FÁBIO RODRIGO AMARAL

**Avaliação de um método baseado em
máquinas de suporte vetorial de múltiplos
núcleos e retificação de imagens para
classificação de objetos em imagens
onidirecionais**

**São Paulo
2010**

FÁBIO RODRIGO AMARAL

**Avaliação de um método baseado em
máquinas de suporte vetorial de múltiplos
núcleos e retificação de imagens para
classificação de objetos em imagens
onidirecionais**

Dissertação apresentada à Escola Politécnica
da Universidade de São Paulo para obtenção
do Título de Mestre em Engenharia Elétrica.

**São Paulo
2010**

FÁBIO RODRIGO AMARAL

**Avaliação de um método baseado em
máquinas de suporte vetorial de múltiplos
núcleos e retificação de imagens para
classificação de objetos em imagens
onidirecionais**

Dissertação apresentada à Escola Politécnica
da Universidade de São Paulo para obtenção
do Título de Mestre em Engenharia Elétrica.

Área de Concentração:
Sistemas Digitais

Orientadora:
Profa. Livre Docente Anna Helena Reali
Costa

**São Paulo
2010**

Agradecimentos

Em primeiro lugar agradeço a Deus, por mais um objetivo alcançado.

À professora Anna Helena Reali Costa, minha orientadora, pela confiança, ajuda, paciência e principalmente pela amizade.

Aos professores Ronaldo Fumio e João Kogler, pelos valiosos apontamentos realizados como examinadores deste trabalho.

À minha esposa Priscila, pelo amor, incentivo e compreensão.

À minha família, pelo apoio incondicional durante todos os desafios até agora enfrentados.

Aos amigos do LTI, Nicolau e Valdinei, pela experiência compartilhada.

Ao amigo André, pelo incentivo mútuo nesta caminhada.

Resumo

Apesar da popularidade das câmeras onidirecionais aplicadas à robótica móvel e da importância do reconhecimento de objetos no universo mais amplo da robótica e da visão computacional, é difícil encontrar trabalhos que relacionem ambos na literatura especializada. Este trabalho visa avaliar um método para classificação de objetos em imagens onidirecionais, analisando sua eficácia e eficiência para ser aplicado em tarefas de auto-localização e mapeamento de ambientes feitas por robôs moveis. Tal método é construído a partir de um classificador de objetos, implementado através de máquinas de suporte vetorial, estendidas para a utilização de Aprendizagem de Múltiplos Núcleos. Também na construção deste método, uma etapa de retificação é aplicada às imagens onidirecionais, de modo a aproximá-las das imagens convencionais, às quais o classificador utilizado já demonstrou bons resultados. A abordagem de Múltiplos Núcleos se faz necessária para possibilitar a aplicação de três tipos distintos de detectores de características em imagens, ponderando, para cada classe, a importância de cada uma das características em sua descrição. Resultados experimentais atestam a viabilidade de tal proposta.

Palavras-chave: RECONHECIMENTO DE PADRÕES, VISÃO COMPUTACIONAL, ROBÓTICA MÓVEL

Abstract

Despite the popularity of omnidirectional cameras used in mobile robotics, and the importance of object recognition in the broader universe of robotics and computer vision, it is difficult to find works that relate both in the literature. This work aims at performing the evaluation of a method for object classification in omnidirectional images, evaluating its effectiveness and efficiency considering its application to tasks of self-localization and environment mapping made by mobile robots. The method is based on a multiple kernel learning extended support vector machine object classifier. Furthermore, an unwrapping step is applied to omnidirectional images, to make them similar to perspective images, to which the classifier used has already shown good results. The Multiple Kernels approach is necessary to allow the use of three distinct types of feature detectors in omnidirectional images by considering, for each class, the importance of each feature in the description. Experimental results demonstrate the feasibility of such a proposal.

Keywords: object recognition, computer vision, mobile robotics

Lista de Figuras

2.1	Exemplos para hiperplanos de separação	21
2.2	Separações de três dados no espaço bidimensional por meio de retas (SCHÖLKOPF; SMOLA, 2001)	22
2.3	Separações de quatro dados no espaço bidimensional por meio de retas (SCHÖLKOPF; SMOLA, 2001)	22
2.4	Hiperplano ótimo para padrões linearmente separáveis	23
2.5	Conjuntos de dados linearmente não separáveis	32
3.1	Algoritmo DoG em execução: à esquerda, imagens suavizadas pelo filtro Gaussiano. Imagens Gaussianas adjacentes são subtraídas para produzir a Diferença de Gaussianas à direita. O processo se repete para outras escalas da imagem de entrada reamostrada (LOWE, 2004)	39
3.2	Imagem suavizada progressivamente no mesmo espaço de escala	39
3.3	Os pontos extremos nas imagens produzidas pela função DoG são detectados comparando um pixel candidato (marcado com X) com seus 26 vizinhos (marcados com círculo) (LOWE, 2004)	40
3.4	Exemplo de construção do descritor SIFT (VEDALDI; FULKERSON, 2005)	42
3.5	Construção do descritor SIFT (VEDALDI; FULKERSON, 2005)	42
3.6	Exemplo de reconhecimento dos pontos obtidos pelo processo SIFT para detecção de um objeto (HESS, 2010)	43
5.1	Classificação através do método avaliado	51
5.2	Processo de retificação da imagem onidirecional (PRONOBIS, 2009)	52
5.3	Exemplo de construção do <i>Geometric Blur</i> . À esquerda, a imagem original; à direita, a distorção realizada em função da distância para o ponto de interesse, marcado em vermelho (BERG; BERG; MALIK, 2005)	53
5.4	Construção do descritor PHOW. A imagem é dividida, e para cada região um histograma de ocorrências das Palavras Visuais é computado. O descritor PHOW é construído concatenando tais histogramas (LAZEBNIK; SCHMID; PONCE, 2006)	54

5.5	Construção do descritor PHOW da aplicação real (BOSCH; ZISSERMAN; MUNOZ, 2007)	55
5.6	Exemplo de localização de descritores SSIM; a) representação gráfica de um descritor; b) localização do descritor em imagens de teste (SHECHTMAN; IRANI, 2007)	56
5.7	Criação do descritor de auto-similaridade (SHECHTMAN; IRANI, 2007)	56
6.1	(a) Modelo de câmera onidirecional catadióptricas de espelho hiperbólico central. (b) Montagem de um dos robôs utilizados na coleta dos dados do projeto COLD (PRONOBIS; CAPUTO, 2009)	61
6.2	Exemplos de imagens convencionais para os objetos avaliados: (a) sofá, (b) monitor, (c) máquina copiadora e (d) extintor de incêndio (GRIFFIN; HOLUB; PERONA, 2007)	62
6.3	Exemplo de imagem onidirecional do conjunto de dados Cogniron (ZIVKOVIC et al., 2008)	63
6.4	Exemplos de imagens onidirecionais para os objetos avaliados: (a) sofá, (b) monitor, (c) máquina copiadora e (d) extintor de incêndio (PRONOBIS; CAPUTO, 2009)	64
6.5	Imagem onidirecional retificada (PRONOBIS; CAPUTO, 2009)	64
6.6	Matriz de confusão para teste da configuração 1 (Treinamento e teste com imagens convencionais)	66
6.7	Matriz de confusão para teste da configuração 2 (Treinamento e teste com imagens onidirecionais)	67
6.8	Matriz de confusão para teste da configuração 3 (Treinamento e teste com imagens retificadas)	68
6.9	Matriz de confusão para teste da configuração 4 (Treinamento com imagens convencionais; teste com imagens onidirecionais)	69
6.10	Matriz de confusão para teste da configuração 5 (Treinamento com imagens convencionais; teste com imagens retificadas)	70

Lista de Tabelas

2.1	Funções Núcleo mais utilizadas	35
6.1	Tabela de contingência para classificação binária	63
6.2	Resultado para teste da configuração 1 (Treinamento e teste com imagens convencionais)	65
6.3	Resultado para teste da configuração 2 (Treinamento e teste com imagens onidirecionais)	66
6.4	Resultado para teste da configuração 3 (Treinamento e teste com imagens retificadas)	68
6.5	Resultado para teste da configuração 4 (Treinamento com imagens convencionais; teste com imagens onidirecionais)	69
6.6	Resultado para teste da configuração 5 (Treinamento com imagens convencionais; teste com imagens retificadas)	70

Lista de Siglas

CCD — *Charge-Coupled Device*

DoG — Diferença de Gaussianas

GB — *Geometric Blur*

GLOH — *Gradient Location and Orientation Histogram*

HSV — *Hue Saturation Value*

MLP — Perceptron de Múltiplas Camadas

MSV — Máquina de Suporte Vetorial

PCA — Análise dos Componentes Principais

PHOW — *Pyramid Histogram Of visual Words*

PTZ — *Pan/Tilt/Zoom*

RBF — *Radial Basis Function*

RGB — *Red Green Blue*

SIFT — *Scale-Invariant Feature Transform*

SLAM — Mapeamento e Localização Simultâneos

SSIM — *Self-Similarity*

SURF — *Speeded Up Robust Features*

Lista de Símbolos

$\tau = \{(\mathbf{x}_i, d_i)\}_{i=1}^N$ — Conjunto de treinamento para uma MSV

d_i — Saída desejada para o i -ésimo item em um conjunto de treinamento

$\mathbf{x}_i$ — Vetor de entrada no i -ésimo item

N — Número de amostras de treinamento em um conjunto

N_s — Número de vetores de suporte em uma MSV

$\mathbf{w}$ — Vetor de pesos em uma MSV

b — Deslocamento (bias) de um hiperplano de separação para uma MSV

ρ — Margem de separação em uma MSV

$\Phi(\mathbf{w})$ — Função custo do problema primal de uma MSV

$J(\mathbf{w}, b, \alpha)$ — Função lagrangiana

α — Multiplicador de Lagrange

$Q(\alpha)$ — Função objetivo do problema dual de uma MSV

ξ — Variável de folga de uma MSV

C — Constante de regularização de uma MSV com margens suaves

μ — Multiplicador de Lagrange introduzidos para forçar a não negatividade das variáveis de folga ξ

$\{\varphi_j(\mathbf{x})\}_{j=1}^{m_1}$ — Conjunto de transformações não lineares

m_1 — Dimensão do espaço de características

$K(\mathbf{x}_i, \mathbf{x}_j)$ — Função Núcleo de uma MSV

η — Vetor de pesos atribuídos aos Núcleos

γ — Vetor de inicialização de η

$T(\eta)$ — Função de custo do problema primal de uma MSV de Múltiplos Núcleos

$W(\eta)$ — Função objetivo do problema dual de uma MSV de Múltiplos Núcleos

$L(x,y,\sigma)$ — Imagem suavizada por um filtro Gaussiano

$G(x,y,\sigma)$ — Filtro Gaussiano

σ — Desvio Padrão

$I(x,y)$ — Imagem de entrada

$D(x,y,\sigma)$ — Diferença de Gaussianas

$\theta(x,y)$ — Orientação do gradiente para construção do descritor SIFT

$m(x,y)$ — Módulo do gradiente para construção do descritor SIFT

Sumário

Lista de Figuras	6
Lista de Tabelas	8
Lista de Siglas	9
Lista de Símbolos	10
Sumário	12
1 Introdução	15
1.1 Robótica Móvel e Visão Onidirecional	17
1.2 <i>Pascal Visual Object Classes Challenge</i>	18
1.3 Objetivos	18
1.4 Justificativas	19
1.5 Organização do Texto	20
2 Máquinas de Suporte Vetorial	21
2.1 Hiperplano ótimo para padrões linearmente separáveis	23
2.2 MSV com Margem Suave	28
2.3 Hiperplano ótimo para padrões não-separáveis	31
2.4 Métodos para Múltiplas Classes	36
2.4.1 Método Um-Contra-Um	36
2.4.2 Método Um-Contra-Todos	36
3 Scale Invariant Feature Transform	37
3.1 Detecção dos Pontos de Interesse	37
3.1.1 Detecção dos extremos no espaço de escala	38
3.1.2 Seleção dos pontos de interesse	39

3.2	Construção do Descritor	40
3.2.1	<i>Determinação da orientação</i>	40
3.2.2	<i>Montagem do descritor</i>	41
3.3	Reconhecimento	41
4	Panorama atual	44
4.1	Marcos Visuais	44
4.2	Máquinas de Suporte Vetorial na Classificação de Objetos	46
4.3	Sistemas de Visão Onidirecional	47
4.4	Considerações finais	48
5	Detalhamento	50
5.1	Retificação da Imagem Onidirecional	51
5.2	Detecção das Características da Imagem	52
5.2.1	<i>Geometric Blur</i>	53
5.2.2	<i>PhowGray / PhowColor</i>	53
5.2.3	<i>Self-Similarity</i>	55
5.3	Múltiplos Núcleos	56
5.4	Aprendizagem	58
5.5	Execução	58
6	Experimentos e Resultados	60
6.1	Ambiente de Teste	60
6.2	Conjuntos de Dados	61
6.3	Métodos para Avaliação do Classificador	63
6.4	Resultados	65
6.4.1	<i>Treinamento e teste com imagens convencionais</i>	65
6.4.2	<i>Treinamento e teste com imagens onidirecionais</i>	66

6.4.3	<i>Treinamento e teste com imagens retificadas</i>	67
6.4.4	<i>Treinamento com imagens convencionais; teste com imagens onidirecionais</i>	67
6.4.5	<i>Treinamento com imagens convencionais; teste com imagens retificadas</i>	69
7	Conclusões e Trabalhos Futuros	71
	Referências Bibliográficas	72

1 Introdução

O reconhecimento de objetos por sistemas computacionais autônomos pode tornar possível um número muito grande de tarefas, além de contribuir para o aprimoramento de várias outras já desenvolvidas. Tarefas mais complexas que exijam interação com o ambiente ou com seres humanos dependem, em sua grande maioria, de tal habilidade que, por sua vez, depende de informações obtidas previamente ou mesmo observadas em tempo real sobre o ambiente em questão. Neste panorama, a **visão computacional** pode oferecer um bom conjunto de informações do ambiente, uma vez que busca a construção de descrições explícitas e significantes de objetos físicos a partir de imagens (BALLARD; BROWN, 1982), ou seja, almeja transformar a informação visual obtida através de sensores visuais (câmeras digitais) em informação útil para tais sistemas autônomos.

Na literatura relacionada ao reconhecimento de objetos encontram-se dois tipos distintos de tarefas: a classificação e a detecção. Na classificação, o objetivo é avaliar na imagem de teste se a mesma contém alguma instância do objeto de interesse, sem a necessidade de indicar sua localização. A tarefa de detecção por sua vez, tem o objetivo de verificar a existência do objeto de interesse, bem como indicar sua localização na imagem. Se várias instâncias do objeto forem localizadas na imagem de teste, a localização de todas estas instâncias precisa ser retornada pelo detector.

Em uma análise superficial, o reconhecimento de objetos é uma tarefa executada por sistemas baseados em técnicas de aprendizagem de máquinas. Por este motivo, um ponto importante a ser considerado aqui é o modelo de aprendizagem utilizado. Uma maneira adotada para categorizar tais modelos (ULUSOY; BISHOP, 2005) realiza uma divisão em duas categorias: modelos gerativos e modelos discriminativos. Ambos modelos são baseados nas teorias de probabilidade mas diferem na maneira em que a aprendizagem é realizada. Os modelos gerativos (FEIFEI; FERGUS; PERONA, 2007; FERGUS; PERONA; ZISSERMAN, 2003; MIKOLAJCZYK; LEIBE; SCHIELE, 2006) baseiam suas decisões sobre as distribuições de probabilidades conjuntas de imagens e classes, as quais podem ser obtidas através do aprendizado em separado, das probabilidades a priori das classes e das verossimilhanças, ou seja, as densidades de probabilidade das imagens condicionadas às classes. Por sua vez, modelos discriminativos (AGARWAL; AWAN; ROTH, 2004; TORRALBA; MURPHY; FREEMAN, 2004; VIOLA; JONES, 2004) introduzem nas probabilidades a posteriori um modelo paramétrico que é estimado com base em imagens rotuladas de treinamento.

Uma técnica de aprendizagem de máquinas amplamente adotada é a Máquina

de Suporte Vetorial (MSV). Esta técnica baseada no modelo discriminativo, foi inicialmente proposta por Vapnik (1995) e atualmente tem sido aplicada com sucesso em diversos problemas de classificação e regressão. Basicamente, uma MSV é uma máquina linear cuja ideia principal consiste em construir um hiperplano como superfície de decisão de tal forma que a margem de separação entre os exemplos rotulados como positivos e negativos seja máxima. As MSVs podem ser aplicadas na classificação de padrões tanto linearmente separáveis quanto linearmente não separáveis, sendo que para o segundo caso, uma etapa inicial é realizada para mapear os vetores de entrada não linearmente separáveis para um *espaço de características* de alta dimensionalidade, onde os padrões passam a ser linearmente separáveis com alta probabilidade. A construção do hiperplano de decisão é então realizada neste novo espaço de características e não mais sobre o espaço de entrada (HAYKIN, 1998).

Para ambas as tarefas relacionadas ao reconhecimento de objetos, encontram-se na literatura uma grande quantidade de propostas, muitas vezes avaliadas sobre conjuntos de imagens como o Caltech 101 (FEIFEI; FERGUS; PERONA, 2007; GRIFFIN; HOLUB; PERONA, 2007) ou mesmo através de competições relacionadas, como o *Pascal Visual Object Classes Challenge* (EVERINGHAM et al., 2009). No entanto, estes conjuntos de imagens sobre os quais os trabalhos são baseados e avaliados possuem apenas imagens obtidas através de câmeras convencionais. Tais imagens são referenciadas neste texto como imagens convencionais. Em áreas como a robótica móvel, outro tipo de câmera tem sido cada vez mais aplicado com sucesso: as câmeras onidirecionais (TAMIMI et al., 2006; YANG et al., 2009; GOEDEMÉ et al., 2007). Tais câmeras permitem obter, em uma única imagem, uma visão quase completa do ambiente, à pena de possíveis distorções de acordo com o modelo utilizado.

Apesar da popularidade das câmeras onidirecionais e da importância do reconhecimento de objetos no universo da robótica, é difícil encontrar na literatura especializada trabalhos que relacionem ambos. Isto motivou o desenvolvimento deste trabalho, visando avaliar um classificador, já aplicado com sucesso em imagens convencionais (VEDALDI et al., 2009b), sobre um conjunto de objetos extraídos de imagens onidirecionais após uma etapa adicional de retificação aplicada a tais imagens. A etapa de retificação da imagem onidirecional, aplicada ao processo de classificação, teve inspiração no trabalho de Puig, Guerrero e Sturm (2008), onde uma análise sobre o casamento de características locais entre imagens convencionais e onidirecionais retificadas mostrou melhores resultados, comparados ao casamento de características locais entre imagens convencionais e onidirecionais não retificadas.

O *Pascal Visual Object Classes Challenge*, que será apresentado formalmente mais adiante neste capítulo, além de ser uma excelente fonte de imagens e anotações para o desenvolvimento de trabalhos baseados em visão, é uma organização onde

os principais grupos de trabalho relacionados a visão computacional do mundo se encontram para uma competição anual. Algumas propostas de sucesso para ambas as tarefas, de detecção e classificação de objetos, podem ser encontradas nesta competição. Portanto o classificador utilizado nas avaliações realizadas para este documento foi extraído de um dos trabalhos vencedores (VEDALDI et al., 2009a; FELZENSZWALB et al., 2007) da competição realizada em 2009 (EVERINGHAM et al., 2009). A proposta de Vedaldi et al. (2009a) para o detector de objetos vencedor faz uso de um classificador que combina a proposta de aprendizado de Múltiplos Núcleos para as MSVs de Varma e Ray (2007) com um conjunto de características locais extraídas das imagens. O detalhamento deste classificador será realizado no Capítulo 5.

Os resultados obtidos neste trabalho foram publicados no III *Workshop on Computational Intelligence* (AMARAL; COSTA, 2010). Avaliações prévias de pouco sucesso com outros métodos de classificação foram também publicadas em (AMARAL; COSTA, 2009).

1.1 Robótica Móvel e Visão Onidirecional

Uma das ramificações da robótica que, de longa data, tem sido foco de pesquisas por seu grande potencial de aplicações inovadoras é a robótica móvel. A integração das duas áreas, visão computacional e robótica móvel, tem sido fortemente explorada. As câmeras têm se tornado um sensor desejado em várias aplicações devido a seu baixo custo, tamanhos reduzidos e facilidade de integração. São vários os modelos de câmeras utilizados como sensores em robôs móveis.

A escolha do modelo de câmera depende da aplicação e do tipo de informação que se deseja obter dela. Aplicações que demandam a informação de profundidade, por exemplo, normalmente fazem uso de sistemas de visão estereoscópicos, composto por duas câmeras alinhadas de maneira que a diferença entre as imagens permite a inferência da topografia da cena e, por consequência, a representação tridimensional dos objetos.

Na robótica móvel, a possibilidade da visão do ambiente todo ao redor do robô tem enorme valor. Tal visualização do ambiente é alcançada com as câmeras onidirecionais. A tarefa de SLAM, do inglês *Simultaneous Localization And Mapping*, é beneficiada com o uso deste sistema de visão, pois marcos visuais de todo o ambiente podem ser obtidos na cena ao longo do percurso do robô, alimentando a construção de um mapa mais detalhado do ambiente. Novas abordagens para o SLAM como em (SELVATICI; COSTA; DELLAERT, 2008) exploram não mais os marcos visuais genéricos do ambiente para a construção do mapa, mas sim objetos identificados no

ambiente. Nesta linha, o presente trabalho endereça a tarefa de classificação de tais objetos na cena.

Podemos ainda encontrar câmeras onidirecionais estereoscópicas, que unem a capacidade de visão completa do ambiente e a inferência da profundidade na cena.

1.2 *Pascal Visual Object Classes Challenge*

PASCAL (sigla para *Pattern Analysis, Statistical Modeling and Computational Learning*) é o nome dado ao grupo fundado, oficialmente, em dezembro de 2003 pela União Européia com o objetivo de desenvolver o conhecimento científico e ajudar na criação ou avanço de novas tecnologias, como interfaces inteligentes e sistemas cognitivos adaptativos. O grupo hoje é formado por pesquisadores e estudantes distribuídos por todo o mundo.

Para alcançar seu objetivo, o PASCAL apóia a colaboração entre os especialistas em aprendizagem de máquinas, estatística e otimização, além de promover o uso de aprendizagem de máquinas em vários domínios de aplicação como: visão computacional, reconhecimento de fala, interface háptica, interfaces cérebro-computador, processamento de linguagem natural, entre outros.

O PASCAL também promove eventos e desafios tecnológicos em todas as áreas de interesse do grupo. Um destes desafios é o *Pascal Visual Object Classes Challenge* (ou simplesmente *Pascal VOC Challenge*), realizado todos os anos e encerrado com um seminário suportado por eventos como *European Conference on Computer Vision (ECCV)* ou *International Conference on Computer Vision (ICCV)*.

Um conjunto de algumas milhares de imagens rotuladas, com vinte categorias de objetos como carro, ônibus, avião, mesa, cadeira, etc., é oferecido aos competidores, que participam em três categorias distintas: classificação, detecção e segmentação. Eles são avaliados e premiados conforme o desempenho de seus trabalhos. Mais detalhes da competição podem ser encontrados em (EVERINGHAM et al., 2009). Para os experimentos realizados neste trabalho, parte das imagens convencionais utilizadas foram extraídas do conjunto de imagens fornecido para esta competição.

1.3 Objetivos

Com base no exposto até aqui, pode-se definir o objetivo principal deste trabalho como sendo a avaliação de um método para a classificação de objetos em imagens onidirecionais, baseado no uso conjugado de um processo de retificação da imagem onidirecional e um classificador construído com Máquinas de Suporte Vetorial de

Múltiplos Núcleos.

Também serão empregadas técnicas já conceituadas como utilização de características densamente amostradas e descritores invariantes a rotação e escala.

Mesmo tendo um vasto domínio de possíveis aplicações para tal sistema, o foco é prover informações para um sistema de SLAM baseado em objetos. Partindo deste classificador, a implementação de um detector de objetos torna-se mais simples, permitindo fornecer a outros sistemas informações como localização e dimensões dos objetos contidos na cena observada.

1.4 Justificativas

O objetivo de tornar este trabalho uma ferramenta acessória para o desenvolvimento de sistemas de navegação autônomos e SLAM na robótica móvel influenciou na decisão sobre as tecnologias aqui aplicadas, sempre considerando as tecnologias encontradas na literatura.

Como fonte de informação visual, a opção pelo uso de um sistema de visão onidirecional catadióptrico se deu ao fato de notória quantidade de informações do ambiente em apenas uma imagem. Outras técnicas como sistemas panorâmicos de visão, também fornecem visualização completa do ambiente, no entanto demandam a utilização de um conjunto de câmeras convencionais ao invés de apenas uma como ocorre no sistema adotado.

A utilização de imagens convencionais dos objetos de interesse para o treinamento do classificador permite que imagens independentes coletadas até mesmo através de ferramentas de busca na internet sirvam de base para o sistema. Uma possível extensão desta característica seria o aprendizado sob demanda do sistema robótico, onde o mesmo, mediante a solicitação de um objeto, coletaria imagens independentes da internet e realizaria seu próprio treinamento.

O tipo de características buscadas nas imagens para a caracterização dos objetos de interesse também é um ponto que demanda a opção por uma tecnologia. A utilização de imagens onidirecionais, mesmo que retificadas, norteiam a escolha destas características. É desejável que estas sejam invariáveis a rotação, escala e luminosidade. Tais alterações são naturalmente encontradas nas imagens utilizadas. A combinação de diferentes tipos de características na descrição de um objeto também pode fortalecer o processo. Por este motivo, neste trabalho foram utilizadas três tipos distintos de características locais, combinadas no modelo de Múltiplos Núcleos da MSV adotada, para a caracterização dos objetos. Estas características serão detalhadas na Seção 5.2. Outros tipos de características como textura, cores, formas, etc., podem

ser avaliadas seguindo o mesmo desenvolvimento, como complemento deste trabalho.

As tecnologias mencionadas acima, bem como a pesquisa na literatura em busca de tais ferramentas, serão detalhadas nos próximos capítulos.

1.5 Organização do Texto

Os tópicos abordados por este trabalho estão organizados da seguinte maneira: nos Capítulos 2 e 3 são apresentadas as tecnologias utilizadas no desenvolvimento deste trabalho. Uma breve revisão bibliográfica é apresentada no Capítulo 4. O algoritmo utilizado no desenvolvimento deste trabalho é finalmente apresentado em mais detalhes na Capítulo 5. O Capítulo 6 traz os experimentos e resultados obtidos. No Capítulo 7 são apresentadas as conclusões sobre os experimentos bem como uma linha para o desenvolvimento de trabalhos futuros.

2 Máquinas de Suporte Vetorial

Proposta por Vapnik (1995), a Máquina de Suporte Vetorial (MSV) é uma máquina linear que visa construir um hiperplano como superfície de decisão de tal forma que a margem de separação entre exemplos positivos e negativos seja máxima. Mais precisamente, a máquina de suporte vetorial é uma implementação do método de **Minimização do Risco Estrutural**. A máquina de suporte vetorial pode fornecer um bom desempenho de generalização em problemas de classificação de padrões, apesar do fato de não incorporar conhecimento do domínio do problema (HAYKIN, 1998).

Para ilustrar tal tarefa, a Figura 2.1 mostra dois conjuntos de dados num espaço bidimensional, rotulados por bolinhas brancas e bolinhas pretas, e três possíveis hiperplanos de separação. O hiperplano H_1 realiza a separação dos dados porém sem maximizar a margem entre as amostras mais próximas de cada conjunto. O hiperplano H_3 , por sua vez, não realiza nem mesmo a separação. Já o hiperplano H_2 , mostra o exato objetivo da MSV, realizando a separação dos dados respeitando a maximização da margem para tal.

Fig. 2.1: Exemplos para hiperplanos de separação

O princípio da Minimização do Risco Estrutural, citado anteriormente, é um princípio indutivo para a seleção de modelos usados na realização de aprendizado baseado em um conjunto finito de dados de treinamento. Ele descreve um modelo geral para o balanceamento entre a complexidade do modelo (dimensão-VC) e a qualidade do resultado do treinamento (erro empírico).

Outro conceito importante da teoria de Vapnik-Chervonenkis é a dimensão-VC (dimensão de Vapnik-Chervonenkis). Ela representa uma medida da complexidade de um classificador estatístico, definida como a maior cardinalidade de um conjunto de

pontos que o classificador pode separar. Na Figura 2.2 observa-se em cada uma das oito configurações, três amostras divididas em duas classes de dados, rotuladas como "círculo" e "triângulo". Verifica-se que para qualquer forma arbitrária que estas três amostras possam assumir, é possível criar uma reta capaz de separar as classes. Porém, para os quatro pontos ilustrados na Figura 2.3, existem combinações que podem ter as classes separadas por uma reta (Figura 2.3a), mas também é possível defini-la tal que uma só reta seja incapaz de realizar esta separação (Figura 2.3b) (HAYKIN, 1998). Para uma divisão binária arbitrária desses quatro pontos, deve-se então recorrer a funções de complexidade superior à das retas. Essa observação se aplica a quaisquer quatro pontos no espaço bidimensional. Portanto, a dimensão-VC do conjunto de funções lineares no espaço bidimensional é 3.

Fig. 2.2: Separações de três dados no espaço bidimensional por meio de retas (SCHÖLKOPF; SMOLA, 2001)

Fig. 2.3: Separações de quatro dados no espaço bidimensional por meio de retas (SCHÖLKOPF; SMOLA, 2001)

O detalhamento da MSV é realizado nas próximas seções, iniciando-se pela construção do hiperplano de separação para dados linearmente separáveis. A seguir, a generalização deste método para dados linearmente não separáveis é descrita, bem como as abordagens para classificação de objetos em múltiplas classes.

2.1 Hiperplano ótimo para padrões linearmente separáveis

Considerando uma amostra de treinamento $\{(\mathbf{x}_i, d_i)\}_{i=1}^N$, onde $\mathbf{x}_i$ é o padrão de entrada para o i -ésimo exemplo e d_i é a resposta desejada correspondente, e assumindo que os padrões (classes) representados pelos subconjuntos $d_i = +1$ e $d_i = -1$ são "linearmente separáveis", a equação de uma superfície de decisão na forma de um hiperplano que realiza esta separação pode ser definida como

$$\mathbf{w}^T \mathbf{x} + b = 0 \quad (2.1)$$

onde $\mathbf{x}$ é um vetor de entrada, $\mathbf{w}$ é um vetor peso ajustável e b é um deslocamento (bias). Pode-se assim escrever

$$\begin{aligned} \mathbf{w}^T \mathbf{x}_i + b &\geq 0 & \text{para } d_i = +1, \\ \mathbf{w}^T \mathbf{x}_i + b &< 0 & \text{para } d_i = -1. \end{aligned} \quad (2.2)$$

Para um dado vetor de peso $\mathbf{w}$ e deslocamento b , a separação entre o hiperplano definido na Equação 2.1 e o ponto de dado mais próximo é denominada margem de separação, representada por ρ . O objetivo de uma Máquina de Suporte Vetorial é encontrar o hiperplano para o qual a margem de separação ρ é máxima. Sob esta condição, a superfície de decisão é referida como hiperplano ótimo. A Figura 2.4 ilustra a construção geométrica de um hiperplano ótimo para um espaço de entrada bidimensional.

Fig. 2.4: Hiperplano ótimo para padrões linearmente separáveis

Considerando que $\mathbf{w}_o$ e b_o representam valores ótimos do vetor peso e do deslocamento, respectivamente, o hiperplano ótimo de decisão pode ser definido por

$$\mathbf{w}_o^T \mathbf{x} + b_o = 0 \quad (2.3)$$

o que é a Equação 2.1 reescrita. A função discriminante

$$g(\mathbf{x}) = \mathbf{w}_o^T \mathbf{x} + b_o \quad (2.4)$$

fornece uma medida algébrica da distância de $\mathbf{x}$ até o hiperplano (DUDA; HART; STORK, 2000). Para visualizar isso pode-se expressar $\mathbf{x}$ como

$$\mathbf{x} = \mathbf{x}_p + r \frac{\mathbf{w}_o}{\|\mathbf{w}_o\|} \quad (2.5)$$

onde $\mathbf{x}_p$ é a projeção normal de $\mathbf{x}$ sobre o hiperplano ótimo, e r é a distância algébrica desejada; r é positivo se $\mathbf{x}$ estiver no lado positivo do hiperplano ótimo e negativo se $\mathbf{x}$ estiver no lado negativo. Como, por definição, $g(\mathbf{x}_p) = 0$, resulta que

$$r = \frac{g(\mathbf{x})}{\|\mathbf{w}_o\|}. \quad (2.6)$$

Em particular, a distância da origem (ex.: $\mathbf{x} = \mathbf{0}$) até o hiperplano ótimo é dada por $\frac{b_o}{\|\mathbf{w}_o\|}$. Se $b_o > 0$ a origem está no lado positivo do hiperplano ótimo; se $b_o < 0$, ela está no lado negativo. Se $b_o = 0$, o hiperplano ótimo passa pela origem.

A questão a resolver é encontrar os parâmetros $\mathbf{w}_o$ e b_o para o hiperplano ótimo, dado o conjunto de treinamento $\tau = \{(\mathbf{x}_i, d_i)\}$ satisfazendo as restrições:

$$\begin{aligned} \mathbf{w}_o^T \mathbf{x} + b_o &\geq 1 & \text{para } d_i = +1, \\ \mathbf{w}_o^T \mathbf{x} + b_o &\leq -1 & \text{para } d_i = -1. \end{aligned} \quad (2.7)$$

Os pontos de dados $(\mathbf{x}_i, d_i)$ para os quais a primeira ou a segunda linha da Equação 2.7 é satisfeita com o sinal de igualdade são chamados de vetores de suporte. Estes vetores desempenham um papel proeminente na operação desta classe de máquinas de aprendizagem. Em termos conceituais, os vetores de suporte são aqueles pontos de dados que se encontram mais próximos da superfície de decisão e são, portanto, os mais difíceis de classificar. Dessa forma, têm uma influência direta na localização ótima da superfície de decisão (HAYKIN, 1998).

Considerando um vetor de suporte $\mathbf{x}^{(s)}$ para o qual $d^{(s)} = +1$, tem-se, por definição

$$g(\mathbf{x}^{(s)}) = \mathbf{w}_o^T \mathbf{x}^{(s)} \mp b_o \mp 1 \quad \text{para } d^{(s)} = \mp 1. \quad (2.8)$$

Da Equação 2.6 a distância algébrica do vetor de suporte $\mathbf{x}^{(s)}$ até o hiperplano ótimo é

$$\begin{aligned} r &= \frac{g(\mathbf{x}^{(s)})}{\|\mathbf{w}_o\|} \\ &= \begin{cases} \frac{1}{\|\mathbf{w}_o\|} & \text{se } d^{(s)} = +1 \\ -\frac{1}{\|\mathbf{w}_o\|} & \text{se } d^{(s)} = -1 \end{cases} \end{aligned} \quad (2.9)$$

onde o sinal positivo indica que $\mathbf{x}^{(s)}$ se encontra do lado positivo do hiperplano ótimo e o sinal negativo indica que $\mathbf{x}^{(s)}$ está do lado negativo do hiperplano ótimo. Considere que ρ represente o valor ótimo da margem de separação entre duas classes que constituem o conjunto de treinamento τ . Então, da Equação 2.9 resulta que

$$\begin{aligned} \rho &= 2r \\ &= \frac{2}{\|\mathbf{w}_o\|}. \end{aligned} \quad (2.10)$$

A Equação 2.10 afirma que maximizar a margem de separação entre classes é equivalente a minimizar a norma euclidiana do vetor peso $\mathbf{w}$.

Agora, o problema de otimização restrito pode ser formulado como:

Dada a amostra de treinamento $\{(\mathbf{x}_i, d_i)\}_{i=1}^N$, encontre os valores ótimos do vetor de peso $\mathbf{w}$ e deslocamento b de modo que satisfaçam as restrições

$$d(\mathbf{w}^T \mathbf{x}_i + b) \geq 1 \quad \text{para } i = 1, 2, \dots, N \quad (2.11)$$

e o vetor de peso $\mathbf{w}$ minimize a função de custo:

$$\Phi(\mathbf{w}) = \frac{1}{2} \mathbf{w}^T \mathbf{w}.$$

O fator de escala $1/2$ é incluído aqui por conveniência de apresentação. Este problema de otimização restrito é chamado de *problema primal* e pode ser caracterizado como

segue:

- A função de custo $\Phi(\mathbf{w})$ é uma função convexa de $\mathbf{w}$.
- As restrições são lineares em relação a $\mathbf{w}$.

Consequentemente, este problema de otimização restrito pode ser resolvido usando o método dos *multiplicadores de Lagrange* (BERTSEKAS, 2007). Primeiro a *função lagrangiana* é construída:

$$J(\mathbf{w}, b, \alpha) = \frac{1}{2} \mathbf{w}^T \mathbf{w} - \sum_{i=1}^N \alpha_i [d_i (\mathbf{w}^T \mathbf{x}_i + b) - 1] \quad (2.12)$$

onde as variáveis auxiliares não negativas α_i são chamadas de multiplicadores de Lagrange. A solução para o problema de otimização restrito é determinada pelo ponto de sela da função lagrangiana, que deve ser minimizada em relação a $\mathbf{w}$ e a b ; ela também tem que ser maximizada em relação a α . Assim as seguintes condições de otimização são obtidas:

$$\begin{aligned} \text{Condição 1: } \frac{\partial J(\mathbf{w}, b, \alpha)}{\partial \mathbf{w}} &= \mathbf{0} \quad e \\ \text{Condição 2: } \frac{\partial J(\mathbf{w}, b, \alpha)}{\partial b} &= 0. \end{aligned}$$

A aplicação da condição de otimização 1 à função lagrangiana da Equação 2.12 produz

$$\mathbf{w} = \sum_{i=1}^N \alpha_i d_i \mathbf{x}_i. \quad (2.13)$$

A aplicação da condição de otimização 2 à função lagrangiana da Equação 2.12 produz

$$\sum_{i=1}^N \alpha_i d_i = 0. \quad (2.14)$$

O vetor solução $\mathbf{w}$ é definido em termos de uma expansão que envolve os N exemplos de treinamento.

É importante notar que no ponto de sela, para cada multiplicador de Lagrange α_i , o produto daquele multiplicador pela sua restrição correspondente desaparece, como mostrado por

$$\alpha_i[d_i(\mathbf{w}^T \mathbf{x}_i + b) - 1] = 0 \quad \text{para } i = 1, 2, \dots, N. \quad (2.15)$$

Dessa forma, apenas aqueles multiplicadores que satisfazem exatamente a Equação 2.15 podem assumir valores não-nulos. Esta propriedade resulta das condição de *Kuhn-Tucker* da teoria da otimização (BERTSEKAS, 2007).

O problema primal lida com uma função de custo convexa e com restrições lineares. A partir deste, o problema dual pode ser construído. Este segundo problema tem o mesmo valor ótimo do problema primal, mas com os multiplicadores de Lagrange fornecendo a solução ótima. Para postular o problema dual para o problema primal, primeiro a Equação 2.12 é expandida termo a termo, como segue:

$$J(\mathbf{w}, b, \alpha) = \frac{1}{2} \mathbf{w}^T \mathbf{w} - \sum_{i=1}^N \alpha_i d_i \mathbf{w}^T \mathbf{x}_i - b \sum_{i=1}^N \alpha_i d_i + \sum_{i=1}^N \alpha_i. \quad (2.16)$$

O terceiro termo do lado direito da Equação 2.16 é zero em virtude da condição de otimização da Equação 2.14. Além disso, da Equação 2.13 tem-se

$$\mathbf{w}^T \mathbf{w} = \sum_{i=1}^N \alpha_i d_i \mathbf{w}^T \mathbf{x}_i = b \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j d_i d_j \mathbf{x}_i^T \mathbf{x}_j.$$

Consequentemente, fazendo a função objetivo $J(\mathbf{w}, b, \alpha) = Q(\alpha)$, a Equação 2.16 pode ser reformulada e o problema dual pode ser definido:

Dada a amostra de treinamento $\{(\mathbf{x}_i, d_i)\}_{i=0}^N$, encontre os multiplicadores de Lagrange $\{\alpha_i\}_{i=1}^N$ que maximizam a função objetivo

$$Q(\alpha) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j d_i d_j \mathbf{x}_i^T \mathbf{x}_j \quad (2.17)$$

sujeita às restrições

1. $\sum_{i=0}^N \alpha_i d_i = 0$
2. $\alpha_i \geq 0 \quad \text{para } i = 1, 2, \dots, N.$

O problema dual é formulado inteiramente em termos dos dados de treinamento e a função $Q(\alpha)$ a ser maximizada depende apenas dos padrões de entrada na forma de um conjunto de produtos escalares, $\{\mathbf{x}_i^T \mathbf{x}_j\}_{(i,j)=1}^N$.

Havendo determinado os multiplicadores de Lagrange ótimos, representados por $\alpha_{o,i}$, o vetor peso ótimo $\mathbf{w}_o$ é calculado usando a Equação 2.13

$$\mathbf{w}_o = \sum_{i=1}^N \alpha_{o,i} d_i \mathbf{x}_i. \quad (2.18)$$

Para calcular o deslocamento ótimo b_o , o $\mathbf{w}_o$ obtido é utilizado na Equação 2.8, relativa ao vetor de suporte positivo, assim escrevendo

$$b_o = 1 - \mathbf{w}_o^T \mathbf{x}^{(s)} \quad \text{para } d^{(s)} = 1. \quad (2.19)$$

2.2 MSV com Margem Suave

A seção anterior tratou da construção do hiperplano de separação para dados linearmente separáveis, no entanto, em problemas reais, é difícil deparar com tais tipos de dados, seja pela natureza dos mesmos ou por medições ruidosas. Diz-se então que a MSV definida anteriormente possui Margem Rígida.

Diz-se que a margem de separação entre classes é Suave se uma amostra $(\mathbf{x}_i, d_i)$ do conjunto de treinamento violar a condição definida na Equação 2.11. Esta violação pode surgir de duas formas:

- A amostra $(\mathbf{x}_i, d_i)$ se encontra dentro da região de separação, mas do lado correto da superfície de decisão.
- A amostra $(\mathbf{x}_i, d_i)$ se encontra do lado errado da superfície de decisão.

Para permitir tal violação, um novo conjunto de variáveis escalares não negativas, $\{\xi_i\}_{i=1}^N$, é introduzido na definição do hiperplano de separação, como mostrado aqui:

$$d_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1 - \xi_i \quad \text{para } i = 1, 2, \dots, N. \quad (2.20)$$

As ξ_i são chamadas de variáveis de folga e medem o desvio de uma amostra do conjunto de treinamento da condição ideal de separabilidade de padrões. Para $0 \leq \xi_i \leq 1$, a amostra se encontra dentro da região de separação, mas no lado correto da superfície de decisão. Para $\xi_i > 1$, ela se encontra do lado errado da superfície de separação. Os vetores de suporte são aquelas amostras de treinamento particulares que satisfazem a Equação 2.20 precisamente, mesmo se $\xi_i > 0$.

O objetivo agora é encontrar um hiperplano de separação para o qual o erro de classificação, cuja média é calculada sobre o conjunto de treinamento, é minimizado. Isso pode ser feito minimizando o funcional

$$\Phi(\xi) = \sum_{i=1}^N I(\xi - 1)$$

em relação ao vetor peso $\mathbf{w}$, sujeito à restrição descrita na Equação 2.20 e a restrição sobre $\|\mathbf{w}\|^2$. A função $I(\xi)$ é uma função indicadora, definida por

$$I(\xi) = \begin{cases} 0 & \text{se } \xi \leq 0 \\ 1 & \text{se } \xi > 0 \end{cases}$$

Por ser um problema de otimização não-convexo, o funcional $\Phi(\xi)$ é aproximado para torná-lo matematicamente tratável. Além disso, o funcional a ser minimizado é formulado em relação ao vetor peso $\mathbf{w}$ como segue:

$$\Phi(\mathbf{w}, \xi) = \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{i=1}^N \xi_i \quad (2.21)$$

Como anteriormente, a minimização do primeiro termo da Equação 2.21 está relacionada com a minimização da dimensão-VC da MSV (VAPNIK, 1995). Assim como para o segundo termo $\sum_i \xi_i$, ele é um limite superior para o número de erros de teste. A formulação da função de custo $\Phi(\mathbf{w}, \xi)$ da Equação 2.21 está, portanto, em perfeito acordo com o princípio da minimização estrutural de risco.

A constante C é um parâmetro de regularização que controla o compromisso entre a complexidade da máquina e o número de pontos não separáveis. Ele pode ser selecionado de duas formas (HAYKIN, 1998):

- Determinado experimentalmente através do uso padrão de um conjunto de treinamento/teste.
- Determinado analiticamente estimando a dimensão-VC e então usando-se limites do desempenho de generalização da máquina baseados na dimensão-VC.

Formalizando o problema primal:

Dada a amostra de treinamento $\{(\mathbf{x}_i, d_i)\}_{i=1}^N$, encontre os valores ótimos do vetor peso $\mathbf{w}$ e do deslocamento b de modo que satisfaçam à restrição

$$d_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1 - \xi_i, \quad \text{para } i = 1, 2, \dots, N \quad \xi_i \geq 0 \quad \text{para todo } i$$

e de modo que o vetor peso $\mathbf{w}$ e as variáveis de folga ξ_i minimizem o funcional de custo

$$\Phi((w), \xi) = \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{i=1}^N \xi_i$$

onde C é um parâmetro positivo.

Novamente, usando o método dos multiplicadores de Lagrange, o problema dual pode ser formulado como:

Dada a amostra de treinamento $\{(\mathbf{x}_i, d_i)\}_{i=1}^N$, encontre os multiplicadores de Lagrange $\{\alpha_i\}_{i=1}^N$ que minimizam a função objetivo

$$Q(\alpha) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j d_i d_j \mathbf{x}_i^T \mathbf{x}_j$$

sujeita às restrições

- $\sum_{i=1}^N \alpha_i d_i = 0$
- $0 \leq \alpha_i \leq C$ para $i = 1, 2, \dots, N$

onde C é um parâmetro positivo.

Nem as variáveis de folga ξ_i nem os multiplicadores de Lagrange aparecem no problema dual. Este caso difere do anterior apenas pelo fato que a restrição $\alpha_i \geq 0$ é substituída pela restrição mais rigorosa $0 \leq \alpha \leq C$. Os vetores de suporte também são definidos exatamente do mesmo modo como anteriormente.

A solução ótima para o vetor peso $\mathbf{w}$ é dada por

$$\mathbf{w}_o = \sum_{i=1}^{N_s} \alpha_{o,i} d_i \mathbf{x}_i \quad (2.22)$$

onde N_s é o número de vetores de suporte. A determinação de valores ótimos do $\mathbf{w}$ também segue um procedimento similar ao descrito anteriormente. As condições de Kuhn-Tucker são agora definidas por

$$\alpha_i [d_i (\mathbf{w}^T \mathbf{x}_i + b) - 1 + \xi] = 0, \quad i = 1, 2, \dots, N \quad (2.23)$$

e

$$\mu_i \xi_i = 0, \quad i = 1, 2, \dots, N. \quad (2.24)$$

A Equação 2.23 é uma forma reescrita da Equação 2.15, exceto pela substituição do termo da unidade por $(1 - \xi_i)$. Como na Equação 2.24, os μ_i são multiplicadores de Lagrange que foram introduzidos para forçar a não negatividade das variáveis de folga ξ_i para todo i . No ponto de sela, a derivada da função lagrangiana para o problema primal em relação à variável de folga ξ_i é zero, produzindo

$$\alpha_i + \mu_i = C. \quad (2.25)$$

Combinando as Equações 2.24 e 2.25, vê-se que

$$\xi_i = 0 \quad \text{se } \alpha_i < C. \quad (2.26)$$

O deslocamento ótimo b_o pode ser tomado como o valor médio resultante da Equação 2.23 sobre todos os pontos de dados $(\mathbf{x}_i, d_i)$ do conjunto de treinamento para o qual $0 < \alpha_{o,i} < C$.

2.3 Hiperplano ótimo para padrões não-separáveis

Para um conjunto de dados de treinamento com padrões linearmente não separáveis, ilustrado na Figura 2.5, um hiperplano de separação não pode ser construído sem que erros de classificação ocorram. Para minimizar tais erros, uma superfície de decisão mais complexa precisa ser adotada. No entanto, para fazer uso de toda a matemática de otimização descrita até agora, uma outra abordagem é praticada para este caso. Os dados de entrada são mapeados para um novo espaço de alta dimensionalidade denominado espaço de características, onde podem ser separados pela MSV linear com alta probabilidade.

Basicamente, esta ideia depende de duas operações matemáticas:

1. O mapeamento não linear de um vetor de entrada para o espaço de características de alta dimensionalidade, que é oculto da entrada e da saída.
2. A construção de um hiperplano ótimo para separar as características descobertas no passo 1.

A operação 1 é realizada de acordo com o teorema de Cover sobre a separabilidade de padrões (HAYKIN, 1998). Considere um espaço de entrada constituído de padrões não-linearmente separáveis. O teorema afirma que este espaço pode ser transformado em um novo espaço de características onde os padrões são linearmente separáveis,

Fig. 2.5: Conjuntos de dados linearmente não separáveis

desde que a transformação seja não-linear e a dimensionalidade do novo espaço seja suficientemente alta.

A operação 2 explora a ideia de construir um hiperplano de separação ótimo de acordo com a teoria descrita anteriormente, porém agora definido como uma função linear de vetores retirados do espaço de características.

Considere que $\mathbf{x}$ represente um vetor retirado do espaço de entrada, que é assumido como tendo dimensão m_0 . Considere que $\{\varphi_j(\mathbf{x})\}_{j=1}^{m_1}$ represente um conjunto de transformações não lineares do espaço de entrada para o espaço de características: m_1 é a dimensão do espaço de características. Assume-se que $\varphi_j(\mathbf{x})$ é definido a priori para todo j . Dado este conjunto de transformações não-lineares, um hiperplano pode ser definido, atuando como a superfície de decisão, como segue:

$$\sum_{j=1}^{m_1} w_j \varphi_j(\mathbf{x}) + b = 0 \quad (2.27)$$

onde $\{w_j\}_{j=1}^{m_1}$ representa um conjunto de pesos lineares conectando o espaço de características com o espaço de saída, e b é o deslocamento. Simplificando o desenvolvimento

$$\sum_{j=0}^{m_1} w_j \varphi_j(\mathbf{x}) = 0 \quad (2.28)$$

onde foi assumido que $\varphi_0(\mathbf{x}) = 1$ para todo $\mathbf{x}$, de modo que w_0 represente o deslocamento b . A Equação 2.28 define a superfície de decisão calculada no espaço de características em termos dos pesos lineares da máquina. A quantidade $\varphi_j(\mathbf{x})$ representa a entrada fornecida ao peso w_j através do espaço de características. Defina

o vetor

$$\varphi(\mathbf{x}) = [\varphi_0(\mathbf{x}), \varphi_1(\mathbf{x}), \dots, \varphi_{m_1}(\mathbf{x})]^T \quad (2.29)$$

onde, por definição tem-se

$$\varphi_0(\mathbf{x}) = 1 \quad \text{para todo } x. \quad (2.30)$$

Na verdade, o vetor $\varphi(\mathbf{x})$ representa a "imagem" induzida no espaço de características pelo vetor de entrada $\mathbf{x}$. Assim, em termos desta imagem, a superfície de decisão na forma compacta pode ser definida:

$$\mathbf{w}^T \varphi(\mathbf{x}) = 0. \quad (2.31)$$

Adaptando a Equação 2.13 à presente situação envolvendo um espaço de características onde a separabilidade "linear" é procurada, pode-se escrever

$$\mathbf{w} = \sum_{i=1}^N \alpha_i d_i \varphi(\mathbf{x}_i) \quad (2.32)$$

onde o vetor de características $\varphi(\mathbf{x})$ corresponde ao padrão de entrada $\mathbf{x}_i$ no i -ésimo exemplo. Dessa forma, substituindo a Equação 2.32 em 2.31, a superfície de decisão calculada no espaço de características pode ser definida como:

$$\mathbf{w} = \sum_{i=1}^N \alpha_i d_i \varphi^T(\mathbf{x}_i) \varphi(\mathbf{x}) = 0. \quad (2.33)$$

O termo $\varphi^T(\mathbf{x}_i) \varphi(\mathbf{x})$ representa o produto escalar de dois vetores induzidos no espaço de características pelo vetor de entrada $\mathbf{x}$ e o padrão de entrada $\mathbf{x}_i$ relativo ao i -ésimo exemplo. Aqui, a função Núcleo (do inglês *Kernel*) é introduzida, representada por $K(\mathbf{x}, \mathbf{x}_i)$ e definida por

$$\begin{aligned} K(\mathbf{x}, \mathbf{x}_i) &= \varphi^T(\mathbf{x}) \varphi(\mathbf{x}_i) \\ &= \sum_{j=0}^{m_1} \varphi_j(\mathbf{x}) \varphi_j(\mathbf{x}_i) \quad \text{para } i = 1, 2, \dots, N. \end{aligned} \quad (2.34)$$

O Núcleo $K(\mathbf{x}, \mathbf{x}_i)$ pode ser usado para construir o hiperplano ótimo no espaço de características sem ter que considerar o próprio espaço de características de forma

explícita. Isto é visto facilmente usando-se a Equação

$$\sum_{i=1}^N \alpha_i d_i K(\mathbf{x}, \mathbf{x}_i) = 0. \quad (2.35)$$

A utilização da função Núcleo $K(\mathbf{x}, \mathbf{x}_i)$ na Equação 2.34 permite construir uma superfície de decisão que é não-linear no espaço de entrada, mas cuja imagem no espaço de características é linear. Com base nessa expansão, a forma dual para otimização restrita de uma máquina de suporte vetorial é formulada como segue:

Dada a amostra de treinamento $\{(\mathbf{x}_i, d_i)\}_{i=1}^N$, encontre os multiplicadores de Lagrange $\{\alpha_i\}_{i=1}^N$ que maximizam a função objetivo

$$Q(\alpha) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j d_i d_j K(\mathbf{x}_i, \mathbf{x}_j) \quad (2.36)$$

sujeitos às restrições:

1. $\sum_{i=1}^N \alpha_i d_i = 0$
2. $0 \leq \alpha_i \leq C$ para $i = 1, 2, \dots, N$

onde C é um parâmetro positivo especificado pelo usuário.

O problema dual formulado tem a mesma forma como no caso de padrões linearmente separáveis com margem suave, exceto pelo fato de que o produto escalar $\mathbf{x}_i^T \mathbf{x}_j$ foi substituído pelo Núcleo $K(\mathbf{x}_i, \mathbf{x}_j)$.

Tendo encontrado os valores ótimos dos multiplicadores de Lagrange, representados por $\alpha_{o,i}$, o valor ótimo do vetor linear de peso $\mathbf{w}_o$ é determinado. Reconhecendo que $\varphi(\mathbf{x}_i)$ desempenha o papel de entrada para o vetor de peso $\mathbf{w}$, $\mathbf{w}_o$ pode ser definido como

$$\mathbf{w}_o = \sum_{i=1}^N \alpha_{o,i} d_i \varphi(\mathbf{x}_i). \quad (2.37)$$

O deslocamento ótimo b_o é representado pela primeira componente de $\mathbf{w}_o$.

É comum empregar a função Núcleo sem conhecer o mapeamento entre os espaços, que é gerado implicitamente. A utilidade dos Núcleos está, portanto, na simplicidade de seu cálculo e em sua capacidade de representar espaços abstratos de alta dimensionalidade. Para garantir a convexidade do problema de otimização formulado e também que o Núcleo represente mapeamentos nos quais seja possível o cálculo

de produtos escalares, utiliza-se Núcleos que seguem as condições estabelecidas pelo teorema de Mercer (MERCER, 1909). De forma simplificada, um Núcleo que satisfaz as condições de Mercer é caracterizado por dar origem a matrizes positivas semi-definidas K , em que cada elemento $K_{i,j}$ é definido por $K_{i,j} = K(\mathbf{x}_i, \mathbf{x}_j)$, para todo $i, j = 1, \dots, n$ (HERBRICH, 2001).

Alguns dos Núcleos mais utilizados na prática são os Polinomiais, os Gaussianos ou Funções de Base Radial (do inglês *Radial Basis Function* - RBF) e os Sigmoidais. Cada um deles apresenta parâmetros que devem ser determinados pelo usuário. O Núcleo Sigmoidal, em particular, satisfaz as condições de Mercer apenas para alguns valores de β_0 e β_1 . Os Núcleos Polinomiais com $p = 1$ também são denominados lineares. A Tabela 2.1 mostra estas funções e os parâmetros determinados a priori pelo usuário.

Tab. 2.1: Funções Núcleo mais utilizadas

Tipo de Núcleo	Função $K(\mathbf{x}, \mathbf{x}_i)$	Parâmetros
Polinomial	$(\mathbf{x}^T, \mathbf{x}_i + 1)^p$	p
Gaussiano (RBF)	$\exp(-\frac{1}{2\sigma^2} \ \mathbf{x} - \mathbf{x}_i\ ^2)$	σ^2
Sigmoidal	$\tanh(\beta_0 \mathbf{x}^T \mathbf{x}_i + \beta_1)$	β_0, β_1

Neste trabalho dois tipos de núcleos são utilizados para o mapeamento dos vetores de entrada, de acordo com a implementação adotada para o classificador. Como será visto no Capítulo 5, a característica *Geometric Blur* é mapeada através do núcleo RBF, mostrado anteriormente. Já as características *Self-Similarity* e PHOW, são mapeadas pelo núcleo $RBF - \chi^2$, que segue a equação

$$K(\mathbf{x}, \mathbf{x}_i) = \exp(-\gamma \chi^2(\mathbf{x}, \mathbf{x}_i))$$

onde γ é a constante de ajuste e a função $\chi^2(\mathbf{x}, \mathbf{x}_i)$ segue a forma

$$\sum_{i=1}^N \frac{(\mathbf{x} - \mathbf{x}_i)^2}{\mathbf{x} + \mathbf{x}_i}.$$

2.4 Métodos para Múltiplas Classes

A proposta original das Máquinas de Suporte Vetorial considera apenas a classificação binária dos dados, podendo distinguir entre duas classes de dados apenas. No entanto, grande parte dos problemas reais exige o reconhecimento entre múltiplas classes. Dois métodos criados para a utilização das MSV em tais problemas são mostrados a seguir.

2.4.1 *Método Um-Contra-Um*

Conceitualmente simples, este método realiza a divisão de um problema com múltiplas classes em vários subproblemas binários onde, para todas as duplas de classes, tem-se uma MSV. O total de máquinas produzidas neste método é $\frac{n(n-1)}{2}$, onde n é o número de classes do problemas.

O dado a ser classificado é testado em todas as MSVs produzidas, ou seja, testado contra todos os pares de classes presentes no problema. A classe com maior número de vitórias entre todos os testes realizados é dada como a classe a que o dado pertence.

Uma desvantagem deste método é que, para problemas com grande número de classes, a quantidade de MSVs necessárias, por ser da ordem de n^2 , pode tornar o problema impraticável, bem como o tempo necessário para classificação também poder ser muito alto, pois depende da execução de todos os classificadores.

2.4.2 *Método Um-Contra-Todos*

Neste método, uma MSV é treinada para cada classe do problema. Cada MSV realizará a classificação entre dois grupos de dados. O primeiro grupo recebe os dados da classe em questão. Já o segundo grupo, recebe os dados de todas as demais classes do problema. Dessa maneira o número de máquinas necessárias para a classificação é n , número muito inferior ao número necessário de máquinas para o método Um-Contra-Um. Com isso, o tempo de execução é também reduzido.

3 *Scale Invariant Feature Transform*

Publicado por David Lowe em 1999 (LOWE, 1999), o algoritmo *Scale Invariant Feature Transform* (SIFT) abriu novos horizontes para a visão computacional podendo ser utilizado em diversas aplicações neste campo. Especificamente, para a tarefa de reconhecimento de objetos, o SIFT tornou possível o reconhecimento de maneira confiável.

O SIFT define um método para detecção de pontos de interesse em imagens, além de um descritor para tais pontos. Essa combinação proporciona invariabilidade a escala, rotação e a pequenas variações de luminosidade. A confiabilidade proporcionada se faz presente até mesmo sob condições de oclusão parcial de objetos, pois apenas 3 pontos SIFT detectados no objeto parcialmente ocluído são suficientes para estimar sua localização.

Como mencionado anteriormente, o algoritmo SIFT define duas tarefas distintas: a detecção do ponto de interesse e a construção do descritor. É possível fazer uso apenas destas tarefas de maneira separada. Alguns trabalhos encontrados na literatura realizam a detecção dos pontos de interesse na imagem de maneira diferente da proposta por Lowe, porém fazem uso dos descritores definidos no SIFT para os pontos detectados. Isso acontece pois a invariabilidade a rotação e escala são características atribuídas ao descritor. Além disso, outros tipos de pontos de interesse podem ser desejados na imagem. Este trabalho faz uso de descritores SIFT em uma das características utilizadas na classificação dos objetos.

As tarefas de detecção dos pontos de interesse e construção do descritor, são detalhadas a seguir.

3.1 Detecção dos Pontos de Interesse

Nesta etapa, o algoritmo SIFT realiza a busca por pontos de interesse na imagem, que na verdade, são pontos de destaque ou característicos à cena ou aos objetos contidos na imagem. Tais pontos precisam ser representativos, pois serão a base para a detecção ou casamento entre as imagens, mesmo sob variações na cena, como luminosidade, rotação, escala ou posicionamento. Duas etapas distintas podem ser definidas aqui:

1. Detecção dos extremos no espaço de escala.
2. Seleção dos pontos de interesse.

Tais etapas são detalhadas nas seções seguintes.

3.1.1 Detecção dos extremos no espaço de escala

A escolha dos pontos de interesse na imagem é feita através da detecção dos extremos máximos e mínimos, da resultante da subtração de uma imagem suavizada por outra menos suavizada. A suavização da imagem é realizada através da filtragem da imagem original por padrões Gaussianos com diferentes desvios padrão. Esta função, denominada Diferença-de-Gaussianas (do inglês *Difference-of-Gaussians* - DoG), é aplicada repetidamente na imagem, em múltiplas escalas, evidenciando pontos de interesse que ocorrem em todo o espaço de escala. A imagem suavizada é dada pela função:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \quad (3.1)$$

onde $G(x, y, \sigma)$ é a função Gaussiana e $I(x, y)$ a imagem original. O resultado da função DoG é dado por:

$$\begin{aligned} D(x, y, \sigma) &= (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma). \end{aligned} \quad (3.2)$$

A função Gaussiana aplicada à filtragem é dada por:

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}. \quad (3.3)$$

A escolha destas funções se deu pelo fato de serem computacionalmente eficientes, pois a imagem filtrada L precisa ser obtida para todo o espaço de escala, e D é obtido através de uma simples subtração de imagens.

A Figura 3.1 mostra a construção de $D(x, y, \sigma)$. A imagem de entrada é filtrada pela convolução com Gaussianas produzindo imagens separadas por um fator constante k no espaço de escala, mostradas na pilha à esquerda. Imagens adjacentes na escala são subtraídas para obter as imagens DoG à direita. A Figura 3.2 mostra uma sequência de imagens suavizadas progressivamente como realizado pelo algoritmo.

A localização dos extremos máximos e mínimos em $D(x, y, \sigma)$ é realizada comparando cada ponto de teste com seus 8 vizinhos na escala e seus 9 vizinhos nas escalas abaixo e acima, como mostrado na Figura 3.3. O ponto é selecionado, ou seja, considerado um marco visual, somente se seu valor for realmente maior ou menor que todos seus vizinhos. O custo de tal comparação é relativamente baixo pois a grande

Fig. 3.1: Algoritmo DoG em execução: à esquerda, imagens suavizadas pelo filtro Gaussiano. Imagens Gaussianas adjacentes são subtraídas para produzir a Diferença de Gaussianas à direita. O processo se repete para outras escalas da imagem de entrada reamostrada (LOWE, 2004)

Fig. 3.2: Imagem suavizada progressivamente no mesmo espaço de escala

maioria dos pontos é eliminada logo nas primeiras comparações.

3.1.2 Seleção dos pontos de interesse

Esta etapa realiza uma análise individual dos pontos encontrados pelo processo anterior. Tais pontos são candidatos a pontos chave na imagem e precisam ser analisados quanto a sua estabilidade. A primeira análise é baseada no contraste de tais pontos em relação a seus vizinhos.

Para garantir a estabilidade dos marcos, não basta apenas a verificação do contraste em relação aos vizinhos. A função DoG possui uma resposta forte ao longo de bordas, porém as bordas não são imunes a ruídos e variações na cena. Portanto os pontos detectados ao longo das bordas são eliminados num processo de filtragem baseado em Harris e Stephens (1988).

Fig. 3.3: Os pontos extremos nas imagens produzidas pela função DoG são detectados comparando um pixel candidato (marcado com X) com seus 26 vizinhos (marcados com círculo) (LOWE, 2004)

3.2 Construção do Descritor

É através do descritor que os pontos são comparados para medir sua similaridade. Para que esta comparação seja simples de ser efetuada, duas etapas são realizadas na construção do descritor:

1. Determinação da orientação.
2. Montagem do descritor.

Estas etapas são detalhadas a seguir.

3.2.1 Determinação da orientação

O descritor para os pontos detectados nas etapas anteriores é representado através de sua orientação na imagem. Por isso, determinar esta orientação de maneira consistente proporciona a ele sua invariância à rotação.

O processo é realizado a partir da imagem filtrada $L(x,y)$ da escala onde o ponto foi localizado. Para cada ponto nesta imagem são calculados módulo $m(x,y)$ e orientação $\theta(x,y)$ do gradiente através da diferença entre os pixels:

$$m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2} \quad (3.4)$$

$$\theta(x,y) = \tan^{-1}((L(x,y+1) - L(x,y-1))/(L(x+1,y) - L(x-1,y))). \quad (3.5)$$

Um histograma da orientação é formado pelas orientações de pontos dentro de uma região em torno do ponto de interesse. Este histograma possui 36 valores que cobrem os 360 graus ao redor do ponto. A orientação de cada um destes pontos do histograma contribui ponderadamente pelo seu módulo para a orientação principal.

3.2.2 Montagem do descritor

As etapas anteriores determinaram a localização, escala e orientação de cada ponto de interesse na imagem. Esta etapa realiza a montagem do descritor propriamente dito. A abordagem de Lowe é inspirada em Edelman, Intrator e Poggio (1997), que propuseram uma representação baseada na visão biológica, em particular, em alguns neurônios complexos do cortex visual. Eles formularam uma hipótese onde estes neurônios complexos são responsáveis pela combinação e reconhecimento de objetos 3D a partir de um conjunto de marcos visuais.

Com a orientação principal determinada na etapa anterior, uma grade com 16 regiões é posicionada com seu centro sobre o ponto de interesse e respeitando a orientação principal. O passo seguinte é a realização do cálculo do módulo e da orientação para cada ponto dentro da janela que envolve o ponto de interesse. Esses valores são ponderados por uma janela Gaussiana também centrada no ponto de interesse. Os valores em cada uma das 16 regiões são então acumulados em histogramas e somados em oito direções.

A Figura 3.4 mostra a janela para construção do descritor com as 16 regiões utilizadas no processo. Tendo os histogramas das 16 regiões, contendo a orientação de cada região de acordo com as oito direções definidas, o descritor é construído concatenando os 128 valores resultantes. Finalmente o vetor construído é normalizado para que alterações leves na iluminação não afetem o processo.

A Figura 3.5 mostra as oito direções consideradas em cada região durante a construção do histograma local.

3.3 Reconhecimento

Uma das principais características que torna o SIFT muito difundido é a facilidade no processo de casamento ou *matching* dos pontos detectados. A verificação de similaridade entre os descritores é realizada através do simples cálculo da **distância Euclidiana**. Lowe em seu trabalho propõe que, para um casamento válido, a localização dos 2 vetores mais próximos ao vetor de teste deve ser realizada. Tendo isso, o segundo vetor deve estar a uma distância superior ao dobro da distância do

Fig. 3.4: Exemplo de construção do descritor SIFT (VEDALDI; FULKERSON, 2005)

Fig. 3.5: Construção do descritor SIFT (VEDALDI; FULKERSON, 2005)

primeiro vetor. Isso reduz a possibilidade de um falso positivo no processo.

A Figura 3.6 mostra o algoritmo SIFT em operação realizando o reconhecimento de um objeto de interesse, exibido na parte superior da imagem, em uma cena complexa. O processo de detecção dos pontos de interesse e construção dos descritores foi realizado para as duas imagens. No exemplo, uma busca dentre os descritores obtidos na imagem de teste, para localizar descritores obtidos previamente do objeto de interesse, foi realizada com sucesso para quase a totalidade dos pontos. O casamento entre os pontos da imagem de teste e do objeto de interesse é exibido através das linhas em violeta.

Fig. 3.6: Exemplo de reconhecimento dos pontos obtidos pelo processo SIFT para detecção de um objeto (HESS, 2010)

4 Panorama atual

Este capítulo mostra uma breve pesquisa na literatura nos principais pontos abordados pelo presente trabalho. Uma descrição das propostas e da evolução da utilização de marcos visuais no reconhecimento de objetos é mostrada na Seção 4.1. A Seção 4.2 discute sobre os modelos probabilísticos encontrados na literatura para o problema abordado, bem como sobre a utilização de Máquinas de Suporte Vetorial neste contexto. Na Seção 4.3, uma análise é realizada sobre alguns trabalhos que fazem uso de sistemas de visão onidirecional em conjunto com Marcos Visuais, aplicados à robótica móvel.

4.1 Marcos Visuais

Os primeiros trabalhos relacionados ao reconhecimento de objetos baseiam-se em processos de correlação de um exemplo – imagem do objeto alvo – sobre a imagem de teste. Estes métodos, muitas vezes eficazes para ambientes totalmente controlados, não são tolerantes a alterações na iluminação, distância de observação, rotação ou outras possíveis transformações no ambiente, além de serem computacionalmente inaplicáveis para grandes quantidades de objetos.

Processos alternativos começaram a surgir, fazendo uso não mais de exemplos completos de objetos alvo mas sim de informações extraídas destes objetos, como bordas, cantos, regiões, etc. A busca nas imagens de teste se tornou mais objetiva uma vez que a correlação de exemplos deu lugar a localização de características, ou pontos de interesse, na imagem. Vale salientar que os termos, pontos de interesse, características ou marcos visuais, fazem referência a regiões da imagem onde uma informação genérica pode ser localizada e utilizada em conjuntos para descrever objetos ou cenas de interesse. Portanto neste trabalho estes termos serão utilizados sem distinção.

Uma deficiência não contornada por esta abordagem é a intolerância às possíveis transformações nos objetos, em escala e rotação, além das demais citadas anteriormente. Iniciou-se então uma busca por características imunes, ou pelo menos parcialmente imunes, a estas transformações.

Cita-se o trabalho de Moravec (1981) como um dos pioneiros no uso de marcos visuais para identificação de objetos em imagens, seguido por Harris e Stephens (1988), que aperfeiçoaram a implementação de Moravec. Ambos fazem uso de regiões da imagem, com grandes gradientes em todas as direções. Por este motivo, estes detectores ficaram conhecidos como detectores de cantos.

O uso do detector de cantos de Harris se estendeu, mas sempre baseado na correlação dos pontos de interesse obtidos nas imagens exemplo com os localizados nas imagens de teste. Fazendo uso do detector de cantos de Harris, Schmid e Mohr (1997) propuseram em seu trabalho um descritor para os pontos de interesse, invariante à rotação, permitindo que estes pontos pudessem ser reconhecidos em diferentes orientações, dispensando o processo de correlação. Além disso, eles demonstraram que a abordagem proposta era tolerante à oclusão parcial e a imagens confusas, pois a localização de grupos de pontos de interesse reconhecidos era suficiente para afirmar a presença e a localização do objeto na imagem.

A alta sensibilidade do detector de cantos de Harris à alterações na escala fez com que a busca por pontos de interesse mais invariantes continuasse. Uma das abordagens mais utilizadas recentemente foi proposta por Lowe (1999). Denominada *Scale-Invariant Feature Transform* (SIFT), ela faz uso da função de Diferença de Gaussianas (DoG) em diferentes escalas da imagem de entrada para obter pontos de interesse fortemente tolerantes a alterações na escala, bem como o uso de um descritor para estes pontos que proporciona tolerância à rotação. O modelo de descritores SIFT é utilizado neste trabalho e é detalhado no Capítulo 3.

Outras propostas surgiram baseadas no mesmo modelo descrito por Lowe, cada uma buscando melhorias em etapas do processo, que muitas vezes sofre com a depreciação de outras. Uma destas propostas é o PCA-SIFT (KE; SUKTHANKAR, 2004), que aplica a análise de componentes principais (PCA) no entorno do ponto de interesse, reduzindo a dimensão do vetor descritor. Esta redução traz melhorias na velocidade de comparação entre os descritores, no entanto o desempenho do cálculo dos descritores sofre perdas. Outra proposta, denominada GLOH (*Gradient Location and Orientation Histogram*) (MIKOLAJCZYK; SCHMID, 2005), produz uma característica ainda mais única que a produzida pelo algoritmo SIFT, no entanto, também faz uso do PCA para a redução do vetor descritor, e portanto, o desempenho no processo de detecção dos pontos e criação dos descritores é reduzido. Preocupado com o problema da velocidade na detecção dos pontos de interesse, o método SURF (*Speeded-Up Robust Features*) apresentado por Bay, Tuytelaars e Gool (2006) baseia-se na soma 2D de *Wavelets* Haar e, em imagens integrais (VIOLA; JONES, 2001) na busca dos pontos de interesse na imagem. O algoritmo SURF é classificado pelos autores como várias vezes mais rápido e suas características mais robustas às transformações quando comparado ao SIFT.

4.2 Máquinas de Suporte Vetorial na Classificação de Objetos

Como dito no Capítulo 1, as propostas para a tarefa de classificação de objetos podem ser divididas segundo o modelo adotado para o classificador, podendo ser ele gerativo ou discriminativo. Os gerativos são baseados nas distribuições de probabilidades conjuntas de imagens e classes. Geralmente fazem uso do algoritmo de Maximização de Expectativa (EM) para maximizar a verossimilhança. Já os discriminativos utilizam um modelo paramétrico, estimado na etapa de treinamento com base em imagens rotuladas. Nesta categoria encontram-se as Máquinas de Suporte Vetorial, as Redes Neurais Artificiais, *Boosting*, entre outros.

Ambos os modelos rendem boas propostas para o problema de classificação de objetos, cada uma destacando sempre as vantagens de um modelo perante o outro. Um consenso observado nos trabalhos, independentemente do modelo adotado, é a utilização de marcos visuais, como os obtidos pelo algoritmo SIFT.

Um comparativo entre os modelos gerativo e discriminativo para a tarefa de reconhecimento de objetos pode ser observado no trabalho de Ulusoy e Bishop (2005). Ambos os modelos foram aplicados no reconhecimento de algumas classes de objetos não rígidos, neste caso, de animais. O autor se mantém imparcial quanto a preferência entre os modelos; no entanto, conclui que a integração dos modelos pode tentar aproveitar as vantagens de cada um. Métodos discriminativos são mais rápidos no reconhecimento, no entanto necessitam de uma quantidade de amostras para treinamento muito maior que para modelos gerativos. Este por sua vez é muito mais lento no reconhecimento comparado aos modelos discriminativos, mas a eficiência no reconhecimento pode ser maior, bem como ser capaz de lidar com dados faltantes e amostras fracamente rotuladas no conjunto de treinamento.

Um modelo gerativo bastante citado na literatura foi proposto em Fergus, Perona e Zisserman (2003). Nele, os objetos são modelados como um conjunto de partes ou pontos de interesse, descritos em parâmetros como aparência, forma e escala relativa. No processo de aprendizagem, tais parâmetros são estimados com base nas imagens rotuladas de treinamento. O algoritmo de Maximização de Expectativa (EM) é utilizado para maximizar a verossimilhança. Como as características utilizadas não são baseadas no SIFT de Lowe, a invariância à rotação não é alcançada e é até mesmo citada como uma melhoria para o modelo. O processo de aprendizagem também é bastante prejudicado com o aumento da quantidade de características extraídas das imagens, por isso essa quantidade é reduzida, prejudicando o reconhecimento.

Seguindo o modelo discriminativo, as MSVs têm despertado grande interesse. Observa-se sua popularidade, por exemplo, entre as propostas competidoras em

eventos como o Pascal VOC (EVERINGHAM et al., 2009). Grande parte dos trabalhos fazem uso de tal ferramenta, como por exemplo os vencedores na categoria detecção (VEDALDI et al., 2009a; FELZENSZWALB et al., 2007). A diferenciação entre as propostas se dá nas características consideradas na imagem e também em variações na implementação da MSV utilizada, como as baseadas em Múltiplos Núcleos ou MSV Latentes, respectivamente.

Soluções que integram os modelos gerativos e discriminativos também são encontradas na literatura, como em Fritz et al. (2005). Um vocabulário de características comum aos dois métodos é construído a partir dos objetos de interesse. A etapa gerativa busca um conjunto de hipóteses para os objetos na imagem, além de estimar a localização e escala para cada um destes. As hipóteses escolhidas são então testadas pela etapa discriminativa. A etapa gerativa, baseada no trabalho de Leibe, Leonardis e Schiele (2004a), faz uso de Modelos de Forma Implícita (do inglês *Implicit Shape Models*) (LEIBE; LEONARDIS; SCHIELE, 2004b), que busca as hipóteses entre os pontos de interesse encontrados na imagem, através de um processo probabilístico de votação de Hough. O resultado é utilizado no processo discriminativo baseado em Máquinas de Suporte Vetorial com Núcleos Locais (WALLRAVEN; CAPUTO; GRAF, 2003) onde a categorização do objeto é realmente realizada.

4.3 Sistemas de Visão Onidirecional

Um detalhe observado durante a pesquisa sobre trabalhos baseados em sistemas de visão onidirecionais foi que a aplicação de algoritmos para reconhecimento de objetos em imagens obtidas nestes sistemas é muito pouco abordada. Por outro lado, as câmeras onidirecionais são amplamente utilizadas na robótica móvel devido à capacidade de visualizar o ambiente quase por completo, através de uma única imagem. Boa parte dos trabalhos relacionam visão onidirecional e detecção de marcos visuais apenas para a tarefa de localização de robôs, como mostrado a seguir.

No trabalho de Tamimi et al. (2006), uma alteração é proposta no estágio do algoritmo SIFT que realiza a busca dos pontos candidatos a marcos visuais na imagem. Essa alteração traz uma redução significativa na quantidade de pontos detectados pelo algoritmo original. A questão levantada aqui refere-se à quantidade de pontos de interesse produzida pelo algoritmo SIFT original, que mostra ser muito superior à quantidade necessária para que um robô identifique sua localização. Como o próprio Lowe afirma em seu trabalho (LOWE, 1999), com apenas três pontos detectados em um objeto, pode-se inferir sua localização em uma nova cena. A alteração proposta no algoritmo SIFT realiza uma análise prévia na imagem e define que a ocorrência de extremos locais é menos provável em regiões planas, ou seja, sem grandes variações

nas cores dos pixels em uma vizinhança extensa. Portanto se a busca por um máximo local estiver ocorrendo em uma região deste tipo, a busca é descartada e se inicia em uma nova região longe desta. O objetivo é tornar o algoritmo mais rápido visando seu uso na localização de robôs. Nenhum tratamento especial é dado ao processo devido à sua aplicação em imagens onidirecionais.

Também focado na localização de robôs, o trabalho de Goedemé et al. (2007) faz uso do SIFT e visão onidirecional. Um mapa topológico do ambiente é construído com base nos pontos detectados. Tais pontos são utilizados para comparação durante o processo de localização, onde o robô – no caso, uma cadeira de rodas autônoma – é questionado sobre sua localização, ou precisa se dirigir a um alvo.

Visando a localização de objetos, o trabalho de Chen et al. (2008) faz uso de uma câmera onidirecional e uma camera PTZ (Pan/Tilt/Zoom). O foco do trabalho é a proposta de uma metodologia para fusão das informações obtidas de ambos sensores visuais. No entanto a localização dos objetos é realizada através de um simples processo de subtração de imagens, ou seja, apenas objetos em movimento são considerados.

Outro uso da visão onidirecional em conjunto com SIFT pode ser visto no trabalho de Zivkovic, Booij e Kröse (2007). Os pontos de interesse detectados nas imagens são utilizadas para agrupar imagens similares e definir ambientes de um mapa. A proposta está associada à tarefa de localização de robôs realizada em ambientes internos como casas e escritórios.

Uma importante referência para este desenvolvimento foi o trabalho de Puig, Guerrero e Sturm (2008), onde é avaliada uma proposta de matriz híbrida para o casamento de descritores SIFT extraídos de imagens convencionais e onidirecionais. Uma análise comparativa entre os resultados do casamento de descritores SIFT entre as imagens convencionais e onidirecionais retificadas, mesmo sem a utilização da matriz híbrida proposta, mostraram que a etapa de retificação já traz benefícios ao processo.

4.4 Considerações finais

Das soluções apresentadas neste capítulo é possível explorar as que possuem características pertinentes a este trabalho, permitindo justificar a utilização das tecnologias aqui aplicadas. Visando a tarefa de localização de robôs, como em Zivkovic, Booij e Kröse (2007), Goedemé et al. (2007), Tamimi et al. (2006), vemos que os sistemas de visão onidirecional despertam grande interesse. Para a tarefa de classificação de objetos em imagens observa-se em eventos como Everingham et al. (2009) a grande popularidade das Máquinas de Suporte Vetorial. Esta técnica de

aprendizagem de máquinas por sua vez baseia-se em informações ou características extraídas das imagens para possibilitar a categorização dos objetos. Tais características necessitam de robustez e invariabilidade a fatores como escala, deslocamento e rotação. O algoritmo SIFT de Lowe, ou mesmo algoritmos baseados neste, tem sido amplamente utilizados para a extração de tais características mesmo em imagens onidirecionais, como mostrado em Zivkovic, Booij e Kröse (2007). No entanto, para tentar reduzir as distorções presentes nas imagens onidirecionais, causadas pelo uso de um espelho convexo na construção deste modelo de câmera, uma etapa de retificação da imagem é realizada, como visto no trabalho de Puig, Guerrero e Sturm (2008). Tais soluções serão consideradas no modelo aqui avaliado para a classificação de objetos em imagens onidirecionais.

5 Detalhamento

O presente trabalho faz uso da implementação do classificador MKL (VEDALDI et al., 2009b), utilizado no detector de objetos proposto por Vedaldi et al. (2009a). As seções a seguir descrevem os detalhes desta implementação bem como detalhes gerais do processo de classificação aqui avaliado.

Três tarefas comuns a todos os classificadores podem ser definidas:

- Detecção das características da imagem
- Aprendizagem
- Execução/Classificação

Para o método avaliado aqui, uma etapa que realiza a retificação da imagem onidirecional foi inserida antes das etapas descritas acima. O objetivo é aproximar a imagem onidirecional de uma imagem em perspectiva. Uma visão resumida da etapa de classificação do método é mostrada na Figura 5.1. Da imagem onidirecional retificada, janelas contendo os objetos de interesse são extraídas e submetidas aos três tipos de detectores de características. Os vetores de características obtidos são então testados na MSV, que retorna o objeto representado na imagem com maior probabilidade. Para a etapa de treinamento da MSV a grande contribuição deste trabalho é proporcionar a utilização de imagens convencionais dos objetos de interesse, imagens estas que podem ser obtidas até mesmo através de ferramentas de busca na internet. Imagens onidirecionais e retificadas nesta etapa também são consideradas nas avaliações realizadas.

A tarefa de detecção das características da imagem sempre antecede as duas outras. Para a aprendizagem, as imagens rotuladas de treinamento passam por este processo, que fornece ao classificador as informações pertinentes ao modelo a ser classificado. Na execução, o processo de detecção é executado sobre as imagens a serem classificadas sem conhecimento prévio. As informações detectadas nas imagens pela implementação utilizada aqui são detalhadas na Seção 5.2.

A utilização de mais de um tipo de característica na representação dos objetos pode, em alguns casos, trazer benefícios à classificação. A combinação destas características de maneira ponderada para a representação de cada objeto pode melhorar ainda mais a qualidade do classificador. Para isso, a implementação aqui aplicada, faz uso de um modelo de Aprendizagem de Múltiplos Núcleos na construção de uma Máquina de Suporte Vetorial. Este modelo será detalhado na Seção 5.3.

Fig. 5.1: Classificação através do método avaliado

Outro detalhe importante a ser considerado é que, por se tratar de um problema de múltiplas classes de objetos, a abordagem Um-Contra-Todos é adotada, onde uma MSV é treinada para cada classe. Cada MSV treinada realiza a separação das características da classe para a qual foi treinada, contra as características obtidas das demais classes. Na etapa de classificação, as características detectadas em uma imagem de teste são testadas em todas as MSVs treinadas. A classe com maior pontuação, ou seja, maior número de classificações, é atribuída à imagem avaliada.

Para detalhar o algoritmo é inicialmente explicada a etapa de retificação da imagem onidirecional, seguida do processo de detecção das características das imagens, bem como pelo detalhamento de cada um dos tipos de características utilizados. Na sequência, a aprendizagem de múltiplos núcleos é explicada e, ao final, o processo de classificação efetivamente é detalhado.

5.1 Retificação da Imagem Onidirecional

A retificação da imagem onidirecional segue o método mostrado na Figura 5.2.

O ponto central da imagem é determinado. Para esta implementação o ponto central é determinado manualmente, uma vez que em todas as imagens de um mesmo conjunto de dados o centro se mantém. Uma transformação de coordenadas polares para cartesianas é realizada. Como a largura da imagem retificada é maior que o perímetro central da imagem onidirecional, é necessário que uma interpolação seja feita para determinar o valor dos pixels faltantes. Esta etapa, proposta por Puig,

Fig. 5.2: Processo de retificação da imagem onidirecional (PRONOBIS, 2009)

Guerrero e Sturm (2008), mostrou melhorias nos resultados para o casamento de descritores SIFT entre imagens convencionais e onidirecionais, como mencionado anteriormente.

5.2 Detecção das Características da Imagem

Nesta etapa, as imagens são submetidas a três processos diferentes que detectam tipos distintos de características utilizadas no processo de classificação. Estas características são:

- *Geometric Blur* (GB)
- PHOWGray / PHOWColor
- *Self-Similarity* (SSIM)

e serão detalhadas a seguir.

A extração das características das imagens resulta em sete conjuntos de dados para cada imagem representando um objeto: um para o detector GB, quatro para o detector PHOWGray / PHOWColor e dois para o detector SSIM. Estes conjuntos de dados serão combinados linearmente através do modelo de aprendizagem de Múltiplos Núcleos para uma MSV. Cada conjunto de dados é associado a um Núcleo. O processo de aprendizagem destes múltiplos núcleos obtidos será descrito na próxima seção.

5.2.1 *Geometric Blur*

Este detector de características foi originalmente proposto por Berg e Malik (2001), com o objetivo de criar um descritor para pontos de interesse na imagem, invariável a distorções geométricas, ou seja, alterações no posicionamento relativo dos objetos na cena. Esta característica é muito utilizada para correspondência entre imagens estereoscópicas. O método para utilização deste detector é melhor explicado em (BERG; BERG; MALIK, 2005). A ideia é obter na imagem pontos de interesse, detectados através do algoritmo SIFT ou, como adotado pela implementação utilizada neste trabalho, através de um detector de bordas. Para cada ponto encontrado, constrói-se o descritor GB, como ilustrado na Figura 5.3. O ponto em vermelho no centro é o ponto de interesse a ser descrito, os demais pontos vizinhos também terão seus valores amostrados para a formação do descritor. Ao invés de simplesmente realizar a amostragem dos pontos, uma etapa anterior é realizada, onde a imagem é suavizada em função da distância do ponto de interesse. Este processo é repetido para a descrição de todos os pontos de interesse, sempre a partir da imagem original, ou seja, a distorção realizada para descrição de um ponto não se acumula para os demais.

Fig. 5.3: Exemplo de construção do *Geometric Blur*. À esquerda, a imagem original; à direita, a distorção realizada em função da distância para o ponto de interesse, marcado em vermelho (BERG; BERG; MALIK, 2005)

Este detector produz um vetor de características que é mapeado na MSV através de um núcleo RBF, como descrito no Capítulo 2.

5.2.2 *PhowGray / PhowColor*

O detector de características *Pyramid Histogram Of visual Words* (PHOW) teve seu conceito inicialmente proposto em (LAZEBNIK; SCHMID; PONCE, 2006) porém foi efetivamente definido em (BOSCH; ZISSERMAN; MUNOZ, 2007). Ele faz uso de descritores SIFT extraídos densamente, ou seja, em uma grade regular a cada N pixels (para este caso N é 5). A etapa de detecção dos marcos visuais não é executada neste processo, isso torna o processo mais rápido; além disso, os pontos obtidos densamente também possibilitam uma boa representação da imagem. Após esta etapa,

os descritores SIFT obtidos são agrupados em 300 grupos através do processo K-means (MACKAY, 2002). Após o agrupamento, cada descritor passa a ter o valor médio de seu grupo. A estes valores é dado o nome de "Palavras Visuais" (do inglês *Visual Words*).

Com o processo de quantização completo, é iniciada a construção dos descritores PHOW. A idéia é dividir a imagem em partes iguais, e para cada parte somar as ocorrências de cada Palavra Visual. Neste caso, a mesma imagem é dividida de duas maneiras, portanto dois descritores PHOW serão construídos. Inicialmente a imagem é dividida em uma matriz 2x2, definindo quatro regiões. O descritor PHOW é construído concatenando os valores acumulados para cada Palavra Visual, em cada uma das regiões definidas. A imagem original é então dividida em uma matriz 4x4, definindo 16 regiões. O processo anterior se repete para a construção do novo descritor PHOW. Este conceito é ilustrado na Figura 5.4. Nela as Palavras Visuais são representadas pelas figuras bolas pretas, cruzes e losangos.

Fig. 5.4: Construção do descritor PHOW. A imagem é dividida, e para cada região um histograma de ocorrências das Palavras Visuais é computado. O descritor PHOW é construído concatenando tais histogramas (LAZEBNIK; SCHMID; PONCE, 2006)

Para a implementação utilizada neste trabalho, a quantização dos descritores SIFT é realizada com 300 Palavras Visuais, seguindo o valor utilizado na implementação original deste detector. A Figura 5.5 ilustra a exata montagem do descritor utilizado aqui. Nela os pontos SIFT obtidos densamente na imagem não são exibidos por serem subentendidos como um reticulado constante a cada 5 pixels em linhas e colunas.

Como as imagens são coloridas, quatro descritores PHOW são obtidos para cada imagem, um apenas para as intensidades, ou níveis de cinzas, *PhowGray*, e três outros para os canais de cores HSV (Tonalidade (*Hue*), Saturação e Brilho (*Value*)), *PhowColor*.

Fig. 5.5: Construção do descritor PHOW da aplicação real (BOSCH; ZISSERMAN; MUNOZ, 2007)

Em uma imagem colorida, o nível de cinza é calculado como $I = 0.3R + 0.59G + 0.11B$, sendo RGB as intensidades individuais das cores na imagem, respectivamente vermelho, verde e azul.

Os quatro descritores PHOW produzidos pelo processo serão linearmente combinados, juntamente com os demais produzidos pelos outros processos de extração de características, como mencionado anteriormente. O mapeamento dos vetores de características deste detector na MSV, é realizado através do núcleo $BBF - \chi^2$, também descrito no Capítulo 2.

5.2.3 Self-Similarity

O detector de características SSIM (do inglês *Self-Similarity*), proposto por Shechtman e Irani (2007), visa definir um modelo para encontrar similaridade geométrica entre as imagens, independente de outras características como textura, luminosidade ou cor. Um exemplo deste descritor é mostrado na Figura 5.6 onde, a partir da extração dos descritores SSIM de uma imagem modelo 5.6a, o mesmo pode ser localizado em diversas outras cenas 5.6b.

A criação dos descritores SSIM é realizada a partir de uma amostragem densa da imagem, ou seja, em todos os pontos da imagem em passos de 5 pixels, como descrito no processo de detecção de características PHOW. Para cada um destes pontos, um descritor é obtido calculando-se a correlação de uma matriz de 5x5 pixels, em janelas de 40x40 pixels, ambas centradas no ponto de interesse. Este processo pode ser observado na Figura 5.7.

Os descritores produzidos para uma imagem são concatenados e quantizados

Fig. 5.6: Exemplo de localização de descritores SSIM; a) representação gráfica de um descritor; b) localização do descritor em imagens de teste (SHECHTMAN; IRANI, 2007)

Fig. 5.7: Criação do descritor de auto-similaridade (SHECHTMAN; IRANI, 2007)

através do processo K-means (MACKAY, 2002), onde 300 agrupamentos de dados são definidos. Cada descritor recebe então o valor médio de seu grupo. Novamente como descrito no processo de construção do descritor PHOW, a imagem é dividida em partes iguais de duas maneiras diferentes, 4 regiões e 16 regiões. O mesmo processo de acúmulo e concatenação das Palavras Visuais em cada região é realizado. O processo resulta em dois descritores SSIM. Da mesma maneira como realizado para o descritor PHOW, os vetores de características deste detector são mapeados na MSV através do uso do núcleo $RBF - \chi^2$.

5.3 Múltiplos Núcleos

Abordagens utilizando Aprendizagem de Múltiplos Núcleos (do inglês *Multiple Kernel Learning* (MKL)) são relativamente recentes (BACH; LANCKRIET; JORDAN, 2004; LANCKRIET et al., 2004; SONNENBURG et al., 2006), no entanto têm despertado grande interesse na sua utilização devido aos bons resultados divulgados (VEDALDI et al., 2009a) mas principalmente, pela sua proposta, que pode ser aplicada em inúmeras áreas de pesquisa e desenvolvimento, para tarefas de classificação de dados. O conceito de Múltiplos Núcleos pode ser aplicado não só para as Máquinas de Suporte Vetorial, mas também para outros algoritmos de aprendizagem de máquinas como, Análise dos Componentes Principais (PCA), agrupamento de dados, Redes

Neurais, etc.

A motivação principal da utilização do MKL no reconhecimento de objetos é a capacidade de combinar, de maneira ponderada, diversas características obtidas nas imagens em questão, para a categorização de objetos, uma vez que cada objeto possui uma característica ou um conjunto destas, que melhor o define. Este grau de importância de cada característica na descrição de um objeto pode ser "aprendido" de maneira individual a cada objeto estudado através do MKL.

Como visto no Capítulo 2, as Máquinas de Suporte Vetorial, quando aplicadas a dados linearmente não-separáveis, são formuladas apenas através da função Núcleo K , deixando implícito o mapeamento dos dados de entrada, do espaço linearmente não separável para o espaço linearmente separável de alta dimensionalidade. A abordagem de Múltiplos Núcleos realiza a combinação linear de um Núcleo para cada tipo de característica extraída da imagem. Isso permite que cada núcleo seja escolhido de maneira a melhor realizar a separação da característica processada.

O problema MKL pode ser formulado como proposto em Varma e Ray (2007). Considerando um conjunto de M tipos de características para serem combinadas, cada uma contendo N amostra rotuladas de treinamento $\{(\mathbf{x}_i, d_i)\}_{i=1}^N$, tem-se para cada característica um Núcleo $K(\mathbf{x}_i, \mathbf{x}_j)$. O objetivo é realizar a combinação linear destes Núcleos em uma matriz de Núcleos ótima $\mathbf{K}_o = \sum_{m=1}^M \eta_m K_m$, onde η_m é o peso atribuído ao Núcleo K_m . Adaptando o funcional de custo da MSV (Equação 2.21), a ser minimizado, obtém-se:

$$T(\eta) = \min_{\mathbf{w}, \xi} \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{i=1}^N \xi_i + \gamma^T \eta \quad (5.1)$$

sujeito às restrições

$$d_i(\mathbf{w}^T \varphi(\mathbf{x}_i) + b) \geq 1 - \xi_i, \quad \text{para } i = 1, 2, \dots, N, \xi_i \geq 0 \quad \text{para todo } i$$

onde C é um parâmetro positivo e γ é um vetor de inicialização para os pesos η .

Usando os multiplicadores de Lagrange, é possível formular o problema dual:

$$W(\eta) = \max_{\alpha} \sum_{i=1}^N \alpha_i + \gamma^T \eta - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j d_i d_j \sum_{m=1}^M \eta_m K_m(\mathbf{x}_i, \mathbf{x}_j) \quad (5.2)$$

sujeita às restrições

- $\sum_{i=1}^N \alpha_i d_i = 0$

- $0 \leq \alpha_i \leq C$ para $i = 1, 2, \dots, N$

onde C é um parâmetro positivo e γ é um vetor de inicialização para os pesos η .

Pelo princípio da dualidade $T(\eta) = W(\eta)$. Além disso, sendo α a solução única que maximiza a função objetivo $W(\eta)$, esta pode ser diferenciada em função de η como segue:

$$\frac{\partial T}{\partial \eta_m} = \frac{\partial W}{\partial \eta_m} = \gamma_m - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j d_i d_j K_m(\mathbf{x}_i, \mathbf{x}_j) \quad (5.3)$$

O processo de aprendizagem envolve as Equações 5.2 e 5.3 e é descrito na próxima seção.

5.4 Aprendizagem

Em uma MSV de Núcleo único, como mostrado na Seção 2.3, o objetivo da etapa de aprendizagem é determinar o valor do vetor de pesos ótimo $\mathbf{w}_o$, que define o hiperplano ótimo de separação. Para o caso da MSV de Múltiplos Núcleos, além do valor do vetor de pesos $\mathbf{w}_o$, deve-se determinar o valor do vetor de pesos η , que pondera a combinação linear dos Núcleos $K(\mathbf{x}_i, \mathbf{x}_j)$ em uma matriz de Núcleos ótima $\mathbf{K}_o = \sum_{m=1}^M \eta_m K_m$.

A aprendizagem é realizada em duas etapas. Na primeira etapa os valores de η e K_m são fixados. A função objetivo W pode ser maximizada como numa MSV de Núcleo único K_m , apresentada na Seção 2.3. Para este problema de otimização, algoritmos como (BACH; LANCKRIET; JORDAN, 2004) podem ser usados.

Na segunda etapa a Equação 5.1 é minimizada através do método do gradiente descendente, fazendo uso da Equação 5.3.

As duas etapas se repetem até a convergência ou até o número de repetições máximo ser atingido.

5.5 Execução

Como a implementação utilizada aqui faz uso da abordagem Um-Contra-Todos para operar com Múltiplas Classes, tem-se neste ponto uma MSV treinada para cada classe. O dado de entrada é testado em todas as MSVs e o resultado final é a classe com maior número de vitórias.

Olhando individualmente, em uma Máquina de Suporte Vetorial treinada para diferenciar entre sua classe e as demais, o processo de classificação é baseado na equação

$$\text{sgn}(\mathbf{w}^T \mathbf{x} + b)$$

onde o sinal *sgn* retornado representa o lado do hiperplano de separação em que o vetor de entrada $\mathbf{x}$ se enquadra. De acordo com este valor, a classe é atribuída ou não ao dado testado.

6 Experimentos e Resultados

Com base no detalhamento descrito até aqui, este capítulo apresenta os experimentos realizados e a análise dos resultados obtidos. Os experimentos foram realizados de maneira a avaliar os resultados obtidos pelo classificador adotado na classificação dos objetos extraídos de imagens onidirecionais, retificadas e não retificadas. O processo faz uso de imagens convencionais e onidirecionais. Cinco configurações entre treinamento e teste de classificação foram testadas. Estas configurações são listadas a seguir:

1. Treinamento e teste com imagens convencionais.
2. Treinamento e teste com imagens onidirecionais.
3. Treinamento e teste com imagens retificadas.
4. Treinamento com imagens convencionais; teste com imagens onidirecionais.
5. Treinamento com imagens convencionais; teste com imagens retificadas.

Inicialmente neste capítulo, o ambiente em que os testes foram realizados é descrito. Na sequência, exemplos das imagens utilizadas para treinamento e testes do classificador são exibidas. As seções seguintes mostram os resultados para cada uma das configurações avaliadas.

6.1 Ambiente de Teste

Todos os experimentos foram realizados no ambiente computacional Matlab 2009.

O algoritmo utilizado nos testes, contendo a implementação do classificador MKL, detalhado na Seção 5.3, foi obtido do projeto *Multiple Kernels for Image Classification* (VEDALDI et al., 2009b). Adaptações no código foram realizadas para a utilização das imagens de treinamento e teste necessárias para estes experimentos. A implementação dos detectores de características utilizados foi obtida do projeto VLFeat.org (VEDALDI; FULKERSON, 2005). Vale destacar que esta é uma biblioteca de código aberto.

Como base de comparação dos tempos de execução do processo, o equipamento utilizado para os experimentos foi um PC Intel Core2Duo 2.0GHz, com sistema operacional Windows 7.

6.2 Conjuntos de Dados

As imagens utilizadas para a realização dos experimentos foram obtidas de quatro conjuntos de dados diferentes. Para as imagens convencionais, os conjuntos de dados Pascal VOC 2008 (EVERINGHAM et al., 2008) e Caltech 101 (GRIFFIN; HOLUB; PERONA, 2007) foram utilizados. Para as imagens onidirecionais foram utilizados os conjuntos COLD (PRONOBIS; CAPUTO, 2009) e Cogniron (ZIVKOVIC et al., 2008).

As imagens onidirecionais foram obtidas, em ambos os conjuntos de dados, com o uso de câmeras onidirecionais catadióptricas de espelho hiperbólico central. A Figura 6.1a mostra o modelo da câmera utilizada. A montagem completa de um dos robôs utilizados na aquisição das imagens do projeto COLD pode ser vista na Figura 6.1b.

Fig. 6.1: (a) Modelo de câmera onidirecional catadióptricas de espelho hiperbólico central. (b) Montagem de um dos robôs utilizados na coleta dos dados do projeto COLD (PRONOBIS; CAPUTO, 2009)

Em todos os conjuntos de dados, imagens representando quatro objetos distintos foram separadas. Os objetos são: **monitor**, **sofá**, **máquina copiadora** e **extintor de incêndio**. Estes objetos foram escolhidos por serem encontrados tanto nos conjuntos de imagens convencionais quanto nos conjuntos de imagens onidirecionais, além de serem possivelmente encontrados em ambientes internos, facilitando estudos

posteriores.

A Figura 6.2 mostra exemplos de imagens convencionais para os objetos avaliados nos experimentos.

Fig. 6.2: Exemplos de imagens convencionais para os objetos avaliados: (a) sofá, (b) monitor, (c) máquina copiadora e (d) extintor de incêndio (GRIFFIN; HOLUB; PERONA, 2007)

A Figura 6.3 representa uma imagem onidirecional do conjunto de dados Cogniron (ZIVKOVIC et al., 2008). A partir destas imagens, quadros contendo os objetos de interesse na cena foram extraídos. Nenhum tratamento foi dado às imagens obtidas neste processo de maneira a manter suas características originais. A Figura 6.4 mostra exemplos dos objetos obtidos nas imagens onidirecionais.

O terceiro tipo de imagem utilizada é a imagem onidirecional retificada. Da mesma forma que a imagem onidirecional não retificada, a imagem retificada contém quase todo o ambiente à volta do ponto de observação, porém, de um modo panorâmico, reduzindo principalmente problemas relacionados à rotação dos objetos. As imagens retificadas foram obtidas a partir das imagens onidirecionais adotadas, aplicadas ao processo de retificação descrito em 5.1. Neste processo, a ferramenta de Pronobis (2009) foi utilizada. O ponto central das imagens foi definido manualmente, uma vez que todas as imagens do mesmo conjunto de dados possuem as mesmas dimensões e a mesma localização do ponto central. Para a definição do valor dos pixels faltantes na imagem retificada, a interpolação bilinear foi adotada. A imagem retificada aproxima-se de uma imagem convencional. A Figura 6.5 mostra um exemplo de imagem retificada utilizada neste trabalho. Os mesmos objetos extraídos das imagens

Fig. 6.3: Exemplo de imagem onidirecional do conjunto de dados Cogniron (ZIVKOVIC et al., 2008)

onidirecionais foram localizados e extraídos das imagens retificadas para a realização dos experimentos. Dessa maneira garante-se que os resultados dos testes não sejam influenciados por diferenças nos conjuntos de dados.

6.3 Métodos para Avaliação do Classificador

Antes de expor os resultados obtidos neste trabalho, vale definir algumas medidas para avaliação de classificadores, comumente utilizadas na literatura. O resultado de uma classificação binária pode pertencer a um dos quatro grupos mostrados na Tabela 6.1.

Tab. 6.1: Tabela de contingência para classificação binária

		Classes Estimadas	
		Positivo	Negativo
Classes Reais	Positivo	Positivos Verdadeiros	Falsos Negativos
	Negativo	Falsos Positivos	Negativos Verdadeiros

A partir daí, algumas medidas são definidas.

Precisão é o número de classificações positivas corretas (Positivos Verdadeiros), dividido pelo número de elementos classificados na mesma classe (a soma dos resultados Positivos Verdadeiros e Falsos Positivos).

Revocação (do inglês *Recall*) é o número de classificações positivas corretas

Fig. 6.4: Exemplos de imagens onidirecionais para os objetos avaliados: (a) sofá, (b) monitor, (c) máquina copiadora e (d) extintor de incêndio (PRONOBIS; CAPUTO, 2009)

Fig. 6.5: Imagem onidirecional retificada (PRONOBIS; CAPUTO, 2009)

(Positivos Verdadeiros), dividido pelo número real de objetos da classe (a soma dos resultados Positivos Verdadeiros e Falso Negativos).

Exatidão é o número de classificações corretas (a soma dos resultados Positivos Verdadeiros e Negativos Verdadeiros) dividido pelo número total de classificações (soma de todos os itens).

A **Matriz de Confusão** representa de maneira gráfica e de fácil entendimento se existe alguma confusão no resultado da classificação. A matriz de contingência da Tabela 6.1 é um caso específico da matriz de confusão para apenas duas classes. É uma matriz quadrada onde valores maiores na diagonal indicam bons resultados na classificação.

Algumas destas medidas serão utilizadas na próxima seção para avaliar os resultados dos experimentos.

6.4 Resultados

Para cada configuração de teste citada anteriormente, três etapas completas de treinamento e classificação foram realizadas, alternando de maneira aleatória os dados de treinamento e teste com base nos conjuntos disponíveis para a configuração definida. Os resultados para cada configuração são exibidos a seguir.

6.4.1 Treinamento e teste com imagens convencionais

Este teste foi realizado para avaliar inicialmente os resultados do classificador adotado no modelo, operando sob condições às quais ele foi proposto, ou seja, operando com o tipo de imagem ao qual ele foi projetado e testado. Foram utilizadas 5 amostras de treinamento escolhidas aleatoriamente do conjunto de imagens convencionais. O restante das imagens, dez amostras de cada objeto, foi utilizado para testar o classificador. Esse processo foi realizado três vezes, alterando aleatoriamente as amostras de treinamento e as imagens de teste.

A exatidão média do classificador nessa configuração é de 75,11%. Individualmente, os valores médios de revocação para as classes são mostradas na Tabela 6.2. O melhor resultado médio de classificação foi alcançado para o objeto extintor com revocação de 93,3%.

Tab. 6.2: Resultado para teste da configuração 1 (Treinamento e teste com imagens convencionais)

Classe	Revocação
Extintor	93,3%
Copiadora	70,0%
Sofá	83,3%
Monitor	63,3%

O valor individual de revocação para as classes não é suficiente para avaliar se o resultado da classificação é satisfatória. Avaliar se algum objeto teve classificações confusas com outros objetos, ou seja, classificações falso positivas, também é necessário. Esta análise pode ser feita através da matriz de confusão, como mostrada na Figura 6.6, obtida para este teste. Observa-se nesta matriz algumas classificações falso positivas, principalmente para o objeto monitor, porém nenhum valor que se sobreponha ao resultado de revocação da classe.

Fig. 6.6: Matriz de confusão para teste da configuração 1 (Treinamento e teste com imagens convencionais)

6.4.2 Treinamento e teste com imagens onidirecionais

O intuito deste próximo experimento foi avaliar o comportamento do classificador adotado, quando o conjunto de treinamento e teste são compostos por imagens onidirecionais. Foram utilizadas apenas 5 amostras de treinamento para cada objeto, escolhidas aleatoriamente do conjunto de imagens onidirecionais, devido ao tamanho total do conjunto ser reduzido. O restante das imagens, dez amostras de cada objeto, foi utilizado para testar o classificador. Esse processo foi realizado três vezes, alterando aleatoriamente as amostras de treinamento e as imagens de teste.

A exatidão média do classificador para esta configuração é de 84,0%. Os valores de revocação média para cada classe neste teste são mostradas na Tabela 6.3. Complementando a análise, a Figura 6.7 mostra a matriz de confusão para esta etapa.

Tab. 6.3: Resultado para teste da configuração 2 (Treinamento e teste com imagens onidirecionais)

Classe	Revocação
Extintor	86,7%
Copiadora	66,7%
Sofá	100,0%
Monitor	100,0%

Este teste foi favorecido por usar imagens obtidas dos mesmos objetos para o treinamento e teste do classificador, no entanto, esta situação pode ocorrer na prática, quando um classificador associado a um robô pode ser treinado com imagens obtidas pelo próprio robô, em seu ambiente de operação.

Fig. 6.7: Matriz de confusão para teste da configuração 2 (Treinamento e teste com imagens onidirecionais)

6.4.3 Treinamento e teste com imagens retificadas

Da mesma maneira que o teste anterior, este teste foi realizado para avaliar o comportamento do classificador quanto treinado e testado com imagens retificadas, porém pertencentes ao mesmo ambiente. Foram utilizadas 5 amostras de treinamento por objeto, escolhidas aleatoriamente do conjunto de imagens retificadas. O restante das imagens, dez amostras de cada objeto, foi utilizado para o teste do classificador. Esse processo foi realizado três vezes, alterando aleatoriamente as amostras de treinamento e as imagens de teste.

A exatidão média do classificador nesta configuração é de 80,0%. Individualmente, os valores da revocação média para as classes são mostradas na Tabela 6.4. Os valores são muito próximos dos obtidos no experimento anterior, e da mesma maneira foram favorecidos pela utilização de imagens obtidas dos mesmos objetos, para treinamento e teste do classificador. Não é possível afirmar para esta situação, que a etapa de retificação introduzida ao processo traga benefícios na classificação. A Figura 6.8 mostra a matriz de confusão para esta etapa.

6.4.4 Treinamento com imagens convencionais; teste com imagens onidirecionais

Uma situação mais real e complexa, enfrentada por um classificador, pode ser mostrada nestes dois últimos testes. Supondo que um robô navegue por um ambiente desconhecido, e seu módulo classificador precise categorizar objetos encontrados na cena, um conhecimento prévio, porém totalmente desvinculado do ambiente a ser

Tab. 6.4: Resultado para teste da configuração 3 (Treinamento e teste com imagens retificadas)

Classe	Revocação
Extintor	80,0%
Copiadora	80,0%
Sofá	100,0%
Monitor	100,0%

Fig. 6.8: Matriz de confusão para teste da configuração 3 (Treinamento e teste com imagens retificadas)

observado, precisa ser adquirido. Dessa maneira, imagens convencionais e de fontes independentes, dos objetos de interesse, são utilizadas no treinamento do classificador. Dotado deste treinamento, este teste avalia o comportamento do classificador quando submetido a imagens de objetos obtidas pelo sistema de visão onidirecional.

No treinamento do classificador foram utilizadas 15 amostras de imagens convencionais para cada objeto, escolhidas aleatoriamente do conjunto de dados. O conjunto de imagens onidirecionais completo foi utilizado para avaliar o classificador. Esse processo foi realizado três vezes, alterando aleatoriamente as amostras de treinamento.

A exatidão média do classificador nessa configuração é de 42,0%. Individualmente, as revocações médias para as classes são mostradas na Tabela 6.5.

A matriz de confusão para este teste, observada na Figura 6.9, mostra no entanto que a classificação de quase todos os objetos tenderam ao objeto copiadora. O objeto monitor nem mesmo obteve alguma classificação válida. O objeto extintor por sua vez, que obteve 33,3% de revocação, obteve um valor superior a 60% das classificações

Tab. 6.5: Resultado para teste da configuração 4 (Treinamento com imagens convencionais; teste com imagens onidirecionais)

Classe	Revocação
Extintor	33,3%
Copiadora	63,3%
Sofá	94,4%
Monitor	0,0%

como objeto copiadora, ou seja, um número muito superior de classificações falso positivas.

Fig. 6.9: Matriz de confusão para teste da configuração 4 (Treinamento com imagens convencionais; teste com imagens onidirecionais)

6.4.5 Treinamento com imagens convencionais; teste com imagens retificadas

Este teste avalia se a etapa de retificação da imagem onidirecional, sugerida nesta proposta, torna viável o cenário exposto no início do teste anterior. As características das imagens retificadas são semelhantes às das imagens convencionais utilizadas no treinamento do classificador.

Como anteriormente, foram utilizadas 15 amostras de treinamento para cada objeto, escolhidas aleatoriamente do conjunto de imagens convencionais. Esse processo foi realizado três vezes, alterando aleatoriamente as amostras de treinamento. O conjunto completo de imagens retificadas foi utilizado para a avaliação do classificador.

A exatidão média do classificador nessa configuração é de 49,33%. Os valores médios da revocação de cada classe avaliada são mostrados na Tabela 6.6. A Figura 6.10 mostra a matriz de confusão para esta etapa.

Tab. 6.6: Resultado para teste da configuração 5 (Treinamento com imagens convencionais; teste com imagens retificadas)

Classe	Revocação
Extintor	46,7%
Copiadora	90,0%
Sofá	77,8%
Monitor	33,3%

Mesmo com uma redução na revocação da classe sofá, sem grandes prejuízos pelo número de classificações corretas continuar superior ao de classificações erradas, a etapa de retificação na imagem onidirecional conferiu ao classificador uma melhoria em sua exatidão média. A melhoria pode também ser notada nos valores de revocação dos demais objetos e na redução das classificações erradas observadas na matriz de confusão.

Fig. 6.10: Matriz de confusão para teste da configuração 5 (Treinamento com imagens convencionais; teste com imagens retificadas)

7 Conclusões e Trabalhos Futuros

Este trabalho apresentou a avaliação de um método para classificação de objetos em imagens onidirecionais, combinando um classificador baseado em máquinas de suporte vetorial com múltiplos núcleos, já aplicado com sucesso na classificação de objetos em imagens convencionais, e uma etapa prévia para retificação da imagem onidirecional, com o intuito de reduzir as distorções causadas pelo espelho hiperbólico utilizado na construção da câmera onidirecional e, por consequência, aproximá-la de uma imagem convencional.

Os experimentos realizados com imagens convencionais puderam comprovar que o conjunto de características adotado possui relevância suficiente para a descrição dos objetos. A utilização destas características se deu ao fato de a extensão da MSV para Múltiplos Núcleos permitir a combinação ponderada destas para a descrição de cada classe de objetos.

O cenário de maior relevância avaliado no experimento, onde o classificador foi treinado com imagens convencionais dos objetos de interesse e posteriormente avaliado com imagens onidirecionais retificadas ou não retificadas, mostrou que a etapa de retificação sugerida neste trabalho traz uma melhoria na exatidão média da classificação. Estes resultados demonstram a viabilidade da utilização de classificadores, aplicados com sucesso no reconhecimento de objetos em imagens convencionais, na construção de sistemas de navegação baseados em câmeras onidirecionais para robôs móveis.

Esta possibilidade de utilização de imagens convencionais independentes do ambiente ao qual o sistema será aplicado, para o treinamento do classificador, pode ser observada como a grande contribuição deste trabalho. Tal característica permite que o classificador seja facilmente treinado com imagens obtidas através de ferramentas de busca na internet, para realizar a classificação de qualquer objeto desejado, mesmo operando através de um sistema de visão onidirecional.

Para a evolução deste trabalho, a utilização de uma matriz híbrida, como proposto por Puig, Guerrero e Sturm (2008), para um melhor mapeamento entre as imagens onidirecionais e convencionais, deve ser avaliada como complemento da etapa de retificação. Além disso, novos tipos de características ou descritores, com melhor invariância às distorções causadas pelos espelhos utilizados na construção das câmeras onidirecionais, devem ser buscados. Com o objetivo de garantir a aplicabilidade deste modelo em tarefas da robótica móvel, avaliações de desempenho em sistemas SLAM para navegação robótica também precisam ser realizadas.

Referências Bibliográficas

- AGARWAL, S.; AWAN, A.; ROTH, D. Learning to detect objects in images via a sparse, part-based representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 26, n. 11, p. 1475–1490, 2004.
- AMARAL, F. R.; COSTA, A. H. R. Object class detection in omnidirectional images. *Workshop de Visão Computacional*, v. 1, 2009.
- AMARAL, F. R.; COSTA, A. H. R. Classificação de objetos em imagens onidirecionais com uso de retificação de imagens e de múltiplos núcleos em máquinas de vetor de suporte. *III Workshop on Computational Intelligence*, v. 1, 2010.
- BACH, F. R.; LANCKRIET, G. R. G.; JORDAN, M. I. Multiple kernel learning, conic duality, and the smo algorithm. In: *Proceedings of the 21st International Conference on Machine Learning*. New York, NY, USA: ACM, 2004. p. 6.
- BALLARD, D. H.; BROWN, C. M. *Computer Vision*. [S.l.]: Prentice Hall, 1982. Hardcover.
- BAY, H.; TUYTELAARS, T.; GOOL, L. J. V. Surf: Speeded up robust features. In: *European Conference on Computer Vision (1)*. [S.l.: s.n.], 2006. p. 404–417.
- BERG, A. C.; BERG, T. L.; MALIK, J. Shape matching and object recognition using low distortion correspondences. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Washington, DC, USA: IEEE Computer Society, 2005. v. 1, p. 26–33.
- BERG, A. C.; MALIK, J. Geometric blur for template matching. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2001. v. 1, p. I-607–I-614 vol.1.
- BERTSEKAS, D. P. *Dynamic Programming and Optimal Control, Vol. II*. [S.l.]: Athena Scientific, 2007.
- BOSCH, A.; ZISSERMAN, A.; MUNOZ, X. Image classification using random forests and ferns. In: *IEEE 11th International Conference on Computer Vision*. [S.l.: s.n.], 2007. p. 1–8.
- CHEN, C.-H. et al. Heterogeneous fusion of omnidirectional and ptz cameras for multiple object tracking. *IEEE Transactions on Circuits and Systems for Video Technology*, v. 18, n. 8, p. 1052–1063, 2008.
- DUDA, R. O.; HART, P. E.; STORK, D. G. *Pattern Classification (2nd Edition)*. 2. ed. [S.l.]: Wiley-Interscience, 2000. Hardcover.
- EDELMAN, S.; INTRATOR, N.; POGGIO, T. Complex cells and object recognition. [Http://kybele.psych.cornell.edu/~edelman/archive.html](http://kybele.psych.cornell.edu/~edelman/archive.html). 1997.
- EVERINGHAM, M. et al. *The PASCAL Visual Object Classes Challenge 2008 (VOC2008) Results*. 2008. [Http://www.pascal-network.org/challenges/VOC/voc2008/workshop/index.html](http://www.pascal-network.org/challenges/VOC/voc2008/workshop/index.html).

- EVERINGHAM, M. et al. *The PASCAL Visual Object Classes Challenge 2009 (VOC2009) Results*. 2009.
[Http://www.pascal-network.org/challenges/VOC/voc2009/workshop/index.html](http://www.pascal-network.org/challenges/VOC/voc2009/workshop/index.html).
- FEIFEI, L.; FERGUS, R.; PERONA, P. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *Computer Vision and Image Understanding*, v. 106, n. 1, p. 59–70, April 2007.
- FELZENSZWALB, P. F. et al. Object detection with discriminatively trained part based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, IEEE Computer Society, Los Alamitos, CA, USA, v. 99, n. 1, 2007.
- FERGUS, R.; PERONA, P.; ZISSERMAN, A. Object class recognition by unsupervised scale-invariant learning. In: *Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2003. v. 2, p. 264–271.
- FRITZ, M. et al. Integrating representative and discriminative models for object category detection. In: *IEEE International Conference on Computer Vision*. [S.l.: s.n.], 2005. p. 1363–1370.
- GOEDEMÉ, T. et al. Omnidirectional vision based topological navigation. *International Journal of Computer Vision*, v. 74, n. 3, p. 219–236, 2007.
- GRIFFIN, G.; HOLUB, A.; PERONA, P. *Caltech-256 Object Category Dataset*. [S.l.], 2007.
- HARRIS, C.; STEPHENS, M. A combined corner and edge detection. In: *Proceedings of The Fourth Alvey Vision Conference*. [S.l.: s.n.], 1988. p. 147–151.
- HAYKIN, S. *Neural Networks: A Comprehensive Foundation (2nd Edition)*. 2. ed. [S.l.]: Prentice Hall, 1998. Hardcover.
- HERBRICH, R. *Learning Kernel Classifiers: Theory and Algorithms (Adaptive Computation and Machine Learning)*. [S.l.]: The MIT Press, 2001. Hardcover.
- HESS, R. *SIFT Library*. 2010. [Http://blogs.oregonstate.edu/hess/](http://blogs.oregonstate.edu/hess/).
- KE, Y.; SUKTHANKAR, R. Pca-sift: A more distinctive representation for local image descriptors. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2004. v. 2, p. 506–513.
- LANCKRIET, G. et al. Learning the kernel matrix with semidefinite programming. *Journal of Machine Learning Research*, v. 5, p. 27–72, 2004.
- LAZEBNIK, S.; SCHMID, C.; PONCE, J. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2006. v. 2, p. 2169–2178.
- LEIBE, B.; LEONARDIS, A.; SCHIELE, B. Combined object categorization and segmentation with an implicit shape model. In: *Workshop on Statistical Learning in Computer Vision - ECCV*. [S.l.: s.n.], 2004. p. 17–32.
- LEIBE, B.; LEONARDIS, A.; SCHIELE, B. Combined object categorization and segmentation with an implicit shape model. In: *Workshop on Statistical Learning in Computer Vision, ECCV*. [S.l.: s.n.], 2004.

- LOWE, D. G. Object recognition from local scale-invariant features. In: *International Conference on Computer Vision*. [S.l.: s.n.], 1999. p. 1150–1157.
- LOWE, D. G. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, v. 60, n. 2, p. 91–110, 2004.
- MACKAY, D. J. C. *Information Theory, Inference & Learning Algorithms*. 1st. ed. [S.l.]: Cambridge University Press, 2002. Hardcover.
- MERCER, J. Functions of positive and negative type and their connection with the theory of integral equations. *Philosophical Transactions of the Royal Society*, 1909.
- MIKOLAJCZYK, K.; LEIBE, B.; SCHIELE, B. Multiple object class detection with a generative model. In: *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2006. v. 1, p. 26–36.
- MIKOLAJCZYK, K.; SCHMID, C. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 27, n. 10, p. 1615–1630, 2005.
- MORAVEC, H. P. Rover visual obstacle avoidance. In: *International Joint Conference on Artificial Intelligence*. [S.l.: s.n.], 1981. p. 785–790.
- PRONOBIS, A. *Unwrapping Omnidirectional Images*. 2009.
[Http://www.pronobis.pro/software/unwrap](http://www.pronobis.pro/software/unwrap).
- PRONOBIS, A.; CAPUTO, B. COLD: COsy Localization Database. *The International Journal of Robotics Research*, v. 28, n. 5, May 2009.
- PUIG, L.; GUERRERO, J.; STURM, P. Matching of omnidirectional and perspective images using the hybrid fundamental matrix. In: *The 8th Workshop on Omnidirectional Vision, Camera Networks and Non-classical Cameras - OMNIVIS*. [S.l.: s.n.], 2008.
- SCHMID, C.; MOHR, R. Local grayvalue invariants for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 19, n. 5, p. 530–535, 1997.
- SCHÖLKOPF, B.; SMOLA, A. J. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond (Adaptive Computation and Machine Learning)*. 1st. ed. [S.l.]: The MIT Press, 2001. Hardcover.
- SELVATICI, A.; COSTA, A. H. R.; DELLAERT, F. Object-based visual slam: How object identity informs geometry. *Workshop de Visão Computacional*, v. 1, p. 82–87, 2008.
- SHECHTMAN, E.; IRANI, M. Matching local self-similarities across images and videos. In: *IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.]: IEEE, 2007. p. 1–8.
- SONNENBURG, S. et al. Large scale multiple kernel learning. *Journal of Machine Learning Research*, JMLR.org, v. 7, p. 1531–1565, 2006.
- TAMIMI, H. et al. Localization of mobile robots with omnidirectional vision using particle filter and iterative sift. *Robotics and Autonomous Systems*, v. 54, n. 9, p. 758–765, 2006.

- TORRALBA, A. B.; MURPHY, K. P.; FREEMAN, W. T. Sharing features: Efficient boosting procedures for multiclass object detection. In: *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2004. v. 2, p. 762–769.
- ULUSOY, I.; BISHOP, C. M. Generative versus discriminative methods for object recognition. In: *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2005. v. 2, p. 258–265.
- VAPNIK, V. N. *The nature of statistical learning theory*. New York, NY, USA: Springer-Verlag New York, Inc., 1995.
- VARMA, M.; RAY, D. Learning the discriminative power-invariance trade-off. In: *IEEE 11th International Conference on Computer Vision*. [S.l.: s.n.], 2007. p. 1–8.
- VEDALDI, A.; FULKERSON, B. *VLFeat Open Source Library*. 2005. [Http://www.vlfeat.org/](http://www.vlfeat.org/).
- VEDALDI, A. et al. Multiple kernels for object detection. In: *Proceedings of the International Conference on Computer Vision*. [S.l.: s.n.], 2009.
- VEDALDI, A. et al. *Multiple Kernels for Image Classification*. 2009. [Http://www.robots.ox.ac.uk/vgg/software/MKL/](http://www.robots.ox.ac.uk/vgg/software/MKL/).
- VIOLA, P.; JONES, M. Rapid object detection using a boosted cascade of simple features. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2001.
- VIOLA, P. A.; JONES, M. J. Robust real-time face detection. *International Journal of Computer Vision*, v. 57, n. 2, p. 137–154, 2004.
- WALLRAVEN, C.; CAPUTO, B.; GRAF, A. B. A. Recognition with local features: the kernel recipe. In: *IEEE International Conference on Computer Vision*. [S.l.: s.n.], 2003. p. 257–264.
- YANG, S. et al. Detecting and tracking moving targets on omnidirectional vision. *Transactions of Tianjin University*, v. 15, n. 1, p. 13–18, February 2009.
- ZIVKOVIC, Z.; BOOIJ, O.; KRÖSE, B. J. A. From images to rooms. *Robotics and Autonomous Systems*, v. 55, n. 5, p. 411–418, 2007.
- ZIVKOVIC, Z. et al. From sensors to human spatial concepts: An annotated data set. *IEEE Transactions on Robotics*, v. 24, n. 2, p. 501–505, 2008.