extremos do Laplaciano ao longo da escala são usados para detectar pontos-chaves. Regiões vizinhas a cada ponto são artificialmente deformadas várias vezes para simular a distorção provocada por

diferentes perspectivas. Essas regiões deformadas são usadas para treinar um conjunto de Árvores Aleatórias (AMIT *et al.*, 1996) que modelam a *aparência* dos pontos-chaves, de forma que uma classe é associada a cada ponto-chave detectado (LEPETIT; FUA, 2006). A grande vantagem desse método é que os pontos podem ser incluídos ou removidos dos classificadores de forma *online*. Claramente, a principal contribuição desse trabalho está na representação dos pontos-chaves através de um novo descritor e no fato de que o conjunto de pontos usados para representar um objeto pode variar durante o seu rastreamento (*tracking*) ao longo de sequências de vídeo.

Kienzle e seus coautores (KIENZLE *et al.*, 2005) usaram o conceito de saliência (KADIR *et al.*, 2004) com base em estatísticas do movimento do olho humano para treinar uma Máquina de Vetores de Suporte (VAPNIK, 1995), um tipo de classificador referenciado na literatura como SVM – do termo original inglês *Support Vector Machine*. No caso, o valor *real* retornado pelo classificador é adotado como uma *medida de saliência*, sendo que geralmente apenas o sinal desse valor é usado para identificar a classe (BURGES, 1998) à qual uma amostra pertence. O objetivo disso é identificar pontos de interesse sob a perspectiva da visão humana. Contudo, o principal objetivo desses autores nesse trabalho é justamente modelar o movimento dos olhos humanos ao invés de obter um classificador que atua de fato como um detector de pontos-chaves. Por esse motivo, esse trabalho não contém sequer uma avaliação da repetibilidade dos pontos obtidos usando o classificador, já que a repetibilidade é um aspecto fundamental para detectores de pontos-chaves. Assim, fica claro que a perspectiva adotada por Kienzle *et al.* para abordar o problema difere radicalmente daquela usada no presente trabalho.

4.4 Conclusões Preliminares

A detecção de pontos-chaves desempenha um papel fundamental na construção de métodos baseados em descritores locais. A utilização de mecanismos de aprendizagem de máquina possui um grande potencial de aplicação neste problema, tanto para que substituam métodos tradicionais quanto para que atuem como uma espécie de *reforço* quando certos tipos de pontos também são desejáveis. De acordo com pesquisa bibliográfica realizada neste contexto, a utilização de métodos supervisionados para detecção de pontos-chaves no espaço de escalas ainda não foi suficientemente explorada. Essa possibilidade será portanto investigada no presente trabalho usando a metodologia proposta no Capítulo 5.

5 *Usando Classificadores Não-Lineares Supervisionados para a Detecção de Pontos-Chaves*

Como visto no Capítulo 4, os métodos supervisionados de aprendizagem de máquina foram pouco explorados para a detecção de pontos-chaves nos métodos baseados em descritores locais. Dessa forma, o presente capítulo destina-se à descrição de uma metodologia proposta para a utilização de classificadores não-lineares supervisionados na detecção de pontos-chaves. Observe-se que as ideias aqui desenvolvidas para a utilização de Máquinas de Vetores de Suporte (VAPNIK, 1995) podem ser estendidas a outros classificadores.

5.1 Premissas e Hipóteses

5.1.1 Adoção do Espaço de Escala Linear

A metodologia proposta adota o *Espaço de Escala Linear*. Essa é a escolha mais adequada, quando não se tem conhecimento *a priori* sobre onde estão localizadas as estruturas de interesse, porque geralmente propicia várias possibilidades mesmo quando a situação requer a detecção de regiões potencialmente anisotrópicas. Nesse caso, o resultado da detecção usando o espaço de escala serve ao menos como uma inicialização para outros detectores.

Em muitos métodos (TUYTELAARS; MIKOLAJCZYK, 2008; MIKOLAJCZYK *et al.*, 2005) voltados a detecção de regiões afins, por exemplo, o Espaço de Escala Linear é considerado, ao menos em um primeiro momento, para que se detectem estruturas. Nesse caso, a configuração inicial dessas estruturas é usada para inicializar algoritmos de ajuste da região local a uma transformação afim (MIKOLAJCZYK; SCHMID, 2005).

5.1.2 Adoção de Classificadores Não-Lineares Supervisionados

De acordo com Witkin (WITKIN, 1983), há fortes evidências de que combinações lineares e não-lineares dos operadores diferenciais geométricos são apropriadas para realizar uma grande diversidade de operações de visão em baixo nível porque essa representação é capaz de expressar simultaneamente propriedades físicas e geométricas (LINDEBERG, 1994).

Sob essa perspectiva, o uso de técnicas não-lineares para aprendizagem de máquina surge como uma opção especialmente interessantes. Essas técnicas são teoricamente capazes de explorar o potencial ainda latente das combinações não-lineares desses operadores. A escolha dessa abordagem torna-se ainda mais racional porque a tarefa de obter combinações adequadas manualmente tende a ser desafiadora e propensa a erros. Assim, é bem provável que não se obtenha uma determinada combinação ótima ou subótima por simples observação devido à complexidade introduzida pela não-linearidade. Por outro lado, é de se esperar que essa tarefa também consuma muito tempo com observações e cálculos.

A seguir são apresentados os principais aspectos sobre classificadores supervisionados, culminando com o estabelecimento da diferença entre *classificadores lineares* e *classificadores não-lineares*.

5.1.2.1 Classificadores Supervisionados

A utilização de um classificador supervisionado tipicamente envolve duas etapas: *treinamento* e *ativação*. Durante a primeira etapa, é fornecido um *conjunto de treinamento* $T = \bigcup \{y_i, \mathbf{x}_i\} \subset \mathbb{U}$. Esse conjunto de amostras é uma porção representativa de um universo $\mathbb{U}$, que em geral, é bem maior. No caso, cada $y_i \in Y \subset \mathbb{Z}$ identifica a categoria (classe) à qual pertence um vetor de características $\mathbf{x}_i \in \mathbb{R}^d$ representando propriedades da entidade que será classificada. É desejável que cada $\mathbf{x}_i$ contenha informações que possibilitem discernir entre as diferentes categorias de objetos.

O objetivo do treinamento é generalizar conceitos "aprendidos" a partir da observação do conjunto de entrada. Na prática, é derivada uma função determinística $f(\mathbf{x}) \rightarrow \mathbb{R}$ pertencente a uma família pré-definida de funções (VAPNIK, 1982) que atuam como superfícies de decisão. Os hiperplanos são exemplos típicos dessas famílias de funções. Uma vez construída, f será usada para mapear cada $\mathbf{x}$ do universo de amostras en sua respectiva classe y de forma consistente com o conhecimento assimilado a partir de T.

Tabela 1: Tabelas-verdades para os operadores lógicos $\wedge$, $\vee$ e $\oplus$, que representam operações básicas da Álgebra de Boole.

a	b	$a \wedge b$	$a \vee b$	$a \oplus b$
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0

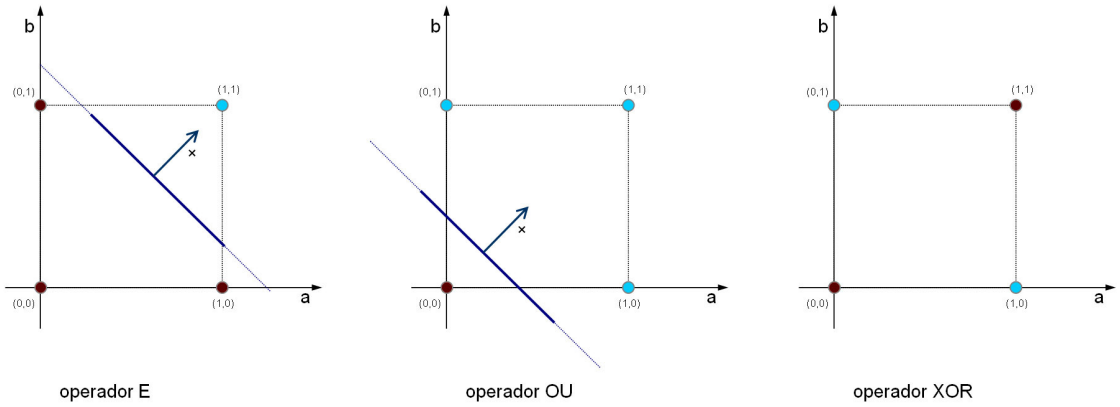

Figura 16: Três conjuntos de treinamento para um classificador linear, construídos usando os operadores lógicos $\wedge$ (esquerda), $\vee$ (centro) e $\oplus$ (direita), cujas tabelas-verdades estão presentes na Tabela 1 . Perceba que nos dois primeiros casos, é possível separar os pontos usando uma linha reta, de forma que os pontos positivos (azuis claros) e negativos (vermelhos escuros) ficam em lados opostos dessa linha – que de fato é um hiperplano. Por outro lado, o operador $\oplus$ é capaz de produzir um conjunto de treinamento que não pode ser satisfeito por reta alguma no plano – este é um exemplo clássico de situação intratável por meio de classificadores lineares.

Em termos computacionais, um método numérico deverá ser utilizado para derivar uma função específica que idealmente satisfaça todos os pontos conhecidos, ou seja, $f(\mathbf{x}_i) = y_i, \forall i$. Dependendo do método numérico utilizado e do conjunto de treinamento, as condições de convergência nem sempre são satisfeitas na prática. Isso ocorre principalmente quando classificadores lineares são usados em situações intratáveis, como visto na Figura 16.

Os classificadores lineares caracterizam-se pela tomada de decisões com base em uma *combinação linear* das características x_j de um vetor $\mathbf{x}$ de características. Mais precisamente, uma função de decisão d é definida em termos de um vetor de pesos $\mathbf{w}$ e um escalar b

$$f(\mathbf{x}) \equiv d\left(b + \sum_j w_j x_j\right), \tag{5.1}$$

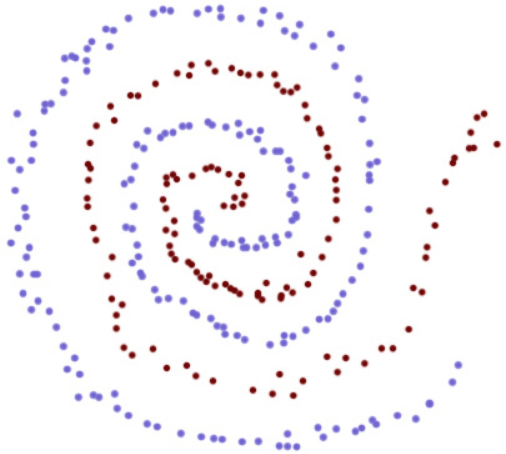

Figura 17: Conjunto de treinamento ilustrando um caso extremo, intratável por classificadores lineares. Perceba como a distribuição das amostras azuis claras e vermelhas escuras estão misturadas ao longo do espaço.

e converte o produto escalar entre $\mathbf{x}$ e $\mathbf{w}$ no valor desejado como saída. Geralmente d é uma função simples que mapeia o sinal de $f(\mathbf{x})$ em duas classes, "sim" $(+1)$ e "não" (-1). Na Figura 16, o vetor de pesos $\mathbf{w}$ define a orientação do hiperplano enquanto o escalar b define o deslocamento do hiperplano em relação à origem. Neurônios do tipo *perceptron* (ROSENBLATT, 1958) são um exemplo clássico desse tipo de classificador.

Claramente, existem conjuntos de treinamento intratáveis que não podem ser satisfeitos usando um classificador linear (vide exemplo construído com o operador $\oplus$ na Figura 16). Essa limitação pode ser contornada ao relaxar as condições de convergência de f sobre o conjunto de treinamento, formulando assim o treinamento do classificador como um problema de otimização no qual os erros de classificação são penalizados. Contudo, isso serve apenas como um paliativo no caso geral.

Na prática, superfícies de decisão mais expressivas fazem-se necessárias para resolver casos mais gerais, nos quais a distribuição dos dados ao longo do espaço é mais complexa. Junções e disjunções lógicas de classificações lineares podem ser usadas com esse propósito, por exemplo. Todavia isso torna o procedimento de treinamento mais intrincado, demandando, portando, algoritmos mais complexos para o treinamento.

O exemplo na figura 17, que ilustra um caso intratável por classificadores lineares. Nesse tipo de situação, as superfícies não-lineares de decisão são ferramentas poderosas para viabilizar a correta classificação das amostras. A Figura 18 ilustra como é possível

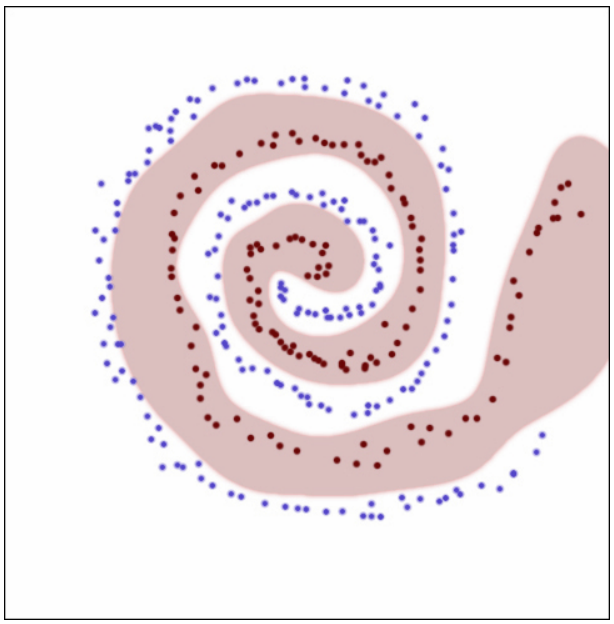

Figura 18: Superfícies não-lineares de decisão possuem um enorme poder de expressão, sendo capazes de classificar corretamente pontos que se confundem no espaço de características. No exemplo acima, um SVM de margem flexível foi treinado usando o *kernel* Gaussiano para categorizar as amostras do conjunto de treinamento apresentado na Figura 17. Observe que a região sólida em vermelho claro na imagem contém os pontos classificados como negativos pela superfície de decisão.

classificar corretamente esse conjunto de treinamento usando um SVM: é preciso selecionar cuidadosamente os parâmetros de treinamento, o que pode ser visto na mesma figura.

5.1.2.2 Máquinas de Vetores de Suporte

As Máquinas de Vetores de Suporte destacam-se entre os demais tipos de classificadores não-lineares porque utilizam uma abordagem estatística construída especialmente para minimização estruturada do risco *real* (VAPNIK, 1995). Modelos construídos usando SVM possuem um comportamento funcional similar ao das Redes Neurais Artificiais (ROSENBLATT, 1958) e das RBFs (*radial basis functions*), técnicas bastante difundidas. Entretanto nenhuma dessas abordagens possui uma base teórica tão bem fundamentada como aquela que forma a base do SVM.

Como resultado, a qualidade da generalização e facilidade de treinamento de modelos SVM está muito adiante das capacidades da maioria dos métodos tradicionais. De fato, SVMs são utilizados com sucesso em várias aplicações: Reconhecimento Óptico de Caracteres (VAPNIK, 1995; BURGES, 1998); Diagnose Automática (WAN; BAO, 2009); e Projeto de Medicamentos (IVANCIUC, 2007), por exemplo.

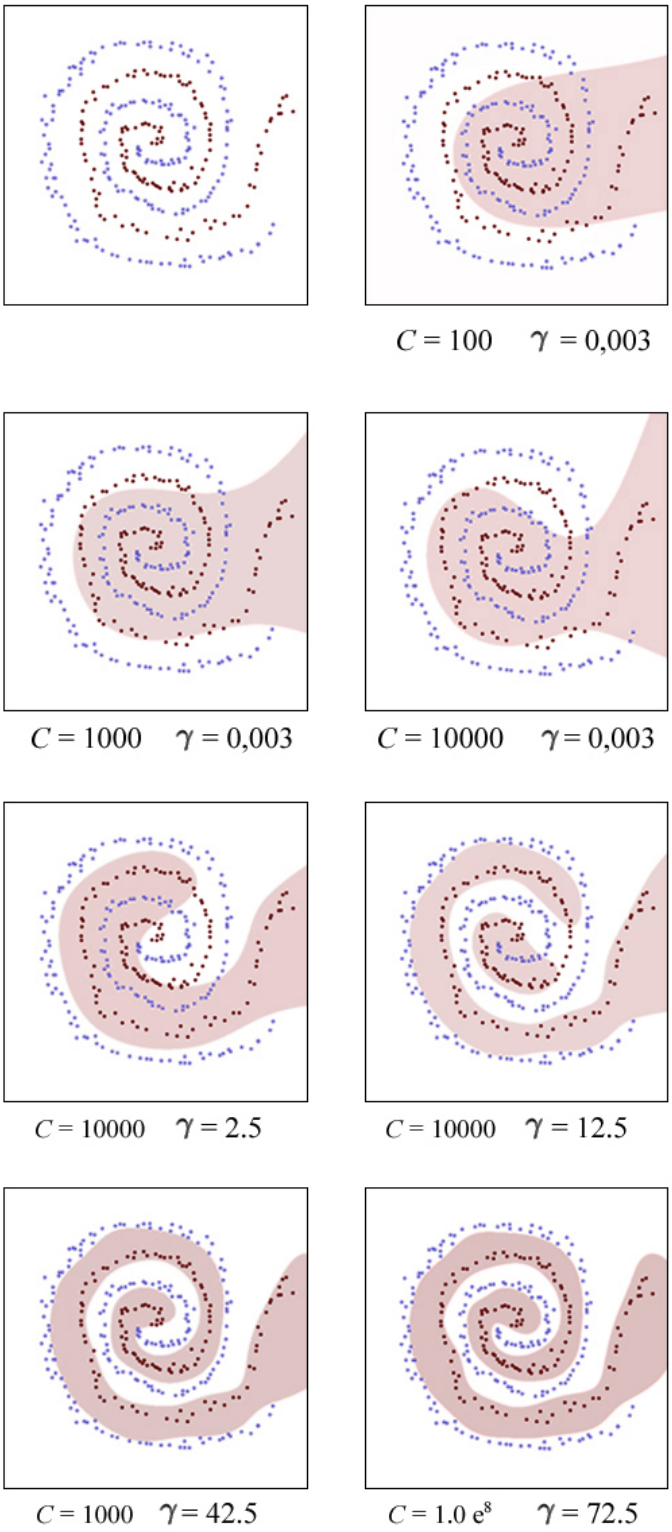

Figura 19: Impacto da seleção de parâmetros na acurácia obtida por um SVM treinado sobre o mesmo conjunto de dados. Observe como o formato da superfície de decisão é influenciado pela variação dos parâmetros γ_i e C_i selecionados. O melhor resultado, no caso, é obtido usando $\gamma = 72.5$ e $C = 10^8$.

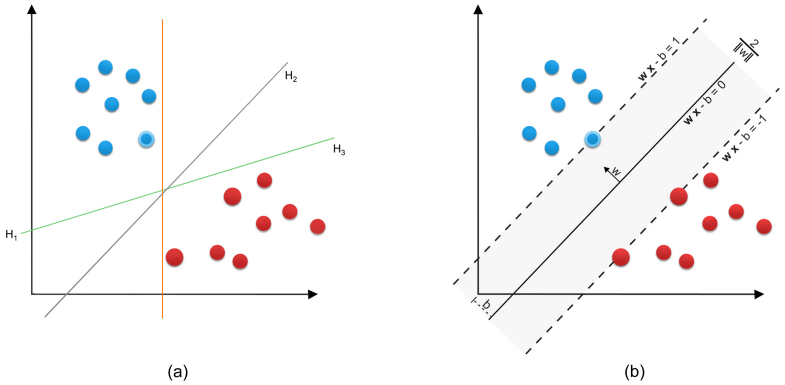

Figura 20: SVM treinado para maximizar a margem de separação. Todos os exemplos positivos (em azul claro) estão do lado positivo do hiperplano, enquanto todos os exemplos negativos (em vermelho escuro). Note-se que a margem de separação é máxima: a distância dos pontos mais próximos de cada classe ao hiperplano é a maior possível.

O problema de treinar um modelo SVM para um conjunto de dados $\{\mathbf{x}_i, y_i\}, \mathbf{x}_i \in \mathbb{R}^d, y_i \in \{-1, +1\}, i \in [1, l] \subset \mathbb{N}$ contendo l amostras consiste em encontrar um hiperplano $\mathbf{w} \cdot \mathbf{x} - b = 0$ tal que $y_i(\mathbf{w} \cdot \mathbf{x}_i - b) \geq 1, \forall i \in [1, l]$. Dessa forma, deseja-se determinar os valores dos pesos w_i e de b para que o hiperplano possa ser usado como um classificador.

Entretanto, desde que as condições de separabilidade existam, inúmeras combinações de $\mathbf{w}$ e b podem ser utilizadas. O melhor hiperplano é aquele que possui a maior distância entre os pontos mais próximos de classes distintas, ou seja, é aquele que *maximiza a margem de separação* entre as duas classes (vide Figura 20). Defina $\|\mathbf{w}\|$ como uma medida básica de distância. Assim, desconsiderando o efeito da escala, pode-se convencionar que a margem de separação é definida como $\frac{2}{\|\mathbf{w}\|}$. Isso implica que a maximização dessa margem pode ser feita *minimizando* $\|\mathbf{w}\|$.

Observe que as condições

$$\begin{aligned} \mathbf{w} \cdot \mathbf{x}_i - b \geq +1 \qquad & \forall y_i = +1 \\ \mathbf{w} \cdot \mathbf{x}_i - b \leq -1 \qquad & \forall y_i = -1 \end{aligned} \tag{5.2}$$

implicam que ponto algum do conjunto de treinamento pode ficar na margem de separação. Além disso, essas condições podem ser reescritas de forma resumida como

$$y_i(\mathbf{w} \cdot \mathbf{x}_i - b) \geq 1 \tag{5.3}$$

Essa formulação pode ser descrita como um problema de otimização: minimize $\|\mathbf{w}\|$ sujeito a $y_i(\mathbf{w} \cdot \mathbf{x}_i - b) \geq 1, \forall i \in [1, l]$. Essa definição é apropriada para a resolução utilizando técnicas de *Programação Quadrática*. Como há interesse em determinar um ponto

de sela, podem-se introduzir multiplicadores de Lagrange α_i para expressar o problema como

$$\min_{\mathbf{w},b} \max_{\boldsymbol{\alpha}} \{\frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{l} \alpha_i [y_i(\mathbf{w} \cdot \mathbf{x_i} - b) - 1]\} \tag{5.4}$$

Nessa formulação, a solução pode ser obtida como uma combinação linear das amostras usadas no treinamento. Mais precisamente,

$$\mathbf{w} = \sum_{i=1}^{l} \alpha_i y_i \mathbf{x_i} \tag{5.5}$$

onde, geralmente, muitos α_i são zero. Os $\alpha_i > 0$ denotam os pontos que estão justamente sobre a margem, e que portanto satisfazem $y_i(\mathbf{w} \cdot \mathbf{x}_i - b) = 1$. Tais amostras são conhecidas como *vetores de suporte*, que são amostras essenciais para o treinamento: todas as demais amostras podem ser removidas do conjunto, de forma que o hiperplano resultante de um retreinamento após essa remoção será o mesmo obtido anteriormente.

Através da substituição dos termos $\|\mathbf{w}\|^2 = w \cdot w$ e $\mathbf{w} = \sum_{i=0}^{l} \alpha_i y_i \mathbf{x_i}$, é possível derivar a forma dual da função objetiva

$$\sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j (\mathbf{x}_i \cdot \mathbf{x}_j) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j) \tag{5.6}$$

que deve ser maximizada com respeito a α_i, tal que $\alpha_i \geq 0$ e $\sum_{i=1}^{l} \alpha_i y_i = 0$. O produto escalar é referenciado como uma função real $k(\mathbf{x}_i, \mathbf{x}_j)$ – isto será útil mais adiante.

Observe que até este ponto o classificador obtido é puramente linear. Além disso, as condições de convergência não podem ser garantidas. Cortes e Vapnik (CORTES; VAPNIK, 1995) propuseram uma nova formulação, relaxada, para permitir a construção de hiperplanos em casos inseparáveis. Para tanto, esses autores introduziram variáveis auxiliares ξ_i para medir o grau de erro em cada amostra $\mathbf{x}_i$. Essa formulação é conhecida na literatura como margem suave ou margem flexível (*soft margin*)

$$y_i(\mathbf{w} \cdot \mathbf{x_i} - b) \geq 1 - \xi_i \tag{5.7}$$

Assim, a função objetiva é modificada para penalizar erros de classificação, obtendo-se um compromisso entre a maximização da margem de separação e erros de classificação. A forma com que os erros influi no resultado final é controlada usando uma constante global C que multiplica a soma das variáveis ξ_i que expressam os erros de classificação

admitidos durante a solução do problema:

$$\text{argmin}\,\frac{1}{2}\|\mathbf{w}\|^2 + C\sum_i \xi_i, \quad y_i(\mathbf{w}\cdot\mathbf{x_i} - b) \geq 1 - \xi_i. \tag{5.8}$$

É possível generalizar a teoria desenvolvida para a construção de hiperplanos usando o conceito de *kernel* (AIZERMAN *et al.*, 1964). *Kernels* são funções especiais que, simultaneamente:

- Mapeiam os pontos $\mathbf{x_i}$ e $\mathbf{x_j}$ no espaço de entrada para um espaço de Hilbert, que possui uma dimensão (potencialmente infinita) superior aquela dos dados de entrada. Essa alta dimensionalidade permite, teoricamente, usar um hiperplano para classificar qualquer configuração de pontos em duas classes (BURGES, 1998);
- Podem ser interpretados como a uma generalização do produto escalar dos pontos mapeados no espaço de Hilbert, na qual duas classes de pontos arbitrariamente distribuídos podem ser separadas usando um simples hiperplano.

Essas propriedades dos *kernels* permitem incrementar significativamente o potencial do método: dimensionalidades maiores permitem separar virtualmente quaisquer conjuntos de amostras usando um hiperplano. Também deve ser observado que a modificação na formulação matemática é mínima (BOSER *et al.*, 1992): os pontos $\mathbf{x_i}$ e $\mathbf{x_j}$ amostrados aparecem apenas em produtos escalares denotados por $k(\mathbf{x_i}, \mathbf{x_j})$, que pode ser interpretado como um *kernel*. Além disso, há outra implicação prática: computacionalmente, a separação independe do espaço de Hilbert no qual o hiperplano é caracterizado. O único requisito para tanto é que $k(\mathbf{x}_i, \mathbf{x}_j)$ seja avaliada para cada par de pontos.

Apesar dos termos núcleo e *kernel* possuírem um certo relacionamento no campo matemático, o termo núcleo será reservado ao contexto dos operadores de convolução enquanto o termo *kernel* será usado para denotar o operador não-linear usado no SVM. As funções mais frequentemente utilizadas como *kernel* são listadas na Tabela 2. Dentre essas opções, destaca-se o *kernel* RBF (*Radial Basis Function*) que também é conhecido como *kernel* Gaussiano. Essa função possui comportamento assintótico semelhante aos demais *kernels* (KEERTHI; LIN, 2003), como aqueles presentes na Tabela 2, de forma que a sua exploração é mais adequada quando a distribuição das amostras a classificar é totalmente desconhecida.

A formulação matemática do SVM foi posteriormente estendida para lidar com outras categorias de problemas, como regressão e estimativas de distribuição (DRUCKER *et al.*,

Tabela 2: Lista de *kernels* frequentemente utilizados para o treinamento de SVMs. Perceba que a natureza do classificador resultante do treinamento (linear ou não-linear) depende diretamente da escolha do *kernel*.

Kernel	**Equação**	**Parâmetros**
Linear	$k(\mathbf{x_i}, \mathbf{x_j}) = \mathbf{x_i} \cdot \mathbf{x_j}$	$\emptyset$
Polinomial	$k(\mathbf{x_i}, \mathbf{x_j}, a, \rho, d) = (a\mathbf{x_i} \cdot \mathbf{x_j} + \rho)^d$	$\{a, \rho, d\}$
Gaussiano	$k(\mathbf{x_i}, \mathbf{x_j}) = e^{-\gamma\|\mathbf{x_i} - \mathbf{x_j}\|}$	$\gamma > 0$
Sigmoide	$k(\mathbf{x_i}, \mathbf{x_j}) = tanh(\mathbf{x_i} \cdot \mathbf{x_j})$	$\{a, \rho\}$

1996). Tais formulações não são abordadas em profundidade neste trabalho – para maiores detalhes, consultar (VAPNIK, 1995) ou (CRISTIANINI; SHAWE-TAYLOR, 2000). Note-se que a seleção de vetores de características e a determinação de parâmetros de treinamento são etapas cruciais para a obtenção de bons resultados usando SVM.

A formulação clássica na qual duas classes de amostras são separadas possui a seguinte complexidade assintótica (BURGES, 1998):

- No pior caso, o treinamento é realizado em $\mathrm{O}\left(n_s^3 + ln_s + dn_s l\right)$, tal que l denota o número de amostras usadas para treinamento, n_s denota o número de vetores de suporte (desconhecido *a priori*) e d denota o número de componentes em cada $\mathbf{x}_i \in \mathbb{R}^d$. Isso corresponde ao maior valor entre l^3 e $d(l^2)$, visto que nesse caso $n_s \approx l$;
- Por sua vez, a ativação é realizada em $\mathrm{O}\left(n_s d\right)$, visto que $\mathbf{w}$ é derivado a partir dos vetores de suporte.

5.2 Visão Geral

A metodologia proposta para a utilização de classificadores não-lineares na detecção de pontos-chaves consiste em:

1. Estratégia para a seleção de vetores de características. Tais vetores devem conter informações que possibilitem uma separação e garantam sua adequada representação para a generalização de conceitos, o que permite simplificar o processo de busca por parâmetros;
2. Método para a seleção de parâmetros para o treinamento de classificadores do tipo SVM não-linear de margem flexível:

   (a) Adoção do *kernel* Gaussiano;

   (b) Combinação de duas heurísticas, baseadas no menor tempo de treinamento e no melhor resultado de classificação, respectivamente;

   (c) Busca de parâmetros em duas etapas. Primeiramente é utilizada uma técnica de subdivisão espacial para determinar regiões interessantes do espaço de busca. Em seguida, uma grade regular adaptativa é usada para refinar a busca.

3. Determinação semiautomática das escalas a serem utilizadas para pontos fornecidos como entrada do procedimento de aprendizagem;

4. Redução do número de contraexemplos considerados no treinamento. Essa é uma etapa importante porque o universo de contraexemplos é tipicamente imenso;

5. Seleção de um classificador específico utilizando o princípio da parcimônia (Navalha de Occam).

As ideias propostas neste trabalho, desenvolvidas a seguir, consideram a adoção de um SVM como classificador não-linear para efeito de estudo de caso. Observe-se, portanto, que não há restrições do ponto de vista teórico que impeçam a utilização destas mesmas ideias no contexto de outros classificadores.

5.3 Construção de Vetores de Características

5.3.1 Seleção de Características Representativas

Há várias informações a respeito de um ponto (x, y, t) no espaço de escala que podem ser úteis para a categorização deste entre pontos-chaves e pontos espúrios. Observações como a posição ou a orientação não favorecem a invariância do detector com respeito a estes aspectos. Tais informações devem ser portanto desconsideradas em uma abordagem geral.

Sendo assim, para fins de treinamento de um classificador, recomenda-se utilizar as seguintes propriedades observadas nas amostras para a construção de vetores de características:

- Grandeza adotada para a supressão de não-máximos ao longo da escala. Caso deseje-se simular comportamento do detector SIFT, por exemplo, essa grandeza corresponde ao valor do operador DoG no ponto (x, y, t) amostrado e nos seus 26

vizinhos. Considerando o detector SURF, por sua vez, essa grandeza corresponde ao valor do operador HoG na vizinhança do ponto observado;

- Sinal do Laplaciano no ponto (x, y, t) de interesse a ser representado. Essa informação básica permite distinguir entre depressões (Laplaciano positivo) e saliências (Laplaciano negativo), o que pode ser interessante dependendo do contexto de uma aplicação;

- Valores dos operadores diferenciais geométricos normalizados com respeito a escala. De forma geral, as derivadas de segunda ordem são informações interessantes para a composição de vetores de características. Isso porque descrevem, mesmo que de forma implícita, as seguintes propriedades locais:

  – O valor do próprio Laplaciano;

  – O valor do Hessiano;

  – A curvatura local do ponto observado.

- Amostras extraídas a partir do próprio espaço de escalas podem ser usadas como componentes. Contudo, suger-se que o uso de tais componentes seja evitado. Isso porque definem propriedades absolutas dos objetos observados, de forma que a informação derivada tende a não possuir um mesmo significado ao longo das escalas. Como consequência, o processo de classificação pode se tornar mais complicado dependendo dessa escolha. Por outro lado, talvez tais informações sejam convenientes para modelar o comportamento de pontos-chaves em uma escala específica.

5.3.2 Normalização de Vetores de Características

É recomendável que os vetores de características sejam normalizados com relação a alguma grandeza. Isso permite que um maior número de exemplos seja expressado da mesma forma, possibilitando assim a redução dos conjuntos usados para treinamento. Além disso, a normalização também favorece a capacidade de generalização do classificador ao passo que aumenta a similaridade entre amostras de uma mesma classe. Do ponto de vista prático, a normalização também é interessante porque beneficia a estabilidade numérica das operações envolvidas.

Eis um conjunto de linhas gerais recomendadas para realizar a normalização dos vetores de características $\mathbf{x}$:

- Não é interessante considerar médias, máximos e mínimos de x_i neste contexto. Além desse procedimento consumir tempo para a computação, essa normalização pode resultar na distorção do significado dos operadores diferenciais geométricos. Nesse sentido, é suficiente utilizar os valores dos operadores normalizados com relação a escala;

- Caso deseje-se reproduzir o resultado de algum classificador específico, é aconselhável dividir o valor da propriedade observada nos vizinhos no espaço e na escala pelo valor observado no ponto (x, y, t). Por questões práticas, é interessante usar $\frac{1}{\tau+c}$, onde τ denota o valor da propriedade observada em (x, y, t), e c é alguma constante positiva selecionada para evitar inconvenientes divisões por zero.

5.4 Redução do Número de Contraexemplos

Os detectores de pontos-chaves conhecidos reportam um número muito pequeno dos pontos analisados durante a avaliação da imagem em várias escalas. O detector baseado na supressão de não-máximos usado pelo método SIFT, por exemplo, tipicamente retorna apenas algumas centenas de pontos-chaves em resposta a uma imagem contendo 800×600 pixels.

Por outro lado, o número de pontos descartados é muito grande. No caso acima, cerca de 2 milhões de pontos são prontamente rejeitados pelo método de detecção. Em consequência disso, faz-se necessário um mecanismo criterioso para a seleção de contraexemplos para fins de treinamento de um classificador. Dois aspectos são fundamentais para a construção desse mecanismo. Primeiramente, deseja-se obter controle sobre o tamanho do conjunto de contraexemplos produzido. Isso permite balancear a razão entre a quantidade de amostras representando o padrão desejado e a quantidade de amostras consideradas como contraexemplos presentes no conjunto de treinamento. Observe que, quando o número de contraexemplos é muito grande, maior é a tendência do classificador em concentrar seu aprendizado neste tipo de exemplos. No caso de SVM, isso pode significar um maior número de vetores de suporte. Por fim, deve-se considerar a representatividade dos contraexemplos selecionados. É importante que os exemplos selecionados sejam capazes de expressar o conceito tão bem quanto o conjunto original, uma vez que um classificador ideal deve ser capaz de absorver esse conhecimento.

O seguinte procedimento é proposto para a construção de um conjunto de contraexemplos que obedeça aos critérios acima mencionados:

- Uma base de imagens deve ser construída para que os contraexemplos sejam extraídos. O maior número possível de contextos e cenários deve ser considerado neste caso. Entretanto, dependendo da aplicação, esse contexto pode ser restringido para que se obtenha um classificador mais eficiente;

- Um universo de amostras representando contraexemplos deve ser extraído a partir da base de imagens. Uma pré-seleção pode ser realizada de forma manual, tal que o usuário do detector aponte quais padrões são indesejáveis. Essas amostras também podem ser pré-selecionadas automaticamente examinando os pontos descartados por um detector, por exemplo;

- O número-alvo de amostras n deve ser estipulado pela aplicação;

- n amostras artificiais podem ser geradas automaticamente a partir do universo de contraexemplos utilizando técnicas de agrupamento. O algoritmo clássico *k-means* (MACQUEEN, 1992) é uma opção interessante nesse caso, pois apresenta complexidade assintótica moderada, $\mathrm{O}(nl^2)$. Além disso, esse algoritmo também pode ser acelerado usando estruturas de subdivisão espacial (MAIA *et al.*, 2007).

5.5 Seleção de *Kernel* e Parâmetros

5.5.1 Seleção de *Kernel*

O *kernel* RBF é adotado por ser teoricamente capaz de efetuar um mapeamento não-linear das amostras para um espaço de alta dimensionalidade, possibilitando assim lidar com casos nos quais a relação entre os vetores de características e as classes é não-linear. O *kernel* linear é um caso especial do *kernel* RBF (KEERTHI; LIN, 2003) para um parâmetro $\tilde{C}$. Além disso, o *kernel* sigmóide também se comporta como o RBF para certos parâmetros. Portanto, o *kernel* RBF é teoricamente capaz de reproduzir os mesmos resultados produzidos por outros *kernels*.

A adoção desse *kernel* é, ainda, conveniente sob outros pontos de vista:

- A avaliação do espaço de busca através de algoritmos é favorecida pela redução do número de parâmetros utilizados. O *kernel RBF* implica na determinação de apenas dois parâmetros (C, γ), permitindo que um maior esforço seja concentrado em buscas refinadas sobre um domínio de menor dimensionalidade;

- A análise de resultados também é favorecida por tornar-se mais simples. É possível construir superfícies $f(C, \gamma)$ descrevendo propriedades como tempo de treinamento, acurácia ou número de vetores de suporte, por exemplo. Isso é interessante porque torna a identificação de parâmetros ótimos mais intuitiva, possibilitando inclusive que o conhecimento humano possa ser utilizado no processo de seleção de parâmetros;

- Por fim, o *kernel* RBF apresenta maior estabilidade numérica. Como o valor de $e^{-\gamma\|\mathbf{x_i}-\mathbf{x_j}\|}$ é atenuado com o crescimento da distância do entre os pontos $\mathbf{x_i}$ e $\mathbf{x_j}$, tem-se que o resultado tende a ocupar a faixa de valores $[0, 1]$, na qual a maioria das operações aritméticas são menos propensas a erros (P754, 1985). Em contraste, o valor do *kernel* polinomial pode tender rapidamente ao infinito ou a zero quando o grau do polinômio é alto.

5.5.2 Seleção de Parâmetros para Treinamento

A determinação de bons parâmetros para treinamento de SVMs é um aspecto decisivo para a adequada utilização desse tipo de classificadores. De acordo com a metodologia proposta, estes são os elementos necessários para se realizar a seleção de parâmetros:

- Construção de dois conjuntos de exemplos S_t e S_e, que serão utilizados para treinamento e avaliação de acurácia, respectivamente. Observe que as funções desses conjuntos podem ser permutadas, produzindo uma validação cruzada dos parâmetros. Tais conjuntos podem ser construídos a partir de um universo maior de amostras $S = S_t \cup S_e$, de forma que cada elemento de S é encaminhado para S_t ou S_e dependendo de uma decisão aleatória;

- Estratégia de percurso no espaço de possibilidades. As seguintes heurísticas serão utilizadas para guiar a busca:

  - Visitar regiões na direção de crescimento da acurácia. Essa é uma escolha natural porque obviamente deseja-se obter o melhor resultado de classificação;
  - Visitar regiões na direção de decrescimento do tempo necessário para o treinamento. Essa escolha baseia-se na intuição de que a escolha de parâmetros adequados implica menores tempos de treinamento, assumindo que o algoritmo de solução tende a convergir mais rapidamente nesses casos.

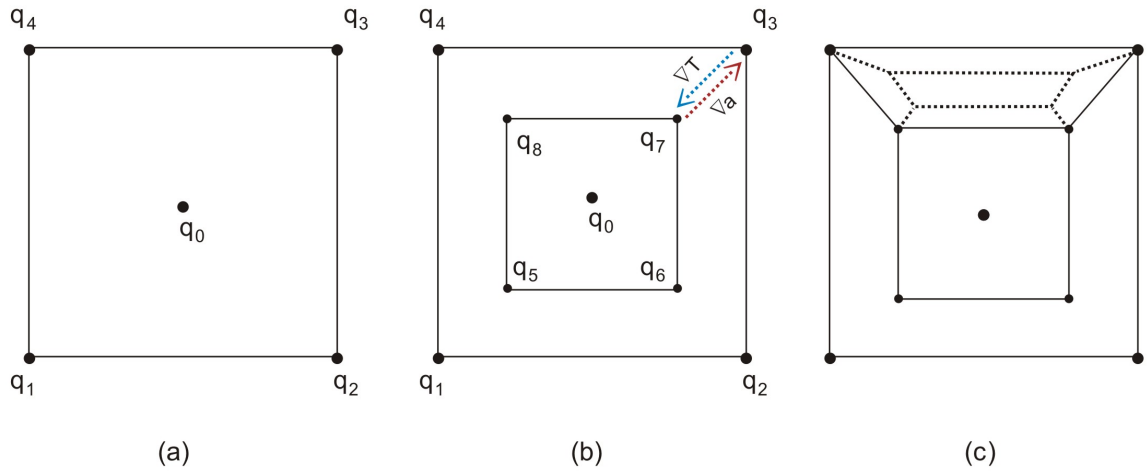

Figura 21: Seleção de parâmetros usando o Algoritmo de Busca em Teia.

- Algoritmo de Busca Inicial. Esse algoritmo tem por objetivo identificar regiões potencialmente ótimas para que se procedam buscas mais refinadas. Os pontos visitados durante esta busca devem ser utilizados para estudar a variação da acurácia e do tempo de treinamento de acordo com a variação dos parâmetros (C, γ). Como tal, a busca inicial deve considerar uma ampla faixa de possíveis valores para C e γ para realizar uma amostragem grosseira sobre o espaço de busca;

- Algoritmo de Busca Refinada. Esse algoritmo realiza uma busca intensiva sobre uma faixa mais específica de valores, restringindo assim a região da busca ao passo que uma análise mais fina é realizada. Assim, esse algoritmo tem por objetivo localizar valores ótimos de C e γ, úteis para a condução do treinamento de modelos finais destinados ao uso na aplicação. Como tal, esse processo é caracterizado por uma busca exaustiva dos melhores parâmetros.

5.5.2.1 Busca Inicial por Parâmetros: Busca em Teia

A estratégia usada pelo algoritmo de busca inicial proposto neste trabalho é ilustrada pela Figura 21. Estas são as principais características dessa estratégia, denominada "Busca em Teia":

- Subdivisão adaptativa do espaço de acordo com um critério de monotonia do desempenho do classificador em função dos parâmetros utilizados. Assim, é possível descartar mais facilmente regiões nas quais os parâmetros produzem classificadores de cujo desempenho é aproximadamente o mesmo. Por outro lado, maior esforço de

busca é concentrado nas regiões mais sensíveis quanto à variação dos parâmetros de treinamento;

- Uso combinado das heurísticas de tempo de treinamento e acurácia. Quando o algoritmo encontra evidências da alteração de alguns destes critérios, isso implica na realização de buscas mais refinadas na região correspondente;
- A busca não utiliza uma grade regular. Ao invés disso, a busca é feita usando uma estrutura que subdivide o espaço em cinco sub-regiões, dando origem a uma estrutura que lembra uma teia (Figura 21-c). A natureza irregular desse tipo de subdivisão permite explorar o espaço de busca de forma mais intensa do que usando grades regulares, pois os pontos visitados estão mais distribuídos ao longo do espaço de busca.

Algoritmo 5.1: "IniciaBusca" em pseudocódigo

```
1  ALGORITMO IniciaBusca
2  ENTRADA
3      g_min, g_max: valores mínimo e máximo para o parâmetro gamma
4      C_min, C_max: valores mínimo e máximo para o fator de penalização de
           erros
5      S_t, S_e : conjuntos de treinamento e de teste para o SVM,
           respectivamente
6      l_t, l_a : limiares de tempo e de acurácia, respectivamente
7  SAÍDA
8      Conjunto Q de pontos visitados nos quais o SVM foi treinado e avaliado
9  INÍCIO
10   Q = vazio
11
12   q[1] = AvaliaSVM( g_min, C_min )
13   q[2] = AvaliaSVM( g_max, C_min )
14   q[3] = AvaliaSVM( g_max, C_max )
15   q[4] = AvaliaSVM( g_min, C_max )
16
17   Q = Q união { q[1], q[2], q[3], q[4] }
18
19   BuscaRecursiva( Q, q[1], q[2], q[3], q[4], S_t, S_e, l_t, l_a  );
20
21   retorne Q;
22 FIM.
```

Devido à natureza recursiva dessa estratégia de busca, a implementação é feita usando dois algoritmos: *IniciaBusca*, responsável por inicializar os parâmetros de busca, descrito pelo Algoritmo 5.1; e *BuscaRecursiva*, detalhado pelo Algoritmo 5.2, e que corresponde ao processo de subdivisão recursiva baseada nos critérios de acurácia e tempo de treinamento. O Algoritmo *IniciaBusca* simplesmente inicializa o procedimento de busca recursiva como descrito a seguir:

1. Os seguintes parâmetros devem ser especificados
   - γ_{min} e γ_{max}, valores mínimo e máximo, respectivamente, admitidos para o parâmetro γ;
   - C_{min} e C_{max}, valores mínimo e máximo, respectivamente, admitidos para o parâmetro C;
   - Conjuntos de amostras S_t e S_e;
   - Escalares l_t e l_a, limiares de tempo e acurácia, respectivamente.
2. Pontos $\mathbf{q}_i = \{\gamma_i, C_i, t_i, a_i, s_i\}$ são usados para armazenar informações sobre cada par de parâmetros (C_i, γ_i) visitado, onde:
   - $t_i \in \mathbb{R}$ denota o tempo em segundos para o treinamento;
   - $a_i \in [0, 100]$ denota a média da acurácia obtida em dois casos: (a) S_t é usado para treinamento e S_e para avaliação; e (b) S_e é usado para treinamento e S_t para avaliação;
   - $s_i \in \mathbb{N}$ denota o número de vetores de suporte no hiperplano treinado.
3. Os pontos $\mathbf{q}_1$, $\mathbf{q}_2$, $\mathbf{q}_3$ e $\mathbf{q}_4$ são computados para os parâmetros (γ_{min}, C_{min}), (γ_{max}, C_{min}), (γ_{max}, C_{max}) e (γ_{min}, C_{max}), respectivamente;
4. O conjunto Q de pontos visitados é inicializado como $\{\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3, \mathbf{q}_4\}$;
5. Por fim, é retornado o conjunto Q de todos os $\mathbf{q}_i$ visitados durante a busca recursiva utilizando o Algoritmo *BuscaRecursiva* sobre os pontos $\mathbf{q}_1$, $\mathbf{q}_2$, $\mathbf{q}_3$ e $\mathbf{q}_4$, considerando os mesmos conjuntos de amostras, bem como os mesmos limiares de tempo e acurácia.

Algoritmo 5.2: "BuscaRecursiva" em pseudo-código

```
1 ALGORITMO BuscaRecursiva
2 ENTRADA
```

```
3     Q : conjunto de pontos visitados
4     S_t, S_e : conjuntos de treinamento e de teste para o SVM,
          respectivamente
5     l_t, l_a : limiares de tempo e de acurácia, respectivamente
6  INÍCIO
7     q[0] = AvaliaSVM( MediaGamma( q[1], q[2], q[3], q[4] ), MediaC( q[1], q
          [2], q[3], q[4] ) )
8     dpSvm = DesvioPadraoSVM( q[0], q[1], q[2], q[3], q[4] )
9     d_t = dpSvm[0]
10    d_a = dpSvm[1]
11
12    se d_t == l_t ou d_a == l_a então
13      q[5] = AvaliaSVM( MediaGamma( q[0], q[1] ), MediaC( q[0], q[1]) )
14      q[6] = AvaliaSVM( MediaGamma( q[0], q[2] ), MediaC( q[0], q[2]) )
15      q[7] = AvaliaSVM( MediaGamma( q[0], q[3] ), MediaC( q[0], q[3]) )
16      q[8] = AvaliaSVM( MediaGamma( q[0], q[4] ), MediaC( q[0], q[4]) )
17      Q = Q uniao { q[5], q[6], q[7], q[8] }
18
19      para i = 1 até 4
20            se GRAD_t( q[i], q[i+4] ) < 0 ou GRAD_q( q[i], q[i+4] ) > 0 então
21                  MarqueVertice( i )
22            fimse
23      fimpara
24      para i = 1 até 4
25            se o quadrilátero c_q_d de vértice q[i] possui vértice marcado
26                  BuscaRecursiva( Q, vertices, S_t, S_e, l_t, l_a );
27            fimse
28      fimpara
29      BuscaRecursiva( Q, q[5], q[6], q[7], q[8], S_t, S_e, l_t, l_a  );
30    fimse
31 FIM.
```

Por sua vez, o processo de busca com subdivisão recursiva do espaço para a avaliação de parâmetros de treinamento utilizando o Algoritmo *BuscaRecursiva* é realizado da seguinte forma:

1. São recebidos os mesmos parâmetros que *IniciaBusca*, além de um conjunto de pontos $\{\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3, \mathbf{q}_4\}$ pré-visitados;

2. O ponto central $\mathbf{q}_0$ é avaliado usando $\left(\frac{\gamma_1+\gamma_2+\gamma_3+\gamma_4}{4}, \frac{C_1+C_2+C_3+C_4}{4}\right)$ como parâmetros de treinamento;

3. É computado o desvio padrão d_t sobre o tempo de treinamento observado nos pontos $\{\mathbf{q}_0, \mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3, \mathbf{q}_4\}$;

4. É computado o desvio padrão d_a sobre a acurácia observada nos pontos $\{\mathbf{q}_0, \mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3, \mathbf{q}_4\}$.

5. Caso $d_t \geq l_t$ ou $d_a \geq l_a$, a direção de crescimento da acurácia e de decrescimento do tempo devem ser consideradas para identificar quais setores do quadrilátero $\mathbf{q}_1\mathbf{q}_2\mathbf{q}_3\mathbf{q}_4$ devem ser visitados recursivamente

   - Computam-se os pontos $\{\mathbf{q}_5, \mathbf{q}_6, \mathbf{q}_7, \mathbf{q}_8\}$;
   - Caso $(t_{i+4} - t_i) < 0$ ou $(a_{i+4} - a_i) > 0$, o par de quadriláteros correspondente é marcado, pois a variação nos valores dos parâmetros implicou na obtenção de um classificador melhor de acordo com a heurística de busca. Assim, os quadriláteros são marcados quando houver indícios de crescimento de acurácia ou de decrescimento no tempo de treinamento;
   - Todos os quadriláteros marcados são visitados usando o Algoritmo *BuscaRecursiva*. O quadrilátero $\mathbf{q}_5\mathbf{q}_6\mathbf{q}_7\mathbf{q}_8$ também é visitado;

5.5.2.2 Busca Refinada usando Subdivisão em Grade

Uma vez determinadas as ordens de grandeza dos parâmetros, podemos proceder com a determinação de parâmetros a um nível mais fino, de forma produzir um classificador mais *forte*. Pontos (C, γ) que apresentem maior acurácia ou ainda menor tempo de treinamento devem ser considerados nesta busca refinada. Também é importante manter um registro do número de vetores de suporte presentes em cada classificador treinado durante a busca refinada. Essa informação é útil para que um par (C, γ) seja escolhido dentre os demais quando da utilização prática.

A busca refinada usando Subdivisão em Grade requer a especificação de intervalos para (C, γ) semelhantes àqueles usados na inicialização do procedimento de busca em teia. Esses intervalos são subdivididos usando uma grande regular, que é explorada de forma exaustiva para treinamento e teste. Células dessa grade podem ser subdivididas recursivamente em vias de concentrar a busca em regiões que produzam resultados mais promissores.

O algoritmo de busca nesse caso é relativamente simples por se tratar de uma exploração exaustiva do espaço. Nesse procedimento, ilustrado pela Figura 22:

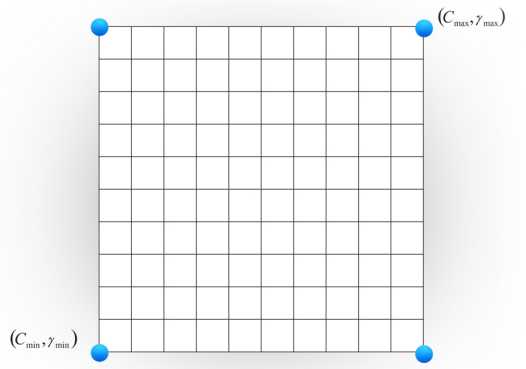

Figura 22: Subdivisão do espaço de busca usando uma grade regular. No caso, 10×10 subamostras são visitadas em cada vértice da subdivisão. As regiões contendo pontos que apresentam os melhores resultados podem ser exploradas recursivamente utilizando o mesmo algoritmo – o que caracteriza uma busca exaustiva.

- O intervalo de interesse $[C_{min}, C_{max}] \times [\gamma_{min}, \gamma_{max}]$ é subdividido usando uma grade regular contendo $m \times m$ amostras;
- Cada ponto (C_i, γ_i) da grade é avaliado com relação as heurísticas de busca (acurácia, tempo e número de vetores de suporte);
- Os resultados obtidos são ordenados de acordo com a qualidade da avaliação com relação as heurísticas. Dessa forma, é suficiente selecionar apenas os n pontos que apresentam o melhor desempenho
  - Cada uma das regiões $[C_j, C_{j+1}] \times [\gamma_j, \gamma_{j+1}]$ selecionadas deve ser usada para iniciar uma nova busca usando Subdivisão em Grade, a fim de refinar os resultados;
  - Os valores obtidos para acurácia, tempo e número de vetores de suporte, oriundos de cada avaliação devem ser armazenados para posterior análise.

Este procedimento pode ser repetido recursivamente, de forma a intensificar a busca. O procedimento de busca refinada também pode considerar a validação cruzada dos dados, procedimento no qual subconjuntos próprios aleatórios das amostras pré-classificadas são utilizados para treinamento e subsequente teste. Isso é feito considerando-se um par (C, γ) por vez. O objetivo é determinar pares (C, γ) que sejam, considerando amostragens gerais, mais adequados aos dados usados no treinamento. Hsu e seus colaboradores (HSU *et al.*, 2003b), por exemplo, utilizaram essa técnica para produzir curvas de níveis descrevendo a acurácia como função dos parâmetros.

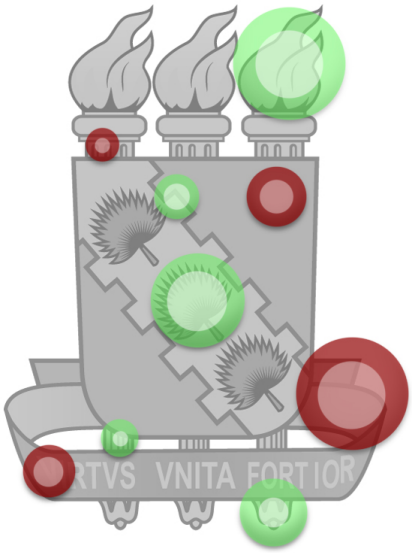

Figura 23: Marcação manual de exemplos. Pontos de interesse exemplificando o conceito de pontos-chaves desejado são denotados pelos círculos verdes (claros). Por sua vez, os contraexemplos são denotados por círculos vermelhos (escuros).

5.6 Seleção Semiautomática de Escalas para Exemplos Manuais

Dependendo da aplicação, pode ser necessário que o usuário do detector de pontos-chaves forneça manualmente um conjunto de exemplos para treinamento. Esses exemplos são extraídos a partir de um conjunto de imagens, podendo corresponder tanto ao conceito de pontos-chaves desejado quanto a contraexemplos representando configurações indesejadas. Isto é ilustrado na Figura 23.

É importante observar que uma estimativa do tamanho da região correspondente aos exemplos de pontos-chaves também deve ser fornecida pelo usuário para cada amostra. A tarefa de marcação manual deve ser realizada de forma minuciosa, e mesmo assim está sujeita a erros humanos. Como resultado, a construção de um conjunto de exemplos incoerentes com respeito a escala pode dificultar a definição do conceito desejado para efeito de se treinar um classificador.

Assim, é desejável que exista uma referência confiável para que se estimem as dimensões das regiões usadas como amostras. Dependendo da aplicação, tal informação pode estar disponível em uma base de imagens. Na ausência de referências explícitas de escala, sugere-se que o usuário utilize a seguinte abordagem para obter tais referências:

- Regiões de referência devem ser obtidas adotando algum detector de pontos-chaves

existente. As regiões correspondentes aos pontos-chaves reportados devem ser exibidas na interface gráfica usada para a marcação das amostras. Essas regiões fornecem uma referência para que o usuário determine a escala de cada uma de suas amostras manuais;

- Alguns dos próprios pontos-chaves detectados podem ser utilizados pelo usuário desde que correspondam ao conceito desejado. Note-se que a existência desses pontos nem sempre é garantida, visto que tais detectores possuem natureza não-supervisionada.

5.7 Seleção de Classificadores Específicos

Uma vez determinada a ordem de grandeza através das técnicas de busca propostas, é preciso selecionar o par de parâmetros (C, γ) para o treinamento do classificador que será utilizado na prática. Como diversos pares de parâmetros podem apresentar o mesmo valor máximo de acurácia, a seleção de um par específico deve considerar dois aspectos:

- Menor tempo de treinamento. Como discutido anteriormente, a experiência adquirida utilizando SVMs em outros domínios de aplicações sugere que a escolha de bons parâmetros implica melhores condições de convergência;

- O princípio da Navalha de Occam, conhecido também como *Lei da Parcimônia*. Esse princípio sugere que teorias usadas para modelar um fenômeno qualquer devem assumir apenas premissas estritamente necessárias à explicação do fenômeno em questão. Assim, quaisquer outras premissas desnecessárias podem e devem ser eliminadas. Em suma, de acordo com a Lei da Parcimônia, a melhor opção dentre um conjunto de teorias que explica um fenômeno é justamente a teoria mais simples do conjunto. No contexto deste trabalho, isso significa escolher o par (C, γ) que produza o hiperplano que possua o menor número possível de vetores de suporte.

De fato, o princípio da parcimônia é um argumento que parece ser razoável sob a perspectiva matemática. A adoção do hiperplano adequadamente treinado – hiperplano que contem o menor número possível de *vetores de suporte* – corresponde à seleção de uma superfície de decisão mais simples. Intuitivamente, hiperplanos mais simples tendem a apresentar um menor número de oscilações que teoricamente são capazes de perturbar qualitativamente o processo de classificação. Outros pesquisadores mostraram que é

possível construir hiperplanos mais simples em termos do número de vetores de suporte sem que a acurácia seja comprometida (ZHAN; SHENB, 2005).

Além disso, a redução do menor número de vetores de suporte implica também a simplificação dos cálculos necessários a classificação de novas amostras. Logo, um classificador computacionalmente mais eficiente também é obtido através da aplicação do princípio da parcimônia. Observe-se que não é possível quantizar esse fator de simplificação *a priori* porque a relação entre o número de amostras e a distribuição dos dados é desconhecida.

5.8 Conclusões preliminares

Neste Capítulo, foram apresentadas as principais ideias intuitivas que inspiram o uso de classificadores não-lineares supervisionados para realizar a detecção de pontos-chaves no contexto dos métodos baseados em descritores locais.

Em seguida, foi proposta uma metodologia para a utilização desse tipo de classificador para tal finalidade. Em particular, foram desenvolvidas abordagens práticas para a utilização de Máquinas de Vetores de Suporte nesse contexto. A metodologia proposta é composta pelos seguintes elementos:

- Estratégias para a seleção de componentes para construção de vetores de características $\mathbf{x_i}$. Além disso, também foi discutido o aspecto de normalização de $\mathbf{x_i}$, o qual possui enorme impacto sobre o resultado da classificação;
- Uso de técnicas de agrupamento para a redução do conjunto de contraexemplos. Isso torna-se necessário porque tais amostras surgem naturalmente em número imensamente maior do que os pontos-chaves detectados em uma imagem;
- Adoção do *kernel* RBF para o treinamento de modelos SVM, observadas as suas propriedades práticas e teóricas;
- Estratégias para a busca de parâmetros de treinamento (C, γ) através da exploração de um espaço de parâmetros;
- Seleção semiautomática de escalas para os exemplos fornecidos manualmente pelo usuário em aplicações específicas;
- Seleção de um par específico de parâmetros (C, γ) com base no Princípio da Parcimônia sobre o número de vetores de suporte dos melhores classificadores obtidos.

A metodologia desenvolvida especialmente para viabilizar a utilização de técnicas não-lineares de aprendizagem de máquina no contexto da detecção de pontos-chaves apresenta várias vantagens teóricas. Destaca-se como principal contribuição a possibilidade de introdução do conhecimento sobre o conceito de estruturas de interesse, o que permite derivar automaticamente novos detectores mais adequados para uso em situações específicas. Também é possível desenvolver novos conceitos sobre pontos espúrios, o que favorece diretamente a detecção e o reconhecimento de objetos em determinados contextos.

Os próximos três capítulos destinam-se à utilização da metodologia proposta para a realização de experimentos computacionais para comprovar sua validade e utilidade no âmbito de aplicações reais.

6 *Aprendendo o Detector SIFT via SVM*

O presente Capítulo destina-se à apresentação dos experimentos computacionais conduzidos com o propósito de avaliar a metodologia proposta em relação à viabilidade prática. Nesse contexto, Máquinas de Vetores de Suporte são usadas como classificadores para reproduzir o comportamento do detector de pontos-chaves usado no método SIFT.

O restante deste capítulo está organizado da seguinte maneira. Primeiramente são apresentados os principais aspectos de implementação da metodologia proposta. Em seguida, o experimento supracitado é detalhado.

6.1 Aspectos de Implementação

Os elementos de software que realizam a metodologia proposta foram implementados usando a linguagem C/C++. Outras linguagens e ambientes foram considerados para o desenvolvimento de protótipos, como por exemplo GNU Octave (EATON, 2008) e MATLAB (MATHWORKS, 2010). Essas opções foram descartadas porque dificultariam a pronta utilização de bibliotecas externas para uma miríade de microtarefas.

Os protótipos iniciais foram desenvolvidos no ambiente Linux, distribuição Ubuntu versão 9.04, utilizando o IDE (Ambiente Integrado de Desenvolvimento, do termo em Inglês *Integrated Development Environment*) Eclipse 3.5 (FOUNDATION, 2009). Contudo, a pronta disponibilidade de drivers para *webcams* atraiu o desenvolvimento de alguns protótipos para o ambiente Windows.

Procedimentos recorrentes de alto nível necessários à realização dos experimentos foram automatizados utilizando a linguagem de programação Lua (IERUSALIMSCHY *et al.*, 2010). Esta linguagem de *script* possui uma sintaxe simples e construções semânticas poderosas. Scripts Lua foram utilizados principalmente para o processamento de arquivos de texto e gerenciar a execução das ferramentas desenvolvidas através do *Bourne shell*.

6.1.1 Implementação de Referência para SIFT

Em se tratando do desenvolvimento de protótipos usando a linguagem C/C++, a implementação SIFT++ (VEDALDI, 2009) foi escolhida porque seu código é aberto e eficiente. Além disso, o autor disponibiliza uma série de relatórios contendo comparativos entre a sua implementação e o código original desenvolvido por Lowe em MATLAB (LOWE, 2004). Sobre esse aspecto, há um ponto digno de ser comentado: é natural que ocorram pequenas divergências porque algumas operações básicas sobre matrizes em MATLAB são implementadas usando técnicas sofisticadas para obter maior estabilidade numérica.

6.1.2 Máquinas de Vetores de Suporte

LIBSVM (CHANG; LIN, 2001) é uma implementação de SVM em C++, escolhida para o desenvolvimento do método tanto por aderir ao desenvolvimento de código aberto e quanto por ser um componente de software suficientemente maduro para utilização no presente trabalho. Essa ferramenta tem sido usada em um grande número de ambientes para a simulação como GNU Octave e MATLAB. LIBSVM incorpora as formulações para classificação, regressão e estimativa de distribuição. No caso da classificação multiclasses, é utilizado a estratégia de comitê, no qual $n * (n-1)/2$ hiperplanos são construídos para separar cada classe das demais. Assim, cada hiperplano produz um voto para uma das duas classes que separa, de forma que a classe contendo o maior número de votos é selecionada como a mais adequada para representar a amostra a ser classificada.

Uma modificação foi considerada necessária nesta biblioteca para reduzir o tempo de operação, visto que vários pares (C, γ) devem ser visitados durante a seleção de parâmetros:

- A biblioteca originalmente opta pelo uso de arquivos de texto sem índices para descrever tanto os conjuntos de treinamento quanto os modelos resultantes do treinamento. Essa abordagem é bastante lenta porque *tokens* presentes no arquivo precisam ser identificados e convertidos, gerando um *overhead* de processamento de texto;

- Foram adicionadas funções para oferecer suporte ao uso de arquivos binários com o objetivo de operar mais eficientemente. Utilizando esse tipo de formato baseado em índices, não apenas o *overhead* de processamento de texto é removido, mas também grandes blocos de dados podem ser lidos de uma só vez. Como resultado,

os processos de manipulação de arquivos são acelerados, em média, por um fator de 50 vezes.

6.2 Aprendendo o Detector SIFT via SVM

O objetivo deste experimento é mostrar que a metodologia proposta é capaz de substituir abordagens não-supervisionadas puramente numéricas durante a detecção de pontos-chaves. Mais especificamente, no contexto do método SIFT, pretende-se mostrar que classificadores não-lineares supervisionados são capazes de:

- Efetuar a supressão de não-máximos sobre combinações de derivadas obtidas ao longo da escala, desempenhando assim o papel de um detector clássico de pontos-chaves com seleção automática de escala;
- Assimilar por conta de sua natureza não-linear o conhecimento necessário para realizar decisões mais complexas. No caso, deseja-se descartar pontos-chaves de má curvatura tão bem quanto o SIFT. Note-se que, originalmente, essa tarefa envolve o cálculo aproximativo da razão entre os autovalores da matriz hessiana (LOWE, 2004);
- Reportar um conjunto formado por pontos tão estáveis quanto aqueles obtidos através do método original. Observe-se que esta propriedade pode ser derivada *diretamente* a partir da confrontação dos pontos reportados pelo SIFT original com os pontos detectados usando um modelo SVM.

Assim, como um modelo SVM será treinado para que aprenda a forma com que o SIFT detecta pontos-chaves, o detector resultante será referido como SVM-SIFT.

6.2.1 Vetor de Características

O vetor de características usado para representar cada ponto analisado contém as seguintes componentes para cada ponto $p_i = (x, y, t)$, que excluem a escala e a orientação associada ao ponto-chave:

- 26 amostras dos valores do operador *DoG*s obtidos nos vizinhos de cada pixel em sua respectiva oitava de Gaussiana. Esses valores são normalizados usando um fator multiplicativo $\frac{1}{\max(1, \|\mathrm{DoG}(x,y,t)\|)}$;

- Os valores das 3 derivadas de segunda ordem $\frac{\delta^2}{\delta^2 x}$, $\frac{\delta^2}{\delta^2 y}$ e $\frac{\delta^2}{\delta x \delta y}$, normalizadas com relação a escala, computados em p_i. A intenção disto é mostrar que a natureza não-linear do SVM é capaz de derivar implicitamente o conceito de *qualidade* de curvatura.

6.2.2 Seleção de Amostras para Treinamento

O conjunto de amostras contém três tipos de pontos, extraídos a partir da execução do método SIFT original com os parâmetros canônicos que aparecem em (LOWE, 2004):

- Pontos-chaves *legítimos*, que são caracterizados como máximo ou mínimo local, e cuja curvatura é considerada regular pelo método proposto em (BROWN; LOWE, 2002);
- Pontos-chaves *ilegítimos*, que foram descartados através da análise de curvatura apesar de serem caracterizados como máximo ou mínimo local. A quantidade desses pontos claramente flutua em função dos parâmetros usados para controlar as respostas do detector nas arestas. Apesar disso, a cardinalidade entre pontos legítimos e ilegítimos é aproximadamente a mesma no caso geral;
- Pontos *espúrios* que não correspondem a um máximo ou a um mínimo local. Observe que há uma enorme quantidade desse tipo de ponto, pois inúmeros pixels da imagem são descartados já na etapa de supressão de não-máximos.

6.2.2.1 Imagens para Teste

A base de imagens utilizada em (MIKOLAJCZYK; SCHMID, 2005) foi adotada para a avaliação do detector SVM-SIFT. Esse conjunto foi considerado adequado para os testes porque, apesar de conter apenas 48 imagens, os cenários expressam situações abrangentes em termos das variações possíveis: iluminação, desfocamento, zoom, rotação, mudança de perspectiva e compressão JPEG (JEONG, 1997). Além disso vários tipos de elementos estão presentes nessas cenas, como materiais orgânicos, sintéticos e texturizados, por exemplo.

Também deve ser observado que, excetuando-se o caso da compressão JPEG, tais variações não foram introduzidas de forma artificial manipulando-se as imagens *a posteriori*, mas pela variação física das condições de aquisição. Estas imagens, ilustradas na Figura 24, possuem as seguintes características:

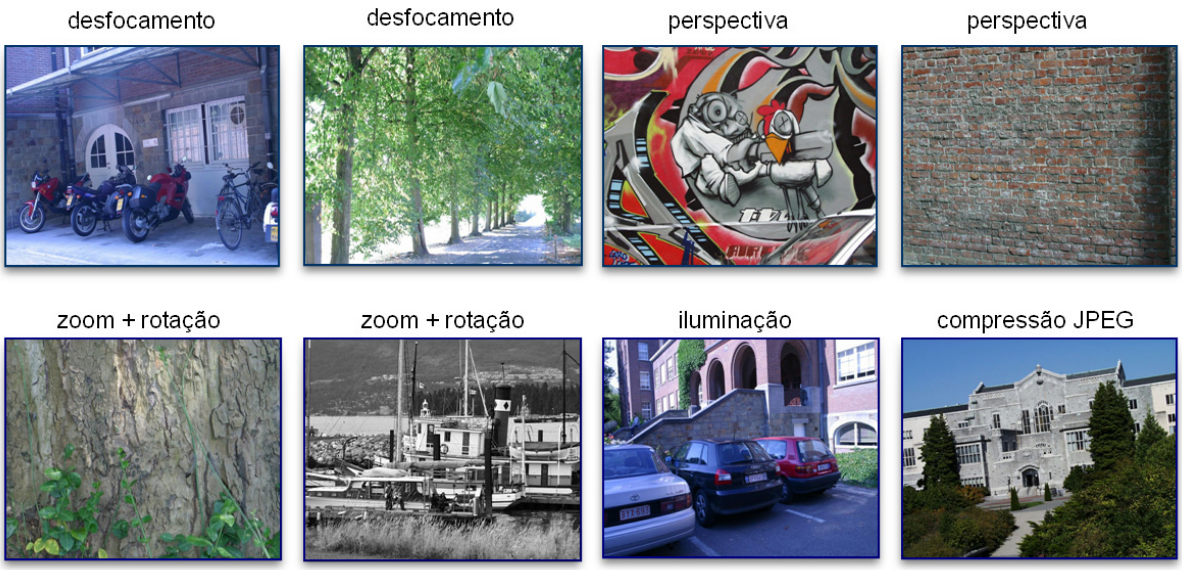

Figura 24: Exemplos de imagens usadas na avaliação realizada em (MIKOLAJCZYK; SCHMID, 2005). Seis variações de cada uma das imagem são usadas para avaliar a robustez dos detectores de pontos-chaves com relação a vários aspectos: desfocamento, mudança de perspectiva, zoom, rotação, iluminação e compressão JPEG.

- Resolução média de 800×640 pixels.
- Contemplam um total de 8 cenários. Uma sequência de 6 imagens foi adquirida por cenário variando as condições de aquisição;
- As variações de escala e foco foram obtidas usando diferentes configurações de zoom e foco. A escala varia até um fator de aproximadamente quatro;
- As variações JPEG foram introduzidas variando o parâmetro de qualidade da imagem entre 40% e 2%;
- A iluminação foi perturbada modificando-se a abertura da câmera;
- As mudanças de perspectiva variam desde uma visão frontal até uma mudança em aproximadamente 60º da câmera em relação à cena.

6.2.2.2 Reduzindo o Número de Pontos Espúrios

Um número excessivo de pontos espúrios é reportado devido à supressão de não-máximos em várias posições e escalas. A partir de uma única imagem, por exemplo, é possível obter centenas de milhares desses pontos. Surge então o problema de como representar esse conjunto de pontos para que o treinamento seja computacionalmente

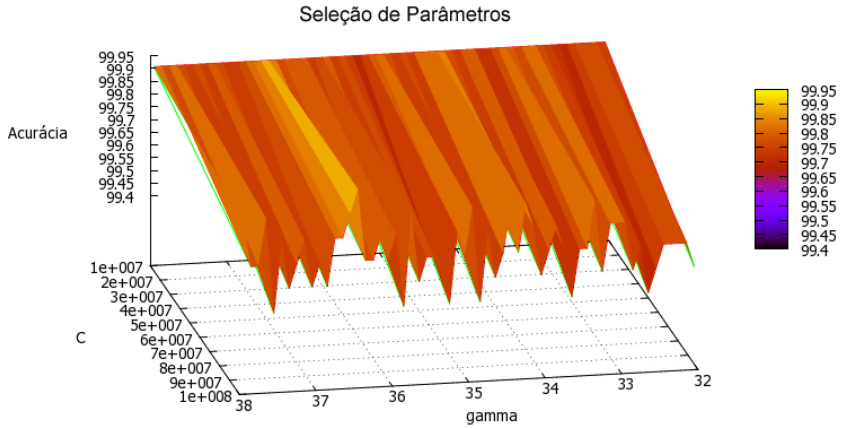

Figura 25: Plotagem 3D dos melhores parâmetros encontrados. (C, γ) variam no intervalo $[10^7, 10^8] \times [32, 38]$, produzindo uma acurácia de pelo menos 99.4%. O treinamento leva, em média, pouco mais de 2 minutos. O treinamento demora apenas 45s em média nos melhores casos.

viável, pois, caso contrário, não haveria sequer como armazenar tantos pontos na memória nem como usá-los para o treinamento do SVM.

Esse problema pode ser contornado da seguinte maneira. Amostras representando pontos espúrios são extraídas a partir de uma base de imagens contendo diversos tipos de cenas: árvores, pedras, faces, animais, objetos, paisagens, etc. A extração considerou 200 imagens VGA selecionadas aleatoriamente, o que resultou em um universo formado por mais de 2 bilhões de amostras. Desse universo, "apenas"1 milhão de amostras foram então selecionadas de forma também aleatória.

Finalmente, o conjunto de amostras resultantes é reduzido usando o algoritmo *k-means* (MACQUEEN, 1992) para produzir um número arbitrário n de *clusters* de amostras, de forma que cada grupo é representado por um centroide s_i. Cada um desses n centroides é na prática uma amostra artificial que é finalmente utilizada para alimentar o processo de treinamento do SVM. Os experimentos realizados mostram que é suficiente utilizar um conjunto composto por 2000 amostras de contraexemplos, que são suficientes para este caso.

O processo de seleção foi implementado em *C++*, sendo executado usando micro-computador com processador Pentium D 3.2 GHz e 2GB de memória RAM. Todo o processamento levou cerca de 5 horas e 45 minutos para ser concluído nessas condições. Acredita-se que o tempo de processamanto pode ser reduzido utilizando várias linhas de execução (*threads*) que colaborem na realização dessa tarefa. A Figura 25 contém a superfície com os melhores parâmetros encontrados.

6.2.3 Resultados

Tabela 3: Tempo em função da resolução para algumas imagens.

Imagem	Pixels	Tempo(s)	Pontos
Brasão	296×380	2.8s	199
Lampião	462×600	9.9s	814
Lena	512×512	8.5s	378
Grafitti	800×640	21.6s	1230

A Tabela 3 resume os resultados da detecção em algumas imagens. Veja as Figuras 26 e 27 para maiores detalhes. Eis algumas considerações importantes sobre os resultados experimentais:

- Os exemplos de treinamento foram obtidos observando o comportamento do detector SIFT com seus parâmetros padrões;
- O método SVM-SIFT é mais lento em termos de tempo de execução, apresentando um tempo de execução 35 vezes maior do que o apresentado pelo SIFT original;
- O hiperplano utilizado nos experimentos possui 673 vetores de suporte;
- Cerca de 9% a 10% dos pontos-chaves genuínos foram suprimidos em média. Dependendo do caso observado, essa taxa pode chegar até 33%. Contudo os pontos-chaves reportados pelo detector SIFT estão entre os mais estáveis. Foi observado que esses descartes ocorrem com maior frequência nas maiores escalas, o que de certa forma é natural. Um número muito menor de pontos-chaves é encontrado nessas condições por qualquer detector. Conjectura-se, assim, que é possível criar um classificador específico para esse caso, que supostamente é mais simples devido à supressão de estruturas menores;
- Alguns pontos-chaves extras são introduzidos devido a erros de classificação e a capacidade de generalização do SVM. Há vários casos nos quais o detector comporta-se de forma mais estável do que o SIFT.

Por fim, deve ser observado que a intenção no desenvolvimento desse experimento foi ilustrar que os classificadores não-lineares são capazes de absorver os conceitos de pontos de interesse observados através do comportamento dos detectores semi-supervisionados. Isso justamente porque neste trabalho não objetiva-se apenas obter um detector que se

Figura 26: Alguns exemplos nos quais os pontos-chaves foram detectados usando a técnica SVM-SIFT. No topo, a esquerda, uma fotografia clássica apresentando bom controle da iluminação. Abaixo, a esquerda, uma imagem sintética. No topo, a direita, uma fotografia capturada sem controle da iluminação. Finalmente, abaixo e a direita um exemplo da base de imagens usada em (MIKOLAJCZYK; SCHMID, 2005).

Figura 27: Pontos-chaves detectados usando a técnica SIFT-SVM para as imagens de entrada ilustradas na Figura 26.

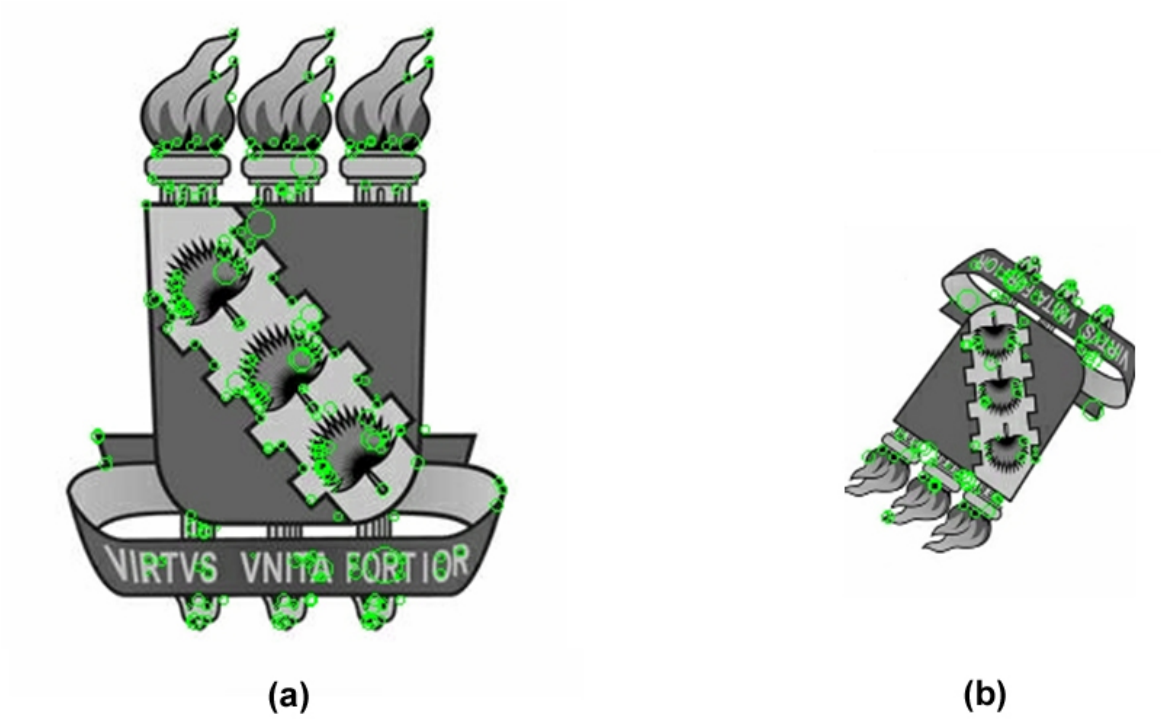

Figura 28: Estabilidade do detector. Pontos chaves detectados em um objeto verticalmente aprumado visualizado em 296 × 380 (a), e depois detectados em uma imagem rotacionada e reduzida para 150 × 193 pixels. Várias regiões preservam a concentração de pontos-chaves mesmo após esta transformação na imagem.

comporte *ips literis* como algum outro existente, mas também permitir que o conhecimento sobre outros conceitos sejam inseridos no contexto de aplicações mais específicas.

6.3 Conclusões Preliminares

Os resultados demonstram que o detector construído com base no treinamento de um classificador SVM é capaz de "imitar"o comportamento do detector usado no método SIFT. A acurácia obtida durante a fase de treinamento do classificador SVM por sí já constitui um forte indício de que isso é possível.

Os resultados experimentais demonstram que é possível de fato obter detectores com comportamento muito próximo aqueles existentes, mesmo sob a presença de operadores mais complexos para a análise da curvatura local. Nesses aspectos, há algumas observações importantes a se considerar:

- A supressão de não-máximos é, sob o ponto de vista de classificação, um problema relativamente simples para se resolver usando um modelo SVM não-linear;

- O foco principal nesse experimento está no fato de que o SVM consegue substituir os complexos cálculos de curvatura através de métodos numéricos, pois, nesse caso, não seria razoável utilizar um SVM em detrimento de outros métodos mais eficientes;
- Outros experimentos realizados com imagens fora do conjunto de treinamento mostram que a capacidade de generalização dos SVMs pode implicar em detectores de pontos mais estáveis. Maiores investigações com respeito a esse aspecto mostram-se promissoras.

7 Localização Automática de Olhos

O presente Capítulo destina-se à apresentação dos experimentos computacionais conduzidos com o propósito de avaliar a metodologia proposta em relação à aplicabilidade no problema de localização automática de olhos humanos. Nesse contexto, Máquinas de Vetores de Suporte são usadas como classificadores para estender o conceito de pontos-chaves a fim de enfatizar a detecção de olhos humanos.

Este experimento tem por objetivo mostrar que é possível influenciar o conceito de pontos de interesse utilizando a metodologia proposta. Mais especificamente, o problema de Localização Automática de Olhos foi selecionado como cenário prático para o estudo desse aspecto. Este é um caso específico no qual a aplicação demanda a introdução de um novo conceito de ponto-chave, pois, dependendo das condições de observação, os centros dos olhos não correspondem a alguma estrutura específica, mas a uma região caracterizada pela combinação de estruturas. Além disso, este caso é um exemplo prático da principal limitação dos detectores não-supervisionados: não há garantias de que os olhos estarão incluídos entre os pontos retornados por esses detectores.

Ao final deste Capítulo, os resultados práticos obtidos para localização automática de olhos são confrontados com os resultados obtidos através de outras técnicas existentes que figuram o estado-da-arte no assunto.

7.1 Definição do Problema

A localização de olhos tem por objetivo analisar uma imagem de entrada, que sabidamente contém uma face, de forma a reportar as posições dos olhos aparecendo na imagem. Este conceito pode ser formalizado através do seguinte mapeamento

$$f(I) = \{C_l, C_r\} = \{(x_l, y_l), (x_r, y_r)\}, \tag{7.1}$$

que exprime uma aproximação da posição dos centros dos olhos, onde $C_l = (x_l, y_l)$ e $C_r = (x_r, y_r)$ denotam os pontos estimados como centros dos olhos esquerdo e direito, respectivamente $x_l < x_r$, tal como estão dispostos na imagem de entrada – como se a imagem fosse um espelho.

7.1.1 Medição da Acurácia

Uma métrica de erro adequada para a avaliação deve ser invariante a escala e a orientação das faces, pois essas podem ocorrer em inúmeras poses e tamanhos nas imagens de entrada. Jesorsky e seus coautores (JESORSKY *et al.*, 2001) propuseram a adoção do pior caso, adotando o limite máximo do erro de localização medido em cada olho de uma dada face. A distância interocular é usada por esses autores como uma referência para a normalização com respeito à escala. A métrica proposta em (JESORSKY *et al.*, 2001) é expressa como

$$d_{eye}(I) = \frac{max(\left\|C_l - \widetilde{C}_l\right\|, \left\|C_r - \widetilde{C}_r\right\|)}{\|C_l - C_r\|}, \tag{7.2}$$

onde $\widetilde{C}_l$ e $\widetilde{C}_r$ denotam as posições legítimas e pré-conhecidas dos olhos em uma dada imagem. Essa métrica d_{eye} foi adotada no presente trabalho para avaliar a acurácia obtida nos experimentos de localização de olhos. Os trabalhos sobre localização de olhos encontrados na literatura (MA *et al.*, 2004; NIU *et al.*, 2006; CAMPADELLI; LANZAROTTI, 2006; CAMPADELLI *et al.*, 2006; KROON *et al.*, 2009) apontam $d_{eye} \leq 0.25$ como um critério razoável para afirmar a ocorrência de *detecção de olhos*, e que $d_{eye} \leq 0.15$ é suficiente para caracterizar a *localização de olhos*. Entretanto, valores menores podem ser mais interessantes para determinadas aplicações, como registro de faces (MAIA *et al.*, 2007).

Sob esse aspecto, uma curva denotando a distribuição do erro pode ser usada para expressar o comportamento da taxa de localização em função do máximo valor de d_{eye} admitido quando um método de localização é avaliado contra uma base de imagens. Essa curva caracteriza uma função monótona na qual o eixo das ordenadas corresponde à taxa de localização obtida naquela base de dados quando a respectiva abcissa é adotada como o erro máximo aceitável.

7.2 Seleção de Amostras

7.2.1 Bases de Imagens

Por conveniência, foram selecionadas bases de imagens pré-existentes e de domínio público que contém imagens de faces. Há várias dessas bases de imagens disponíveis, das quais foram selecionadas as seguintes devido às variações apresentadas em cada conjunto de imagens:

- A base **AR** (MARTINEZ; BENAVENTE, 1998) contém mais de 4.000 imagens coloridas, das quais apenas 3.368 estão disponíveis *online*, com as seguintes características
  - Resolução 768 × 768 pixels;
  - Corresponde às faces de 126 indivíduos, das quais 70 são homens e 56 são mulheres;
  - Presença de diferentes expressões faciais, condições de iluminação e oclusões.
- A base **BioID** (JESORSKY *et al.*, 2001) consiste em 1.521 imagens em tons de cinza, apresentando as seguintes características
  - Resolução de 384 × 286 pixels;
  - Corresponde às faces de 23 indivíduos em várias escalas;
  - Há várias ocorrências de indivíduos usando óculos;
  - A aquisição foi realizada sem o controle das condições de iluminação;
  - Essa base é tida como difícil pela literatura (CAMPADELLI *et al.*, 2005).
  - Contém a marcação manual dos centros dos olhos. Entretanto, constatamos a existência de alguns erros nesta marcação.
- A base **CALTECH** (WEBER, 1999) composta por 450 imagens coloridas
  - Resolução de 896 × 592 pixels
  - Fotografados 27 indivíduos;
  - Presença de diferentes condições de iluminação, expressões faciais, e cenários complexos.
- A base **JAFFE** (LYONS *et al.*, 1999),*Japanese Female Facial Expression*, contém 213 imagens em tons de cinza, apresentando as seguintes características

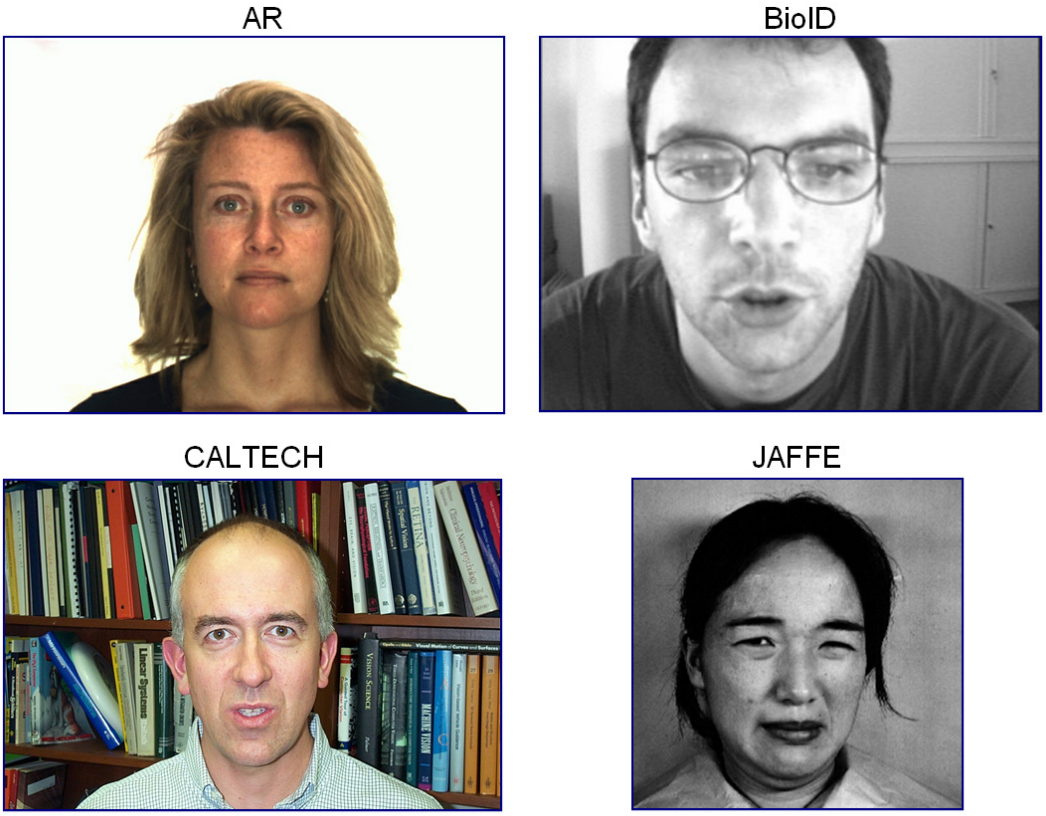

Figura 29: Amostras de magens presentes nas bases de dados AR, BioID, CALTECH e JAFFE. Apenas 20% das imagens de cada base foram selecionadas para treinamento, de modo que os testes de acurácia foram conduzidos utilizando os 80% restantes. Exemplos de todas as bases foram usados para o treinamento de um classificador SVM aplicado posteriormente em cada base de imagens.

- Resolução de 256×256 pixels;
- Contém 7 expressões faciais fotografadas a partir de 10 modelos femininas Japonesas.

Exemplos de imagens destas bases são ilustrados na Figura 29. Um total de 20% das imagens de cada base foi selecionado para treinamento, de modo que os testes de acurácia foram conduzidos utilizando as demais imagens. Vetores de características idênticos àqueles utilizados no detector SVM-SIFT foram utilizados.

7.2.2 Marcação e Extração de Amostras

A marcação das imagens foi usada para estimar automaticamente a escala das amostras positivas utilizadas no treinamento. No caso, 25% da distância interocular em cada face foi usada como uma aproximação para o raio da região correspondendo ao centro do olho.

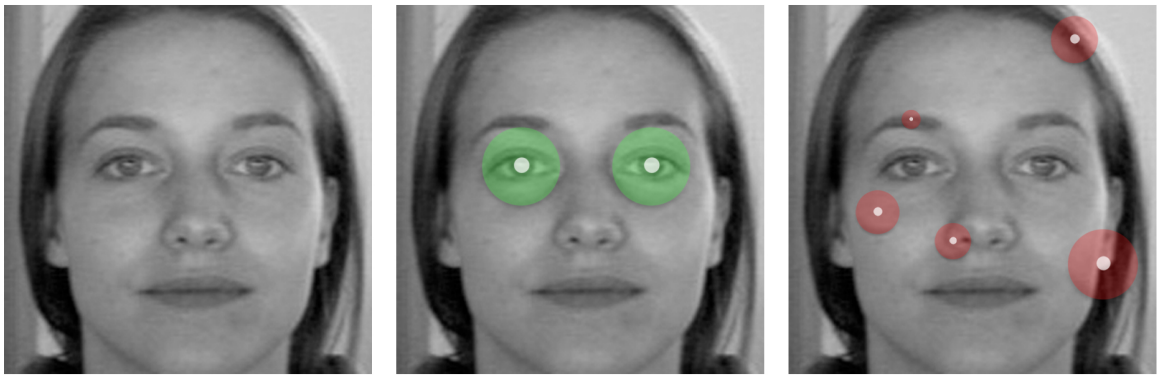

Figura 30: Seleção de vetores de características para o treinamento de um modelo SVM. Dada uma face cuja posição dos olhos é pré-conhecida (a), são amostradas duas regiões centralizadas nos olhos e de raio igual a 25% da distância interocular (b). Contraexemplos são amostrados em pontos aleatórios ao longo da face (c), tal que o raio varia entre 5% e 50% da distância interocular – obviamente essa amostragem descarta pontos sorteados próximos aos centros dos olhos.

Observe-se que é necessário estimar o valor do parâmetro t em função do raio r. No caso, $t = \frac{r}{\sqrt{2}}$ é uma aproximação adequada. Em seguida, t é usado para obter o nível de detalhe necessário para computar o vetor de características. Mais especificamente, é determinando o nível de detalhe correspondente a t' que mais se aproxima de t, presente em nas oitavas da pirâmide Gaussiana.

O conjunto de contraexemplos foi construído da seguinte maneira. Assumindo que a detecção da face já ocorreu, é suficiente coletar amostras em posições aleatórias sobre a região da imagem contendo a face. Tais amostras incluem escalas correspondendo a valores dentro da faixa partindo de 5% a 50% da distância interocular de cada face. O processo de construção de amostras para treinamento a partir de uma face é ilustrado pela Figura 30. Os contraexemplos são agrupados usando o algoritmo *k-means* da mesma forma que no experimento anterior, e acrescido dos contraexemplos computados anteriormente nesse experimento. A seleção de parâmetros é realizada usando a estratégia discutida no Capítulo anterior.

7.3 Localização Automática de Olhos

O seguinte procedimento é utilizado para a localização de olhos utilizando a metodologia proposta:

- Computam-se todos os descritores SIFT $\mathbf{sift}_i$ para os exemplos de olhos amostrados anteriormente;

- Aplica-se um detector de faces à imagem original. No caso, o detector de faces de Viola-Jones implementado na biblioteca OpenCV (BRADSKY *et al.*, 2006);

- Como o detector de faces utilizado reporta apenas faces verticalmente aprumadas, é suficiente buscar os olhos na região superior da imagem. Computam-se os pontos-chaves usando o modelo SVM previamente treinado, considerando apenas a metade superior da imagem. Além disso, os olhos direito e esquerdo ficam nos quadrantes superior esquerdo e direito da face, respectivamente. Dessa forma, a localização dos olhos procede como se segue para cada quadrante superior:

  - Computa-se o descritor do ponto-chave candidato $\mathbf{sift}_j$. Compute a correspondência desse descritor com aqueles conhecidos usando alguma métrica de distância. Preservam-se apenas os pontos cuja similaridade seja menor do que um valor máximo. O valor 0.5 foi obtido experimentalmente, sendo adequado na maioria dos casos;
  - Caso vários pontos resultem desse processo, utiliza-se a distância ao centro da imagem como critério de desempate, selecionando o ponto de distância mínima. Caso nenhum ponto seja reportado, assume-se que o método falhou em localizar os olhos.

7.4 Resultados

Tabela 4: Taxas de localização de faces reportadas pelo método desenvolvido sobre as bases de imagens BioID e JAFFE assumindo $d_{eye} \leq 0.1$. As taxas reportadas em outros trabalhos nesta mesma condição também estão incluídas, quando publicadas sobre uma mesma base de imagens. Os valores associados ao método proposto foram extraídas a partir dos gráficos da Figura 31.

Método de Localização	BioID	JAFFE
SVM-SIFT	94.6%	97.3%
(JESORSKY *et al.*, 2001)	93.0%	*n.d.*
(ZHOU; GENG, 2004)	94.8%	97.2%
(NIU *et al.*, 2006)	93.0%	100.0%
(CAMPADELLI *et al.*, 2006)	87.6%	*n.d.*
(KROON *et al.*, 2009)	97.9%	*n.d.*
(WANG *et al.*, 2005)	99.0%	*n.d.*

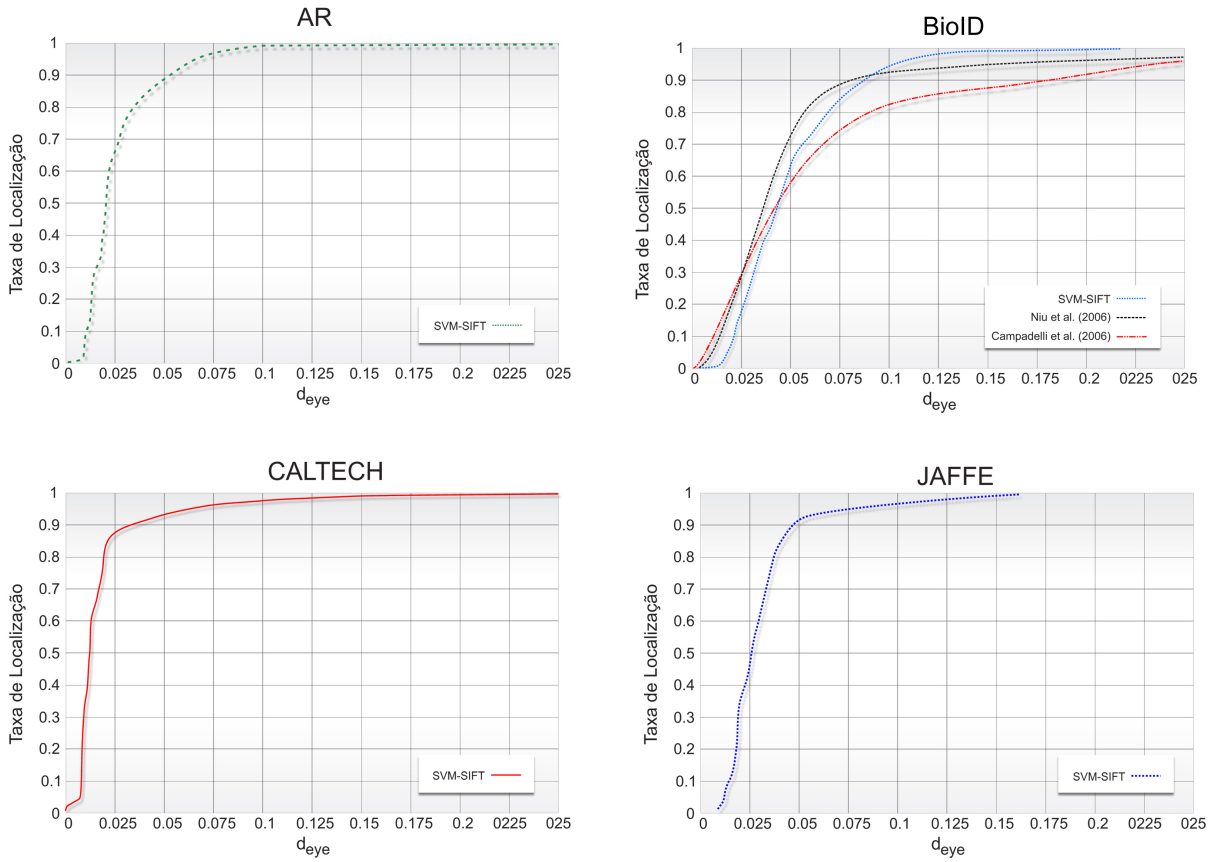

Figura 31: Resultados para a localização de olhos utilizando método proposto sobre as bases AR, BioID, CALTECH e JAFFE. Os resultados obtidos para a base BioID em (NIU *et al.*, 2006) e (CAMPADELLI *et al.*, 2006) foram incluídos para efeito de comparação. Uma taxa considerável de localização é obtida para $d_{eye} \leq 0.125$ em todos os quatro casos. Em particular, o localizador de olhos proposto apresenta os melhores resultados sobre a base BioID no intervalo $0.085 \leq d_{eye} \leq 0.15$.

Como o problema de localização é estabelecido sobre a pré-detecção de faces, os resultados consideram apenas os casos nos quais as faces foram encontradas com sucesso (CAMPADELLI *et al.*, 2005). As curvas de distribuição do erro obtidas aplicando o localizador de olhos proposto considerando a métrica de erro d_{eye} estão presentes na Figura 31. A Tabela 4 contém um resumo comparativo entre os resultados obtidos e aqueles publicados por outros pesquisadores nas mesmas bases, quando possível. Especial enfoque é dado a base de dados BioID, que constitui um cenário interessante devido as condições reais de aquisição.

7.5 Conclusões Preliminares

O problema de localização de olhos consiste em determinar a posição dos olhos em imagens que contém uma face. Esse problema é um exemplo prático da principal limitação dos detectores não-supervisionados de pontos-chaves: não há garantias de que os pontos retornados são aqueles importantes no contexto da aplicação. Nesse caso, essa etapa é crucial para o sucesso da localização, pois a ocorrência de falhas não é permitida em nenhum dos olhos.

A inadequação dos detectores não-supervisionados para a localização de olhos é explicada da seguinte forma. Há diversos casos nos quais os olhos não são estruturas caracterizadas como um máximo ou um mínimo local de combinações de operadores diferenciais geométricos. Os principais exemplos disso são: presença de óculos e de obstruções parciais. Nesses casos, a seleção de escala também é comprometida.

O método proposto apresenta boa acurácia com respeito aos resultados publicados por outros pesquisadores usando outras abordagens. É importante observar que o algoritmo de localização é extremamente simples e produz resultados tão bons quanto outros métodos encontrados na literatura, principalmente quando o erro de localização admitido está no intervalo $0.085 \leq d_{eye} \leq 0.15$. Dessa forma, acredita-se que a utilização de estratégias mais sofisticadas para aprimorar o localizador de olhos proposto é capaz de produzir resultados bastante promissores nessa área.

Conhecimento *a priori* sobre toda a base de imagens foi usado para construir uma segunda versão do algoritmo de localização capaz de operar de forma interativa. As faces detectadas são redimensionadas para 256×256 pixels, faixas de valores conhecidos nessa escala para o operador *DoG* são usadas para descartar rapidamente pontos espúrios. A partir desse ponto a detecção é realizada da mesma forma como foi descrita anteriormente.

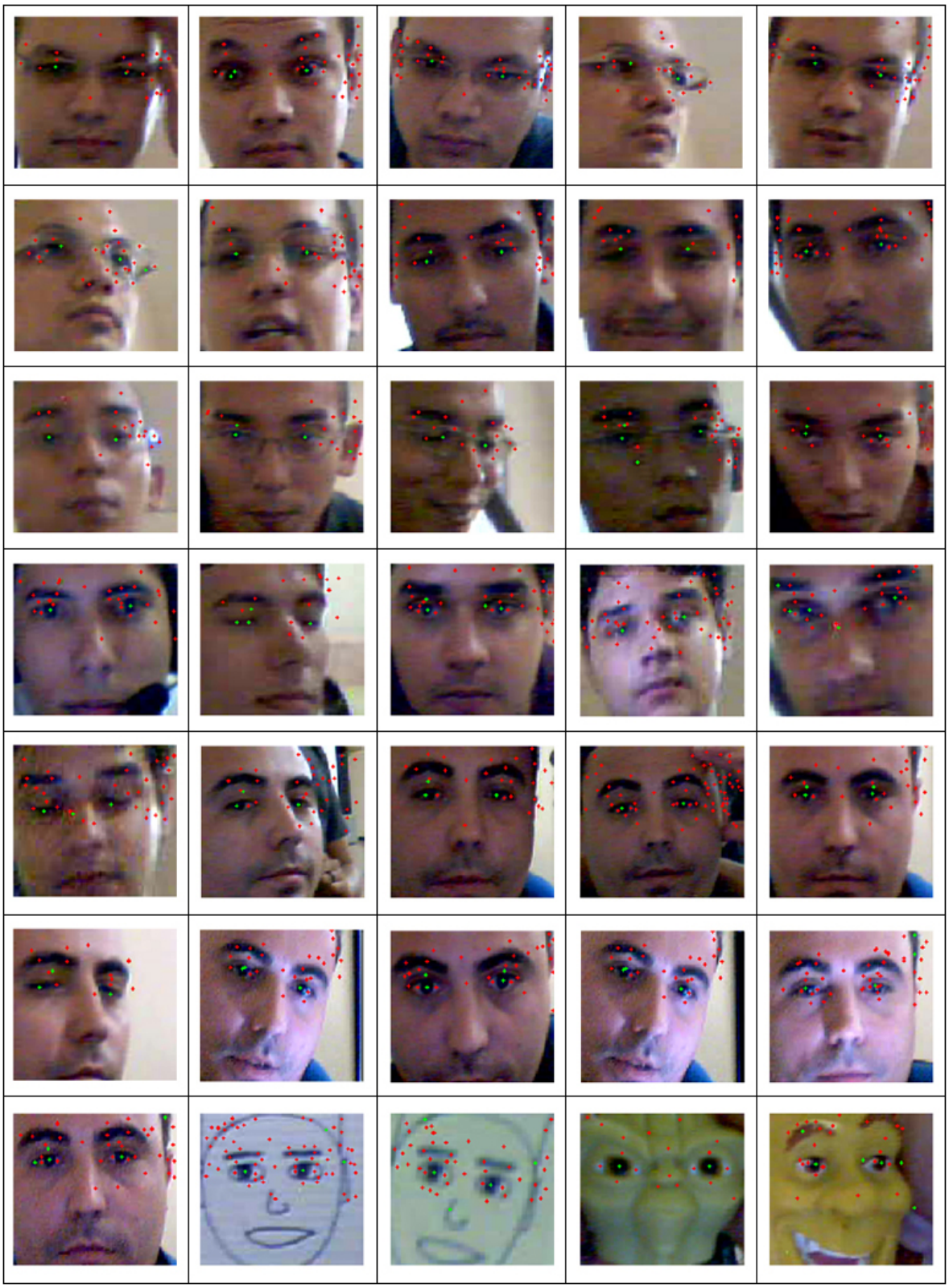

Figura 32: Localização de olhos a partir de imagens capturadas por uma *webcam* padrão. Os pontos detectados via SIFT estão em vermelho, enquanto pontos detectados pelo método SVM-SIFT estão em verde. O conjunto de treinamento considera imagens de apenas um indivíduo, capturadas na pose frontal com variação no estado dos olhos (aberto e fechados) e pequenas variações de perspectiva.

Alguns resultados obtidos por esse protótipo são ilustrados na Figura 32, no qual apenas um indivíduo foi considerado durante o treinamento: isto demonstra a capacidade de generalização do localizador proposto.

As principais limitações e deficiências observadas nesse localizador sob o ponto de vista da localização de olhos são:

- Há alguns casos classificados incorretamente, principalmente obstruções parciais dos olhos – pela ponte do nariz e por acessórios. Talvez a utilização de outros tipos de *kernels* (HSU *et al.*, 2003a), principalmente pré-computados, possa ajudar a contornar essa limitação. Além disso, tais casos podem ser usados como exemplos adicionais para reforçar o treinamento;

- Eficiência moderada em termos de tempo de processamento. Outros detectores apresentam acurácia similar com melhores tempos de detecção. Contudo, tais abordagens não apresentam a mesma capacidade de generalização, sendo necessário utilizar um grande número de amostras para produzir um resultado aceitável. Por outro lado, desempenho não foi o principal foco no desenvolvimento deste trabalho. A utilização de outros tipos de classificadores não-lineares pode ser estudada para essa finalidade;

- Surgimento de pontos espúrios. Algumas regiões correspondendo a ruído ou a estruturas aleatórias podem ser detectadas em decorrência da adoção de um classificador, o que pode comprometer a eficácia da metodologia no contexto das aplicações. Acredita-se que isso pode ser contornado reforçando o treinamento com contraexemplos dessa natureza, ou ainda usando outro classificador mais específico para essa situação.

O localizador de olhos proposto apresenta uma taxa de localização comparável aos métodos desenvolvidos por outros pesquisadores. Contudo, alguns casos oriundos de obstruções parciais dos olhos geram situações ambíguas. O localizador proposto também requer um tempo considerável de processamento. Além disso, pontos espúrios podem ser reportados pelo detector. O número de tais candidatos a centros dos olhos é prontamente reduzido utilizando os descritores locais computados em cada um desses pontos.

8 Reconhecimento de Faces

Ao longo do presente Capítulo são apresentados os experimentos computacionais realizados especialmente para avaliar a metodologia proposta à luz do problema de reconhecimento de faces humanas. No caso, Máquinas de Vetores de Suporte são usadas como classificadores para refinar a acurácia de uma técnica simples para reconhecimento de faces usando descritores locais.

8.1 Definição do Problema

Reconhecer faces é uma tarefa que pode ser formalizada da seguinte maneira. Deseja-se obter uma função $m(F_i, F_j)$ representando uma medida de similaridade entre as faces presentes em duas imagens F_i e F_j. Por simplicidade, assume-se que a imagem de $m(F_i, F_j)$ corresponde ao intervalo $[0, 1] \in \mathbb{R}$, tal que:

- O mínimo valor de similaridade é quantificado pelo valor 0, o valor 1 denota o valor máximo de similaridade entre duas faces;
- Adotando-se uma função de similaridade apropriada, é razoável utilizar um escalar s como o valor mínimo admissível que caracteriza a correspondência entre as duas faces. Assim, pode-se desenvolver um algoritmo simples para avaliar se F_i e F_j pertencem à mesma pessoa, bastando verificar a condição $m(F_i, F_j) \geq s$.

8.2 Algoritmo de Reconhecimento

Desenvolveu-se um algoritmo simples para o reconhecimento de faces, que funciona da seguinte maneira. As faces são detectadas usando o método de Viola-Jones (VIOLA; JONES, 2004) e representadas por descritores locais. Apenas os pontos-chaves situados no interior do círculo contendo a face devem ser considerados. Isso é feito com dois propósitos:

- Em primeiro lugar, deseja-se reduzir a influência do fundo da imagem na representação das faces;

- Além disso, é obtida uma considerável redução no esforço computacional associado à detecção de pontos-chaves e aàcomputação dos seus respectivos descritores locais.

A seguinte função de similaridade foi adotada para o reconhecimento de faces neste experimento

$$m_D(F_i, F_j) = \frac{\|D(F_i, F_j)\|}{min\left(\|k_D(F_i)\|, \|k_D(F_j)\|\right)}, \tag{8.1}$$

onde D identifica um método baseado em descritores locais, $k_D(F)$ denota o detector de pontos-chaves adotado no método D, e $D(F_i, F_j)$ denota o conjunto de correspondências $(\mathbf{p}_i, \mathbf{p}_j)$ entre pontos chaves $\mathbf{p}_i \in k_A(F_i), \mathbf{p}_j \in k_A(F_j)$. Assumindo que F_i e F_j são tais que $\|k_D(F_i)\| \leq |k_D(F_j)\|$, cada $\mathbf{p}_i$ deve ser comparado contra vários $\mathbf{p}_j$, que ocorrem em maior número. Isto permite reduzir o número de múltiplas correspondências para um mesmo ponto.

Observe-se que $m_D(F_i, F_j)$ é uma função relativamente simples e que depende diretamente da obtenção de correspondências entre as imagens. Logo, a métrica de similaridade pode ser prejudicada pela ocorrência de falsas correspondências entre os descritores locais das faces. Falsas correspondências entre as faces também poderão ser obtidas nessa situação. No caso específico dos algoritmos de reconhecimento de faces, é altamente preferível repudiar pares de faces genuínas do que aceitar pares ilegítimos (HWANG *et al.*, 2007). Este último caso teria um impacto fulminante em aplicações financeiras, por exemplo. Dessa forma, um algoritmo ideal de reconhecimento deve possuir duas propriedades fundamentais:

- Deve ser capaz de obter a maior taxa possível de aceitação de pares genuínos (GAR, *Genuine Acceptance Rate*). Isso evita que os usuários de uma aplicação de reconhecimento de faces sejam repudiados, por exemplo;

- Deve prevenir a ocorrência de situações que levem a ambiguidades, reduzindo assim a taxa de aceitação de falsos positivos (FAR, *False Acceptance Rate*).

8.2.1 Conceito de Pontos Indesejáveis

Assumindo que $m_D(F_i, F_j)$ é a função de similaridade adotada, a supressão de falsas correspondências contribui para reduzir o valor de GAR. O conceito de pontos indesejáveis

pode ser construído com base em amostas $(\mathbf{p}_i, \mathbf{p}_j)$ que produzem esse efeito no processo de reconhecimento. Um conjunto de tais amostras pode ser construído a partir da observação desse evento. Contudo, resta-nos definir o conceito oposto, no qual todos os demais pontos-chaves estão incluídos.

8.2.2 Seleção de Amostras para Treinamento

Há duas abordagens para construir vetores de características para o treinamento de um classificador a partir de um par de pontos-chaves $(\mathbf{p}_i, \mathbf{p}_j)$ erroneamente associados. A primeira abordagem consiste em usar as propriedades observadas em cada um dos pontos $\mathbf{p}_i$ e $\mathbf{p}_j$. Essa seria uma decisão inconveniente e um tanto ingênua, pois a dimensionalidade dos vetores de características seria dobrada. Além disso, o conjunto de treinamento deveria considerar a inclusão de duas permutações, por par, resultando assim em quatro vezes mais dados.

Por outro lado, a utilização de cada um dos pontos $\mathbf{p}_i$ e $\mathbf{p}_j$ mal-associados como um contraexemplo parece mais interessante. Essa abordagem é mais simples do que a opção anterior. Observando que ocorrem casos nos quais múltiplos pontos $\mathbf{p}_i$ são associados a um ponto $\mathbf{p}_j$, essa abordagem também colabora para a redução do tamanho do conjunto de treinamento. Além disso, essa construção de vetores de características favorece ainda a redução do número de pontos-chaves dos quais as correspondências devem ser obtidas assumindo que o classificador é utilizado como um pós-detector responsável por descartar pontos-indesejáveis.

8.2.3 SVM de Classe Única

Um SVM de classe única (*one-class*) foi utilizado neste experimento porque apenas o conceito de pontos indesejáveis foi desenvolvido. Essa formulação permite derivar uma função que é positiva no espaço de distribuição estimado e negativa em seu complemento (DRUCKER *et al.*, 1996; SCHöLKOPF *et al.*, 2001). Essa formulação requer a introdução de um novo parâmetro $\nu \in [0, 1]$, cujo valor está associado à taxa de vetores de suporte e à taxa do erro admitido durante o treinamento. Contudo, é razoável fixar $\nu = 0.001$ porque há interesse em obter taxas de classificação acima de 99%, sem que isso afete a seleção de parâmetros.

Figura 33: Exemplos de imagens da base Olivetti (SAMARIA; HARTER, 1994), a qual contém faces com diferentes expressões e poses, o que caracteriza essa base como difícil. Há 10 fotos para cada uma das 40 pessoas fotografadas. As imagens na última linha ilustram as variações para um mesmo indivíduo.

8.3 Bases de Imagens Selecionadas

Os exemplos de pontos indesejáveis foram extraídos a partir da base de dados BioID (JESORSKY *et al.*, 2001), da qual pares de pontos mal associados foram detectados manualmente, considerando 5 imagens das faces de cada um dos 23 indivíduos. A maioria dos pontos marcados correspondem a detalhes de pequena importância e que estão sujeitos a repetições. Exemplos desses detalhes são: barba, dentes e cabelo, dentre outros. Foram utilizados vetores de características idênticos àqueles utilizados no método SVM-SIFT.

Para os testes foi usada a base de faces AT&T (SAMARIA; HARTER, 1994). Esta base de imagens, ilustrada na Figura 33, é de domínio público e possui as seguintes características:

- Contém 400 imagens em tons de cinza;
- Resolução de 92 por 112 pixels;
- Corresponde a um total de 40 pessoas, sendo 10 de cada pessoa;

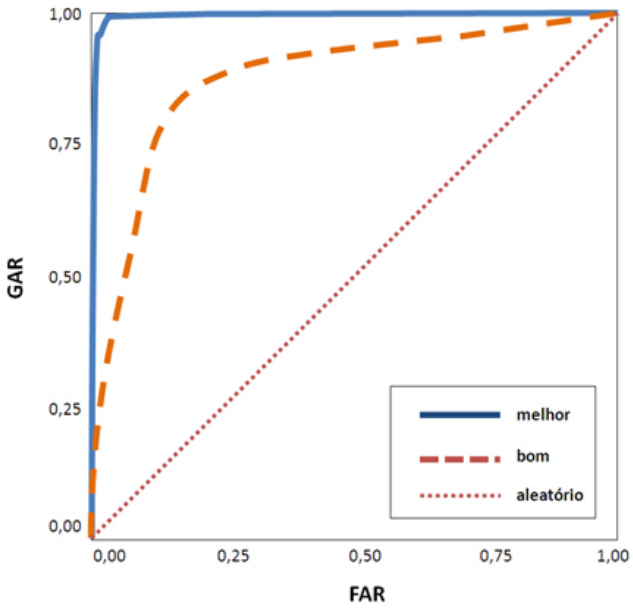

Figura 34: Exemplos de curva ROC (*Receiver Operating Characteristic*) na qual relaciona-se $FAR \times GAR$. No gráfico, o comportamento de um classificador aleatório é denotado como uma linha reta (pontilhado fino) expressando a sua má qualidade. Um classificador mais interessante apresenta uma curva (pontilhado grosso) que tende a passar próximo ao ponto $(0, 1)$, que, por sua vez, caracteriza um classificador ideal (linha sólida).

- Variações na pose da face e presença de acessórios;
- Ocorrência de distorções geométricas, o que ocorre supostamente devido ao redimensionamento das faces.

Essa base foi escolhida porque apresenta características interessantes que a tornam relativamente difícil de se classificar usando métodos padrões. Para efeito de testes, cada indivíduo é representado pelos descritores obtidos em 3 faces, de forma que as demais são usadas para o reconhecimento de exemplos autênticos. As mesmas 3 faces são utilizadas para computar os descritores que são comparados com cada uma das 10 faces dos demais 39 indivíduos para o cálculo da taxa de aceitação de falsos positivos.

8.4 Resultados

Os resultados são exibidos na forma de curvas ROC (*Receiver Operating Characteristic*), mais adequadas para a análise do poder de discriminação de um reconhecedor de faces. Uma curva ROC expressa a variação da taxa de aceitação de genuínos em função da taxa admitida de falsos positivos que permite selecionar um ponto de operação para o algoritmo por consistir em um gráfico FAR $\times$ GAR. Entenda-se como "ponto

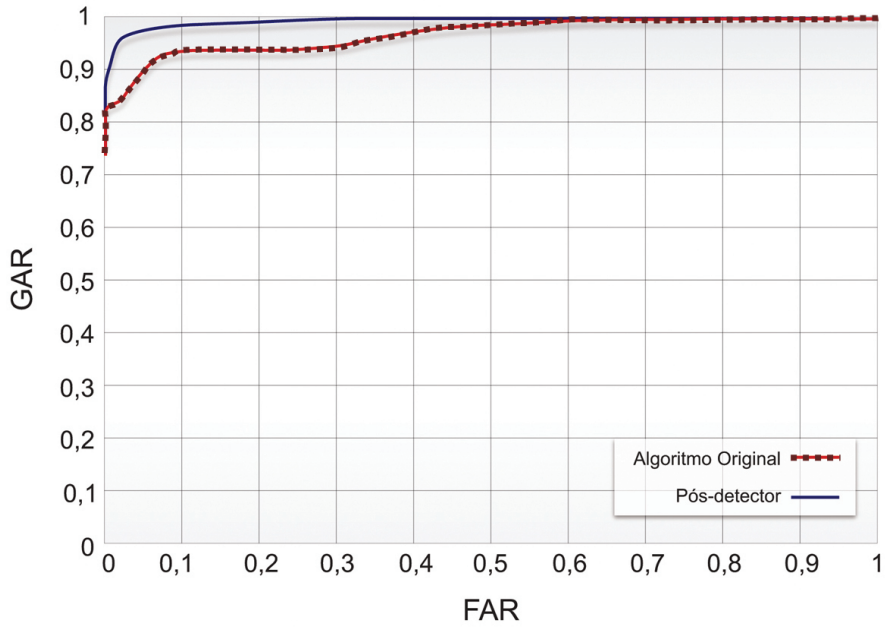

Figura 35: Curvas ROC obtidas para o reconhecimento de faces sobre a base de imagens AT&T. A curva pontilhada em vermelho (abaixo) corresponde aos resultados obtidos pelo algoritmo básico, cuja ocorrência de pontos indesejáveis introduz ruído na função de similaridade. O número de ocorrências deste caso é reduzido quando o pós-detector é utilizado, o que permite reduzir o valor de FAR em uma ordem de grandeza ao passo que o valor de GAR tende a ser o mesmo. Este segundo caso é denotado pela curva sólida em azul (acima).

de operação"todos os parâmetros e a qualidade da solução em um determinado ponto. Exemplos desse tipo de curva são ilustrados na Figura 34. Essa curva é obtida variando os parâmetros que interferem na classificação pelo, mas especificamente o limiar s mínimo de similaridade.

A Figura 35 ilustra as curvas ROC obtidas para dois casos: utilização do algoritmo básico (abaixo, em vermelho) e adoção do pós-detector de pontos indesejáveis (acima, em azul). Foram detectados em média 97 pontos-chaves por imagem utilizando o algoritmo básico. Em média, a introdução da etapa de pós-detecção implica na remoção de 6 pontos-chaves que introduzem ruído na função de similaridade. A introdução do conceito de pontos indesejáveis permite manter aproximadamente o mesmo valor de GAR enquanto o valor de FAR é reduzido em uma ordem de grandeza. Isto pode ser constatado através das curvas presentes na Figura 35.

8.5 Conclusões Preliminares

O reconhecimento facial é um problema de interesse em muitos domínios de aplicações. Esse cenário foi utilizado para mostrar que a metodologia proposta pode ser útil em outras situações nas quais o conceito de ponto de interesse não necessariamente surge: há aplicações que podem se beneficiar pela introdução do conceito de *pontos indesejáveis*.

Os resultados experimentais obtidos pela adoção do pós-detector proposto demonstram que os resultados obtidos através de métodos baseados em descritores locais podem ser significativamente melhorados utilizando a metodologia proposta. No caso do reconhecimento facial, o descarte dos pontos-chaves que confundem o processo de classificação implica na redução do número de falsos positivos.

Poucas limitações foram encontradas dada a simplicidade dessa técnica, apenas alguns pontos-chaves genuínos são classificados erroneamente como espúrios. O impacto disto praticamente não é percebido quando o método é aplicado em situações reais de uso. Além disso, como o classificador é aplicado após a detecção dos pontos-chaves, o mesmo não apresenta um grande impacto sobre a eficiência porque um número reduzido de pontos é classificado.

A introdução do conceito de *pontos indesejáveis* pode ser estendida a outros domínios de aplicações. Na construção de imagens panorâmicas (BROWN; LOWE, 2007), por exemplo, espera-se que o método numérico usado para calcular homografias entre pares de imagens venha a convergir em um maior número de casos.

Surpreendentemente, menos limitações foram encontradas no contexto do reconhecimento de faces. De uma forma geral, os pontos de interesse são mal classificados em um número pequeno de casos, o que pouco influencia o desempenho geral do método de reconhecimento. A introdução do conceito de pontos indesejáveis nesse caso é capaz de melhorar significativamente o desempenho, pois os pontos-chaves levando às situações mais ambíguas de classificação são eliminados antes que o processo de reconhecimento seja realizado.

9 Conclusões

O arcabouço matemático provido pela Teoria do Espaço de Escalas permite o desenvolvimento de modelos computacionais capazes de detectar e reconhecer objetos em imagens de forma virtualmente invariante à posição, à orientação e ao tamanho dos objetos aparecendo nessas imagens. Os métodos baseados em descritores locais são exemplos notáveis desses modelos computacionais, pois apresentam diversos atrativos e menor número de limitações.

A utilização de métodos baseados em descritores locais consiste em três etapas: detecção de pontos-chaves, cálculo de descritores e obtenção de correspondências. A primeira etapa é uma tarefa essencial por resultar em conjuntos de partes notáveis da imagem (os pontos-chaves), e que são a base da representação dos objetos utilizando essa abordagem. Essa tarefa é tipicamente realizada usando mecanismos não-supervisionados. Tais mecanismos impõem limitações que dificultam a plena utilização desse tipo de método em domínios específicos, principalmente quando o conceito de regiões de interesse pode ser associado a alguma aplicação.

Visto que a etapa de detecção é fundamental e determinante para o sucesso dos métodos baseados em descritores locais, o presente trabalho dedica-se à proposição de uma metodologia para a utilização de mecanismos supervisionados de aprendizagem de máquina no contexto da detecção de pontos-chaves. Tal metodologia é inovadora porque investiga a utilização de combinações não-lineares de operadores diferenciais geométricos, que são obtidas de forma implícita através do treinamento de um classificador não-linear – uma Máquina de Vetores de Suporte.

Essa metodologia foi desenvolvida especialmente para viabilizar a introdução de conhecimento externo sobre o conceito de pontos de interesse. Nesse contexto foram propostas estratégias para a construção de vetores de características, a construção de conjuntos de amostras para treinamento, a seleção de parâmetros para treinamento e a seleção de modelos específicos.

Os experimentos realizados mostram que a metodologia proposta é adequada para realizar as seguintes tarefas: (a) reproduzir o comportamento dos detectores não-supervisionados existentes; (b) obter detectores mais adequados para domínios específicos de aplicações; e(c) introduzir o conceito de pontos "indesejáveis"para reforçar a capacidade de reconhecimento. Os experimentos realizados consideraram importantes aplicações de Visão Computacional:

- Localização Automática de Olhos. Os resultados obtidos são comparáveis àqueles publicados por outros pesquisadores. Em particular, acredita-se que outras tarefas de processamento de faces podem se beneficiar com as ideias apresentadas neste trabalho;

- Reconhecimento de Faces. Essa importante aplicação tem sido tema recorrente em muitas pesquisas nas duas últimas décadas. Foram obtidos bons resultados de reconhecimento, em particular pelo protótipo desenvolvido.

Em particular, observou-se que a capacidade de generalização dos SVMs permite detectar pontos-chaves de forma estável. Acredita-se que este é um tema frutífero para novas pesquisas. Entretanto, a principal limitação da metodologia proposta neste trabalho consiste no tempo de processamento exigido para o treinamento e para a classificação, utilizando Máquinas de Vetores de Suporte. Mesmo assim há uma considerável quantidade de aplicações que podem se beneficiar prontamente da metodologia proposta, principalmente aquelas nas quais não há interação em tempo-real. Desta forma, sugere-se a otimização do desempenho como um trabalho futuro.

Esperamos que as ideias apresentadas neste trabalho permitam a criação de novas técnicas para lidar com problemas de Visão Computacional, principalmente no tocante às tarefas de visão em baixo nível e ao reconhecimento de padrões visuais.

Por fim, sugere-se o desenvolvimento dos seguintes temas como trabalhos futuros, os quais são considerados interessantes e frutíferos:

- Buscar mecanismos mais sofisticados para a seleção de contraexemplos para o treinamento do classificador;

- Adaptar a implementação usada nos experimentos para que utilizem outras alternativas mais eficientes, como por exemplo, as modernas Unidades de Processamento Gráfico (GPU, *Graphics Processing Unit*) atualmente disponíveis em plataformas

para computação de propósito geral (GROUP, 2009) ou ainda o conceito de Computação em Nuvem (ARMBRUST *et al.*, 2009). A primeira abordagem é imediatamente possível, visto que Catanzaro e seus coautores (CATANZARO *et al.*, 2008) conseguiram implementar o treinamento de Máquinas de Vetores de Suporte usando a GPU;

- Avaliar a utilização da metodologia proposta em outras aplicações de Visão Computacional;
- Investigar o potencial de outros classificadores não-lineares, tais como as Árvores de Decisão;
- Combinar a metodologia proposta com outros tipos de detectores;
- Realizar testes mais minuciosos sobre a estabilidade envolvendo novos conceitos. Acredita-se que novas métricas de estabilidade e metodologias de avaliação, possivelmente específicas para aplicações, podem representar uma contribuição impactante nesta linha de pesquisa;
- Adequar ou estender a metodologia proposta para a utilização de Espaços de Escala não-Lineares. Nesse caso, a pesquisa resultará em benefícios para inúmeras aplicações em outros segmentos da ciência, principalmente em áreas da Biologia, Bioquímica e Medicina.

Referências

AIZERMAN, M.; BRAVERMAN, E.; ROZONOER, L. Theoretical foundations of the potential function method in pattern recognition learning. *Automation and Remote Control*, v. 1, n. 25, p. 821–837, 1964.

AMIT, Y.; AUGUST, G.; GEMAN, D. Shape quantization and recognition with randomized trees. *Neural Computation*, v. 9, n. 1, p. 1545–1588, 1996.

ARMBRUST, M. *et al. Above the Clouds: A Berkeley View of Cloud Computing.* Berkeley, CA, February 2009.

BABAUD, J. *et al.* Uniqueness of the gaussian kernel for scale-space filtering. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-8, n. 1, p. 26–33, jan. 1986. ISSN 0162-8828.

BARTLETT, M. S.; MOVELLAN, J. R.; SEJNOWSKI, T. Face recognition by independent component analysis. *IEEE Transaction on Neural Networks*, v. 13, n. 1, p. 1450 – 1464, 2002.

BAY, H. *et al.* SURF: Speeded up robust features. v. 110, n. 3, p. 346–359, June 2008.

BENTLEY, J. L. Multidimensional binary search trees used for associative searchings. *Communications of the ACM*, ACM, New York, NY, USA, v. 18, n. 9, p. 509–517, 1975. ISSN 0001-0782.

BOARD, O. A. R. *et al. OpenGL(R) Programming Guide: The Official Guide to Learning OpenGL(R) Version 2.1.* [S.l.]: Addison-Wesley Professional, 2007. ISBN 0321481003, 9780321481009.

BOOMGAARD, R. van den; SMEULDERS, A. W. M. The morphological structure of images, the differential equations of morphological scale-space. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 16, n. 11, p. 1101–1113, November 1994.

BOSER, B. E.; GUYON, I. M.; VAPNIK, V. N. A training algorithm for optimal margin classifiers. In: *Annual ACM Workshop on Computational Learning Theory.* [S.l.]: ACM Press, 1992. p. 144–152.

BRADSKY, G. R.; PISAREVSKY, V.; BOUGUET, J. *Learning OpenCV: Computer Vision with the OpenCV Library.* [S.l.]: Springer, 2006.

BREIMAN, L. *et al. Classification and Regression Trees.* 1. ed. San Diego, CA: Chapman and Hall, 1984. ISBN 0412048418.

BROCKETT, R. W.; MARAGOS, P. Evolution equations for continuous-scale morphology. *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, v. 1, n. 1, p. 125–128, 1992.

BROWN, M.; LOWE, D. Invariant features from interest point groups. In: *In British Machine Vision Conference*. [S.l.: s.n.], 2002. p. 656–665.

BROWN, M.; LOWE, D. G. Automatic panoramic image stitching using invariant features. *International Journal of Computer Vision*, Kluwer Academic Publishers, Hingham, MA, USA, v. 74, n. 1, p. 59–73, 2007. ISSN 0920-5691.

BURGES, C. J. C. A tutorial on support vector machines for pattern recognition. *Data Min. Knowl. Discov.*, Kluwer Academic Publishers, Hingham, MA, USA, v. 2, n. 2, p. 121–167, 1998. ISSN 1384-5810.

BURT, P. J. Fast filter transforms for image processing. *CGIP*, v. 16, n. 1, p. 20–51, May 1981.

CAMPADELLI, P.; LANZAROTTI, R. Eye localization: a survey. IOS Press, Amsterdam, Netherlands, 2006.

CAMPADELLI, P.; LANZAROTTI, R.; LIPORI, G. Face localization in color images with complex background. In: *CAMP '05: Proceedings of the Seventh International Workshop on Computer Architecture for Machine Perception*. Washington, DC, USA: IEEE Computer Society, 2005. p. 243–248. ISBN 0-7695-2255-6.

CAMPADELLI, P.; LANZAROTTI, R.; LIPORI, G. Precise eye localization through a general-to-specific model definition. In: *Proceedings of the 17th British Machine Vision Conference*. [S.l.: s.n.], 2006. v. 1, n. 1, p. 187–196.

CATANZARO, B.; SUNDARAM, N.; KURTKEUTZER. Fast support vector machine training and classification on graphics processors. In: *International Conference on Machine Learning*. [S.l.: s.n.], 2008. p. 104–111.

CHANG, C.-C.; LIN, C.-J. *LIBSVM: a library for support vector machines*. [S.l.], 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

CHEN, C.-H.; ROCKETT, P. Bayesian labelling of corners using a grey-level corner image mode. In: . [S.l.: s.n.], 1997. v. 1, n. 1, p. 687–690.

CORTES, C.; VAPNIK, V. Support-vector networks. *Machine Learning*, Springer, v. 20, n. 1, p. 291–400, 1995.

CRISTIANINI, N.; SHAWE-TAYLOR, J. *An Introduction to Support Vector Machines and other kernel-based learning methods*. [S.l.]: Cambridge University Pres, 2000. ISBN 978–0521780193.

CROW, F. C. Summed-area tables for texture mapping. In: . [S.l.: s.n.], 1984. v. 1, n. 1, p. 207 – 212. ISBN 0-89791-138-5.

CROWLEY, J. L. *A Representation for Visual Information*. Pittsburgh, PA, November 1981.

DIAS, P.; KASSIM, A.; SRINIVASAN, V. A neural network based corner detection method. In: . [S.l.: s.n.], 1995. v. 4, n. 1, p. 2116–2120.

DRUCKER, H. *et al.* Support vector regression machines. In: *NIPS*. [S.l.: s.n.], 1996. p. 155–161.

EATON, J. W. *GNU Octave User Manual*. [S.l.], August 2008. Disponível em: <http://www.gnu.org/software/octave/doc/interpreter/>.

ETEMAD, K.; CHELLAPPA, R. Discriminant analysis for recognition of human face images (invited paper). In: *AVBPA '97: Proceedings of the First International Conference on Audio- and Video-Based Biometric Person Authentication*. London, UK: Springer-Verlag, 1997. p. 127–142. ISBN 3-540-62660-3.

FOUNDATION, E. *Eclipse Galileo Documentation*. [S.l.], 2009. Software available at `http://www.eclipse.org`.

GROUP, K. O. W. *The OpenCL Specification*. Oregon, CA, February 2009.

HAAR, A. Theorie der orthogonalen funktionensysteme. *Mathematische Annalen*, v. 69, n. 1, p. 331–371, 1910.

HARRIS, C.; STEPHENS, M. A combined corner and edge detector. In: . [S.l.: s.n.], 1988. v. 1, n. 1, p. 147–151.

HSU, C. W.; CHANG, C. C.; LIN, C. J. *A practical guide to support vector classification*. Taipei, Taiwan, 2003. Disponível em: <http://www.csie.ntu.edu.tw/~cjlin/papers/guide-/guide.pdf>.

HSU, C. wei; CHANG, C. chung; LIN, C. jen. *A Practical Guide to Support Vector Classification*. [S.l.], 2003. Disponível em: <http://www.csie.ntu.edu.tw/~cjlin/papers-.html>.

HUBEL, D. H. *Eye, Brain, and Vision*. New York, NY, USA: Scientific American Library, 1988. ISBN 978-0716760092.

HUBEL, D. H.; WIESEL, T. N. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *Journal of Physiology*, Kluwer Academic Publishers, v. 160, n. 1, p. 106–154, 1962. ISSN 0022-3751. Disponível em: <http://view.ncbi.nlm.nih.gov/pubmed/14449617>.

HWANG, M.-C. *et al.* Person identification system for future digital TV with intelligence. *Consumer Electronics, IEEE Transactions on*, v. 53, n. 1, p. 218–226, february 2007. ISSN 0098-3063.

IERUSALIMSCHY, R.; FIGUEIREDO, L. H. de; CELES, W. *Lua 5.2 Reference Manual*. [S.l.], 2010. Software available at `http://www.lua.org`.

IVANCIUC, O. Applications of support vector machines in chemistry. *Reviews in Computational Chemistry*, Wiley-VCH, v. 23, n. 1, p. 291–400, 2007.

JACKWAY, P. Morphological scale-space. In: . [S.l.: s.n.], 1992. v. 1, n. 1, p. 252–255.

JEONG, J. The jpeg standard. McGraw-Hill, Inc., Hightstown, NJ, USA, p. 91–99, 1997.

JESORSKY, O.; KIRCHBERG, K. J.; FRISCHHOLZ, R. Robust face detection using the hausdorff distance. In: *AVBPA '01: Proceedings of the Third International Conference on Audio- and Video-Based Biometric Person Authentication*. London, UK: Springer-Verlag, 2001. p. 90–95. ISBN 3-540-42216-1.

KADIR, T.; BRADY, M. Saliency, scale and image description. *International Journal of Computer Vision*, Kluwer Academic Publishers, Hingham, MA, USA, v. 45, n. 2, p. 83–105, 2001. ISSN 0920-5691.

KADIR, T.; ZISSERMAN, A.; BRADY, M. An affine invariant salient region detector. In: *ECCV (1)*. [S.l.: s.n.], 2004. p. 228–241.

KEERTHI, S. S.; LIN, C.-J. Asymptotic behaviors of support vector machines with gaussian kernel. *Neural Computation*, v. 15, n. 7, p. 1667–1689, 2003.

KIENZLE, W. *et al.* Learning an interest operator from human eye movements. In: . [S.l.: s.n.], 2005. v. 1, n. 1, p. 1–8.

KLINGER, A. Patterns and search statistics. *Optimizing Methods in Statistic*, Academic Press, New York, NY, USA, v. 1, n. 1, p. 303–337, 1971.

KOENDERIK, J. J.; DOORN, A. van. The structure of images. In: . [S.l.: s.n.], 1984. v. 50, n. 1, p. 363 – 370.

KOENDERINK, J. J.; van Doorn., A. J. Operational significance of receptive field assemblies. *Biological Cybernetics*, Springer, Berlim, Alemanha, v. 58, n. 3, p. 163–171, 1988. ISSN 0340-1200.

KROON, B. *et al.* Eye localization in low and standard definition content with application to face matching. *Computer Vision and Image Understanding*, v. 113, n. 8, p. 921 – 933, 2009. ISSN 1077-3142. Disponível em: <http://www.sciencedirect.com-/science/article/B6WCX-4VYP9C3-1/2/840994ffdab1ad4d19e2e8e9bab0d038>.

LEPETIT, V.; FUA, P. Keypoint recognition using randomized trees. In: . [S.l.: s.n.], 2006. v. 28, n. 9, p. 1465–1479.

LIENHART, R.; MAYDT, J. An extended set of haar-like features for rapid object detection. In: *Image Processing. 2002. Proceedings. 2002 International Conference on*. [s.n.], 2002. v. 1, p. 900–903. Disponível em: <http://dx.doi.org/10.1109/ICIP.2002-.1038171>.

LINDEBERG, T. On the behaviour in scale-space of local extrema and blobs. In: *SCIA91*. [S.l.: s.n.], 1991. p. 8–17.

LINDEBERG, T. Scale-space theory: A basic tool for analysing structures at different scales. *Journal of Applied Statistics*, v. 21(2), p. 224–270, 1994. Disponível em: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.49.4689>.

LINDEBERG, T. Feature detection with automatic scale selection. *International Journal of Computer Vision*, Kluwer Academic Publishers, Hingham, MA, USA, v. 30, n. 2, p. 79–116, 1998. ISSN 0920-5691.

LOWE, D. G. Object recognition from local scale-invariant features. In: *ICCV '99: Proceedings of the International Conference on Computer Vision-Volume 2*. Washington, DC, USA: IEEE Computer Society, 1999. p. 1150–1157. ISBN 0-7695-0164-8.

LOWE, D. G. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, Kluwer Academic Publishers, Hingham, MA, USA, v. 60, n. 2, p. 91–110, 2004. ISSN 0920-5691.

LYONS, M. J.; BUDYNEK, J.; AKAMATSU, S. Automatic classification of single facial images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 21, n. 12, p. 1357–1362, 1999.

MA, Y. *et al.* Robust precise eye location under probabilistic framework. In: . [S.l.: s.n.], 2004. v. 1, n. 1, p. 339–344.

MACQUEEN, J. B. Some methods for classification and analysis of multivariate observation. *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, University of California Press, v. 1, n. 1, p. 281–297, 1992.

MAIA, J. G. R.; GOMES, F. de C.; SOUZA, O. de. Automatic eye localization in color images. In: *XX Brazilian Symposium on Computer Graphics and Image Processing*. [S.l.: s.n.], 2007. v. 1, n. 1, p. 195–204. ISSN 1530-1834.

MARTINEZ, A. M.; BENAVENTE, R. *The AR face database*. West Lafayette, IN, June 1998.

MATHWORKS. *MATLAB User Manual*. [S.l.], January 2010. Disponível em: <http:/-/www.mathworks.com/access/helpdesk/help/techdoc/matlab_product_page.html>.

MIKOLAJCZYK, K.; SCHMID, C. An affine invariant interest point detector. In: *Proceedings of the 7th European Conference on Computer Vision, Copenhagen, Denmark*. Springer, 2002. p. 128–142. Copenhagen. Disponível em: <http://perception.inrialpes.fr-/Publications% -/2002/MS02>.

MIKOLAJCZYK, K.; SCHMID, C. Scale & affine invariant interest point detectors. *International Journal of Computer Vision*, Kluwer Academic Publishers, Hingham, MA, USA, v. 60, n. 1, p. 63–86, 2004. ISSN 0920-5691.

MIKOLAJCZYK, K.; SCHMID, C. A performance evaluation of local descriptors. *IEEE Trans. Pattern Anal. Mach. Intell.*, IEEE Computer Society, Washington, DC, USA, v. 27, n. 10, p. 1615–1630, 2005. ISSN 0162-8828.

MIKOLAJCZYK, K. *et al.* A comparison of affine region detectors. *International Journal of Computer Vision*, Kluwer Academic Publishers, Hingham, MA, USA, v. 65, n. 1–2, p. 43–72, 2005. ISSN 0920–5691.

NIU, Z. *et al.* 2d cascaded adaboost for eye localization. In: *ICPR '06: Proceedings of the 18th International Conference on Pattern Recognition*. Washington, DC, USA: IEEE Computer Society, 2006. p. 1216–1219. ISBN 0-7695-2521-0.

OLIVEIRA, M. M.; BISHOP, G.; MCALLISTER, D. Relief texture mapping. In: *Proceedings of SIGGRAPH 2000 (vol. 1)*. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 2000. p. 359–368. ISBN 1-58113-208-5.

OZUYSAL, M.; FUA, P.; LEPETIT, V. Fast keypoint recognition in ten lines of code. In: . [S.l.: s.n.], 2007. v. 1, n. 1, p. 1–8.

OZUYSAL, M. *et al.* Feature harvesting for tracking-by-detection. In: *Proceedings of the European Conference on Computer Vision*. [S.l.: s.n.], 2006. v. 3953, n. 1, p. 592–605.

P754, I. T. *ANSI /IEEE 754-1985 Standard for Binary Floating-Point Arithmetic*. [S.l.], August 1985.

PERONA, P.; MALIK, J. Scale-space and edge detection using anisotropic diffusion. In: . [S.l.: s.n.], 1992. v. 12, n. 1, p. 629 – 639.

QUINLAN, J. R. Induction of decision trees. *Machine Learning*, Kluwer Academic Publishers, Hingham, MA, USA, v. 1, n. 1, p. 81–106, 1986. ISSN 0885-6125. Disponível em: <http://dx.doi.org/10.1023/A:1022643204877>.

ROSENBLATT, F. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, v. 65, n. 6, p. 386–408, 1958.

ROSTEN, E.; T.DRUMMOND. Fusing points and lines for high performance tracking. In: . [S.l.: s.n.], 2005. v. 1, n. 1, p. 1508–1511.

ROSTEN, E.; T.DRUMMOND. Machine learning for high-speed corner detection. In: . [S.l.: s.n.], 2006. v. 1, n. 1, p. 430–443.

SAMARIA, F.; HARTER, A. Parameterisation of a stochastic model for human face identification. In: *IEEE Workshop on Applications of Computer Vision*. Sarasota, Florida: [s.n.], 1994. p. 138–142. ISBN 0-8186-6410-X.

SCHöLKOPF, B. *et al.* Estimating the support of a high-dimensional distribution. *Neural Computation*, MIT Press, Cambridge, MA, USA, v. 13, n. 7, p. 1443–1471, 2001. ISSN 0899–7667.

SLOT, K.; KIM, H. Keypoints derivation for object class detection with SIFT algorithm. In: . [S.l.]: Springer, 2006. v. 4029, n. 1, p. 850–859. ISBN 354035748–3.

TEIXEIRA, R. C. *Introdução aos Espaços de Escala*. [S.l.]: Instituto de Matemática Pura e Aplicada, 2001.

TOMPSETT, M. F. *et al.* Charge-coupled imaging devices: Experimental results. *IEEE Transactions on Electron Devices*, v. 18, n. 11, p. 992–996, 1971. ISSN 0018-938.

TSAI, D. M. Boundary-based corner detection using neural networks. In: . [S.l.: s.n.], 1997. v. 30, n. 1, p. 85–97.

TURK, M.; PENTLAND, A. Face recognition using eigenfaces. In: . [S.l.: s.n.], 1991. v. 1, n. 1, p. 586–591.

TUYTELAARS, T.; GOOL, L. V. Matching widely separated views based on affine invariant regions. *International Journal of Computer Vision*, Kluwer Academic Publishers, Hingham, MA, USA, v. 59, n. 1, p. 61–85, 2004. ISSN 0920-5691.

TUYTELAARS, T. *et al.* Matching of affinely invariant regions for visual servoing. In: . [S.l.: s.n.], 1999. v. 2, n. 1, p. 1601 –1606 vol.2.

TUYTELAARS, T.; MIKOLAJCZYK, K. Local invariant feature detectors: a survey. *Foundation and Trends on Computer Graphics and Vision*, Now Publishers Inc., Hanover, MA, USA, v. 3, n. 3, p. 177–280, 2008. ISSN 1572-2740.

TUYTELAARS, T.; Van Gool, L. Wide baseline stereo matching based on local affinely invariant regions. In: . [S.l.: s.n.], 2000. v. 1, n. 1, p. 412–425.

VAPNIK, V. *Estimation of Dependences Based on Empirical Data*. [S.l.]: Springer-Verlag, 1982. ISBN 0-387-98780-0.

VAPNIK, V. *The Nature of Statistical Learning Theory*. New York, USA: Springer-Verlag, 1995. ISBN 0–387–98780–0.

VEDALDI, A. *SIFT++: A lightweight C++ implementation of SIFT*. [S.l.], 2009. Software available at `http://www.vlfeat.org/~vedaldi/code/siftpp.html`.

VELHO, L.; TEIXEIRA, R.; GOMES, J. Introdução aos espaços de escala. In: *Escola de Computação*. [S.l.: s.n.], 2000.

VIOLA, P.; JONES, M. J. Robust real-time face detection. *International Journal of Computer Vision*, Kluwer Academic Publishers, Hingham, MA, USA, v. 57, n. 2, p. 137–154, 2004. ISSN 0920-5691.

WAN, L.; BAO, W. Research and application of animal disease intelligent diagnosis based on support vector machine. In: . [S.l.]: Cis, 2009. v. 2, n. 1, p. 66–70.

WANDELL, B. A. *Foundations of Vision*. Sunderland, Massachusetts: Sinauer Associates Inc., 1995.

WANG, P. *et al.* Automatic eye detection and its validation. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE, v. 3, n. 1, p. 164–164, June 2005. ISSN 1063-6919.

WEBER, M. *CALTECH face database*. [S.l.], August 1999. Disponível em: <http:/-/www.vision.caltech.edu/html-files/archive.html>.

WITKIN, A. P. Scale-space filtering. In: *Proceedings of the International Joint Conference on Artificial Intelligence of Biological Systems*. [S.l.: s.n.], 1983. v. 1, n. 1, p. 1019–1022.

YOUNG, R. A. The gaussian derivative model for machine vision: Visual cortex simulation. *Publication GMR-5323*, General Motors Research Laboratories, Computer Science Dept., 30500 Mound Road,Box 9055, Warren,Michigan, USA, v. 1, n. 1, 1986.

ZHAN, Y.; SHENB, D. Design efficient support vector machine for fast classification. *Pattern Recognition*, v. 38, n. 1, p. 157–161, 2005.

ZHOU, Z.-H.; GENG, X. Projection functions for eye detection. *Pattern Recognition*, v. 37, n. 5, p. 1049–1056, 2004.