



Universidade Federal da Bahia
Programa de Pós-graduação em Mecatrônica

**UMA ABORDAGEM PARA VISÃO ARTIFICIAL EM ROBÓTICA
MÓVEL BASEADA EM FUSÃO DE SENSORES**

Luciano Rebouças de Oliveira

Salvador
2005

Universidade Federal da Bahia

Luciano Rebouças de Oliveira

**UMA ABORDAGEM PARA VISÃO ARTIFICIAL EM ROBÓTICA
MÓVEL BASEADA EM FUSÃO DE SENSORES**

Trabalho apresentado ao Programa de Pós-graduação em Mecatrônica, programa conjunto entre o Departamento de Engenharia Mecânica e o Departamento de Ciência da Computação, da Universidade Federal da Bahia como requisito parcial para obtenção do grau de Mestre em Mecatrônica.

Orientador: *Prof. Dr. Augusto César Pinto Loureiro da Costa*

Co-orientador: *Prof. Dr. Leizer Schnitman*

Salvador
2005

Universidade Federal da Bahia

Luciano Rebouças de Oliveira

**UMA ABORDAGEM PARA VISÃO ARTIFICIAL EM ROBÓTICA
MÓVEL BASEADA EM FUSÃO DE SENSORES**

Banca Examinadora

Prof. Dr. Augusto César Pinto Loureiro da Costa (Orientador)

Prof. Dr. Leizer Schnitman (Co-orientador)

Prof. Dr. Anna Helena Reali Costa (USP)

Prof. Dr. Jês de Jesus Fiais Cerqueira (UFBA)

Prof. Dr. Iuri Muniz Pepe (UFBA)

Salvador
2005

Dedico às mulheres da minha vida (minha esposa, minha mãe e minha irmã) que me deram força e suportaram todos os momentos de afastamento durante esse projeto de dissertação.

*Quanto mais uma pessoa se eleva, menor ela parece às pessoas que
não sabem voar.*

—FRIEDRICH NIETZSCHE

AGRADECIMENTOS

É sempre uma tarefa injusta a de agradecer à ajuda de outros. Sempre corremos o risco de esquecer alguém ou de não usarmos eficientemente as palavras para descrever o real auxílio recebido. Mesmo levando isso em consideração e sabendo que muitas pessoas, sejam elas anônimas ou não, servem de apoio para um projeto que exige dedicação quase que exclusiva durante dois anos, tentarei agradecer de maneira sincera.

Começo citando minha família que suportou um duro afastamento de minha parte, em especial minha esposa e minha mãe. Minha irmã, que mesmo longe, fez algumas vezes o impossível para estar perto de mim, emocional ou fisicamente. Meu pai que me ensinou a ser forte diante da vida.

Os orientadores quase sempre são apenas citados nesta seção com alguma referência tímida. Como costumo extrair sempre algo que ajude em meu desenvolvimento, lembro (mais que apenas cito) dessas duas figuras como indispensáveis para o meu projeto. Seja com experiências diretamente fortes ou com um apoio sincero com o objetivo de meu crescimento, tive realmente sorte em aprender com os dois, de diversas formas e em muitas situações diferentes.

Devo agradecer ao apoio financeiro do projeto AeroUfba e ao apoio técnico do Laboratório de Propriedades Óticas (LAPO). Dois professores foram de extrema importância para o meu projeto, Prof. Dr. Hermam Lepkison e Prof. Dr. Iuri Pepe. Agradeço não somente ao apoio financeiro ou técnico, mas também à amizade, conversas e apoio nos momentos difíceis.

Agradeço, em especial, a Luiz Simões do LAPO que, sem nenhum envolvimento formal no projeto, me auxiliou na construção do protótipo físico, de maneira dedicada e com alto nível técnico.

À Lúcia Lago, secretária do Mestrado, que ao longo desse tempo se mostrou mais que uma pessoa de eficiência ímpar e indispensável para o curso, mas também uma grande amiga.

Agradeço também à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pelo apoio financeiro durante os dois anos de Mestrado.

Enfim, sinto-me sinceramente devedor a todos que, direta ou indiretamente, me apoiaram e me ensinaram algo relevante para a conclusão desse trabalho de dissertação.

RESUMO

Cada sub-sistema de um robô móvel possui aspectos próprios, que devem estar integrados a fim de capacitar o robô para a realização de uma tarefa particular. Dentre estes sub-sistemas, o sensoriamento é responsável por construir uma representação dos marcos do ambiente a fim de fornecer informações relevantes para a navegação do robô. Em geral, a construção de uma representação do ambiente possui alto custo computacional, devido a inaptidão nas estimativas de medição dos sensores.

A proposta deste trabalho é a construção de um sistema de sensoriamento, baseado em fusão de sensores, de baixo custo computacional. O objetivo do sistema é o reconhecimento e localização espacial de objetos. A abordagem para o sistema se baseia na fusão redundante e complementar de dados dos sensores envolvidos (uma câmera e sensores de distância), utilizando Lógica Difusa. Para a classificação e localização dos objetos na imagem, o método de Máquina de Vetores de Suporte é utilizado e os resultados obtidos apontam para um bom desempenho nestas tarefas, com a utilização de apenas uma amostra de treinamento e sem qualquer tipo de processamento de imagem prévio. Ao final do processo, uma descrição dos objetos na cena é enviada para o robô, com informações detalhadas sobre cada objeto detectado no quadro considerado.

O sistema de sensoriamento proposto é aplicado no ambiente de experimentação de futebol de robôs, cujas regras determinam as restrições para a arquitetura do sistema.

Palavras-chave: Robótica Móvel, Classificação de Imagens, Fusão de Sensores, Máquina de Vetor de Suporte, Visão Artificial, Lógica Difusa

ABSTRACT

Each subsystem of a mobile robot owns aspects that must be integrated in order to enable the robot to the accomplishment of a particular task. Amongst these subsystems, the sensing is responsible for constructing a representation of landmarks of the environment in order to supply relevant information for the navigation of the robot. In general, to construct a representation of the environment represents a high computational cost due to ineptitude to estimate sensor measurements.

The proposal of this work is a sensing system, based on sensor fusion, of low computational cost. The objective of the system is the recognition and spatial object localization. The approach for the system is based on the redundant and complementary fusing of data of the involved sensors (a camera and sensors of distance), using Fuzzy Logic. For the classification and localization of objects, the method of Support Vector Machine is used and the results point out to a good performance in these tasks, with the use of only one sample of training and without any type of previous image processing. To the end of the process, a description of objects in the scene is sent to the robot, with detailed information about each object detected.

The considered vision system is applied in the experimentation environment of robot soccer, whose rules determine the constraints for the system architecture.

Keywords: Mobile Robotic, Image Classification, Sensor Fusion, Support Vector Machine, Artificial Vision, Fuzzy Logic

SUMÁRIO

Capítulo 1—Introdução	1
1.1 Trabalhos correlatos	3
1.2 Escopo da dissertação	4
Capítulo 2—Contextualização	6
2.1 Ambiente de experimentação	7
2.2 Arquitetura do agente no Axebot	10
2.3 Proposta	11
2.3.1 Contribuições	12
2.4 Considerações finais	13
Capítulo 3—Sistemas de visão artificial	15
3.1 Planejamento físico de sistemas de visão artificial	17
3.1.1 Tecnologias de sensores de visão	20
3.1.1.1 CMOS	21
3.1.1.2 CCD	22
3.2 Classificação de padrões em imagem	22
3.2.1 Padrões em imagem	24
3.2.1.1 Cor	25
3.2.1.2 Textura	25
3.2.1.3 Forma	26
3.2.2 Métodos para classificação de padrões em imagem	26
3.2.2.1 Métodos estruturais	27
3.2.2.2 Métodos especialistas	28
3.2.2.3 Métodos matemáticos	30
3.3 Requisitos matemáticos para o classificador utilizado	33
3.3.1 Definição	35
3.3.2 Espaços linearmente separáveis	35
3.3.3 Teoria da Otimização	38
3.3.4 Kernels	41
3.3.5 Princípio da minimização de risco	42
3.3.6 Dimensão VC	44
3.3.7 Espaços não linearmente separáveis	45
3.4 Considerações finais	46

Capítulo 4—Fusão de sensores	47
4.1 Arquiteturas	50
4.2 Técnicas de fusão de sensores	52
4.2.1 Filtro de Kalman	53
4.2.2 Dempster-Shafer	53
4.3 Lógica difusa	55
4.3.1 Definição	55
4.3.2 Teoria geral	56
4.3.3 Sistemas de inferência	59
4.4 Considerações finais	61
Capítulo 5—Solução proposta: implementação e resultados	62
5.1 Arquitetura do sistema de visão artificial	62
5.1.1 Aquisição de dados	63
5.1.1.1 Câmera	64
5.1.1.2 Sensor de distância	67
5.1.2 Reconhecimento e localização de objetos	69
5.1.2.1 Construção do classificador	70
5.1.2.2 Localização de objetos	76
5.1.3 Processador dos sensores de distância	80
5.1.4 Processador de fusão	82
5.2 Protótipo físico	85
5.3 Características e propriedades do sistema	86
5.4 Experimentos e testes	91
5.4.1 Resultados	92
Capítulo 6—Conclusões e Perspectivas	98
6.1 Perspectivas	99
6.2 Publicações	99

LISTA DE TABELAS

3.1	Configuração física de sistemas de visão artificial	18
3.2	Vantagens e desvantagens de uma RNA	34
5.1	Características da CMUCam2	65
5.2	Comando CR da CMUCam2	66
5.3	Especificação técnica do sensor Sharp GP2D02	67
5.4	Avaliação do procedimento de <i>Validação cruzada em 5-dobras</i> – RGB .	73
5.5	Avaliação do procedimento de <i>Validação cruzada em 5-dobras</i> – YCrCb	74
5.6	Resultados em diferentes espaços de imagem	76
5.7	Número de VS para cada <i>kernel</i> na regressão	78
5.8	Distância x valores decimalizados dos sensores	82
5.9	Características do sistema de visão artificial proposto	88
5.10	Resultados em diferentes faixas de luz	93
5.11	Dados levantados para 4 ângulos de cabeça de visão	94
5.12	Dados levantados para 4 ângulos de cabeça de visão	95
5.13	Correção do erro absoluto em cada ângulo da cabeça de visão	96

LISTA DE FIGURAS

2.1	Estrutura de um robô móvel	7
2.2	Esquema físico da liga F-180 utilizando visão global	8
2.3	Robô móvel com visão local	9
2.4	Arquitetura do agente autônomo concorrente (fluxo de informações)	10
2.5	Nível reativo	11
2.6	Integração entre o sistema de visão artificial e o agente autônomo concorrente que controla o Axebot	12
3.1	Modelo geral de sistemas de visão artificial	15
3.2	Tipos de sistemas de visão artificial quanto ao grau de movimentação	20
3.3	Tecnologia de um sensor de visão. Figura adaptada de (GROTTA, 2003)	21
3.4	Estrutura de um sensor de visão com tecnologia CMOS. Adaptada de (GROTTA, 2003)	22
3.5	Estrutura de um sensor de visão com tecnologia CCD. Adaptada de (FORSYTH; PONCE, 2003)	23
3.6	Exemplos de padrões	25
3.7	Representação do espaço de cor RGB	26
3.8	Exemplo de uma textura em imagem	26
3.9	Letras “As” com 3 estruturas diferentes. Figura retirada de (SHAPIRO; STOCKMAN, 2000)	27
3.10	Grafo representando a relação das estruturas de uma letra A. Figura adaptada de (SHAPIRO; STOCKMAN, 2000)	28
3.11	Modelo geral de uma Rede Neural Artificial	32
3.12	Modelo geral de um neurônio	33
3.13	Modelo geral de uma Máquina de Vetor de Suporte	36
3.14	Hipersuperfície separadora (w,b) para um conjunto de treinamento bidimensional	37
3.15	Problema de classificação com hiperplano de margem maximal	37
3.16	Mapeamento de um espaço de entrada não-linearmente separável para um espaço de características	41
4.1	Modelo JDL de fusão de dados de sensores. Adaptado de (HALL; LLINAS, 2001)	48
4.2	Arquitetura flexível de fusão de sensores. Adaptado de (DASARATHY, 1997)	52
4.3	Modelo geral de um sistema difuso	57
4.4	Estrutura de uma variável linguística	58

5.1	Arquitetura do sistema de visão artificial	63
5.2	CMUCam2	64
5.3	Sensor de distância Sharp GP2D02	67
5.4	Ligação entre o computador e o sensor de distância	68
5.5	Protocolo de comunicação do sensor Sharp GP2D02. Retirada de (GP2D02, 2005)	68
5.6	Interface gráfica construída para o gerenciamento do sistema de visão artificial proposto	71
5.7	Amostras utilizadas	73
5.8	Imagens segmentadas	75
5.9	Pixels descartados para a melhoria no tempo de classificação da imagem	76
5.10	Determinação da relação entre pixel e ângulo	77
5.11	Ângulos da cabeça móvel utilizados para a determinação da função de mapeamento pixel-ângulo. O ângulo zero representa a posição inicial da câmera quando o robô é ativado	78
5.12	Centróides da bola e do robô (ponto branco)	79
5.13	Determinação da relação entre altura e distância	79
5.14	Processo de calibração dos sensores de distância	80
5.15	Curvas dos 5 sensores de distância	81
5.16	Linha de tempo dos sensores envolvidos no processo de fusão	83
5.17	Determinação dos conjuntos difusos a partir da geometria dos sensores de distância no protótipo	84
5.18	Conjuntos difusos para a variável linguística distância	85
5.19	Protótipo do robô com a cabeça de visão artificial	86
5.20	Determinação do ângulo de abertura da câmera	87
5.21	Determinação dos limites de percepção do sistema proposto	87
5.22	Conjuntos difusos para a variável linguística velocidade	89
5.23	Determinação da velocidade difusa	90
5.24	Suporte de calibração utilizado para avaliação da precisão do processador de fusão de sensores	92
5.25	Resultado da classificação. Objetos classificados pela MVS. Cada cor representa um objeto classificado	93
5.26	Gráfico do ângulo da cabeça de visão x correção do erro absoluto	96

LISTA DE ACRÔNIMOS

RMAI	Robô Móvel Autônomo e Inteligente	1
MVS	Máquina de Vetor de Suporte	2
CCD	<i>Charge Coupled Device</i>	20
CMOS	<i>Complementary Metal Oxide Semiconductor</i>	20
RNA	Redes Neural Artificial	31
PMR	Princípio de Minimização de Risco	43
VC	<i>Vapnik-Chervonenkis</i>	43
JDL	<i>Joint Directors of Laboratories Data Fusion Workgroup</i>	48
TSK	Takagi-Sugeno-Kang	59
SF	<i>Send Frame</i>	64
CR	<i>Camera Register</i>	65
DS	<i>Down Sample</i>	66
TC	<i>Track Color</i>	69
RF	<i>Read Frame</i>	70
BM	<i>Buffer Mode</i>	70

LISTA DE SÍMBOLOS

x_i^K	Vetor de característica i de um objeto k	24
$U(\lambda)$	Comprimento de onda de uma cor λ	24
P_i	Primitivas de uma cor i	25
$fc_i(\cdot)$	Função de correspondência de uma cor	25
c	Classes	26
τ	Conjunto de pares para treinamento	26
G	Grafo	29
N	Nós de um grafo	29
X	Vetor de entrada	31
W	Vetor de pesos	31
Y	Vetor de saída	36
γ	Margem de separação de uma MVS	37
$f(\cdot)$	Função objetiva	38
$g_i(\cdot); h_i(\cdot)$	i -ésimas relações de restrição	38
ξ	Variável de folga	40
F	Espaço de característica	41
$K(\cdot)$	Função <i>kernel</i>	42
$R[\cdot]$	Função de risco	43
E	Esperança matemática	44
Θ	Quadro de discernimento	53
S_i	Sensor i	53
m_i	Combinação de observações i	54
E_k	Evidência de uma observação k	54
Z	Conjunto ordinário	55
z	Elemento de um conjunto ordinário Z	55
$\mu_A(\cdot)$	Função característica de um conjunto ordinário	55
U	Universo de discurso	56
$\mu_A(\cdot)$	Função de pertinência do conjunto A	56
$A_i; B_i$	i -ésimos conjuntos difusos	57
V	Variável linguística	58
C	Critério de parada	72

LISTA DE SÍMBOLOS

(θ, d)	Coordenadas polares dos objetos localizados	75
$a(\cdot)$	Função ângulo	85
$d(\cdot)$	Função distância	85
$V_{min}; V_{max}$	Velocidades mínima e máxima de detecção	87
$Z_{min}; Z_{max}$	Limites do sensor de distância	87

CAPÍTULO 1

INTRODUÇÃO

O campo da robótica móvel tem evidenciado vasta possibilidade de pesquisas e experimentos reais. Robôs móveis ¹ que apresentam comportamentos inteligentes são constituídos de sub-sistemas que isoladamente demonstram potenciais de estudo (KORTENKAMP; BONASSO; MURPHY, 1998).

Segundo Franca (2003), os problemas cinemáticos fundamentais da robótica móvel são: desviar de obstáculos e seguir pela melhor trajetória até a sua meta. O primeiro caso deve ser tratado visando a segurança do robô; no segundo, o conhecimento dos obstáculos deve influenciar no planejamento da melhor trajetória.

Considerando os problemas anteriores, um Robô Móvel Autônomo e Inteligente (RMAI) deve possuir elementos eletrônicos e computacionais adequados a fim de realizar seu objetivo de forma eficaz. Neste contexto, o planejamento de sensores torna-se de fundamental importância por prover técnicas para a determinação da configuração de sensores, a fim de extrair informações necessárias para a navegação segura do robô.

Um trabalho de levantamento dos aspectos importantes para o planejamento de sensores em robótica móvel é realizado por Tarabanis, Allen e Tsai (1995). Neste estudo, os autores propõem um modelo baseado no direcionamento da tarefa a ser realizada e definido pela quantificação e entendimento da relação entre objetos a serem detectados e os sensores. A compreensão desta relação tem como objetivo prover o sistema de controle do robô com a maior quantidade de informações sobre o ambiente. Para esse objetivo, sistemas de visão artificial têm demonstrado grande potencial e alguns trabalhos recentes podem ser citados (KATO; NINOMIYA; MASAKI, 2002; LIN et al., 2002; LINGRAND, 2002).

¹Robôs de base móvel

Por captar projeções do espaço em um plano, um sensor de visão requer a integração com outros tipos de sensores que forneçam a noção de distância, quando se deseja obter a localização espacial de objetos numa cena. Desta forma, o estudo de técnicas de fusão de dados de sensores torna-se uma ferramenta útil para a combinação de medições de sensores presentes num RMAI. O objetivo da fusão de dados de sensores é obter redundância e complementaridade de informações e, portanto, reduzir a natureza comumente inapta dos sensores em geral (FACELI; SARCINELLI; FREITAS, 2002; FRANCA, 2003; GU et al., 2002; KATO; NINOMIYA; MASAKI, 2002; WECHESSE; DILMAN, 1997).

Observa-se que pesquisas em visão artificial têm concentração em diversas áreas de aplicação, tais como: navegação de veículo não-tripulado, robôs guias de museus, robôs detectores de bombas terrestres, entre outras. Além disso, aplicações em robótica móvel são usualmente baseadas em tarefa, ou seja, para cada tarefa proposta a um robô móvel, uma configuração física deve ser adotada e, por conseguinte, uma estratégia computacional. Para este trabalho, o futebol de robôs, a partir das definições da categoria F-180 da *Robocup Federation*, é utilizado como ambiente dinâmico de experimentação (ROBOCUP, 2005).

Para um time de robôs participar de uma partida de futebol, várias áreas de conhecimento devem estar presentes: princípios de projetos de agentes autônomos, colaboração multi-agente, robótica, fusão de sensores, entre outros. A utilização do futebol de robôs tem como objetivo principal, portanto, a análise de múltiplas áreas do conhecimento que servem de suporte a problemas socialmente significativos e na área industrial.

O presente trabalho concentra-se no sistema de visão artificial de um RMAI e, portanto, visa o fornecimento de uma descrição instantânea da cena sensorizada para a correta navegação do robô. O sistema de sensoriamento proposto é referido como a integração entre o sensor de visão e os sensores de distância (infra-vermelho) com o objetivo de prover a localização espacial dos objetos em relação à unidade robótica. A

identificação e a localização dos objetos na cena imageada são obtidas através de uma técnica denominada Máquina de Vetor de Suporte (MVS), que possui como princípio de funcionamento o reconhecimento dos padrões na imagem a partir de um método determinístico-estatístico. Para a fusão de sensores é utilizada a lógica difusa.

A arquitetura utilizada para controlar o RMAI é baseada no agente autônomo concorrente proposto em (COSTA; BITTENCOURT, 1999). O objetivo desta arquitetura é descentralizar o processo decisório, através de três níveis concorrentes: reativo, cognitivo e instintivo. O nível responsável pela aquisição de percepção do agente robótico é o reativo e é para este nível que são enviadas informações do ambiente percebido.

A arquitetura adotada para o RMAI, bem como o sistema de visão artificial proposto, serão aplicados no projeto do robô Axebot (AXEBOT, 2005), projeto de pesquisa para construção de robôs móveis autônomos multidirecionais.

1.1 TRABALHOS CORRELATOS

Embora sua proposta seja recente, muitos trabalhos também recentes na literatura apontam para a robustez da técnica de MVS para a classificação de padrões de imagem. Diversos estudos comprovam esta idéia através de experimentos aplicados. Como exemplo, Wallraven, Caputo e Graf (2003) relatam um estudo sobre aplicações de MVS em imagens através de dois tipos de representações distintas: pixels e histograma. Os autores realizam uma comparação entre os desempenhos de uma MVS com um *kernel* desenvolvido por eles e a técnica de classificação de “Vizinho Mais Próximo” (*Nearest Neighbor*), que é conhecida por seu alto desempenho como classificador. As duas técnicas são aplicadas nos dois tipos de representação de imagem e, em todos os casos, a MVS demonstra uma taxa de erro inferior. Em outra aplicação, Quinlan e Middleton (2003) aplicam uma MVS para a classificação de cor em um robô Sony Aibo. Ao invés de utilizar uma técnica para multiclassificação, uma MVS binária é empregada para cada cor a ser reconhecida e uma tabela de busca (*Lookup Table*) é criada para ser utilizada a identificação dos objetos, em tempo de execução.

Na área de fusão de sensores em robótica móvel, alguns trabalhos podem ser citados: Bastos, Sarcinelli e Freitas (1999) apresentam um trabalho de integração de dados de sonares e uma câmera CCD para detecção e localização de objetos. Em (BONCI; LEO; LONGHI, 2002), outra aplicação de integração de dados entre uma câmera CCD e sonares é discutida. Neste último, o sistema proposto é aplicado para detecção de segmentos de linhas do ambiente visando a construção de mapas por um robô móvel.

Um trabalho relevante na área de fusão de sensores aplicado na indústria é apresentado por Nashman et al. (1998). Os autores propõem um sistema para inspeção dimensional de máquinas utilizadas para medição de coordenadas de peças mecânicas. A informação de uma câmera e vários sensores de toque são fundidas a fim de realizar um comparativo visual para aferição destas máquinas.

1.2 ESCOPO DA DISSERTAÇÃO

O Capítulo 2 tem como principal objetivo contextualizar o sistema proposto, o qual é aplicado em um projeto destinado a capacitar robôs móveis a execução de tarefas autônomas. Neste capítulo, a estrutura geral de um robô móvel, a arquitetura que será utilizada para integração dos diversos sub-sistemas do Axebot e a comunicação entre o sistema proposto neste projeto e o Axebot são apresentados.

Os Capítulos 3 e 4 representam o núcleo dessa dissertação. O Capítulo 3 trata do modelo geral de um sistema de visão artificial e três aspectos gerais são abordados: planejamento físico de sensores para robótica móvel, as principais metodologias de classificação/reconhecimento de padrões e os requisitos matemáticos para o classificador utilizado neste projeto. O Capítulo 4 discute o método de fusão de dados de sensores; duas arquiteturas comumente referidas na literatura são apresentadas e duas técnicas conhecidas são brevemente discutidas. A teoria geral da lógica difusa é abordada, com o objetivo de fundamentar a teoria aplicada no processador de fusão utilizado.

No Capítulo 5, os aspectos de implementação da arquitetura proposta são analisados. A arquitetura do sistema de visão artificial é apresentada e seus módulos são discutidos em seções. A análise dos resultados obtidos pelo classificador utilizado é realizada através de dados obtidos por experimentos práticos e o desempenho do classificador é avaliado. O algoritmo para a fusão de dados dos sensores e seus aspectos principais são apresentados e discutidos. As características e principais propriedades do sistema proposto, bem como o protótipo utilizado para testes, são descritos.

O Capítulo 6 realiza uma conclusão sobre o sistema proposto e resultados obtidos, bem como perspectivas para trabalhos futuros.

CAPÍTULO 2

CONTEXTUALIZAÇÃO

Este trabalho de dissertação é aplicável a robôs móveis autônomos ¹, particularmente, robôs que apresentam certo grau de inteligência.

A construção de tais robôs, aqui denominados RMAI e algumas vezes citados apenas como robôs móveis, apresenta um desafio especial devido a dificuldade de inserir, nas unidades autônomas, a tomada de decisão. Quanto mais dinâmico o ambiente, mais difícil torna-se o projeto de um RMAI do ponto de vista do processo decisório.

No que se refere a comportamento inteligente, existem alguns conceitos distintos aplicados a um RMAI: para alguns pesquisadores, este conceito está relacionado à emergência de tal fenômeno através do binômio percepção–ação (??). Para outros, a inteligência deve possuir uma representação ² e deve ser modelada como unidades de processamento de informações independentes (MARR, 1982).

A inteligência de um RMAI encontra-se implementada em seu centro de processamento e controle. Este módulo pode ser composto por um ou mais processadores e sistemas computacionais. É neste módulo que o robô obtém os dados dos sensores e controla os atuadores, conforme ilustrado na Fig. 2.1.

Os sensores têm como função fazer a interface entre o robô e o ambiente ao seu redor. A extração de informações dos sensores visa a construção de um modelo do ambiente para uma navegação adequada do robô. Dessa forma, quanto maior a robustez das informações providas pelos sensores, maior o grau de liberdade no processo decisório.

¹Compreende-se autonomia como as propriedades que permitem a execução de determinadas tarefas sem a necessidade de interferência externa durante o processo (HORN, 1986)

²Neste caso, entende-se por representação “um sistema formal a fim de produzir certas entidades explícitas ou tipos de informação, junto com uma especificação de como o sistema produz isso; como um exemplo o alfabeto permite uma representação escrita de palavras” (MARR, 1982)

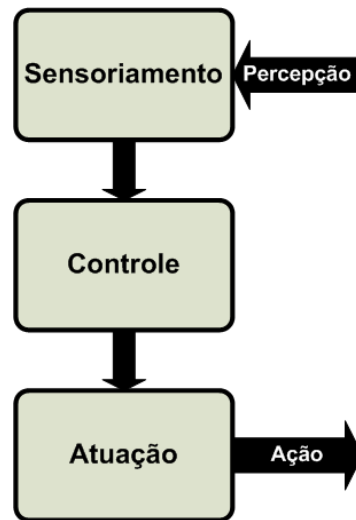


Figura 2.1. Estrutura de um robô móvel

Outro aspecto importante é que os sub-sistemas mecânico, eletrônico e computacional de um RMAI devem estar integradas de modo a permitir que este robô realize tarefas específicas. Assim, os objetivos de um determinado problema, onde se pretende utilizar a unidade robótica, devem sugerir as restrições físicas e computacionais de cada RMAI. Em vista disso, as seções seguintes abordam aspectos da área de aplicação do sistema de visão artificial proposto: a Seção 2.1 apresenta alguns detalhes sobre o ambiente de experimentação que definirá o escopo da arquitetura do sistema de visão e do projeto do robô; na Seção 2.2, a estrutura do agente concorrente para controle do robô, onde será aplicado o sistema de visão artificial, é descrita e a Seção 2.3 introduz, em linhas gerais, a proposta desta dissertação.

2.1 AMBIENTE DE EXPERIMENTAÇÃO

A Robocup é um projeto de integração internacional com o objetivo de promover as áreas de Inteligência Artificial, Robótica e áreas afins. O tópico central para o desenvolvimento deste objetivo é o Futebol de Robôs, visto que para um time de robôs participar de uma partida de futebol, várias tecnologias devem estar presentes: princípios de projetos de agentes autônomos, colaboração multi-agente, robótica, fusão

de sensores, entre outros (ROBOCUP, 2005).

Há cinco ligas principais dentro das competições da Robocup:

- **Robôs simulados** – times de robôs jogam futebol em um ambiente simulado em computador.
- **Robôs de pequeno porte (F-180)** – robôs com limite máximo de 18 cm de diâmetro.
- **Robôs de médio porte (F-2000)** – robôs com limite máximo de 50 cm de diâmetro.
- **Robôs quadrúpedes** – times de robôs de quatro pés (robôs AIBO da Sony).
- **Robôs humanóides** – robôs competem em partidas de "andar-chutar" ou disputas de penalidades máximas.

Os RMAIs deste projeto serão construídos para a liga de robôs F-180. Nesta liga, o sistema de visão artificial dos robôs pode ser global ou local. Na primeira modalidade, apenas uma câmera é utilizada para cada time. Um computador central é responsável pelo processamento da imagem e envio de comandos, através de rádio-frequência, para cada robô, conforme ilustrado na Fig. 2.2.

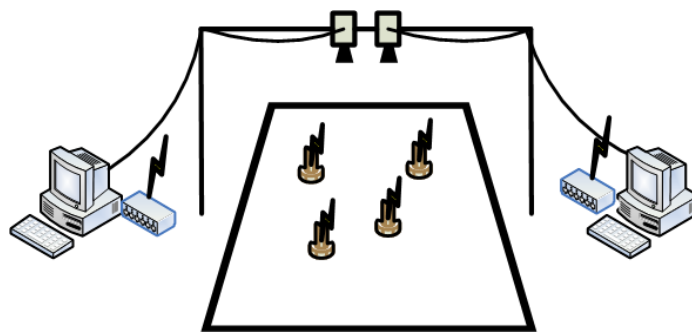


Figura 2.2. Esquema físico da liga F-180 utilizando visão global

Na modalidade de sistema de visão local, cada robô possui o seu próprio sistema de visão artificial e o processamento das informações de imagem pode ser realizado

localmente ou de forma distribuída. A Fig. 2.3 ilustra um exemplo de um robô com visão local.

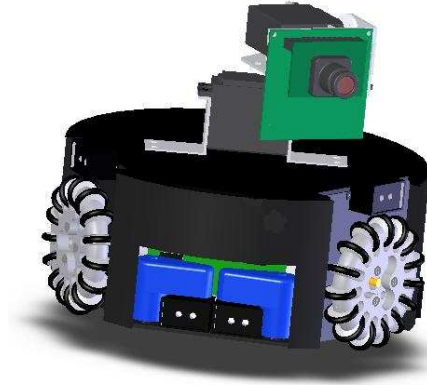


Figura 2.3. Robô móvel com visão local

Em sistemas de visão local, as definições para o reconhecimento dos objetos são determinadas a partir das seguintes regras da Robocup (ROBOCUP, 2004):

- **Quanto à bola** – esférica, de cor laranja, aproximadamente 46 gramas de massa e 43 mm de diâmetro.
- **Quanto aos robôs** – deve caber em um cilindro de 180 mm de diâmetro e altura máxima de 225 mm; esferas de 50 mm de diâmetro, designando cada time, e de cor azul ou amarela, devem estar no topo de cada robô; robôs podem usar as cores branca ou preta, sem restrição, porém o corpo desses robôs são usualmente de cor preta.

Em particular, o sistema de visão artificial proposto neste trabalho terá processamento local e embarcado em cada robô. O sistema proposto será aplicado ao projeto de um robô, denominado Axebot (AXEBOT, 2005), projeto desenvolvido pelo grupo de pesquisa Mecateam (MECATEAM, 2005), que será utilizado como base para o desenvolvimento de um time de futebol de robôs para as competições da categoria F-180 da Robocup (ROBOCUP, 2005).

2.2 ARQUITETURA DO AGENTE NO AXEBOT

Para a integração dos vários módulos que compõem o Axebot, propõe-se a utilização de um agente ³ autônomo concorrente, definido por Costa e Bittencourt (1999). Os processos que compõem o agente utilizam uma abordagem de programação *multi-thread* e são ilustrados na Fig. 2.4.

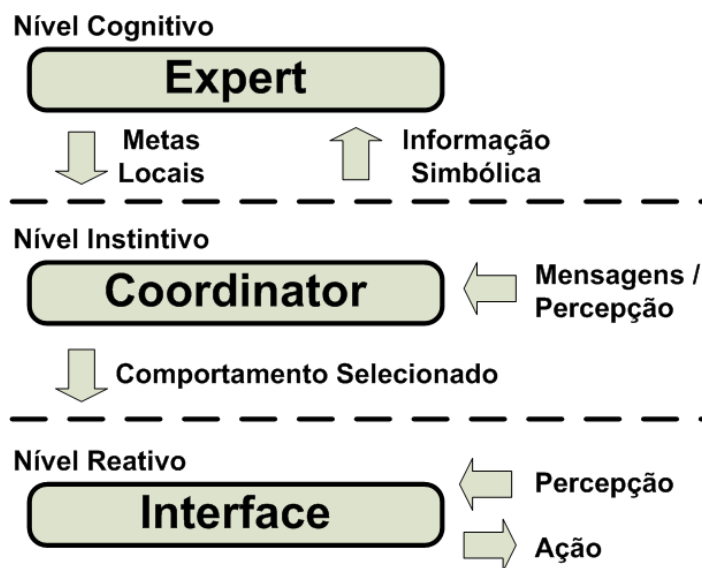


Figura 2.4. Arquitetura do agente autônomo concorrente (fluxo de informações)

O nível Cognitivo consiste em um sistema baseado em conhecimento simbólico que manipula tanto as informações recebidas do nível Instintivo quanto as mensagens assíncronas recebidas dos demais agentes, gerando metas locais e globais. A execução das metas locais do agente e a geração da informação simbólica para atualização da base de conhecimento do nível Cognitivo são implementados no nível Instintivo.

O nível Instintivo também é responsável por identificar os estados do ambiente e escolher o comportamento reativo mais adequado ao estado e a meta correntes.

Como o nível Reativo é responsável pelo processamento das mensagens de per-

³Compreende-se agente como uma entidade computacional, inserida em um dado ambiente, que possui a percepção deste ambiente através de sensores, possui a capacidade de raciocínio e age de forma autônoma através de atuadores, de forma a desempenhar uma dada função para a qual foi projetado

cepção do RMAI, este se torna o ponto de comunicação entre o agente e o sistema de visão artificial. O nível Reativo é ilustrado na Fig. 2.5 e é composto por controladores difusos, filtro de entrada, filtro de saída e *mailbox* ⁴.

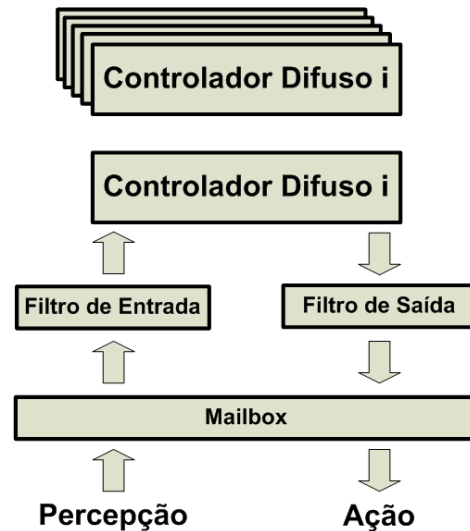


Figura 2.5. Nível reativo

O filtro de entrada extrai da informação visual difusa o valor das variáveis linguísticas utilizadas pelos controladores difusos, enquanto o filtro de saída verifica o valor das saídas do controlador difuso e as combina a fim de definir a ação para os atuadores. A inferência, neste nível, é realizada pelo processo de Interface a partir das informações contidas no *mailbox*. O processo de Inferência é responsável pelo controle dos elementos do nível Reativo e se encontra ortogonal ⁵ à Fig. 2.5

2.3 PROPOSTA

O objetivo deste trabalho de dissertação é a identificação dos objetos no ambiente do futebol de robôs. Além disso, tais objetos devem ser representados internamente com informações úteis para a tomada de decisão do RMAI. Uma vez geradas tais representações, estas são enviadas para o nível reativo do agente autônomo concorrente

⁴Todas as mensagens enviadas pelo sistema de visão artificial são armazenadas no *mailbox*

⁵Perpendicular ao plano da figura

embarcado no Axebot.

De acordo com a Fig. 2.6, primeiramente, as informações da câmera e dos sensores de distância são obtidas. Cada quadro de imagem é processado a fim de que os objetos, neste quadro, sejam identificados e seus atributos de decisão, extraídos. Em seguida, os atributos de decisão são fundidos com aqueles também extraídos a partir dos sensores de distância. Por fim, uma descrição de cada objeto, utilizando *quadro* (*frame*) (MINSKY, 1975), é gerada e enviada para o agente autônomo concorrente responsável por controlar o Axebot.

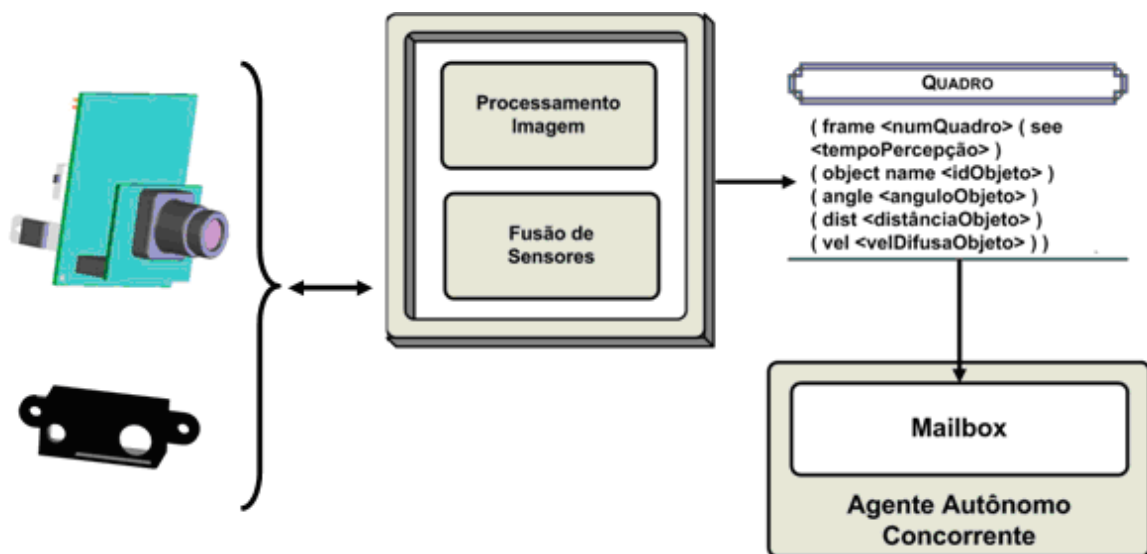


Figura 2.6. Integração entre o sistema de visão artificial e o agente autônomo concorrente que controla o Axebot

A abordagem para a fusão dos dados dos sensores é a lógica difusa. A escolha de tal abordagem se justifica por sua simplicidade e eficiência no processo de integração dos dados. Além disso, o conhecimento aplicado de um especialista, utilizando a lógica difusa, pode tornar o processo de fusão mais flexível diante da diversidade de situações do ambiente dinâmico do futebol de robôs.

2.3.1 Contribuições

As contribuições dessa dissertação podem ser listadas abaixo:

- Um sistema de sensoriamento de baixo custo computacional. O custo da identificação dos objetos em um quadro sensoriado é de aproximadamente 55 ms.
- Eficiência no reconhecimento e localização dos objetos. Sem a utilização de qualquer processamento de imagem prévio, a aplicação do conjunto de técnicas no sistema de sensoriamento robótico demonstrou eficiência tanto para o reconhecimento dos objetos caracterizados por cor (utilizando apenas uma amostra de treinamento), quanto para a determinação do ângulos destes objetos relativos ao robô.
- Um sistema de reconhecimento de objetos pouco sensível às condições de variação de luminosidade, através da utilização de um MVS no espaço de cor YCrCb.
- Simplicidade no processo de integração dos dados dos sensores. O conhecimento da disposição física e geometria dos sensores determina também como os seus dados serão integrados e os conjuntos e regras difusos são definidos.
- Construção do protótipo físico, o que que permite a validação das tarefas de reconhecimento e localização de objetos.

2.4 CONSIDERAÇÕES FINAIS

Neste capítulo, foram discutidos a área de aplicação e os elementos estruturais para a comunicação com o sistema de visão proposto. Inicialmente, o modelo geral de um RMAI é apresentado a fim de ilustrar as funções de seus módulos (sensoriamento, controle e atuação). Em seguida, as principais restrições do ambiente de experimentação de futebol de robôs servem de base para a compreensão do escopo deste trabalho. A arquitetura do agente autônomo concorrente proposto para o controle do RMAI é abordada e, por fim, a proposta deste trabalho é apresentada.

No Capítulo 3, as principais características de um sistema de visão artificial são discutidas com o objetivo de fornecer os fundamentos para o módulo de classificação

2.4 CONSIDERAÇÕES FINAIS

de padrões e extração de atributos de decisão da arquitetura proposta.

CAPÍTULO 3

SISTEMAS DE VISÃO ARTIFICIAL

A visão é considerada um dos sentidos mais completos do ser humano, pois provê uma vasta quantidade de informações indispensáveis para a interação do homem com o ambiente.

Atualmente, muitas atividades humanas utilizam sistemas de visão artificial. Sensores de visão juntamente com sistemas computacionais com funcionalidades específicas são capazes de extrair informações do ambiente a fim de capacitar máquinas para a tomada de decisão.

Na área de robótica móvel, os sistemas de visão são essenciais para garantir uma navegação segura do robô, a partir do reconhecimento e localização dos objetos, em um determinado ambiente.

Em geral, pode-se apresentar um modelo de sistemas de visão artificial conforme a Fig. 3.1.

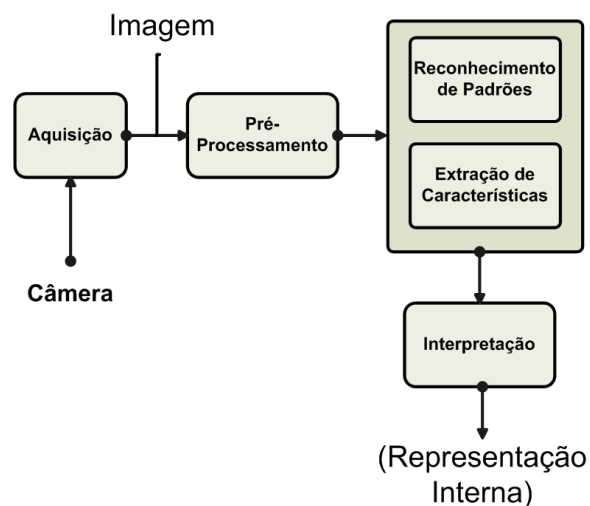


Figura 3.1. Modelo geral de sistemas de visão artificial

Inicialmente, os bits que representam cada quadro de imagem são obtidos através

do protocolo de comunicação com a câmera. Em seguida, a imagem é pré-processada com o objetivo de retirar ruído e efeitos indesejáveis usualmente associados à condição de iluminação. Um extrator de características ¹ é utilizado para reduzir o espaço da imagem em elementos significativos, e uma certa representação reduzida da imagem é utilizada a fim de auxiliar os algoritmos de classificação de padrões. Ao final do processo de percepção, uma representação interna dos objetos é fornecida pelo interpretador, o qual é responsável pela compreensão do ambiente e provisão de uma consciência descritiva dos objetos.

O limite conceitual entre o módulo de extração de característica e classificação de padrões é arbitrário. Muitos pesquisadores compreendem o extrator de características como um facilitador do processo de classificação e muitas vezes este é aplicado através de algum processo de segmentação da imagem ² (FANG; QIU, 2003; LAMPINEN; OJA, 1995; MITZIAS, 2004).

Em geral, o que diferencia os sistemas de visão artificial são os objetivos a serem alcançados em suas aplicações. Dessa forma, pode-se considerar dois paradigmas principais no que se refere à percepção de máquina: reconstrutivista e propositivo.

O paradigma reconstrutivista propõe que o principal objetivo de um sistema de visão é construir uma representação a partir da forma do mundo (JOLION, 1994; MARR, 1982; REALI et al., 1996). Este paradigma tem como principal precursor David Marr (MARR, 1982). Este defendia que a cena vista por um sistema de visão deveria ser totalmente reconstruída e propunha três níveis de projeto para isso (JOLION, 1994):

- **Teoria computacional** – definição dos papéis funcionais da estratégia adotada para reconstrução da cena;
- **Representação e algoritmo** – definição da estrutura da representação e dos algoritmos utilizados para a cena reconstruída;
- **Implementação no hardware** – realização física dos dois níveis anteriores.

¹Do Inglês, *feature extractor*

²Segmentação consiste em dividir a imagem em regiões que dizem respeito ao mesmo conteúdo de interesse na imagem

O conjunto das teorias do paradigma reconstrutivista contribui principalmente para o desenvolvimento de técnicas matemáticas para extração de características geométricas e estruturais de imagens (JOLION, 1994).

Por outro lado, surge o questionamento: “*É necessário a reconstrução de toda a cena para realizar determinadas tarefas?*”. Para alguns pesquisadores, a resposta é não. Segundo o paradigma propositivo, um sistema de visão deve ser construído com base nas tarefas que realiza e, portanto, baseado em propósitos específicos (JOLION, 1994; REALI; BIANCHI, 1996; RIVLIN; ALOIMONOS; ROSENFELD, 1991).

Considerando o dinamismo do ambiente de aplicação – futebol de robôs – e os requisitos temporais de resposta dos RMAIs nesse ambiente, este trabalho tem por base o paradigma propositivo.

As seções seguintes abordam temas relevantes para o projeto de sistemas de visão artificial. Na Seção 3.1, alguns requisitos para o planejamento dos componentes físicos para tais sistemas são apresentados. A Seção 3.2 descreve algumas abordagens utilizadas para a classificação de padrões, com o objetivo de auxiliar o projeto lógico do classificador. Na Seção 3.3, os requisitos matemáticos para a compreensão do classificador utilizado neste projeto são discutidos e analisados.

3.1 PLANEJAMENTO FÍSICO DE SISTEMAS DE VISÃO ARTIFICIAL

A área de planejamento de sensores de visão procura compreender e quantificar a relação entre objetos a serem vistos e os sensores, observando a adequação sensor-objeto segundo um modelo baseado na tarefa a ser realizada. Segundo Tarabanis, Allen e Tsai (1995), o objetivo de se planejar um sistema de sensoriamento visual é “gerar configurações apropriadas, baseadas em informações conhecidas previamente”.

Na processo de planejamento de sensores, diferentes técnicas são empregadas para o posicionamento e orientação da câmera, sendo possível determinar dois tipos de parâmetros a serem planejados:

- **Geométricos** – referentes à disposição dos sensores no espaço do robô;
- **Óticos** – referentes às características óticas dos sensores de visão que podem ser: aproximação (*zoom*), foco e abertura da câmera.

Sob o ponto de vista da configuração física (parâmetros geométricos), os sistemas de visão podem ser, em geral, classificados por: visão monocular, visão estereoscópica, visão panorâmica e fusão de dados da câmera e sensores de distância.

A Tabela 3.1 apresenta as principais características de cada uma das configurações físicas, suas vantagens e desvantagens.

Tabela 3.1. Configuração física de sistemas de visão artificial

Tipo	Vantagens	Desvantagens	Referências
Visão Monocular	-Baixo custo financeiro -Baixo custo computacional	-Ambiguidade de detecção 3D	(JOLION, 1994)
Estereoscopia	-Permite extração 3D	-Alto custo computacional	(FRANCA, 2003)
Fusão de dados	-Baixo custo financeiro -Baixo custo computacional	Depende dos sensores usados	(GU et al., 2002)

Os sistemas que utilizam somente uma câmera (visão monocular) possuem restrições quanto à extração de informações tridimensionais (3D). É possível inferir a distância entre robô e objeto, utilizando uma determinada função de interpolação entre a distância do objeto-robô e a altura do objeto na imagem, por exemplo. Contudo, qualquer tipo de oclusão ou interferência na extração de uma dessas características pode comprometer a qualidade da informação obtida.

Sistemas de visão estereoscópica podem ser compostos por duas ou mais câmeras (LIN et al., 2002; MAKI; NORDLUND; EKLUNDH, 2000; DORNAIKA; CHUNG, 2000). A extração de informações espaciais dos objetos no ambiente é realizada pela aplicação de

geometria epipolar (ZHANG, 1996). A determinação dessas informações é um problema mal-posto³ e, portanto, sujeito a restrições. Quanto mais câmeras, menor a quantidade de restrições que o sistema está sujeito, porém maior o custo computacional na extração de tais informações.

A fusão de dados de sensores de visão e de distância possui as vantagens das configurações anteriores. São sistemas que podem ser construídos com baixos requisitos computacionais. Apesar deste tipo de configuração ser dependente da qualidade dos sensores de visão e de distância, é possível minimizar tal problema através de técnicas adequadas (HOSEINNEZHAD; ASHARIF, 2002). A integração entre a câmera e os sensores de distância fornece duas vantagens intrínsecas: complementaridade entre as informações da câmera e os sensores de distância, e redundância entre as informações dos sensores em geral, aumentando a confiabilidade do sistema de sensoriamento.

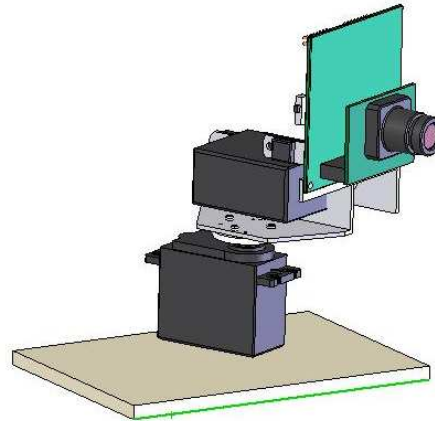
Os sistemas ainda podem ser classificados, quanto ao grau de movimentação, como: sistema de visão panorâmica ou sistema servo-acionado. O primeiro utiliza, em muitas aplicações, uma ou mais câmeras apontadas para um espelho côncavo, a fim de obter uma imagem total do ambiente (HUNDELSHAUSEN; BEHNKE; ROJAS, 2001; KROSE et al., 2004), extraindo da imagem refletida as informações necessárias; um exemplo desse tipo de sistema é ilustrado na Fig. 3.2(a). O segundo pode utilizar um ou mais servomotores (ou motores de passo) a fim de obter uma movimentação do sensor de visão em um ou mais eixos (Fig. 3.2(b)).

As características óticas estão relacionadas tanto à tecnologia do sensor de visão quanto às lentes utilizadas para a detecção do espectro de cor da imagem. A seção seguinte aborda alguns desses aspectos, bem como o princípio de aquisição de dados em cada tecnologia.

³Diversas soluções



(a) Exemplo de sistema de visão panorâmica. Figura retirada de (POLIUSP, 2001)



(b) Exemplo de sistema de visão artificial servo-acionada

Figura 3.2. Tipos de sistemas de visão artificial quanto ao grau de movimentação

3.1.1 Tecnologias de sensores de visão

O princípio básico de funcionamento de sensores de visão é a detecção do espectro de cor da imagem, a partir de uma matriz de detectores de luz analógicos, e a posterior conversão analógica-digital. Estes sensores formam pixels que são usualmente dispostos em linhas e colunas. A estrutura dos pixels varia em cada sensor, porém, o princípio de funcionamento básico é similar em todos os projetos, conforme será explicado mais adiante.

Um sensor de visão é constituído, essencialmente, de foto-detectores e microlentes. As microlentes determinam as características de aproximação e abertura da câmera.

Um ponto importante a ser observado é que todo sensor de imagem é um dispositivo que registra a intensidade de luz entre dois limites: iluminado (branco) e o obscuro (preto) (GROTTA, 2003). Para adicionar a informação de cor, uma camada de filtro de cor é aplicada na janela de entrada de luz desses detectores, conforme ilustrado na Fig. 3.3. Cada filtro permite a passagem apenas do espectro de cor referente àquele filtro.

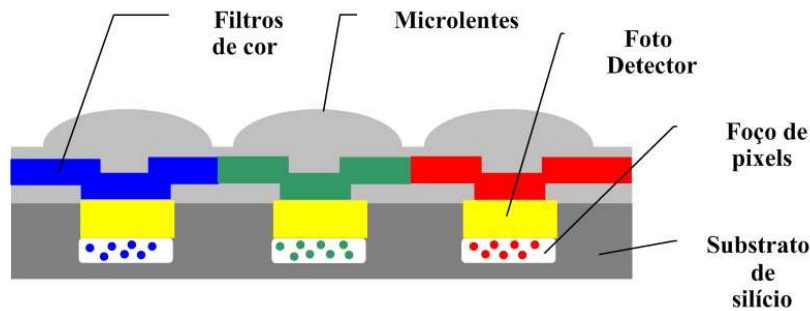


Figura 3.3. Tecnologia de um sensor de visão. Figura adaptada de (GROTTA, 2003)

As principais tecnologias de sensores digitais de imagem são: CCD (*Charge Coupled Device*) e CMOS (*Complementary Metal Oxide Semiconductor*). Inicialmente, estas tecnologias eram aplicadas à integração de dispositivos do estado sólido e, posteriormente, foram utilizadas em sensores de visão.

A diferença principal entre as duas tecnologias é a forma em que os elétrons são transferidos do sensor e o fato dos sensores tipo CMOS poderem incorporar mais funções em um único circuito integrado.

3.1.1.1 CMOS Um sensor de visão do tipo CMOS pode executar processamento da imagem, detecção de borda, redução de ruído e conversão analógico-digital, em um único circuito integrado. Esta característica de alta integração torna o preço de tais sensores menores que os baseados na tecnologia CCD.

A Fig. 3.4 mostra o esquema básico de um sensor de imagem do tipo CMOS. Conforme ilustrado, os pixels são endereçados por dois decodificadores de endereço (um mapeia as linhas e o outro as colunas da matriz), o que significa que cada pixel pode ser lido individualmente ou em coluna. Esta característica e o encapsulamento em um único circuito integrado, torna a tecnologia em questão mais veloz para a leitura da informação da imagem que a tecnologia CCD.

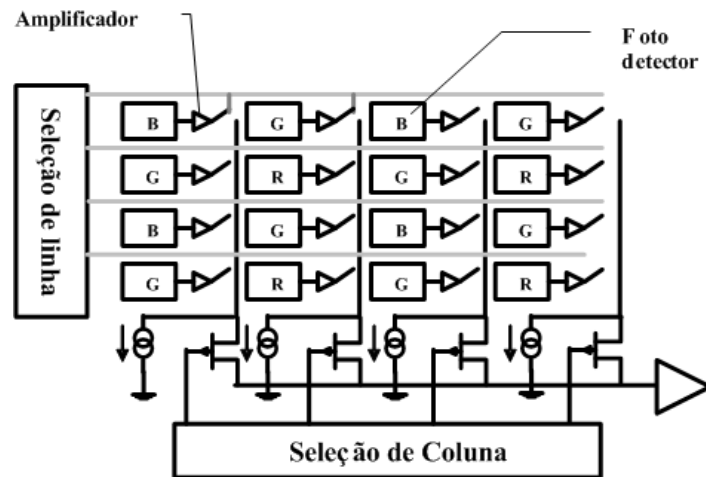


Figura 3.4. Estrutura de um sensor de visão com tecnologia CMOS. Adaptada de (GROTTA, 2003)

3.1.1.2 CCD A principal vantagem da tecnologia CCD, em relação à CMOS, é a qualidade da imagem obtida. Além disso, um sensor de imagem CCD para possuir as mesmas funções que um sensor CMOS, deve ocupar um maior volume.

A grande desvantagem da tecnologia CCD é o acesso aos dados obtidos da imagem, conforme ilustrado na Fig. 3.5. O endereçamento é feito por linha e, portanto, não permite uma compactação da imagem registrada como realizada pela tecnologia CMOS (GROTTA, 2003; FORSYTH; PONCE, 2003).

3.2 CLASSIFICAÇÃO DE PADRÕES EM IMAGEM

Classificação de padrões ⁴ é considerado um problema de inferência, ou seja, deve-se inferir informações úteis a partir do dado bruto (bits) (SHAPIRO; STOCKMAN, 2000).

Com relação a imagens, o problema é encontrar uma relação entre um conjunto de $M \times N$ pixels e Q classes, extraindo características e associando-as a cada classe $c_i \in Q$, $i = 1, \dots, Q$, onde c_i representa a i -ésima classe.

Os sistemas de sensoriamento que utilizam sensores de visão são muito eficientes

⁴Também denominado reconhecimento de padrões (SHAPIRO; STOCKMAN, 2000)

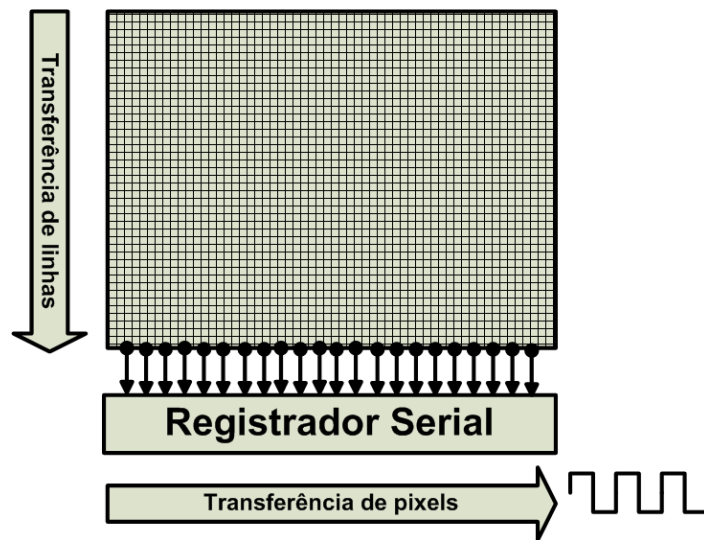


Figura 3.5. Estrutura de um sensor de visão com tecnologia CCD. Adaptada de (FORSYTH; PONCE, 2003)

devido a alta densidade de informação espaço-temporal que se pode obter, porém, mais complexos devido à necessidade de extração de dados a partir das imagens obtidas. Sob esse ponto de vista, para tais sistemas, um método adequado deve ser selecionado para a determinação dos padrões de imagens. Tal método deve ser escolhido a partir da análise do custo computacional e da ordem de complexidade dos algoritmos envolvidos, bem como a tarefa a que se propõe.

A fim de se obter uma classificação de padrões adequada para cada tarefa, Webb (2002) propõe um conjunto de critérios a serem analisados:

- **Formulação do problema** – obter uma clara compreensão dos pontos a serem investigados, planejando cada estágio;
- **Coleção de dados** – realizar medidas das variáveis apropriadas para a tarefa;
- **Extração de características** – selecionar as variáveis do conjunto de medidas e escolher o melhor método de classificação;
- **Classificação dos dados** – aplicar o método escolhido no estágio anterior a fim de determinar as classes;

- **Teste dos resultados** – neste estágio, um novo conjunto de dados deve ser aplicado ao classificador a fim de aferir a qualidade do método escolhido;
- **Interpretação** – obter uma *consciência descritiva* para as classes encontradas.

A escolha do melhor método de classificação de padrões deve residir na determinação do padrão apropriado para a tarefa proposta e da análise dos métodos propostos para cada padrão. As seções a seguir discutem estes aspectos. A Seção 3.2.1 formaliza o conceito de padrão em uma imagem e, na Seção 3.2.2, as abordagens mais comumente utilizadas para classificação de padrões em imagens são discutidas.

3.2.1 Padrões em imagem

Seja K a dimensão do espaço de entrada (imagem). Um padrão, numa imagem, pode ser representado como um vetor $X_i^k = (x_1^k, x_2^k, \dots, x_z^k)^T$ de medidas, onde $k = 1, \dots, K$ representa cada componente do espaço de entrada, enquanto x_i é a medida das características do i -ésimo objeto. O conjunto X_i^k representará as características do objeto as quais serão avaliadas no processo de classificação. Exemplos de padrões são: cor, textura ou forma. Cada um desses padrões deve ser extraído por métodos específicos.

As Fig. 3.6(a), 3.6(b) e 3.6(c) ilustram exemplos de imagens que utilizam padrões específicos.

Na Fig.3.6(a), os objetos são distinguidos pela cor. Nesse caso, um classificador que utilize o padrão de cor, reconhecerá os objetos sem ambiguidade. Na Fig. 3.6(b), o padrão de cor não seria o ideal para o classificador, visto que a cor do ambiente e do animal são semelhantes, porém a textura do animal o define unicamente. Na Fig. 3.6(c), o padrão forma pode ser utilizado. É possível também utilizar mais de um padrão para a separação do espaço da imagem.



Figura 3.6. Exemplos de padrões

3.2.1.1 Cor A descrição de uma cor λ , num espaço de cor, é dada a partir de três componentes das primitivas utilizadas para representá-la. Em outras palavras, a cor de um objeto $U(\lambda)$ pode ser representado pela soma ponderada de três comprimentos de onda, conforme a Eq. (3.1):

$$U(\lambda) = fc_1(\lambda_1)P_1 + fc_2(\lambda_2)P_2 + fc_3(\lambda_3)P_3 \quad (3.1)$$

onde P_i , $i = 1, 2, 3$ representam as primitivas em que uma cor pode ser definida e $fc_i(\lambda_i)$, $i = 1, 2, 3$ as funções de correspondências de cor ⁵ λ_i – que definem a correspondência das componentes de cor e sua radiância espectral ⁶.

Exemplos de espaços de cor usuais são: CIE XYZ, RGB, HSV. Dentre todos os espaços, o espaço de cor RGB (*Red*, *Green* e *Blue*) é, em geral, o mais conhecido e utilizado. Neste espaço, cada cor é definida a partir de três comprimentos de ondas primitivos: vermelho (645,16 nm), verde (526,32 nm) e azul (444,44 nm). Cada cor é representada por uma coordenada em um cubo de cor RGB, conforme ilustrado na Fig. 3.7.

3.2.1.2 Textura Em geral, texturas em imagens consistem de padrões organizados por subelementos regulares (algumas vezes denominados *textons*). Na Fig. 3.8, a

⁵Do Inglês, *color matching functions*

⁶Quantidade de luz emitida na faixa de seu comprimento de onda

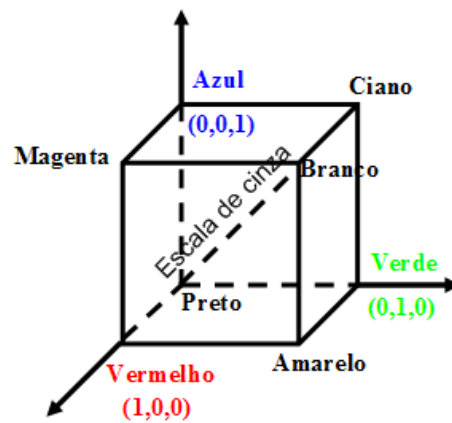


Figura 3.7. Representação do espaço de cor RGB

representação de uma textura é determinada por pequenos triângulos.



Figura 3.8. Exemplo de uma textura em imagem

3.2.1.3 Forma O padrão “forma” é representado em uma imagem através de seus componentes geométricos: linhas, curvas, superfícies, entre outros.

Tais padrões são comumente utilizados para reconstrução 3D dos objetos de uma imagem e são analisados através dos contornos dos objetos e suas inter-relações (FORSYTH; PONCE, 2003).

3.2.2 Métodos para classificação de padrões em imagem

Uma tarefa de classificação de padrão é determinar c_j classes às quais os componentes x_i^k de um conjunto de K objetos, com i componentes, podem pertencer. Um conjunto de pares $\tau = (x^k, c_j)$ é então fornecido pelo classificador.

No que se refere aos métodos utilizados para classificação de padrões em imagens e a partir da análise de alguns trabalhos (DUDA; HART; STORK, 2000; JAIN; DUIN; MAO, 2000; MACHADO, 1994; SHAPIRO; STOCKMAN, 2000; WEBB, 2002), pode-se categorizar três métodos: estruturais, especialistas e matemáticos. Cada um desses métodos será discutido nas seções seguintes.

3.2.2.1 Métodos estruturais No reconhecimento de padrões estruturais, uma entidade é representada por suas partes primitivas, seus atributos e as relações entre as partes. Estes métodos são utilizados quando não é suficiente representar uma característica por números ou símbolos. Um exemplo retirado de (SHAPIRO; STOCKMAN, 2000) é ilustrado na Fig. 3.9.

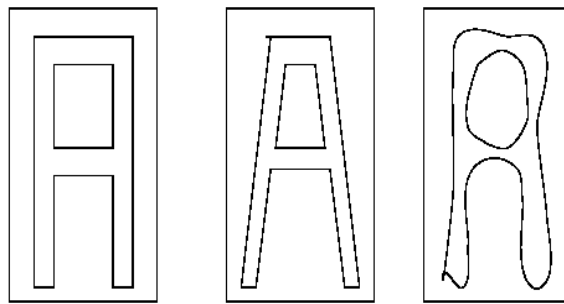


Figura 3.9. Letras “As” com 3 estruturas diferentes. Figura retirada de (SHAPIRO; STOCKMAN, 2000)

No exemplo da Fig. 3.9, pode-se observar três tipos de letras “A”. Apesar de estruturas semelhantes, o reconhecimento por um método numérico ou simbólico seria de difícil aplicação.

Em geral, em métodos estruturais, uma entidade é representada por um *grafo* de interligação dos elementos primitivos.

Na Fig. 3.9, por exemplo, observa-se as seguintes estruturas primitivas:

- **Buraco** – pode ser um lago (caso todo fechado) L ou uma baía B (caso aberto em algum trecho);
- **Traço** – linha horizontal ou vertical S .

Para essas primitivas, pode-se ainda definir os seguintes relacionamentos:

- **CON** – especifica a conexão entre dois traços;
- **ADJ** – especifica que uma região de um traço é imediatamente adjacente a um lago ou baía;
- **ACIMA** – especifica que um buraco está acima de outro.

A partir dessas definições, pode-se definir o grafo, ilustrado na Fig. 3.10, que representa cada “A” da Fig. 3.9.

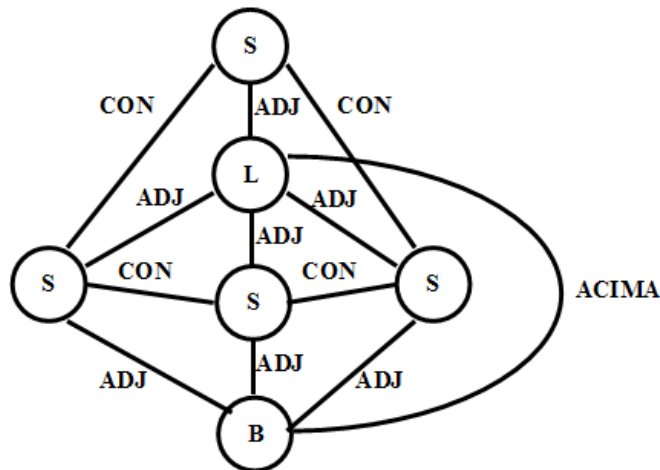


Figura 3.10. Grafo representando a relação das estruturas de uma letra A. Figura adaptada de (SHAPIRO; STOCKMAN, 2000)

O relacionamento entre duas primitivas pode ser considerado como uma característica atômica ⁷ do objeto e, portanto, utilizado em um vetor de características.

Alguns trabalhos utilizando métodos estruturais são encontrados em (ANISIMOVICH et al., 1997; KITADAI; NAKAGAWA, 2002; LIU; JAEGER, 2004).

3.2.2.2 Métodos especialistas Estes métodos utilizam o conhecimento de um especialista para o reconhecimento do padrão na imagem. Nesta categoria, encontram-se as técnicas: árvores de decisão, diagrama de influência, entre outros.

⁷Não pode ser mais decomposta em outros elementos estruturais

Com relação a árvores de decisão, seja um conjunto de treinamento τ e um conjunto de classes $C = c_1, c_2, \dots, c_n$. Uma árvore de decisão é construída a fim de determinar os objetos de τ que pertencem a um ou mais elementos de C . Para isso, os seguintes passos gerais podem ser seguidos:

- τ contém um ou mais exemplos todos pertencentes à mesma classe c_j . Nesse caso, a árvore conterá apenas um nó folha identificando c_j ;
- τ não contém exemplos. A árvore conterá apenas um nó folha com informações da classe utilizada pelo nó pai, por exemplo;
- τ contém exemplos que pertencem a várias classes. Nesse caso, τ é refinado em subconjuntos de exemplos que são cada um pertencentes a uma única classe c_j .

Uma árvore de decisão, portanto, determina todos os cenários de um problema de decisão (nesse caso, reconhecimento de padrões) e especifica a preferência que o sistema decisório atribui a cada cenário. Uma aplicação para reconhecimento de faces e imagens via satélite aplicando árvore de decisão pode ser encontrado em (BALA et al., 1995).

Com relação a diagrama de influências, este representa uma alternativa para árvores de decisão. Esta técnica possibilita o estudo das relações de causa e efeito entre os elementos de um problema, o que não ocorre nas árvores.

Um diagrama de influência é um grafo direcionado e acíclico $G = (N, E)$, sendo $N = P \cup D \cup V$ e E o conjunto de arcos. N é representado por:

- **Nós de probabilidade P :** cada nó N possui uma tabela de probabilidades condicionais, dada por $\phi_A = P(A|pa(A))$, onde $pa(A)$ representa um nó pai de A .
- **Nós de decisão D :** são pontos de escolhas de ações. Seus nós pais podem ser outros nós de decisões ou nós de probabilidade.

- **Nós de utilidade** U : são funções de utilidade. Cada nó possui uma tabela contendo a descrição da utilidade do decisor como função das variáveis associadas aos seus nós pais. Seus pais podem ser nós de decisão ou nós de probabilidade.

Para classificação de padrões em imagens, os diagramas de influência são comumente aplicados em forma de Redes de Bayes (MOGHADDAM; JEBARA; PENTLAND, 2000; SHAMAIE, 2004).

Outras técnicas que adotam conhecimento de especialista para classificação de padrões podem ser encontradas. Muitas destas são aplicações híbridas entre técnicas determinísticas e especialistas (KULKARNI, 2001; PAL; MANDAL; KUMAR, 1999).

3.2.2.3 Métodos matemáticos Os métodos matemáticos para classificação de padrões em imagem utilizam-se de funções de decisão para particionar (ou classificar) o espaço de entrada em classes. Estas funções podem ser de dois tipos: estatísticas ou determinísticas.

Os métodos que utilizam as funções de decisão estatísticas dão origem ao chamado reconhecimento de padrão estatístico (SHAPIRO; STOCKMAN, 2000; WEBB, 2002). Estes métodos ainda se dividem em dois: os baseados em funções de distância e os baseados em funções de similaridade (MACHADO, 1994). Para o primeiro tipo, as distâncias comumente empregadas são a euclidiana (DAHOT; CHARBONNIER; HEITZ, 2000) e a de Mahalanobis (KATO et al., 1999).

No segundo tipo, as funções de similaridade (ou fronteiras de decisão) são determinadas pelas distribuições de probabilidade de um padrão pertencente a cada classe (que podem ser aprendidas ou especificadas) (JAIN; DUIN; MAO, 2000; WEBB, 2002). A determinação da classificação do espaço de entrada é dada pela Eq. (3.2):

$$p(c_j|x_k) = p(c_j) \frac{p(x_k|c_j)}{p(x_k)} \quad (3.2)$$

onde c_j representa a j -ésima classe e x_k a k -ésima amostra extraída do espaço de

entrada.

Caso a probabilidade $p(c_j)$ seja conhecida, $p(x_k)$ é independente de c_j e, portanto, a classificação é definida pela estimativa da probabilidade de $p(x|c_j)$. Caso $p(c_j)$ não seja conhecida, então deve-se encontrar uma estimativa de c_j , baseada nas amostras x_k .

Para as funções de decisões determinísticas, 2 técnicas se destacam: Redes Neural Artificial (RNA) e MVS.

Uma RNA, em sua arquitetura básica, é formada por 2 camadas ⁸: camada escondida e camada de saída (HAYKIN, 2001, p. 47). Cada camada é composta por um conjunto de neurônios definidos por funções de ativação. Quando as funções de ativação da camada escondida são não-lineares, existem N neurônios na camada escondida e a camada de saída é linear, a RNA funciona como um aproximador universal.

Em geral, pode-se identificar três tipos de arquiteturas básicas para uma RNA: redes alimentadas adiante ⁹ com camada única, redes alimentadas adiante com múltiplas camadas e redes recorrentes. As duas primeiras são redes supervisionadas (ou seja, redes cujo processo de aprendizagem depende de um conjunto de treinamento) e as do último tipo são redes não-supervisionadas (não possuem treinamento prévio).

As redes neurais não supervisionadas não são utilizadas para a classificação de padrões em imagens, conforme conceito apresentado nesta seção. Estas redes são utilizadas para aplicações de agrupamento ¹⁰ (DUDA; HART; STORK, 2000). Portanto, não são de interesse para este trabalho. Nesta seção, apenas as RNAs supervisionadas são apresentadas.

A Fig. 3.11 ilustra a arquitetura básica de uma RNA com alimentação adiante e múltiplas camadas.

⁸Segundo Haykin (2001), o vetor de entrada não é considerado uma camada, pois não possui qualquer processo computacional

⁹Do Inglês, *feed forward*

¹⁰Do Inglês, *clustering*. Processo de separação de uma imagem em componentes similares mas sem identificação de um padrão, apenas um rótulo que define cada grupo (DUDA; HART; STORK, 2000)

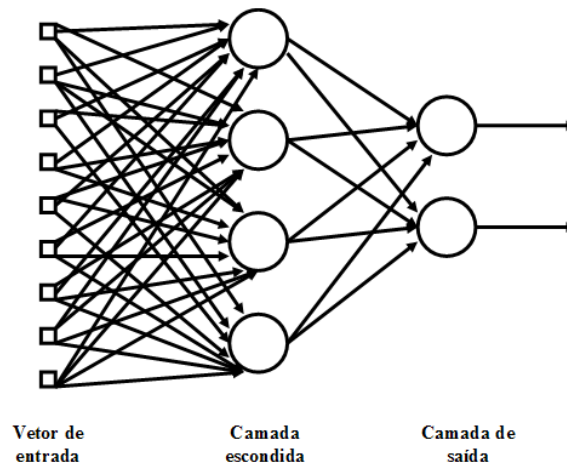


Figura 3.11. Modelo geral de uma Rede Neural Artificial

Conforme definido anteriormente, cada camada possui um ou mais neurônios. Em termos matemáticos a saída y de um neurônio i equivale ao produto interno entre o vetor de entrada X e o vetor de pesos sinápticos W mais um componente de polarização B ¹¹, conforme Eq. (3.3):

$$y_i = W \cdot X + B = \sum_{i=1}^n w_i x_i + b_i \quad (3.3)$$

onde i representa o i -ésimo neurônio.

O componente b_i realiza uma transformação *afim* à i -ésima saída y . O modelo de um neurônio é ilustrado na Fig. 3.12.

A interpretação geométrica da Eq. (3.3) é dada através de hipersuperfícies de separação a partir das amostras de treinamento.

Algumas aplicações de RNAs para classificação de padrões de imagens podem ser encontradas em (FERAUD et al., 2001; KURITA; TAKAHASHI; IKEDA, 2002).

¹¹Do Inglês, *bias*

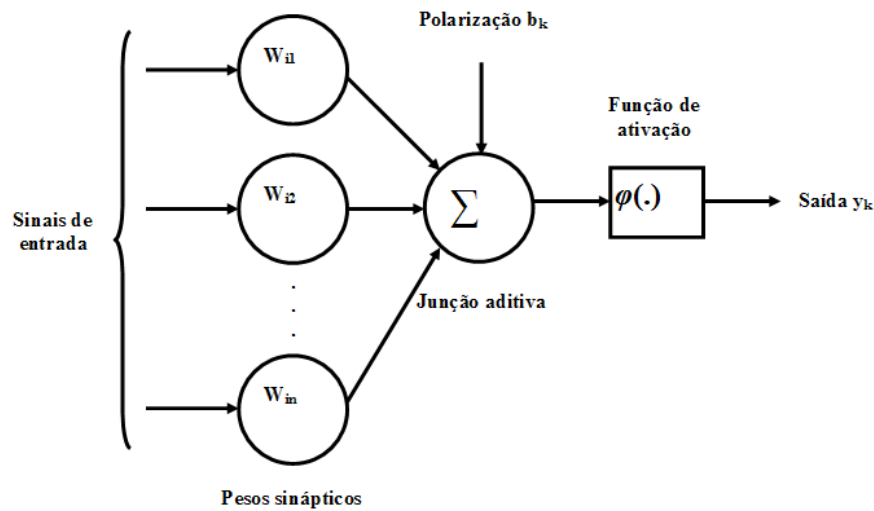


Figura 3.12. Modelo geral de um neurônio

3.3 REQUISITOS MATEMÁTICOS PARA O CLASSIFICADOR UTILIZADO

Visto que os objetos no campo de futebol de robôs são bem definidos através de cores distintas, um método estrutural seria custoso do ponto de vista computacional. Além disso, a aplicação de um método especialista poderia comprometer a generalização da tarefa de reconhecimento de padrões frente a variações de luminosidade (áreas de sombra, iluminação não uniforme, etc).

Dessa forma, o classificador adotado utiliza um método matemático, já que possui uma característica de generalização que deve ser bem explorada num ambiente dinâmico como o futebol de robôs. Dentre os métodos matemáticos, a escolha de um método determinístico sobre um estatístico é justificada pelo fato do primeiro não necessitar de um conhecimento da função de decisão para a classificação do espaço de entrada. Os métodos determinísticos possuem a facilidade de aprendizagem a partir de amostras de treinamento.

Entre os métodos determinísticos, a decisão é ponderada entre duas técnicas: RNA e MVS. Neste caso, a técnica de MVS é escolhida devido as considerações apresentadas a seguir.

A Tabela 3.2 ilustra as vantagens e desvantagens das RNAs segundo Kecman (2001).

Tabela 3.2. Vantagens e desvantagens de uma RNA

Vantagens	Desvantagens
Aprendizagem a partir de dados	Tempo de treinamento longo para grande volume de dados
Pode aproximar qualquer função não-linear multivariada	Tendência a <i>overfitting</i>
Não requerem um conhecimento profundo do problema sendo estudado	Pouco ou nada é oferecido sobre a estrutura de uma RNA ou o tipo de rede para um problema em particular
Robustas à presença de ruído	
Possui estrutura paralela e pode ser implementada em hardware	

As principais desvantagens das RNAs são a susceptibilidade a *overfitting*, ou seja, a possibilidade de uma rede não generalizar, e poucas definições sobre a estrutura desta para cada problema. Apesar de uma MVS possuir as mesmas vantagens de uma RNA, mostradas na Tabela 3.2, esta técnica reduz as chances de ocorrência das principais desvantagens de uma RNA (CRISTIANINI; SHAW-TAYLOR, 2003; KECMAN, 2001).

Em comparação a uma RNA, uma MVS é também mais robusta no que se refere a generalização devido ao fato de não possuir ocorrência de mínimos locais no processo de treinamento (solução de problemas de otimização convexos e quadráticos) (CRISTIANINI; SHAW-TAYLOR, 2003; KECMAN, 2001).

Uma MVS é uma técnica híbrida entre uma abordagem estatística e determinística. Isto significa que, para encontrar o melhor espaço de hipóteses de classificação, a MVS não utiliza uma distribuição de probabilidade conhecida *a priori*, mas determina uma função de probabilidade a partir do espaço de entrada.

3.3.1 Definição

MVSs são sistemas de aprendizado que utilizam um espaço de hipóteses de funções lineares em um espaço de características de dimensão maior, treinado (espaço de hipóteses) através de algoritmos de aprendizagem derivados da Teoria da Otimização que implementa uma polarização (*bias*) fornecida pela Teoria da Aprendizagem Estatística (CRISTIANINI; SHAW-TAYLOR, 2003).

A etapa responsável por encontrar as hipersuperfícies de separação, ou seja, aquelas que dividem um espaço de entrada, sempre define hipersuperfícies lineares (hiperplanos), assim como a resolução dos problemas de programação quadrática são os mesmos para espaços linearmente e não linearmente separáveis (ver seção 3.3.2). A diferença reside onde tais problemas são aplicados, seja no espaço de entrada, seja no espaço de características.

A figura 3.13 mostra um modelo geral para a utilização das MVSs.

3.3.2 Espaços linearmente separáveis

Seja $\tau = \{(x_i, d_i)\}_{i=0}^N$ uma amostra de treinamento, onde x_i é o padrão de entrada para o i -ésimo exemplo e d_i é a resposta desejada correspondente representada por dois valores distintos $\{+1, -1\}$ deste padrão. Considerando uma superfície de decisão dada pela Eq. (3.3) igual a zero, a amostra τ é linearmente separável se:

$$\begin{aligned} w_i x_i + b &\geq 0, & \text{para } d_i &= +1 \\ w_i x_i + b &< 0, & \text{para } d_i &= -1 \end{aligned} \tag{3.4}$$

A interpretação geométrica deste tipo de situação (hipótese) é que o espaço de entrada será particionado em dois, por uma hipersuperfície definida pela equação $w \cdot x + b = 0$. Tal hipersuperfície é um subespaço afim de dimensão $n - 1$, onde n representa a dimensão do espaço de entrada, e pode ser ilustrado conforme a figura

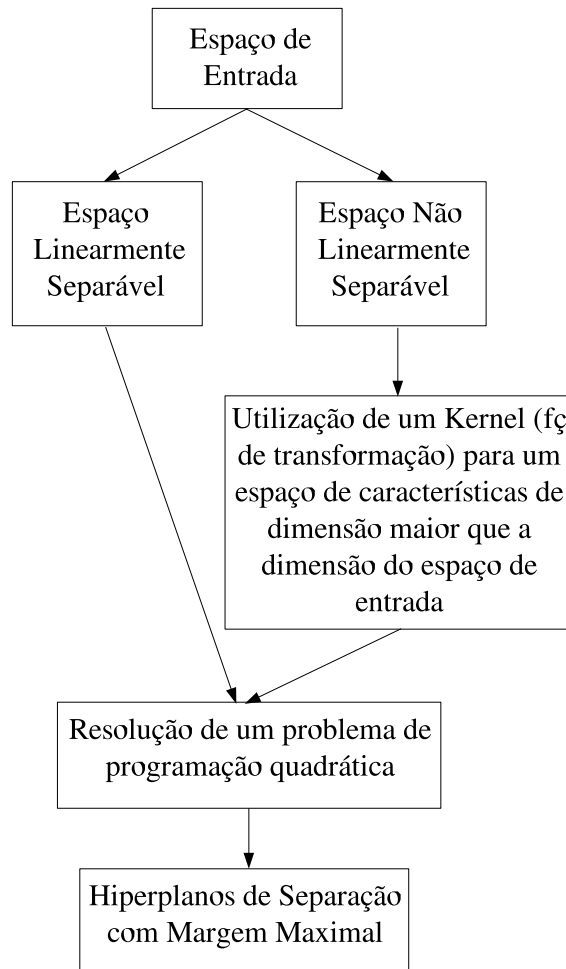


Figura 3.13. Modelo geral de uma Máquina de Vetor de Suporte

3.14, onde o vetor w define uma direção perpendicular ao hiperplano, enquanto o valor de b move o hiperplano de separação paralelamente a si mesmo.

A hipersuperfície de separação procurada é denominado de *discriminante linear*, e é determinada somente por uma fração dos dados de entrada, os chamados *vetores de suporte*.

MVS é uma técnica de treinamento supervisionado, ou seja, suas funções de discriminação lineares são determinadas a partir dos dados de entrada e suas respectivas saídas esperadas. Seja X o espaço de entrada e Y o espaço de saída, correspondendo às classes propostas, então para uma classificação binária $Y = \{-1, 1\}$, e para uma

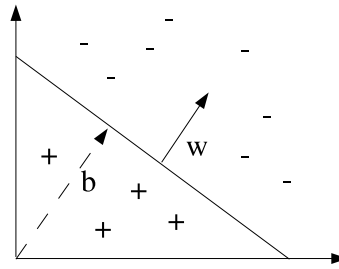


Figura 3.14. Hipersuperfície separadora (w,b) para um conjunto de treinamento bidimensional

multiclassificação $Y = \{1, 2, \dots, m\}$.

Os hiperplanos separadores do espaço de entrada não somente são obtidos, como também as margens maximais que separam tal espaço, através dos vetores de suporte \mathbf{o} , ilustrado na Fig. 3.15. Os vetores de suporte são definidos em pontos mais próximos do hiperplano de separação e denominados *margem de separação* γ . O objetivo de uma MVS é encontrar o hiperplano particular para o qual a margem de separação γ é máxima. Sob esta condição, a superfície de separação é referida como hiperplano ótimo de margem maximal.

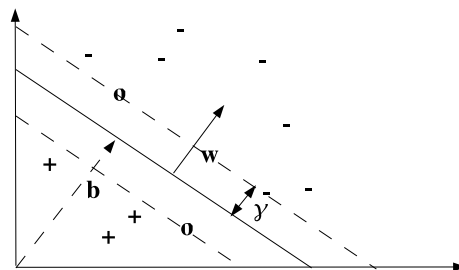


Figura 3.15. Problema de classificação com hiperplano de margem maximal

Na figura 3.15, a variável γ corresponde à máxima distância entre o hiperplano e sua margem e é obtido através da distância entre os vetores de suporte \mathbf{o} e o hiperplano obtido.

Para um problema de multiclassificação com o domínio de saída $Y = \{1, 2, \dots, m\}$, para cada uma das m classes é associado um vetor de entrada e um viés, (w_i, b_i) , com $i \in (1, \dots, m)$, e a função de decisão é dada, agora, por:

$$c(x) = \arg \max_{1 \leq i \leq m} (\langle w_i \cdot x \rangle + b_i)$$

Geometricamente, isto é equivalente a associar um hiperplano a cada duas classe.

3.3.3 Teoria da Otimização

A Teoria da Otimização aplicada à uma MVS provê as técnicas matemáticas necessárias para encontrar os hiperplanos que separam ou o espaço de entrada ou o espaço de características, otimizando as medidas de tais hiperplanos (CRISTIANINI; SHAW-TAYLOR, 2003).

Esta teoria provê não somente as técnicas algorítmicas, mas também define as condições necessárias e suficientes para uma função de custo dada ser uma solução para um problema quadrático convexo (CRISTIANINI; SHAW-TAYLOR, 2003). Esta etapa faz parte da etapa de treinamento de uma MVS.

O objetivo principal da formulação de problemas de otimização (para as MVSs) é encontrar o mínimo de uma função, tipicamente sujeito a algumas restrições. Assim, uma problema de otimização tem a seguinte forma geral:

Problema de Otimização Primal Dada as funções $f_i(\cdot)$, $g_i(\cdot)$, $i= 1, \dots, k$, e $h_i(\cdot)$, $i= 1, \dots, m$, definidas em um domínio $\Omega \subseteq R^n$,

$$\begin{aligned} \text{minimizar} \quad & f_i(w), & w \in \mathfrak{R}^n, \\ \text{sujeito a} \quad & g_i(w) \leq 0, & i = 1, \dots, k, \\ & h_i(w) = 0, & i = 1, \dots, m \end{aligned} \tag{3.5}$$

onde $f_i(\cdot)$ é chamado de *função objetiva* ou *função de custo* e as relações de restrição $g_i(\cdot) \leq 0$ e $h_i(\cdot) = 0$ são denominadas de restrições de desigualdade e igualdade, respectivamente. O valor ótimo da função objetiva é chamado de *valor do problema de otimização*.

Uma solução global de um problema de otimização é um ponto $w^* \in \mathfrak{R}^n$ tal que

não existem outros pontos $w \in \mathfrak{R}$ para os quais $f(w) < f(w^*)$, ou seja, $w^* \in \mathfrak{R}^n$ é um *mínimo global* para a função f .

Na fase de treinamento, um problema de otimização deve ser elencado, onde a função objetiva é uma função quadrática e suas restrições são lineares. Neste caso, o problema de otimização é chamado de *problema de otimização quadrática*. Além disso, as funções objetivas solucionadas são convexas, ou seja, para $w \in \mathbb{R}^n$ se, $\forall w, u \in \mathfrak{R}^n$, e para qualquer $\theta \in (0, 1)$, então:

$$f(\theta w + (1 - \theta)u) \leq \theta f(w) + (1 - \theta)f(u) \quad (3.6)$$

Neste caso, o problema de otimização é denominado convexo e quadrático (a função objetiva e suas restrições são convexas), e, portanto, só possui mínimos globais (uma demonstração matemática mais detalhada pode ser encontrada em (CRISTIANINI; SHAWE-TAYLOR, 2003; KECCMAN, 2001)).

É importante observar que para solucionar tais problemas de otimização de aprendizagem ¹², deve-se utilizar algumas ferramentas matemáticas para cada situação. Para os casos em que não há restrições de desigualdade, os multiplicadores de *Lagrange* são empregados a fim de obter o valor do problema de otimização (descartando as restrições); para os casos em que as restrições podem ser de desigualdades, o método de *Karush-Kuhn-Tucker* permite obter um problema de otimização equivalente, sem restrições de desigualdade, denominado problema de otimização dual. A relação entre encontrar as hiperplanos de separação e a solução dos problemas de otimização reside em:

Dada uma amostra de treinamento τ , o hiperplano (w, b) é encontrado a partir da solução do problema de otimização primal com a seguinte forma:

¹²Nesse caso, espera-se aprender a encontrar os hiperplanos que melhor separam os dados de entrada com maior margem maximal

$$\begin{aligned}
 \text{minimizar}_{w,b} \quad & \langle w \cdot w \rangle, \\
 \text{sujeito a} \quad & y_i(\langle w \cdot x_i \rangle + b) \geq 1 \\
 & i = 1, \dots, \ell
 \end{aligned} \tag{3.7}$$

Como ilustrado na Fig. 3.13, para o caso de um espaço não-linearmente separável, os vetores de entrada são transformados, primeiramente, para um espaço de características (através de uma função – *kernel*) e, portanto, o problema de otimização de hiperplanos neste espaço apresenta uma forma primal diferente da anterior:

$$\begin{aligned}
 \text{minimizar}_{w,b,\xi} \quad & \frac{1}{2}w^T w + C \sum_{i=1}^{\ell} \xi \\
 \text{sujeito a} \quad & y_i(K(x_i, x_j) + b) \geq 1 - \xi_i \\
 & \xi_i \geq 0
 \end{aligned} \tag{3.8}$$

onde C é o parâmetro de penalidade do termo de erro, ou seja, o parâmetro de parada de treinamento. O termo $K(x_i, x_j)$ representa o *kernel* que realiza o mapeamento para o espaço de características (descrito com maior detalhe na seção 3.3.4) e, ξ_i são as variáveis de folga para se determinar as margens maximais dos hiperplanos no espaço de características.

Uma MVS realiza apenas uma classificação binária, ou seja, obtém apenas um hiperplano linear que separa o espaço em dois. Na prática, algumas técnicas para encontrar múltiplos hiperplanos são utilizadas, tais como: um-contra-um, um-contra-todos, DAGSVM (HSU; LIN, 2002).

Os hiperplanos de separação são denominados hipóteses e formam um espaço de hipóteses H de funções lineares (hiperplanos), onde na fase de treinamento são encontrados, a partir da solução de problemas de otimização, a fim de criar um *modelo* para a fase de predição, onde as amostras de teste são classificadas.

3.3.4 Kernels

A utilização de funções que transformam o espaço de entrada para um espaço de característica de dimensão maior é a estratégia utilizada para lidar com dados não-linearmente separáveis. No espaço de características, a solução para os problemas de otimização do espaço de hipóteses se resume em encontrar hiperplanos. A transformação referida é realizada através de funções denominadas *kernels* e podem ser vistas como:

$$x = (x_1, \dots, x_m) \mapsto \phi(x) = (\phi_1(x), \dots, \phi_N(x)) \quad (3.9)$$

kernels transformam o espaço de entrada para um novo espaço, $F = \{\phi(x) | x \in X\}$. A partir do espaço F , a teoria de otimização é aplicada como em um espaço linearmente separável na forma de 3.8. A Fig. 3.16 ilustra esta idéia:

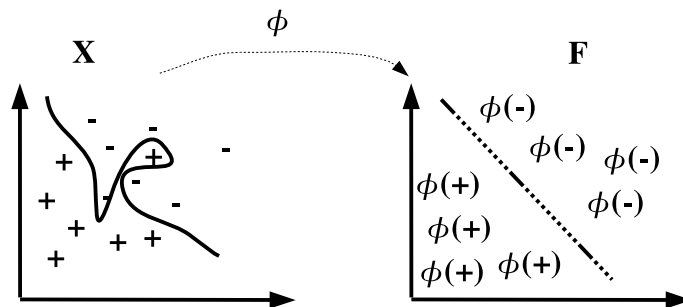


Figura 3.16. Mapeamento de um espaço de entrada não-linearmente separável para um espaço de características

Após o mapeamento, uma MVS linear pode ser aplicada na forma de:

$$f(x) = \sum_{i=1}^n \phi(w_i x_i) + b \quad (3.10)$$

Uma função é considerada um *kernel* se é uma função simétrica e obedece ao *Teorema de Mercer* (a apresentação e demonstração do Teorema de Mercer podem ser encontradas em (KECMAN, 2001; CRISTIANINI; SHAWE-TAYLOR, 2003)).

Os tipos de *kernels* mais utilizados em MVSs são (HSU; LIN, 2002):

1. linear: $K(x_i, x_j) = x_i^T x_j$
2. polinomial: $K(x_i, x_j) = (\gamma x_i^T x_j + r)^d, \gamma > 0$
3. RBF (Radial Basis Function): $K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2), \gamma > 0$
4. sigmóide: $K(x_i, x_j) = \tanh(\gamma x_i^T x_j + r)$

onde γ , r e d são parâmetros dos kernels. Como nas RNAs, a estrutura de uma rede pode se modificar de um problema para o outro, também os parâmetros do *kernel* devem ser ajustados em relação a cada conjunto de amostras de treinamento.

3.3.5 Princípio da minimização de risco

Diferentemente dos classificadores estatísticos clássicos, que se baseiam em distribuições previamente definidas antes do processo de classificação, as MVSs se baseiam na idéia de distribuição livre, ou seja, a distribuição é definida a partir dos dados de entrada.

Como mencionado anteriormente, MVS é uma técnica em parte determinística, em parte estatística, o que lhe concede algumas vantagens em relação a técnicas puras, tais como:

1. não ter a rigidez de distribuições estatísticas, ou seja, a natureza dos dados não precisa ser conhecida a priori;
2. a arquitetura das MVSs são bem definidas, bastando apenas que se encontre o *kernel* adequado e seus parâmetros ótimos para uma boa classificação (conforme será descrito no Capítulo 5);
3. o processo de se definir os hiperplanos de separação é realizado através de problemas de otimização convexos e quadráticos, o que evita o problema de mínimos locais, encontrados em RNAs.

MVS é uma técnica que teve sua origem no PMR (Princípio de Minimização de Risco) ¹³ e na teoria das fronteiras VC (*Vapnik–Chervonenkis*), o que originaram a Teoria do Aprendizado Estatístico (VAPNIK, 1995).

O PMR se baseia na idéia de que “aprendizagem pode ser considerada um problema de encontrar o melhor estimador f utilizando os dados disponíveis” (KECMAN, 2001). Uma medida apropriada deve ser encontrada a fim de se medir este “melhor estimador”.

Além disso, os dados de treinamento são formados por dois conjuntos: as variáveis de entrada x_i com probabilidade $P(x_i)$ e a resposta do sistema y_i , com probabilidade $P(y_i|x_i)$. Durante a fase de treinamento, y_i é denotado por d_i (valor desejado). Portanto, a probabilidade de se encontrar um par de pontos (x, d) será:

$$P(x_i, d_i) = P(x_i)P(y_i|x_i) \quad (3.11)$$

A resposta observada do sistema é probabilística, e isto é descrito pela probabilidade condicional $P(y_i|x_i)$ que define que para um mesmo x_i pode ser gerado um y diferente.

O que o PMR propõe é uma ferramenta estatística para determinar o erro entre d_i e y_i . Em outras palavras, explicando como e porquê os dados de treinamento são classificados, pode-se criar estratégias para minimizar os erros de classificação. Assim, pode-se determinar o risco esperado da função estimadora (conseguida pela resolução de um problema de otimização) através da Eq. (3.12):

$$R[f] = E[(y - f(x))^2] \quad (3.12)$$

O objetivo é encontrar o melhor elemento de $f(\cdot)$ que minimiza $R[f]$, através da esperança da diferença quadrática $(y - f(x))^2$

Depois de determinada a função de risco $R[f]$, pode-se afirmar que encontrar seu

¹³Compreende-se risco como a diferença entre valor desejado e o valor obtido

mínimo global (o risco mínimo) é um problema “mal-posto”, isto é, existem infinitas soluções. Para resolver este problema, o PMR propõe que o espaço de hipóteses H seja criado através de um processo de crescimento do espaço e do seu número de parâmetros (definidos como peso w), de modo que se forme a seguinte estrutura aninhada:

$$H_1 \subset H_2 \subset H_3 \subset \dots \subset H_{(n-1)} \subset H_n \subset \dots \subset H \quad (3.13)$$

Na estrutura 3.13, cada conjunto de funções H_i sempre contém as funções do conjunto anterior $H_{(i-1)}$. Neste caso, diz-se que o Princípio de Minimização de Risco é estrutural e pode-se observar, portanto, seu princípio indutivo. Em outras palavras, o que se espera é encontrar um modelo entre todos os H_i que represente minimamente os dados de entrada, com dimensão VC mínima.

3.3.6 Dimensão VC

Dimensão VC pode ser definida como: “o maior número de pontos h que pode ser separado de todas as maneiras possíveis” (KECMAN, 2001). Por exemplo, um conjunto de ℓ pontos, para um problema de classificação binária, pode ser classificado em 2^ℓ maneiras.

Para se determinar a dimensão VC de um espaço é necessário determinar também o número de *Vetores de Suporte*. Neste caso, encontrar os hiperplanos orientados, dentro do menor espaço de hipótese definido pela função de estimação de risco, que determinam a fronteira dos dados classificados.

A probabilidade de se classificar erroneamente os dados de teste, depois de encontrados os vetores de suporte, pode ser dada pela Eq. (3.14):

$$E_\ell[P(error)] \leq \frac{E[\text{nro. de vetores de suporte}]}{\ell} \quad (3.14)$$

onde E_ℓ denota a expectativa sobre o conjunto de dados de entrada de tamanho ℓ . Pela

Eq. (3.14), caso a MVS possua um número pequeno de vetores de suporte, possuirá também uma boa capacidade de generalização (HAYKIN, 2001; KEEMAN, 2001).

A dimensão VC de um conjunto de hiperplanos de separação em um espaço de dimensão m é igual a $m + 1$. Entretanto, para se aplicar o PMR, é necessário construir um conjunto de hiperplanos de separação de dimensão VC variável tal que o erro empírico e a dimensão VC sejam minimizados ao mesmo tempo.

Através da teoria da Dimensão VC, Vapnik (1995) demonstrou que é possível exercer controle sobre a dimensão VC do hiperplano ótimo, independentemente da dimensão do espaço de entrada, escolhendo adequadamente a margem de separação γ .

3.3.7 Espaços não linearmente separáveis

Para o tratamento das MVSs para espaços não-linearmente separáveis, o seguinte modelo é proposto (CRISTIANINI; SHAW-TAYLOR, 2003; KEEMAN, 2001):

1. Realizar um mapeamento do espaço de entrada para o espaço de característica através de uma função denominada *kernel*;
2. Resolução de um problema de otimização quadrática sobre o espaço de características a fim de se encontrar os hiperplanos de separação e, conseqüentemente, os vetores de suporte para a classificação dos dados;
3. Encontrar o modelo que melhor representa os dados de entrada com risco mínimo para as hipóteses encontradas (hiperplanos ou funções de discriminação)

Antes de seguir os passos propostos acima, deve-se verificar se o espaço de entrada é linearmente separável ou não, conforme será demonstrado a seguir. A partir da Eq. (3.4), deve-se encontrar os parâmetros (w_o, b_o) para o hiperplano ótimo, dado o conjunto de treinamento τ . O par (w_o, b_o) deve satisfazer a restrição 3.15:

$$\begin{aligned} w_o x_i + b_o &\geq 0, & \text{para } d_i &= +1 \\ w_o x_i + b_o &< 0, & \text{para } d_i &= -1 \end{aligned} \tag{3.15}$$

Caso a Eq. (3.4) seja válida, isto é, os padrões forem linearmente separáveis, é sempre possível escalar w_o e b_o de modo que a Eq. (3.15) seja válida.

3.4 CONSIDERAÇÕES FINAIS

Neste capítulo, aspectos de projeto de sistemas de visão artificial, metodologias de reconhecimento de padrões e os requisitos matemáticos para o classificador utilizado são abordados. As escolhas para o módulo de reconhecimento de objetos do sistema proposto são determinadas a partir da base teórica aqui apresentada.

No capítulo seguinte, os requisitos teóricos para a construção do módulo de processamento de fusão são analisados, bem como as arquiteturas e técnicas usualmente utilizadas na área de fusão de sensores.

CAPÍTULO 4

FUSÃO DE SENSORES

Fusão de dados de sensores, ou simplesmente fusão de sensores, representa o processo de combinação de dados ou informações de diversos sensores para estimativa ou predição de estados de entidades (objetos de medição) (HALL; LLINAS, 2001).

Tipicamente, alguns objetivos para diversos cenários de aplicações de fusão de sensores podem ser citados: detecção da presença de um objeto ou condição ambiental, reconhecimento de um objeto, rastreamento de objetos ou monitoramento de um evento continuado, combinação de informações para tomada inteligente de decisão, entre outros.

A combinação de dados de múltiplos sensores podem apresentar três vantagens imediatas:

- **Redundância** – dados provenientes de vários sensores semelhantes provêm N observações independentes e a combinação destas resultará em uma estimativa melhorada das medidas, assumindo que estes dados serão combinados de maneira adequada.
- **Cooperatividade** – a disposição relativa e a movimentação de múltiplos sensores auxilia no processo de inferência na observação. Por exemplo, dois sensores que medem direção angular de um objeto permitem determinar a posição espacial deste por triangulação em relação ao objeto.
- **Complementaridade** – dados provenientes de dois sensores diferentes podem ser combinados a fim de se obter uma estimativa mais completa do estado de uma entidade, especialmente quando esta estimativa não pode ser obtida por qualquer um dos sensores isoladamente.

Na área de fusão de sensores, o principal órgão que regulamenta a terminologia é o JDL (*Joint Directors of Laboratories Data Fusion Workgroup*). Este órgão foi estabelecido em 1986 e define um modelo de processos para fusão de dados, através de níveis de informação (Fig. 4.1), além de um dicionário de termos relativos a esta área de conhecimento (HALL; LLINAS, 2001).

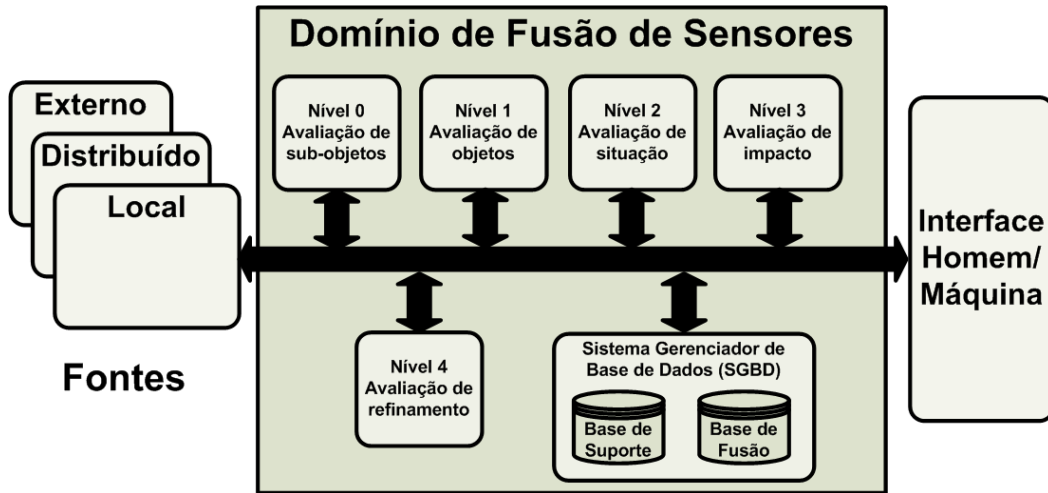


Figura 4.1. Modelo JDL de fusão de dados de sensores. Adaptado de (HALL; LLINAS, 2001)

Na Fig. 4.1, observa-se quatro níveis paralelos de processamento de informação:

- **Nível 0** – trata das informações em termos de dados (bits);
- **Nível 1** – é responsável pela estimativa e predição dos estados das entidades (medidas), baseados em inferências das observações;
- **Nível 2** – a estimativa e predição são realizadas através das relações entre as entidades;
- **Nível 3** – os impactos das estimativas anteriores são medidos;
- **Nível 4** – ocorre o refinamento do processo para realização dos objetivos.

Os níveis mencionados anteriormente podem aparecer isoladamente ou em conjunto. De fato, cada nível N pode conter um processo de refinamento do nível $N - 1$, porém um nível N também pode ser aplicado isoladamente para um problema ou

situação específica. Portanto, com base no modelo proposto, o JDL determina três alternativas para métodos de fusão de dados:

- **Fusão direta dos dados:** os dados são retirados dos sensores e integrados diretamente a partir das denominadas técnicas clássicas de estimativa, como por exemplo, filtro de Kalman (DROLET; MICHAUD; COTE, 2000);
- **Fusão de características:** representação dos dados dos sensores através de um vetor de características (objetos identificados e seus atributos), com subsequente fusão desse vetor. Geralmente são utilizadas técnicas para extração de características a partir dos dados dos diversos sensores e estes dados são então combinados em um único vetor;
- **Fusão de decisões:** processamento de cada sensor a fim de se obter inferências ou *decisões*¹ de alto nível (como identificação de um objeto e sua localização), com subseqüentes combinações. Técnicas de votação, inferência clássica, inferência Bayesiana (SINGHAL; BROWN, 1997), técnica de Dempster-Shafer (WU et al., 2002; WU; SIEGEL; ABLAY, 2003), lógica difusa, entre outras, podem ser citadas.

Após essa primeira abordagem proposta pela JDL, outras são apresentadas como uma evolução: a Seção 4.1 descreve estas arquiteturas. Em seguida, na Seção 4.2, algumas técnicas de fusão de dados de sensores são apresentadas. Por fim, na Seção 4.3, a técnica utilizada neste projeto é abordada.

¹A palavra decisão é utilizada aqui não como atuação, mas como inferência de informações

4.1 ARQUITETURAS

Modelos ² e arquiteturas ³ para fusão de dados de sensores têm sido caracterizados pela literatura sob diversas formas, em termos de: níveis de informação a qual a fusão é aplicada, objetivos do processo de fusão, domínio de aplicação, disposição dos sensores, entre outros.

Dentre diversos trabalhos sobre arquiteturas para fusão de sensores, dois se destacam como modelos genéricos de fusão de sensores: um trabalho proposto por Bruder, Faronq e Bayoumi (1991) e outro por Dasarathy (1997).

Bruder, Faronq e Bayoumi (1991) propõem quatro arquiteturas de fusão de sensores quanto ao alinhamento temporal, a partir da análise de diversas aplicações na área:

- **Fusão centralizada** – possui um único processador para integração das informações dos sensores;
- **Fusão descentralizada** – para cada sensor existe um processador e todas as informações são integradas por uma linha de tempo definida pelos tipos de sensores;
- **Fusão sequencial** – tipo de fusão descentralizada, porém os dados de cada processador são obtidos de forma sequencial, sem a necessidade de uma linha de tempo;
- **Fusão estática** – tipo de fusão descentralizada, porém os dados obtidos de cada processador são somados através de uma ponderação.

As configurações acima possuem uma relação intrínseca com os tipos de sensores envolvidos no processo de fusão. Caso o sistema de sensoriamento seja composto so-

²Descrição abstrata de um conjunto de funções ou processos que podem ser componentes de um sistema, sem indicação de software ou implementação física (HALL; LLINAS, 2001)

³Estrutura de componentes, seus relacionamentos e os princípios e guias que regem seus projetos e evolução no decorrer do tempo (HALL; LLINAS, 2001)

mente por sensores passivos ⁴, ou um sensor ativo ⁵ e N sensores passivos, a fusão centralizada é a mais adequada. No caso de somente sensores ativos, a fusão descentralizada (seja esta pura, sequencial ou estática) é a mais comumente utilizada. No segundo caso, a integração dos dados dos sensores é mais complexa, devido à necessidade de alinhamento temporal.

Dasarathy (1997) apresenta um trabalho sobre uma nova proposta de arquitetura para fusão de sensores. Esta arquitetura é baseada em seis categorias de processos de fusão:

- **(DaE – DaS), entra dado – sai dados**
- **(DaE – CaS), entra dado – sai característica**
- **(CaE – CaS), entra característica – sai característica**
- **(CaE – DeS), entra característica – sai decisão**
- **(DeE – DeS), entra decisão – sai decisão**
- **Fusão temporal**

As categorias propostas por Dasarathy (1997) se baseiam nos três níveis hierárquicos da JDL (dados, característica e decisão), porém, são utilizadas como modos de processamento de entrada/saída. A partir dos níveis anteriores, Dasarathy (1997) propõe uma arquitetura flexível, ilustrada na Fig. 4.2.

Pode-se observar que a última categoria – fusão temporal – não aparece na Fig. 4.2. Isto se deve, pois este tipo de fusão de sensores trata da integração de dados no decorrer de um período de tempo, conforme apresentado no trabalho de Bruder, Faronq e Bayoumi (1991) e, portanto, deve aparecer como um componente ortonormal à arquitetura proposta.

⁴Sensores que não fornecem retroalimentação de informação, exemplo GPS (*Global Positioning System*)

⁵Sensores que fornecem retroalimentação, exemplo sensores de distância

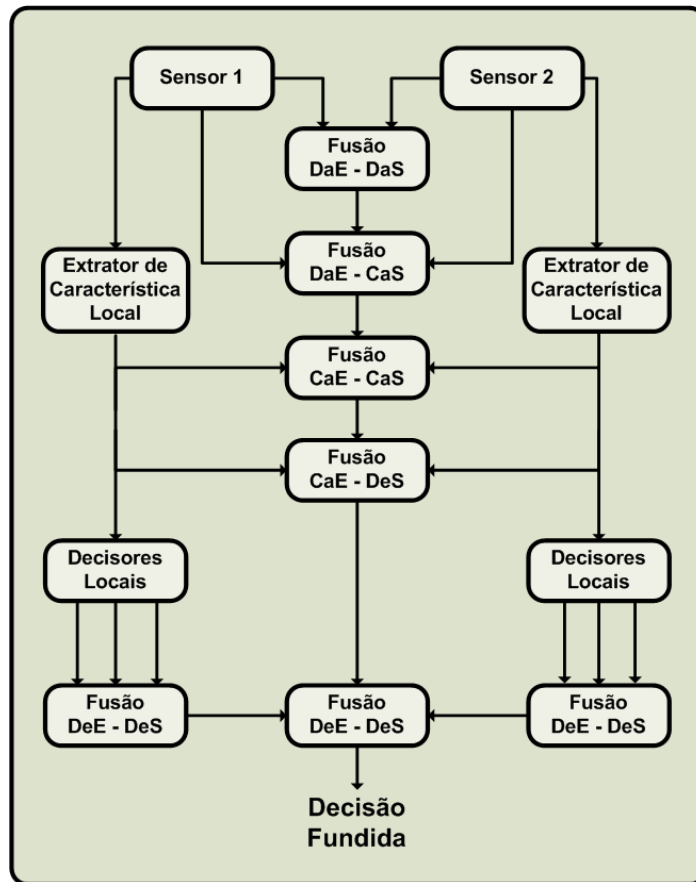


Figura 4.2. Arquitetura flexível de fusão de sensores. Adaptado de (DASARATHY, 1997)

4.2 TÉCNICAS DE FUSÃO DE SENSORES

Historicamente, as técnicas na área de fusão de sensores residem inicialmente no campo militar e, somente algum tempo depois, têm seu desenvolvimento associado às aplicações civis, principalmente na área da robótica (HALL; LLINAS, 2001).

Dentre as principais técnicas, pode-se destacar três: filtro de Kalman, Dempster-Shafer e lógica difusa. A primeira é comumente utilizada para fusão direta de dados dos sensores; as duas últimas são mais adequadas para fusão de sensores cuja entrada é decisão. A seguir, alguns aspectos destas técnicas são apresentados.

4.2.1 Filtro de Kalman

Filtro de *Kalman* é uma forma de estimativa caracterizada por uma avaliação dinâmica e um modelo interno do sistema estimado.

Entre as alternativas do JDL, esta técnica é usualmente empregada para fusão direta de dados de sensores. O filtro de *Kalman* provêm da teoria estocástica de sistemas dinâmicos e pode ser utilizado para melhorar a qualidade das estimativas de quantidades desconhecidas. A partir dos dados de vários sensores, uma nova estimativa é encontrada, filtrando as estimativas prévias a fim de determinar uma previsão melhorada.

Aplicações de filtro de *Kalman* para fusão de sensores podem ser encontradas em (ENRIQUEZ; ROBLES, 2004; LOY et al., 2002): no primeiro, uma aplicação da técnica é utilizada para rastreamento de objetos em situação de oclusão. No segundo, é apresentado um algoritmo para fusão de múltiplos sensores em um veículo subaquático controlado remotamente, com o objetivo de tornar a navegação do veículo adaptável a diferentes situações de obstáculos.

4.2.2 Dempster-Shafer

A teoria de *Dempster-Shafer* é considerada uma generalização da teoria *Bayesiana*, a qual é comumente utilizada para inferência estatística. Esta técnica permite suporte distribuído para proposições simples e união de proposições. Em um sistema de *Dempster-Shafer*, todos os eventos mutuamente exclusivos do mesmo tipo são enumerados em um quadro de “discernimento” Θ .

Cada sensor S_i , contribuirá com uma observação através da definição de sua crença sobre Θ . Esta função de definição é denominada “função de reunião de probabilidade”⁶ do sensor S_i , denotada por m_i . Assim, de acordo com a observação do sensor S_i , a

⁶Do Inglês, probability mass function

probabilidade que um objeto detectado seja O é indicada pelo intervalo:

$$[\text{Crença}_i(O), \text{Plausabilidade}_i(O)] \quad (4.1)$$

O limite inferior do intervalo é a confiança da crença, que é definida por todas as evidências E_k que suportam a proposição “objeto O ” e dada pela Eq. (4.2):

$$\text{Crença}_i(O) = \sum_{E_k \subseteq O} m_i(E_k) \quad (4.2)$$

O limite superior do intervalo é a confiança da plausabilidade, que é definida por todas as observações m_i que não suportam a proposição ”objeto O ”, dada pela Eq. (4.3):

$$\text{Plausabilidade}_i(O) = 1 - \sum_{E_k \cap O = \emptyset} m_i(E_k) \quad (4.3)$$

Para cada possível proposição “objeto O ”, a técnica fornece uma regra de combinação \oplus das observações m_i com evidência E_k dos sensores S_i e observações m_j com evidências $E_{k'}$ dos sensores S_j , dada pela Eq. 4.4:

$$(m_i \oplus m_j)(O) = \frac{\sum_{E_k \cap E_{k'} = O} m_i(E_k)m_j(E_{k'})}{1 - \sum_{E_k \cap E_{k'} = \emptyset} m_i(E_k)m_j(E_{k'})} \quad (4.4)$$

A Eq. 4.4 pode ser tratada também de forma recursiva, ou seja, m_j não sendo uma observação do sensor S_j mas uma combinação de observações de outros dois sensores S_l e S_p .

Pode-se observar, portanto, que há uma tentativa de refinamento das observações dos sensores através da combinação de suas crenças para uma dada proposição. Visto que a fusão ocorre depois que cada sensor realiza uma determinação preliminar da identidade do objeto, sua localização e atributos, esta técnica é adequada para proces-

sadores de fusão cuja entrada é decisão (WU et al., 2002; WU; SIEGEL; ABLAY, 2003).

4.3 LÓGICA DIFUSA

Como mencionado no início deste capítulo, o objetivo da fusão de sensores é obter redundância e/ou complementaridade a partir da integração dos dados de diversos sensores. Esta tarefa de integração possui, em geral, a dificuldade do tratamento de incerteza e regiões de ponderação na combinação dos dados sensoriais.

Dentre as diversas técnicas disponíveis na literatura, optou-se pela lógica difusa devido a simplicidade no tratamento de tais incertezas no processo de fusão, além de possuir fácil implementação algorítmica e baixo custo computacional.

O processador de fusão proposto é baseado em lógica difusa, visto que permite a modelagem do raciocínio humano no processamento e na integração dos dados dos sensores, tornando assim a tarefa de fusão de sensores flexível em regiões de ponderação.

Os principais elementos desta técnica são discutidos a seguir.

4.3.1 Definição

Seja um conjunto ordinário Z representando um *universo de discurso*, definido por um conjunto de elementos $z \in Z$ e um conjunto A , tal que $A \subseteq Z$. O elemento z pode ser avaliado pela Eq. (4.5):

$$\mu_A(z) \equiv \begin{cases} 1 & \iff z \in A \\ 0 & \iff z \notin A \end{cases} \quad (4.5)$$

A função μ_A é denominada *função característica* e avalia a pertinência de $z \in Z$ ao conjunto A . O par $(0, 1)$ é denominado de *conjunto de avaliação*. Portanto, o conjunto A pode ser definido como um par $(z, 0)$ ou $(z, 1)$, indicando $z \notin A$ ou $z \in A$. A Eq.

(4.5) pode ser então reescrita como a Eq. (4.6):

$$\mu_A(z) : Z \rightarrow 0, 1 \quad (4.6)$$

A lógica difusa provém de uma generalização da teoria dos conjuntos ordinários e tem suas origens no trabalho de (ZADEH, 1965). No lugar do conjunto de avaliação, tem-se agora um intervalo de avaliação $[0, 1]$ e a Eq. (4.6) é modificada para definir não mais uma função de característica, mas uma *função de pertinência*:

$$\mu_A(u) : U \rightarrow [0, 1] \quad (4.7)$$

onde um conjunto A definido em U , universo de discurso, é definido como $\{(u, \mu_A(u))\}$, onde cada par $(u, \mu_A(u))$ é denominado de *singleton*⁷.

Para U discreto, então A pode também ser definido como na Eq. (4.8):

$$A = \sum_{u_i \in U} \mu_A(u_i)/u_i \quad (4.8)$$

onde \sum indica a união de todos os *singletons* e “/” é apenas um marcador.

Para U contínuo, A é então definido como na Eq. (4.9):

$$A = \int_U \mu_A(u_i)/u_i \quad (4.9)$$

onde \int indica a união de todos os *singletons* sobre o conjunto contínuo X .

4.3.2 Teoria geral

Em geral, um sistema difuso possui o modelo ilustrado na Fig. 4.3. Um elemento do universo de discurso U (valor de entrada) é *fuzificado*⁸, ou seja, através das funções

⁷Esta palavra não possui equivalência na língua portuguesa

⁸Do Inglês, *fuzzified*

de pertinência de cada conjunto difuso, um valor preciso é transformado em um valor difuso. Em seguida, um mecanismo de inferência é aplicado a fim de encontrar um valor difuso baseado nos valores das regras ativadas para aquele valor preciso. Por fim, o valor encontrado através do processo de inferência é *defuzificado*⁹, ou seja, transformado novamente em um valor preciso.

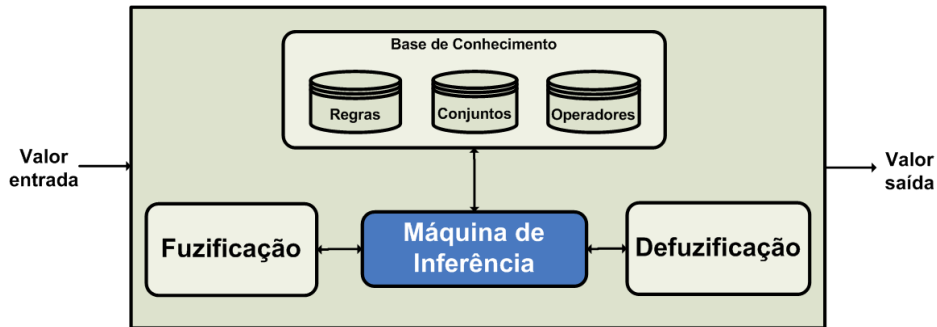


Figura 4.3. Modelo geral de um sistema difuso

A base de conhecimento é formada por: regras, conjuntos difusos e operadores. As regras são determinadas a partir de variáveis linguísticas que são compostas por conjuntos difusos, os quais são definidos por funções de pertinência. Uma regra difusa possui a seguinte forma geral:

$$\text{SE } x = A_i \quad \text{ENTÃO } y = B_i \quad (4.10)$$

onde A_i e B_i são conjuntos difusos, no eixo x e y , respectivamente. $x = A_i$ é denominado de antecedente e $y = B_i$ de conseqüente da regra.

A regra em (4.10) é *disparada* através de operadores lógicos. Em uma regra, os operadores podem ser: complemento, união ou intersecção. Os operadores de intersecção (E-lógico) são denominados *T-normas* e de união (OU-lógico) *T-conormas* ou *S-normas*. As Eqs. (4.11), (4.12) e (4.13) apresentam exemplos de operadores de complemento, união e intersecção, respectivamente, para dois conjuntos difusos A e B :

⁹Do Inglês, *defuzzified*

$$\mu_A^C(x) = 1 - \mu(x) \quad (4.11)$$

$$\mu_{A \vee B} = \text{MAX}(\mu_A, \mu_B) \quad (4.12)$$

$$\mu_{A \wedge B} = \text{MIN}(\mu_A, \mu_B) \quad (4.13)$$

Deve-se observar também que N regras distintas são avaliadas através de um operador de união. Além disso, no antecedente de uma regra, mais de um conjunto difuso pode ser avaliado por um operador de união ou de intersecção.

A estrutura de uma variável linguística é ilustrada na Fig. 4.4.

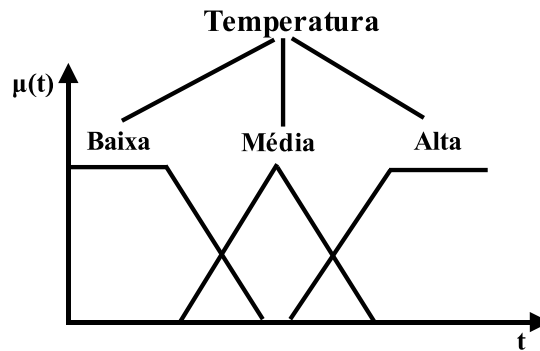


Figura 4.4. Estrutura de uma variável linguística

A variável linguística ilustrada na Fig. 4.4, denominada temperatura, pode ser definida como um tupla $(V, U, R(V, u))$, onde V é o nome da variável, U é o universo de discurso, u é um elemento genérico de U e $R(V, u)$ é um subconjunto de U que representa uma restrição sobre os valores de u imposta por V (ZADEH, 1975).

4.3.3 Sistemas de inferência

Um sistema de inferência difuso é responsável pela ativação das regras adequadas para cada valor de entrada.

Três modelos gerais de mecanismos de inferência podem ser apresentados: modelo de Mamdani, modelo de Takagi-Sugeno-Kang (TSK) e modelo de Tsukamoto. A principal distinção entre estes modelos reside nos consequentes das regras e, por conseguinte, nos procedimentos de defuzzificação ¹⁰.

O modelo de inferência Mamdani foi proposto em (MAMDANI; ASSILIAN, 1975), como tentativa de controlar uma máquina a vapor. Este modelo utiliza a função de MIN para a T-norma e MAX para a T-conorma. O valor de saída de um mecanismo de inferência é um valor difuso e, portanto, deve ser defuzzificado a fim de se determinar o valor preciso (discussões sobre vantagens e desvantagens dos principais métodos de defuzzificação podem ser encontrados em (KECMAN, 2001; JANG; SUN; MIZUTANI, 1997; TSOUKALAS; UHRIG, 1997)).

O modelo de inferência TSK é proposto em (TAKAGI; SUGENO, 1985). Uma regra, neste modelo, possui um formato diferente de (4.10):

$$\text{SE } x = A_i \quad \text{E } y = B_i \quad \text{ENTÃO } z = f(x, y) \quad (4.14)$$

onde A_i e B_i são conjuntos difusos como em (4.10) e $z = f_i(x, y)$ é uma função não difusa que descreve uma relação entre as entradas (x, y) e a saída z .

Usualmente, $f(x, y)$ é uma função polinomial, mas pode ser qualquer função que descreva apropriadamente a relação entre as entradas. Caso as $f_i(x, y)$ sejam funções polinomiais de primeira ordem então o mecanismo de inferência é denominado *TSK de ordem um*. Caso as $f_i(x, y)$ sejam funções constantes então o mecanismo de inferência é denominado *TSK de ordem zero*.

¹⁰Do Inglês, *defuzzification*

No sistema de inferência TSK, as *saídas precisas* não necessitam de uma defuzificação mas somente uma agregação das regras avaliadas. A saída deste sistema de inferência pode ser definida pela Eq. (4.15).

$$S = \frac{\sum \psi_i z_i}{\sum \psi_i} \quad (4.15)$$

onde ψ_i é a T-norma ou T-conorma de cada antecedente e z_i é o resultado da função $f(x, y)$.

Sem o consumo de tempo e sem a aplicação da operação matemática de defuzificação do modelo de Mamdani, o sistema de inferência TSK é o candidato mais adequado para modelagem difusa de sistemas baseados em amostras de dados e, portanto, é utilizado no presente trabalho. Para a codificação de conhecimento especialista, Mandani apresenta um melhor desempenho (JANG; SUN; MIZUTANI, 1997).

No modelo de inferência proposto por Tsukamoto (TSUKAMOTO, 1979), o *consequente* de cada regra difusa é representado por um função de pertinência monotônica M . Uma regra neste mecanismo de inferência é definida em 4.16:

$$\text{SE } x = A_i \quad \text{ENTÃO } y = M_i \quad (4.16)$$

onde A_i é um conjunto difuso e M_i é o valor da função monotônica M para a regra.

A *saída precisa* não necessita de defuzificação, já que o resultado da inferência é dado através de uma média ponderada:

$$S = \frac{\sum f_i w_i}{\sum w_i} \quad (4.17)$$

onde f_i é a saída de cada regra induzida pelo peso w_i e a função monotônica do consequente M_i .

4.4 CONSIDERAÇÕES FINAIS

O objetivo da fusão de dados de sensores é fornecer redundância e complementaridade aos sistemas com múltiplos sensores a fim de aumentar a confiabilidade geral das informações sensoriais. Para isso, a representação dos dados (dados diretamente dos sensores, características ou decisão), arquitetura do sistema e técnicas adequadas para a integração dos dados dos sensores devem ser analisados. Cada aspecto mencionado é escolhido a partir do problema proposto e definido através dos dados fornecidos pelos sensores.

A lógica difusa se destaca por modelar informações com um certo grau de ponderação em determinadas regiões. Tal modelagem é determinada através do conhecimento prévio do problema e, a partir do mapeamento não linear entre entradas e saídas, é possível construir um sistema adaptável ao dinamismo do ambiente.

O capítulo anterior e este formam o núcleo deste trabalho e apresentam a teoria geral para a compreensão da arquitetura utilizada no sistema de visão artificial proposto. O capítulo seguinte apresenta a arquitetura do sistema, suas características, propriedades e resultados obtidos sobre seu desempenho.

CAPÍTULO 5

SOLUÇÃO PROPOSTA: IMPLEMENTAÇÃO E RESULTADOS

O sistema de visão artificial proposto será aplicado no projeto em desenvolvimento de um robô, denominado Axebot (AXEBOT, 2005), que será utilizado como base para a criação de um time de futebol de robôs, para as competições da categoria F-180 da Robocup (ROBOCUP, 2005). Como cada parte do RMAI representa um projeto de pesquisa, o objetivo é integrar todos os módulos – sensoriamento, controle e atuação – utilizando a arquitetura proposta por Costa e Bittencourt (1999).

Neste capítulo são descritos o contexto da aplicação, os detalhes de implementação, testes e resultados obtidos do sistema de visão artificial. A arquitetura do sistema proposto bem como os aspectos e funções de cada módulo são apresentados na Seção 5.1. O protótipo construído para os testes é descrito na seção 5.2. Na Seção 5.3, as propriedades do sistema são elencadas. Os resultados obtidos pelo algoritmo de classificação e extração de características encontram-se na Seção 5.4.

5.1 ARQUITETURA DO SISTEMA DE VISÃO ARTIFICIAL

A arquitetura proposta para o sistema é ilustrada na Fig. 5.1 e consiste em quatro módulos: aquisição de dados, extração de característica/classificação de padrões, fusão de dados dos sensores e representação interna de conhecimento dos objetos.

Os módulos *Aquisitores* são responsáveis pela obtenção dos dados da câmera e sensores de distância. Uma vez obtido um quadro de imagem, os objetos contidos neste quadro são classificados através de uma MVS. O centróide de cada objeto extraído é calculado e enviado para o *Seletor de Decisão*¹ a fim de se obter informações como

¹Como mencionado na Seção 4, decisão se refere à inferência de informações, ou seja, informações

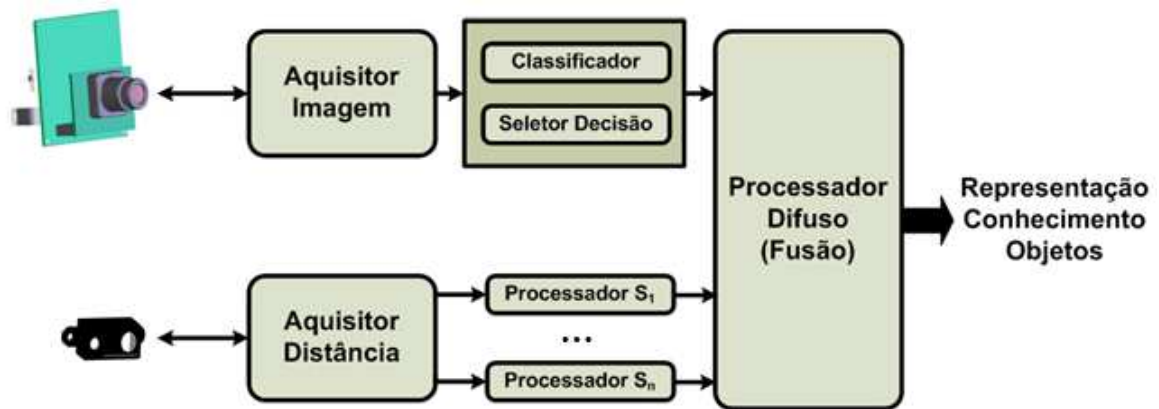


Figura 5.1. Arquitetura do sistema de visão artificial

ângulo e distância estimada de cada objeto. Ao mesmo tempo, os dados adquiridos de todos os sensores de distância são separados e enviados para cada *Processador S_i* , onde $i \in [1, 5]$, e uma função de interpolação é aplicada com o objetivo de encontrar a distância a partir do valor decimal de cada um destes sensores. Por fim, todas estas informações são enviadas para o *Processador de Fusão* a fim de gerar uma representação de conhecimento de cada objeto com informações relevantes para o filtro de entrada do nível reativo.

Os processadores S_i , bem como o Classificador e Seletor de Decisão, são implementados em duas *threads* distintas e sincronizados no Processador de Fusão, visto que a fusão é realizada através de sensores ativos (câmera e sensores de distância).

O sistema proposto é implementado utilizando a linguagem de programação C++ em ambiente Linux, explorando as características de modularização e reutilização de código.

5.1.1 Aquisição de dados

O projeto de cada RMAI contará com um processador StrongARM PXA255 da Intel (INTEL, 2003), responsável tanto para o sistema de controle e navegação, quanto inferidas a partir de algum nível de relevância para o sistema

para o processamento das informações do sistema de sensoriamento. Como o grupo Mecateam ainda não dispõe deste processador, um computador pessoal é utilizado para o processamento das informações de visão e o controle do robô.

As portas de comunicação serial e paralela são utilizadas para a comunicação com a câmera e sensores de distância, respectivamente. A aquisição dos dados da câmera e dos sensores de distância se faz através de um protocolo de comunicação específico para cada sensor, conforme será discutido nas seções a seguir.

5.1.1.1 Câmera A câmera utilizada é a CMUCam2, projetada pela Universidade Carnegie-Mellon (CMUCAM2, 2005) e consiste de um microprocessador SX52 fabricado pela Uvicom e um sensor de visão CMOS OV6620 (ROWE, 2003). Os dados deste sensor podem ser obtidos tanto por uma interface do tipo RS232 quanto por uma porta serial padrão *Transistor Transistor Logic* (TTL). Neste trabalho optou-se pelo padrão RS232.

A Fig. 5.2 ilustra a CMUCam2 e um servo-motor que pode ser acionado pela câmera e a Tabela 5.1 apresenta as principais especificações da CMUCam2.

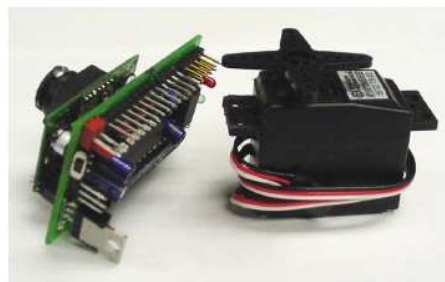


Figura 5.2. CMUCam2

A comunicação com a CMUCam2 se faz por comandos em formato *American Standard Code for Information Interchange* (ASCII). Para se obter um quadro de imagem, o comando SF (*Send Frame*) é enviado para a câmera e um pacote de dados tipo F é retornado na forma:

Tabela 5.1. Características da CMUCam2

Tipo	Característica
Taxa de transmissão	-50 quadros/s para rastreamento de cor -26 quadros/s para rastreamento de movimento
Informações fornecidas	-Centróide de dados rastreados por cor -Média e variância de um conjunto de pixels -Histograma da imagem
Resolução	Até 176 x 144
Capacidade de transmissão (bps)	115.200 57.600 38.400 19.200 9.600 4.800 2.400 1.200
Geral	-Permite ajuste de propriedades da câmera -Controla até 5 servo-motores -Permite redimensionamento da imagem adquirida
Especificações elétricas	Tensão de alimentação 6 a 15 V Corrente de operação 200 mA

$$F \ 1 \ xTam \ yTam \ 2 \ rgb \ rgb \ rgb \ \dots \ rgb \ 2 \ rgb \ rgb \ rgb \ \dots \ rgb \ 3 \quad (5.1)$$

onde $xTam$ é a quantidade máxima de pixels em linha, $yTam$ a quantidade máxima de pixels em coluna, 1 é um delimitador para início do quadro, 2 um delimitador para uma nova coluna e 3 um delimitador de final de quadro. Os valores r , g e b residem na faixa de 16 a 240 e correspondem a valores de cor no espaço RGB.

Antes de se obter um quadro de imagem, alguns registradores da câmera devem ser configurados a fim de se definir uma comunicação adequada entre a unidade de processamento e a câmera. Para isso, o comando CR [reg1 valor1 reg2 valor2 ... reg16 valor16], (*Camera Register*) é utilizado e os valores $reg_i, valor_i, i=1...16$, correspondem aos registradores e seus respectivos valores, ilustrados na Tabela 5.2.

Na Tabela 5.2, o símbolo (*) indica valores padrões da câmera. Para o sistema proposto, o único valor alterado é do registrador 18 para o qual é atribuído o valor 36. Este valor configura a câmera para o espaço de cor YCrCb, que possui menor sensibilidade a variações de luminosidade do ambiente, tornando as cores da imagem obtida mais uniformes. O espaço de cor YCrCb é definido por um percentual do espaço

Tabela 5.2. Comando CR da CMUCam2

Registrador	Valor	Efeito
5	0–255	Contraste
6	0–255	Brilho
18	36	YCrCb, Balanço de branco ligado
	32	YCrCb, Balanço de branco desligado
	44	RGB, Balanço de branco ligado
	40	*RGB, Balanço de branco desligado
17	0	*50 quadros/s
	1	26 quadros/s
	2	17 quadros/s
	3	13 quadros/s
	4	11 quadros/s
	5	9 quadros/s
	6	8 quadros/s
	7	7 quadros/s
	8	6 quadros/s
	10	5 quadros/s
19	32	Auto ganho ligado
	33	*Auto ganho desligado

de cor RGB, segundo a Eq. (5.2):

$$\begin{aligned} Y &= 0.59G + 0.31R + 0.11B \\ Cr &= 0.713(R - Y) \\ Cb &= 0.564(B - Y) \end{aligned} \tag{5.2}$$

Além das configurações para os registradores internos da câmera, a resolução padrão é modificada pelo comando DS [xfator yfator] (*Down Sample*), cujos parâmetros *xfator* e *yfator* correspondem aos valores que dividem o tamanho máximo padrão de imagem enviada pela câmera. Para o sistema proposto, os valores escolhidos são 4 e 4, respectivamente, e tornam a nova resolução da câmera igual a 44x36 pixels.

Após a aplicação das configurações dos registradores e redução do tamanho dos quadros de imagem, o tempo médio de aquisição de um quadro obtido é de aproximadamente 245 ms, através de um canal serial de 115.200 bps. A Eq. 5.3 define a relação

entre o canal serial da câmera, a quantidade de dados e o tempo médio de aquisição de um quadro de imagem:

$$TM = \frac{44 \times 36 \times \text{pixel}}{115.200} \simeq 245ms \quad (5.3)$$

onde TM representa o tempo médio de aquisição e $pixel$ corresponde a 24 bits de informação: 8 bits para o vermelho, 8 bits para o verde e 8 bits para o azul.

5.1.1.2 Sensor de distância Os sensores de distância utilizados são os GP2D02 fabricados pela Sharp (GP2D02, 2005) baseados em informações de infra-vermelho. A faixa de detecção de cada sensor reside entre 10 e 80 cm. A Fig. 5.3 apresenta o aspecto físico do sensor GP2D02. As especificações técnicas destes sensores estão listadas na Tabela 5.3.

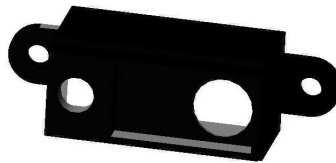


Figura 5.3. Sensor de distância Sharp GP2D02

Tabela 5.3. Especificação técnica do sensor Sharp GP2D02

Parâmetro	Símbolo	Faixa
Tensão de alimentação	V_{cc}	-0.3 a +10 V
Tensão no pino de entrada	V_{in}	-0.3 a +3 V
Tensão no pino de saída	V_{out}	-0.3 a +10 V
Temperatura de operação	T_{op}	-10 a +60 °C
Temperatura de armazenamento	T_{st}	-40 a +70 °C

Cada sensor dispõe de quatro pinos de conexão: V_{in} , pino de entrada dos sinais de controle do sensor; V_{out} , pino de saída, que fornece os valores de distância; GND , pino de terra e V_{cc} pino de alimentação. O sensor vem acompanhado de um diodo para ser montado entre o pino de V_{in} e o pino do conector da interface de comunicação com o

sensor, conforme Fig. 5.4.

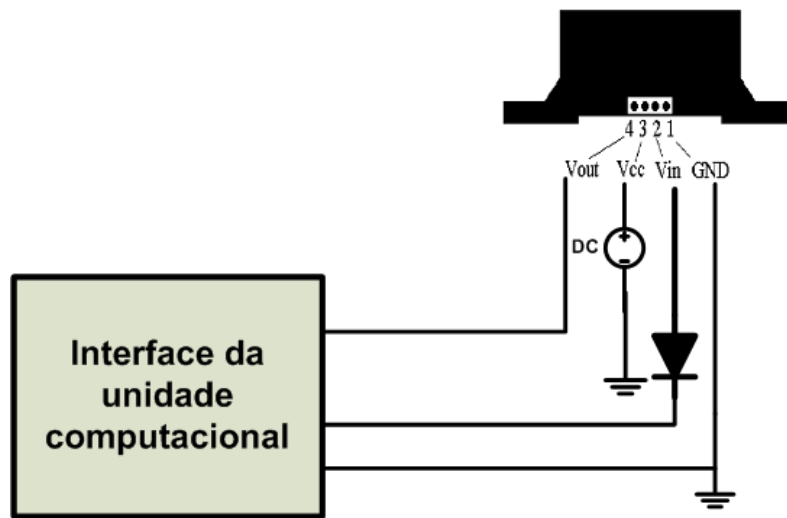


Figura 5.4. Ligação entre o computador e o sensor de distância

Como o pino V_{in} do sensor está montado em configuração de dreno aberto, este pino é habilitado quando a corrente é drenada e, portanto, o diodo tem a função de estabelecer um sentido único para a corrente deste pino (GP2D02, 2005). O sensor de distância GP2D02 consome uma corrente de $3 \mu A$ quando inativo e $22 mA$ quando em operação.

O protocolo de comunicação com o sensor durante a aquisição de dados de distância é ilustrado pelo diagrama temporal da Fig. 5.5.

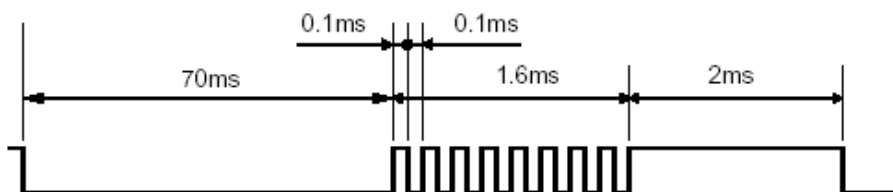


Figura 5.5. Protocolo de comunicação do sensor Sharp GP2D02. Retirada de (GP2D02, 2005)

Inicialmente, deve-se enviar um nível lógico zero durante, no mínimo, 70 ms para ligar o sensor. Em seguida, oito pulsos de, no mínimo, 0,2 ms cada, são enviados para o pino V_{in} a fim de se obter cada um dos oito bits da palavra de distância (o primeiro

bit é o mais significativo). Por fim, envia-se um pulso de duração de 2 ms para o desligamento do sensor.

O algoritmo implementado para o protocolo da Fig. 5.5 é descrito em Algoritmo (1).

Algoritmo 1 Comunicação com o sensor Sharp GP2D02

```
Envia 0 para o  $V_{in}$ 
Aguarda 70 ms
for  $i = 0$  to 8 do
  Envia 0
  Envia 1
  Obtem 1 bit do  $V_{out}$ 
end for
Envia 1 para o  $V_{in}$ 
Aguarda 2 ms
Envia 0 para o  $V_{in}$ 
```

5.1.2 Reconhecimento e localização de objetos

O módulo de Classificação de padrões de imagem e Seleção de decisão é responsável pelo processamento dos dados provenientes do módulo Aquisitor da imagem (ver Fig. 5.1).

A classificação de padrões na imagem tem como objetivo a identificação dos objetos móveis no campo de futebol. Esses objetos podem ser: robôs do time, robôs adversários e bola.

A câmera CMUCam2 possui a função rastreamento de cores, bastando para tanto receber um comando TC [R_{min} R_{max} G_{min} G_{max} B_{min} B_{max}] (*Track Color*)². Este comando utiliza a técnica de limiar direto³ a fim de determinar a cor a ser rastreada. R_{min} , G_{min} e B_{min} são as componentes da cor, no espaço RGB, no limite inferior e R_{max} , G_{max} e B_{max} as componentes da cor no limite superior. Todas as cores, num quadro de imagem, que estão entre os limites inferior e superior, são rastreadas por

²O comando TC pode rastrear tanto uma cor no espaço RGB quanto no espaço YCrCb

³Do Inglês, *thresholding*

este comando, e a resposta é enviada para o controlador da câmera através de um pacote tipo T.

$$T \quad mx \quad my \quad x1 \quad y1 \quad x2 \quad y2 \quad pixels \quad (5.4)$$

onde mx e my são as coordenadas do centróide do retângulo formado por $x1$, $y1$, $x2$, $y2$, que circunda os *pixels* com cores rastreadas.

Através do comando BM [valor] (*Buffer Mode*) é possível rastrear mais de uma cor ao mesmo tempo, onde a variável *Valor* pode ser zero (padrão) ou um: caso seja igual a um, somente um quadro por vez permanece no *armazenador de quadro*⁴. Neste caso, é possível rastrear mais de uma cor em um quadro e depois utilizar o comando RF (*Read Frame*) para ler o próximo quadro.

A desvantagem desta função oferecida pela câmera é a alta sensibilidade à variação da luminosidade do ambiente. Qualquer variação de luminosidade, pode fazer com que os pixels entre os dois limiares não sejam encontrados. Além disso, caso seja configurada uma faixa de limiar muito grande, há a possibilidade de se classificar, erroneamente, pixels fora da faixa de cor dos objetos procurados. Em vista disso, uma MVS é utilizada, a partir da biblioteca proposta em (CHANG; LIN, 2001), com o objetivo de implementar um classificador mais robusto às variações de luminosidade.

5.1.2.1 Construção do classificador Para a aplicação de uma MVS no espaço de cor (ver Capítulo 3) da imagem obtida, quatro hipóteses foram analisadas, a fim de avaliar o melhor espaço para classificação:

- Imagens no espaço de cor RGB
- Imagens no espaço de cor YCrCb
- Imagens no espaço de cor RGB com segmentação

⁴Do Inglês, *frame buffer*

- Imagens no espaço de cor YCrCb com segmentação

Inicialmente, imagens nos espaços RGB e YCrCb sem nenhum tipo de segmentação são avaliadas quanto ao *kernel* de menor número de vetores de suporte, na fase de treinamento. Neste estágio, os objetos são rotulados por cores distintas (azul para bola, roxo para corpo do robô, marrom para robô do mesmo time e vermelho para robô adversário) através da interface gráfica de usuário utilizando o *mouse*, conforme Fig. 5.6.

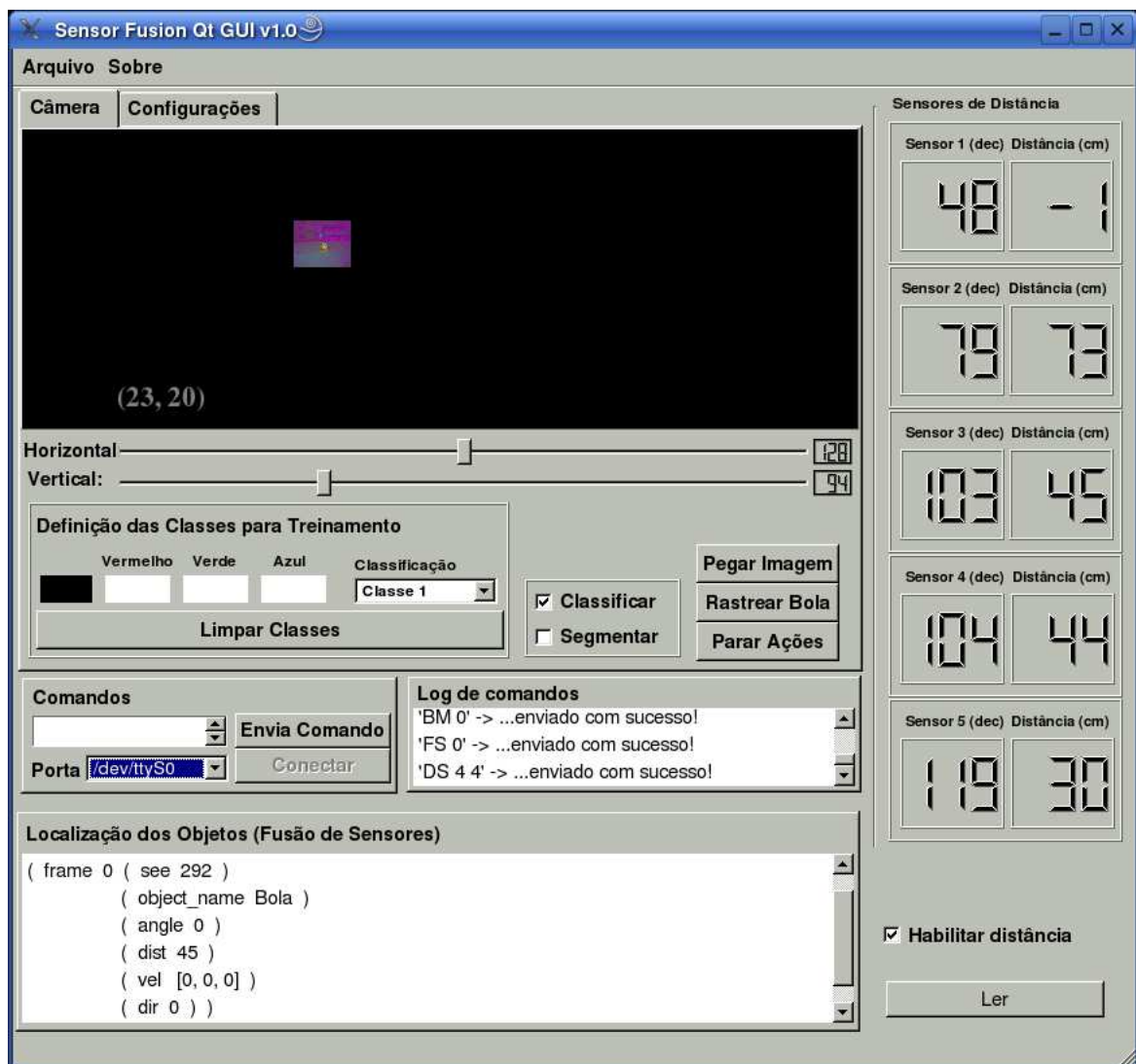


Figura 5.6. Interface gráfica construída para o gerenciamento do sistema de visão artificial proposto

Em seguida, a partir do conjunto de *pixels* rotulados da imagem é gerado um arquivo de treinamento. Cada componente de cor neste arquivo é normalizado no intervalo de $[-1, +1]$, aplicando a Eq. 5.5.

$$C_{normal} = l + (u - l) * \frac{(C_i - \min(C_i))}{\max(C_i) - \min(C_i)} \quad (5.5)$$

onde C_{normal} é o componente normalizado (R, G ou B) da cor; l representa o menor valor do intervalo de normalização, neste caso $l = -1$; u representa o maior valor do intervalo de normalização, neste caso $u = +1$; C_i representa o componente da i_{esima} cor que será normalizado; $\min(C_i)$ e $\max(C_i)$ são os valores mínimo e máximo, respectivamente, de cada componente C_i das cores do arquivo da amostra de treinamento.

A escolha de um *kernel* apropriado é determinante para uma MVS, visto que este influencia na forma em que os dados são separados no novo espaço de característica. Outra escolha crítica é a seleção dos parâmetros de um *kernel*, que consiste em encontrar C e γ , onde C é o critério de parada de treinamento e γ um parâmetro específico do *kernel*.

A fim de determinar os melhores parâmetros do *kernel*, um algoritmo de validação cruzada em k -dobras⁵ é aplicado. Neste caso, $k = 5$ (esse valor é obtido empiricamente através dos testes). O método consiste em dividir uma amostra de treinamento em 5 subconjuntos de iguais tamanhos e validar todos entre si. A idéia geral do método é ilustrada no Algoritmo 2.

Algoritmo 2 Validação cruzada em 5 dobras

```
for  $i = 0$  to  $k$  do  
  treine subconjunto[s] com pares de  $(C, \gamma)$   
   $\forall$  subconjunto[i],  $i \neq s$ , encontre a taxa de erro  
  selecione a menor taxa de erro  
  teste com novos parâmetros, incrementando-os em potência de dois  
end for
```

⁵Do Inglês, *K-fold cross-validation*

As Figs. 5.7(a) e 5.7(b) ilustram as amostras de treinamento, obtidas pelo sistema de visão proposto, utilizando uma iluminação ambiente de 600 lux⁶ com um fundo branco, a fim de se obter a máxima separação dos atributos de cor na fase de treinamento (esta condição é relaxada na fase de testes). Apenas uma amostra em cada espaço de cor é utilizada para a classificação.

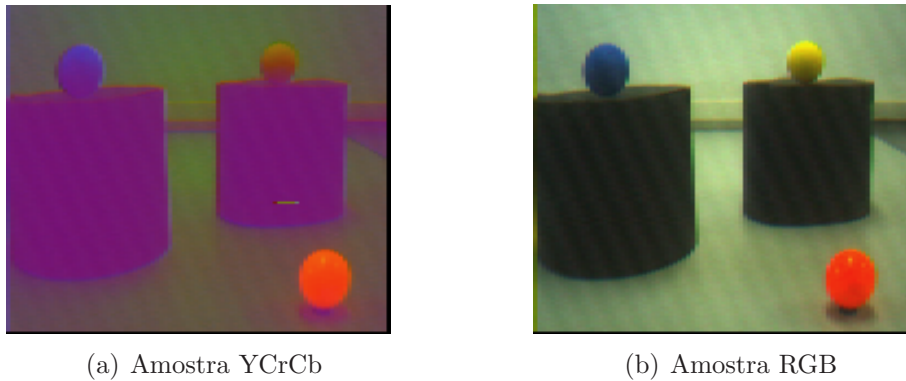


Figura 5.7. Amostras utilizadas

A fim de se determinar o melhor *kernel* para o treinamento, duas tabelas são construídas. As Tabelas 5.4 e 5.5 ilustram os resultados de um treinamento com e sem o método de validação cruzada e usando os espaços de cor RGB e YCrCb, respectivamente. Os parâmetros obtidos no procedimento são $\gamma = 8$ e $C = 32.768$.

Tabela 5.4. Avaliação do procedimento de *Validação cruzada em 5-dobras* – RGB

Sem validação cruzada			Com validação cruzada		
Kernel	Precisão	nVS	Kernel	Precisão	nVS
RBF	71,7%	820	RBF	76,3%	694
linear	72,5%	810	linear	73,2%	755
polinomial	73,6%	818	polinomial	75,0%	810
sigmóide	68,8%	900	sigmóide	43,8%	1687

Pode-se observar a partir dos dados da Tabela 5.5 que o *kernel* RBF demonstrou o melhor desempenho, com uma precisão de classificação de 84,6% e utilizando apenas

⁶O valor de luminosidade foi conseguido por um luxímetro e a iluminação ambiente não possuía qualquer controle

Tabela 5.5. Avaliação do procedimento de *Validação cruzada em 5-dobras* – YCrCb

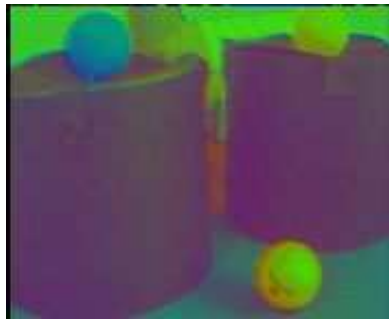
Sem validação cruzada			Com validação cruzada		
Kernel	Precisão	nVS	Kernel	Precisão	nVS
RBF	76,7%	1311	RBF	84,6%	656
linear	77,5%	1295	linear	77,6%	1220
polinomial	77,2%	1421	polinomial	78,5	1313
sigmóide	75,9%	1347	sigmóide	64,2%	1553

656 vetores de suporte (nVS). Devido ao fato de que a quantidade de pixels na imagem da bola e na imagem que diferencia os times, ser menor que a quantidade de pixels da imagem do corpo dos robôs, são aplicados diferentes pesos para estes diferentes objetos: 10 para a bola, 25 para a marca do robô do mesmo time, 25 para o robô do time adversário e 1 para o corpo do robô. Estes pesos aplicados durante a fase de treinamento não alteram o número de vetores de suporte totais, porém auxiliam na obtenção de vetores de suporte em áreas de fronteira, que melhoram a classificação.

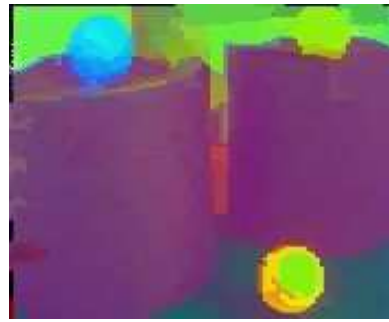
Uma vez definido o melhor *kernel*, duas imagens no espaço RGB e YCrCb são segmentadas utilizando os algoritmos de normalização compreensiva proposto por Finlayson, Schiele e Crowley (1998) e a segmentação por deslocamento de média, proposto por Comaniciu e Meer (2002). Exemplos de amostras utilizando os algoritmos acima são ilustradas na Fig. 5.8.

Em seguida, cada imagem é treinada com uma MVS utilizando *kernel* RBF e parâmetros definidos pelo algoritmo de validação cruzada. A Tabela 5.6 é construída para avaliação do número de vetores de suporte e, conseqüentemente, da qualidade da classificação.

Dessa forma, pode-se avaliar os efeitos de uma segmentação de cor no desempenho do classificador. A utilização dos algoritmos de segmentação não auxilia o trabalho do classificador. Além disso, o próprio espaço YCrCb já normaliza, de uma certa forma, o espaço RGB, diminuindo os efeitos de variação de luminosidade, o que explica,



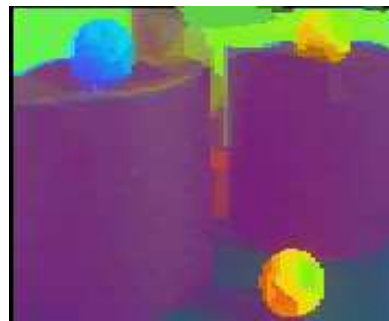
(a) Segmentação por Normalização Compreensiva (YCrCb)



(b) Segmentação por Deslocamento de Média (YCrCb)



(c) Segmentação por Normalização Compreensiva (RGB)



(d) Segmentação por Deslocamento de Média (RGB)

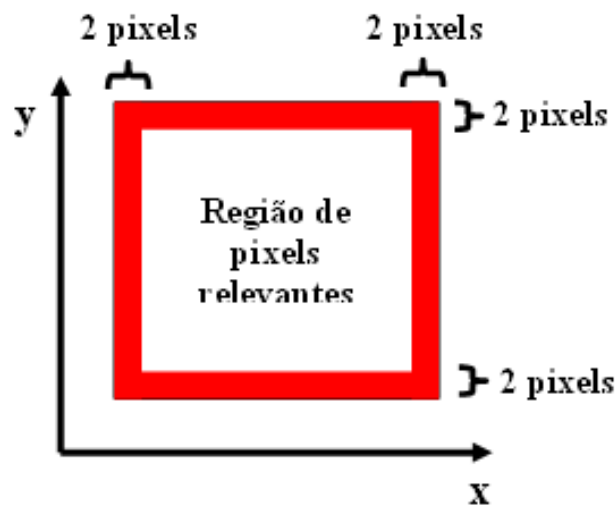
Figura 5.8. Imagens segmentadas

portanto, a melhor classificação utilizando imagens em YCrCb puro.

Ao final da fase de treinamento, um arquivo contendo os vetores de suporte para a classificação é gerado. Em seguida, o espaço de pixels da imagem é reduzido a fim de diminuir o tempo de classificação. O critério usado para descartar os pixels irrelevantes é definido pela probabilidade de não haver objetos ou partes de objetos numa dada região da imagem. A Fig. 5.9 apresenta os pixels descartados. Na região onde os pixels são considerados relevantes, apenas os pixels pares são submetidos à classificação, visto que o objetivo final é determinar o centróide de cada objeto e nem todos os pixels são necessários para tanto. Os pixels das bordas da imagem também são descartados e, por fim, um tempo médio de classificação de 55 ms é obtido.

Tabela 5.6. Resultados em diferentes espaços de imagem

RGB		YCrCb	
Imagem	nVS	Imagem	nVS
Pura	694	Pura	656
Compreensão normalizada	965	Compreensão normalizada	752
Deslocamento de média	1705	Deslocamento de média	1111

**Figura 5.9.** Pixels descartados para a melhoria no tempo de classificação da imagem

5.1.2.2 Localização de objetos A partir das características extraídas da imagem (área do objeto, altura, largura, centróide), a localização dos objetos é determinada através de suas coordenadas polares (θ, d), onde θ representa o ângulo e d a distância relativos ao centro de massa do robô.

Para a determinação das coordenadas polares, duas MVS de regressão são utilizadas: para o ângulo θ , é determinada uma função de mapeamento de cada pixel a seu ângulo correspondente, e, para a distância d , um mapeamento da altura do objeto na imagem e sua respectiva distância ao centro do robô.

Para a obtenção da função pixel-ângulo, raios e linhas paralelas são definidos em uma folha branca de papel. A bola é colocada na intersecção entre cada raio e linha,

onde a origem dos raios é o centro do robô. A Fig. 5.10 ilustra esta situação.

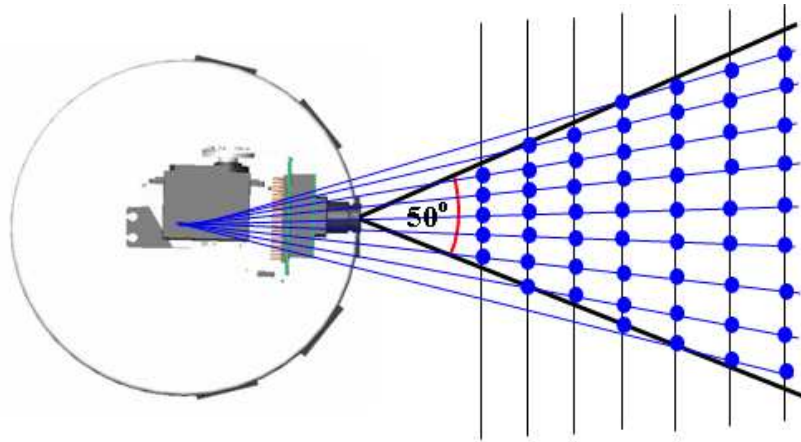


Figura 5.10. Determinação da relação entre pixel e ângulo

Um conjunto de pares (pixel, ângulo) é então obtido a partir do centróide de cada objeto e sua respectiva localização no espaço com ângulo relativo ao centro do robô.

A cabeça de visão artificial proposta é controlada por dois servo-motores e, portanto, possui dois graus de liberdade de movimentação (o movimento panorâmico feito pela rotação em torno do eixo y do sistema de coordenadas da câmera e o movimento vertical, ou seja, rotação em torno do eixo z). A fim de reduzir o efeito da distorção angular da imagem, ocorrido quando a cabeça de visão artificial se encontra em diferentes ângulos em relação ao eixo z , um valor de ângulo vertical (da cabeça) é adicionado ao par (pixel, ângulo) obtido anteriormente. A Fig. 5.11 ilustra os ângulos utilizados.

Para cada ângulo (20° , 35° , 50° e 65°) um conjunto de pares (pixel, ângulo) é determinado e a *tupla* (pixel, ângulo, ângulo da cabeça) é submetida ao classificador.

Para o treinamento, uma MVS de *kernel* linear e um modelo com 30 vetores de suporte são utilizados. A razão da utilização de um *kernel* linear é o número de vetores de suporte (nVS), como ilustrado na Tabela 5.7 e, conseqüentemente, sua habilidade de generalização.

A Fig. 5.12 mostra as centróides de cada objeto. Estas são obtidas pela média entre o menor e o maior valores das coordenadas x e y dos pixels rotulados para um

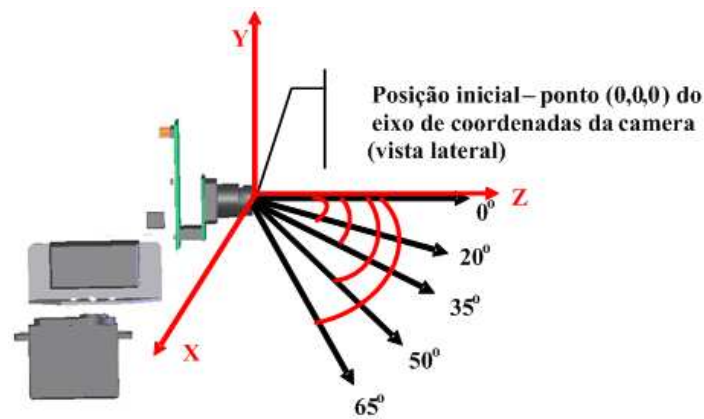


Figura 5.11. Ângulos da cabeça móvel utilizados para a determinação da função de mapeamento pixel-ângulo. O ângulo zero representa a posição inicial da câmera quando o robô é ativado

Tabela 5.7. Número de VS para cada *kernel* na regressão

Kernel	nVS
linear	30
RBF	32
polinomial	32
sigmóide	32

determinado objeto.

A informação de distância extraída da imagem é utilizada apenas como estimativa bruta, útil para o caso em que algum objeto se encontre em áreas de sombra dos sensores de distância (uma análise sobre a disposição dos sensores de distância é feita na Seção 5.2). À medida que o robô se locomove, este pode corrigir as informações de distância rotacionando seu corpo, por exemplo, a fim de obter maior confiabilidade na informação.

A distância de um objeto a partir da imagem é obtida através de uma função de mapeamento entre a altura h_1 do objeto na imagem (obtida pelo contorno do objeto definido pelos pixels classificados) e distância d do objeto ao robô em relação a sua origem, conforme ilustrado na Fig. 5.13.

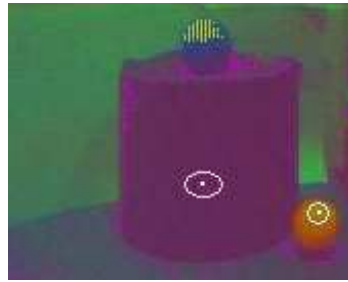


Figura 5.12. Centróides da bola e do robô (ponto branco)

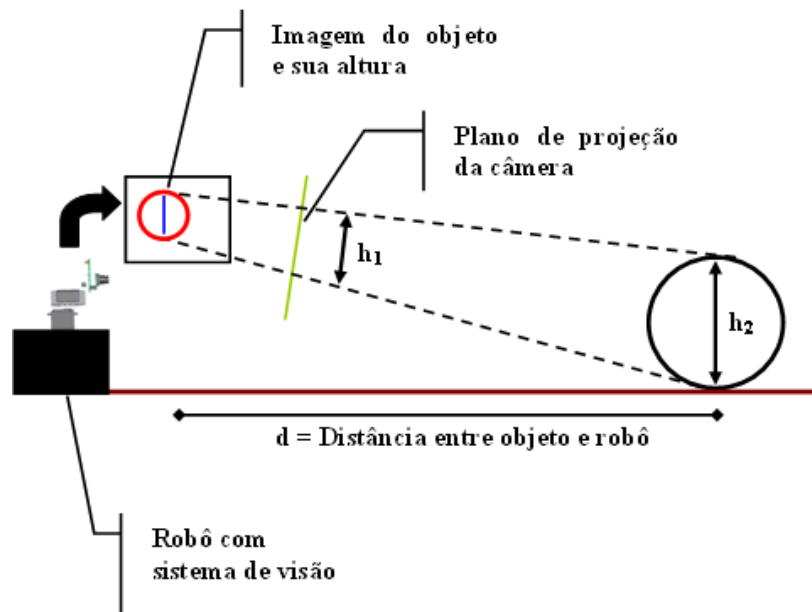


Figura 5.13. Determinação da relação entre altura e distância

Na Fig. 5.13, a altura real h_2 do objeto é projetada no plano de projeção da câmera, obtendo-se a altura h_1 da imagem em pixels. A relação entre h_1 e d é "aprendida" através de uma MVS de *kernel* linear, agora sem a necessidade de avaliação de precisão, visto que o próprio método de determinação da distância é impreciso e apenas se aplica em circunstâncias especiais (objetos localizados na área de sombra dos sensores de distância).

Por fim, uma etapa de pós classificação dos pixels classificados é realizada. Esta etapa é responsável por diminuir o número de pixels classificados fora do contorno dos objetos e, portanto, definidos como ruído de classificação. O objetivo principal dessa etapa é aumentar a precisão da determinação dos atributos de centróide e altura dos

objetos. O algoritmo de pós classificação é ilustrado em Algoritmo 3.

Algoritmo 3 Rotina de pós classificação

```
Dividir a imagem em N subregiões ( $N = 6$ )
Para cada região, calcular a frequência  $F_i$  de pixels de cada classe  $c_i$ 
if  $F_i \leq TH$  ( $TH = 3$ ), onde TH é um limiar definido pelo especialista then
  Retire os pixels da classe
end if
```

5.1.3 Processador dos sensores de distância

O processador dos sensores de distância é responsável por separar os dados obtidos pela *thread* de aquisição dos dados dos sensores e determinação da distância correspondente aos valores decimais de cada sensor (contagens do contador interno).

Uma vez obtidos os dados dos sensores, estes são submetidos a uma função polinomial de interpolação. Estes polinômios são responsáveis pela conversão do valor decimalizado de uma medida de um sensor em um valor de distância. Tais polinômios são obtidos pela interpolação dos pontos discretos medidos no procedimento de calibração (valor decimalizado versus distância). O processo de calibração de cada sensor de distância é ilustrado na Fig. 5.14.

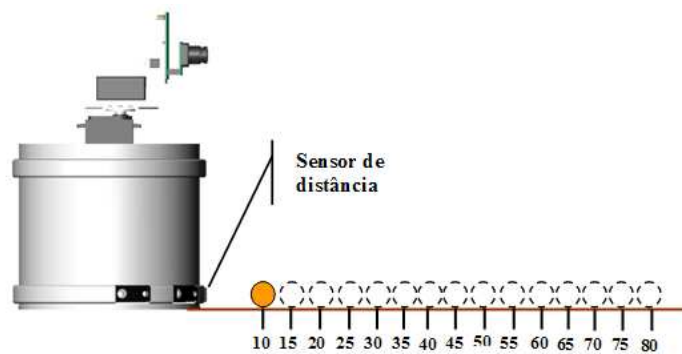
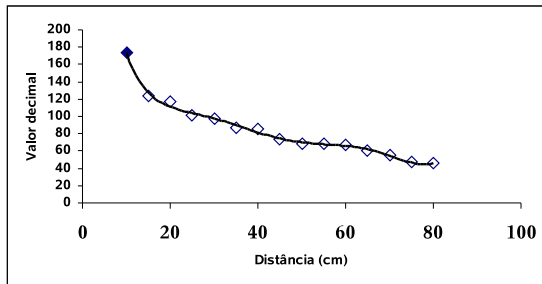


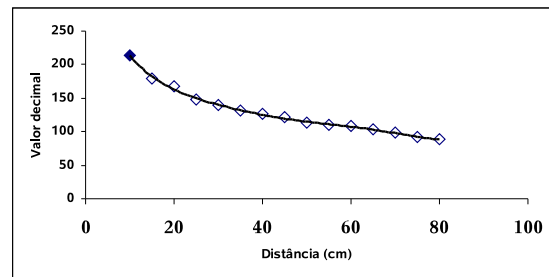
Figura 5.14. Processo de calibração dos sensores de distância

Para uma interpolação com maior precisão, dois polinômios foram definidos em duas faixas, para cada sensor: um para a faixa entre 10 e 35 cm e outro para a faixa entre 35 e 80 cm.

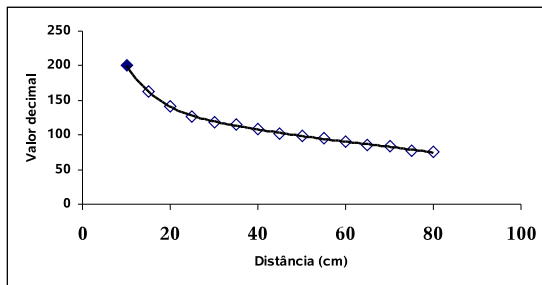
A resposta de cada um dos cinco sensores de distância utilizados são apresentadas na Fig. 5.15. Os pontos foram obtidos durante o processo de calibração de sensores. As curvas em linha cheia são o ajuste dos pontos experimentais por um polinômio de grau seis.



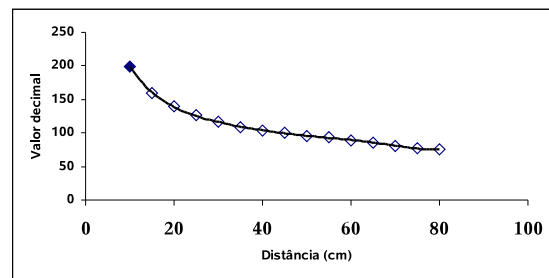
(a) Curva do sensor1



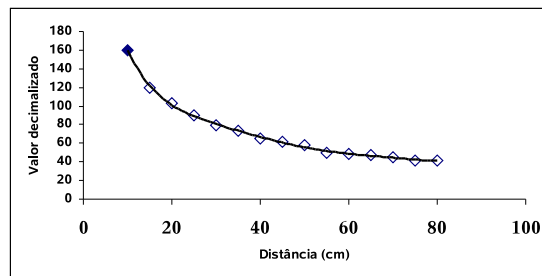
(b) Curva do sensor2



(c) Curva do sensor3



(d) Curva do sensor4



(e) Curva do sensor5

Figura 5.15. Curvas dos 5 sensores de distância

Os valores da Fig. 5.15 foram obtidos por cinco medições em três suportes diferentes de papéis (branco, preto e cinza) utilizados em frente aos sensores. O valor decimal utilizado para cada sensor corresponde a média dos três valores obtidos. Os testes demonstram a robustez dos sensores GP2D02 da Sharp quanto à superfície de reflexão para a medida de distância, conforme Tabela 5.8.

Tabela 5.8. Distância x valores decimalizados dos sensores

	S1			S2			S3			S4			S5		
D	B	P	C	B	P	C	B	P	C	B	P	C	B	P	C
10	174	174	174	214	214	214	201	201	201	199	199	199	160	160	160
15	124	124	124	179	179	179	162	162	162	160	160	160	120	120	120
20	117	117	117	168	168	168	142	142	142	139	139	139	103	103	103
25	101	101	101	148	148	148	127	127	127	126	126	126	90	90	90
30	98	98	98	139	139	139	118	118	118	117	117	117	79	79	79
35	87	87	87	132	132	132	115	115	115	108	108	108	74	74	74
40	85	85	85	126	126	126	109	109	109	104	104	104	65	65	65
45	74	74	74	121	121	121	102	102	102	100	100	100	62	62	62
50	69	69	69	114	114	114	98	102	102	96	96	96	58	58	58
55	68	68	68	110	110	110	95	95	95	94	94	94	50	50	50
60	67	65	66	108	108	108	90	91	89	88	87	87	48	48	48
65	61	61	61	103	101	102	86	87	85	86	85	87	47	45	46
70	55	53	54	98	95	97	84	84	84	81	81	81	45	44	45
75	47	46	48	92	90	91	78	77	74	77	77	76	42	42	42
80	46	45	47	89	89	89	75	75	75	75	74	74	41	40	39

D = distância do objeto ao sensor (em centímetros)

S1 ... S5 = sensor 1 a sensor 5 de acordo com Fig. 5.17

B = papel branco, P = papel preto, C = papel cinza

5.1.4 Processador de fusão

Uma vez selecionados os valores de decisão de cada sensor (objeto, ângulo e distância), tais valores são submetidos ao processador de fusão. A fusão dos dados dos sensores é realizada em duas etapas distintas: seleção do ângulo e distância a partir da imagem e fusão entre estes e os valores obtidos a partir dos sensores de distância. A primeira etapa é fundamental a fim de sincronizar as linhas de tempo dos sensores. Visto que o sensor de visão (câmera) é o mais lento dos sensores utilizados e possui maior densidade de dados espaço-temporais por aquisição, este é utilizado como sincronizador do processador de fusão.

A linha de tempo das duas classes de sensores utilizados é ilustrada na Fig. 5.16. O tempo de aquisição de um valor de distância e conversão para centímetros é de

aproximadamente 75 ms. Isto corresponde a aproximadamente quatro leituras de distância em um processamento de um quadro de imagem e seleção de variáveis de decisão na imagem (300 ms). Em vista disso, as três primeiras leituras dos sensores de distância são descartadas e somente a última representa as distâncias dos objetos reconhecidos. Isto é ilustrado através de linhas pontilhadas na Fig. 5.16.

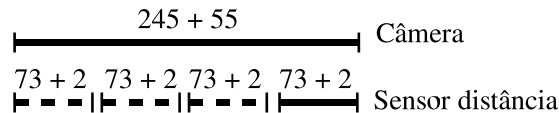


Figura 5.16. Linha de tempo dos sensores envolvidos no processo de fusão

O alinhamento temporal das informações dos sensores é necessário, pois os sensores utilizados são do tipo ativo. Após realizado este alinhamento, os valores de decisão são utilizados para a fusão de sensores propriamente dita. Para isso, os conjuntos difusos (Fig. 5.18) são determinados a partir da disposição física dos sensores de distância no protótipo, conforme Fig. 5.17.

Na Fig. 5.17, o sensor três (S3) é utilizado como referência ilustrativa para o processo de determinação dos conjuntos difusos utilizados para o processo de fusão em todos os sensores. Quando os objetos se encontram nas áreas de sombra, a distância utilizada é a obtida pela imagem. Nas demais situações, a distância e o ângulo finais são determinados pelo sistema difuso com sistema de inferência de *Takagi-Sugeno* de ordem zero (TAKAGI; SUGENO, 1985). A razão da escolha é determinada pelo desempenho deste sistema para processamento de amostras de dados, conforme exposto na Seção 4.3.3.

Uma vez determinado o ângulo através da câmera, seu valor pode ser refinado utilizando as informações dos sensores de distância. Visto que a disposição dos sensores de distância é conhecida, por questões de simplicidade, os ângulos são fornecidos a partir de uma linha imaginária (linha hachurada na Fig. 5.17) que parte da origem do robô e passa por cada ponto central dos sensores de distância.

Pode-se observar na Fig. 5.17, que quanto mais longe o objeto se encontra do

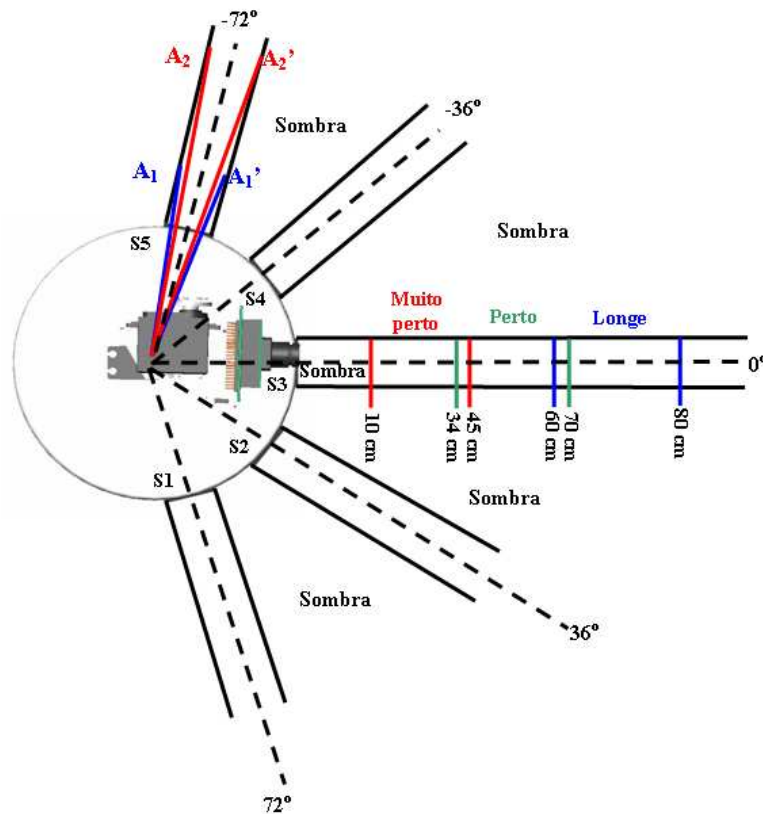


Figura 5.17. Determinação dos conjuntos difusos a partir da geometria dos sensores de distância no protótipo

sensor de distância, mais preciso seu ângulo é determinado. Isto se verifica da seguinte forma: já que o ângulo sólido sensível de cada sensor de distância é um cilindro de 3 cm de diâmetro, que corresponde ao espaço entre o receptor e emissor, um objeto entre A_1 e A_1' possuirá um ângulo menos preciso do que um objeto entre A_2 e A_2' .

A partir da geometria dos sensores, obtém-se os conjuntos difusos ilustrados na Fig. 5.18 e, então, oito regras difusas são propostas:

$$R_1: \text{SE distancia} = \text{SOMBRA ENTÃO distancia} = d(\text{camera})$$

$$R_2: \text{SE distancia} = \text{SOMBRA ENTÃO angulo} = a(\text{camera})$$

$$R_3: \text{SE distancia} = \text{MUITOPERTO ENTÃO distancia} = d(S_i)$$

$$R_4: \text{SE distancia} = \text{MUITOPERTO ENTÃO angulo} = a(\text{camera})$$

$$R_5: \text{SE distancia} = \text{PERTO ENTÃO distancia} = d(S_i)$$

$$R_6: \text{SE distancia} = \text{PERTO ENTÃO angulo} = a(\text{camera}) * 0,5 + a(S_i) * 0,5$$

R_7 : SE distancia = LONGE ENTÃO distancia = $d(S_i)$

R_8 : SE distancia = LONGE ENTÃO angulo = $a(S_i)$

As funções $a(\cdot)$ e $d(\cdot)$ representam, respectivamente, ângulo e distância obtidos pela câmera e pelos sensores de distância.

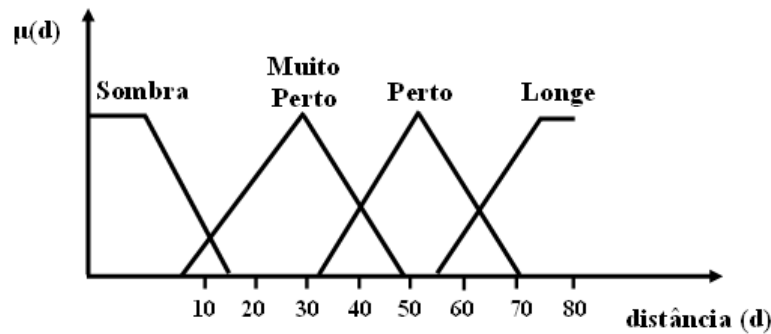


Figura 5.18. Conjuntos difusos para a variável linguística distância

Os valores reais das distâncias e ângulos, após a avaliação das regras, são determinados através da Eq. (4.15).

5.2 PROTÓTIPO FÍSICO

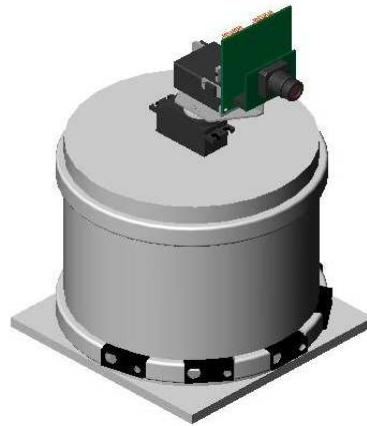
A elaboração de um protótipo serve de base para a aplicação dos conceitos gerais do sistema de visão artificial proposto. O projeto e o desenho do protótipo realizado no sistema de projetos mecânicos 3D, Solidworks, são ilustrados nas Figs. 5.19(a) e 5.19(b), respectivamente.

Para a construção do protótipo, um cilindro de 17 cm de diâmetro e 12 cm de altura é utilizado. Pode-se observar que a cabeça de visão artificial se encontra exatamente no centro do cilindro, acima. Esta decisão tem como objetivo simplificar a determinação dos ângulos dos objetos relativos ao centro da base do cilindro do robô. Dessa forma, tanto os ângulos obtidos pela câmera quanto pelos sensores de distância tem como origem o centro da base do cilindro.

Os sensores de distância estão dispostos no semi-círculo do anel inferior, cobrindo



(a) Protótipo físico



(b) Protótipo em SolidWorks

Figura 5.19. Protótipo do robô com a cabeça de visão artificial

180° de circunferência. Como o comprimento da semi-circunferência do cilindro é igual a $S_c = \pi * R = \pi * 17/2 \cong 29,7\text{cm}$, onde R é o raio da base do cilindro, e o comprimento total de cada sensor de distância é igual a 4,5 cm, pode-se posicionar até 5 sensores inteiros.

Apesar da possibilidade de posicionamento de sensores ao redor de todo o cilindro, obtendo, assim, uma varredura total do ambiente, opta-se, aqui, por utilizar um semi-anel de sensores de distância. Isto se deve ao fato do semi-anel traseiro do robô⁷ servir apenas como leitura complementar e não trazer maiores benefícios para a varredura do ambiente, já que o servo panorâmico varre apenas 164° e não teria como identificar objetos na parte traseira.

5.3 CARACTERÍSTICAS E PROPRIEDADES DO SISTEMA

As características definidas para o sistema proposto são: ângulo de abertura, cone de visão, velocidades máximas para detecção de um objeto pela câmera e limites para a atuação dos motores servo-acionados que movimentam a câmera.

A Fig. 5.20 ilustra como é determinada o ângulo de abertura da câmera. Este é

⁷Considera-se o sistema de chute como o ponto de referência para a frente do robô

obtido pela determinação dos pontos limites da imagem de um objeto conhecido e do local na imagem (x, y) de seu centróide nestes limites. Em seguida, uma triangulação dessas informações com o centro da lente da câmera é realizada. A abertura da câmera, neste caso, é igual a 50° .

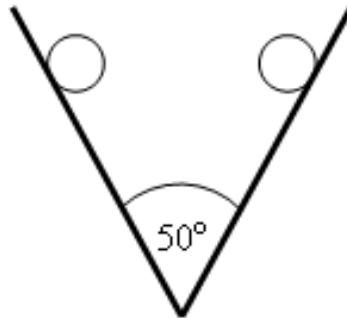


Figura 5.20. Determinação do ângulo de abertura da câmera

O cone de visão é definido pelo limite mínimo Z_{min} e máximo Z_{max} dos sensores de distância e a abertura da câmera, conforme Fig. 5.21.

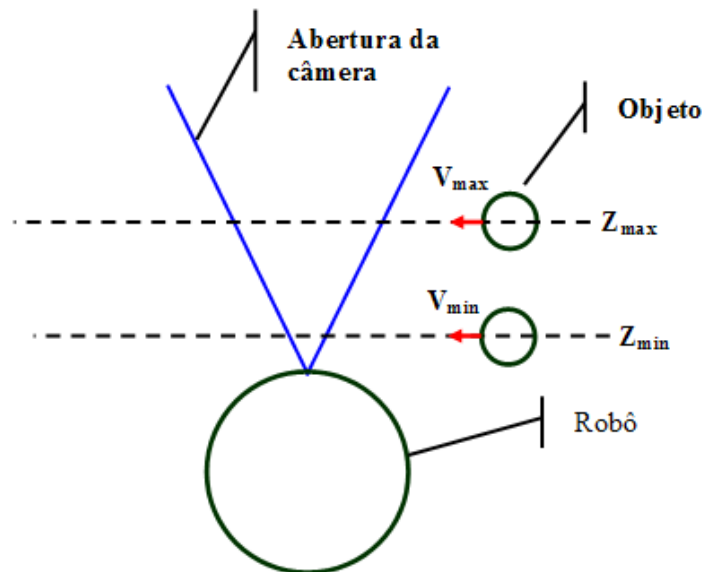


Figura 5.21. Determinação dos limites de percepção do sistema proposto

Para os sensores utilizados, Sharp GP2D02, o Z_{min} é igual a 10 cm e Z_{max} é igual a 80 cm (GP2D02, 2005). A partir dessas características medidas pode-se então calcular as velocidades para uma distância mínima V_{min} e para uma distância máxima V_{max} nas quais um objeto pode ser percebido pelo sistema, através da Eq. (5.6).

$$V_{min} = \frac{Z_{min}}{\hat{T}} \quad \text{e} \quad V_{max} = \frac{Z_{max}}{\hat{T}} \quad (5.6)$$

onde \hat{T} representa o tempo médio estimado e é igual a soma dos tempos de aquisição (245 ms), classificação dos objetos na imagem e determinação dos atributos de decisão (55 ms).

A Eq. (5.6) fornece, portanto, para o pior caso (um objeto passando paralelamente à lente da câmera), um $V_{min} = 0,33m/s$ para um objeto a 10 cm de distância da câmera e $V_{max} = 2,3m/s$ para um objeto a 80 cm.

As características gerais do sistema de visão artificial proposto são resumidas na tabela 5.9.

Tabela 5.9. Características do sistema de visão artificial proposto

Tipo	Característica
Câmera	CMUCam2
Sensores de distância	Sharp GP2D02 (5 sensores)
Abertura da câmera	50°
Faixa de sensoriamento de distância	entre 10 e 80 cm
Velocidades máximas de detecção	0,33 m/s (a 10 cm) 2,3 m/s (a 80 cm)
Taxa de transmissão máxima de imagens	~3 quadros/s
Taxa de transmissão máxima de informação de distância	13 distâncias/s (restrita pela taxa da aquisição de imagens na fusão)
Varredura da cabeça de visão artificial	164° panorâmico 65° vertical

As propriedades do sistema proposto são: reconhecimento e localização dos objetos, representação de conhecimento dos objetos identificados/localizados e rastreamento da bola. Cada objeto identificado e localizado é representado internamente pela estrutura definida a seguir.

Algoritmo 4 Estrutura do quadro da saída do sistema para cada objeto

```
( frame <numQuadro> ( see <tempoPercepção> )
  ( object_name <idObjeto> )
  ( angle <ânguloObjeto> )
  ( dist <distânciaObjeto> )
  ( vel <velDifusaObjeto> ) )
```

Para cada quadro de imagem $\langle \text{numQuadro} \rangle$, um objeto rotulado por um $\langle \text{idObjeto} \rangle$ é identificado e três características são fornecidas pelo sistema de visão: ângulo relativo ao centro da base do robô $\langle \text{ânguloObjeto} \rangle$, distância do objeto à frente do robô $\langle \text{distânciaObjeto} \rangle$ e velocidade difusa do objeto $\langle \text{velDifusaObjeto} \rangle$. O processo de determinação das duas primeiras características é apresentado na Seção 5.1.2.

Com relação à velocidade, esta é definida a partir de uma variável linguística difusa, ilustrada na Fig. 5.22.

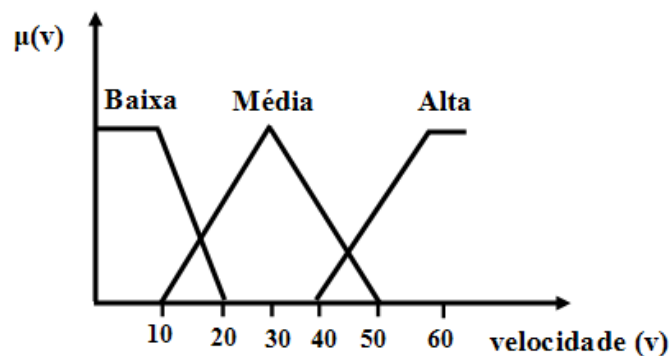


Figura 5.22. Conjuntos difusos para a variável linguística velocidade

Os conjuntos difusos da variável velocidade são definidos pela variação da posição dos pixels do centróide da imagem de um dado objeto e pelo tempo de processamento entre dois quadros de imagem. A Fig. 5.23 ilustra esta situação.

Como a resolução da imagem utilizada é de 44x36 pixels, isto implica que de um quadro de imagem para outro, o centróide de um objeto pode se mover por no máximo 63 pixels (tamanho da diagonal do retângulo definido pelo quadro da imagem). Dessa forma, o conjunto velocidade.baixa é definido entre 0 a 20 pixels percorridos, o conjunto velocidade.média entre 10 e 50 e o conjunto velocidade.alta entre 40 e 63. Assim, a

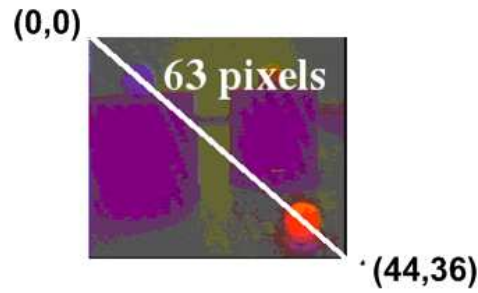


Figura 5.23. Determinação da velocidade difusa

variável $\langle \text{velocidadeDifusaObjeto} \rangle$ é definido através da Eq. (5.7). Os processos de inferência e *defuzificação* da variável velocidade do objeto serão realizados pelo nível reativo do RMAI.

$$\langle \text{velDifusaObjeto} \rangle = [\mu_{\text{baixa}}(\text{difPixel}), \mu_{\text{media}}(\text{difPixel}), \mu_{\text{alta}}(\text{difPixel})] \quad (5.7)$$

onde $\mu_i(\text{difPixel})$ representa as funções de pertinência, onde i representa cada um dos conjuntos difusos da variável linguística velocidade e difPixel é a diferença da localização do centróide de cada objeto em relação aos quadros n e $n - 1$.

Para o rastreamento da bola, os servo-motores são acionados a partir da localização do centróide da imagem do objeto. Os servo-motores são conectados à placa de processamento de imagem da CMUCam2 e, através de comandos SV [numServo pos] (*Servo*), um servo de número numServo é enviado para a posição pos . Cada servo pode ser acionado entre 46 e 210 graus, sendo 128 a posição central e para onde os servos são deslocados durante o ciclo de inicialização da câmera. Sendo assim, a varredura máxima para cada servo acionado pela câmera é de 164° . Para o sistema proposto, por limitações dos servos, a varredura panorâmica é máxima e de 65° em relação ao eixo vertical, conforme Tabela 5.9.

Considerando os limites de atuação da cabeça de visão utilizando a CMUCam2 e o objetivo de posicionar a imagem da bola no centro do quadro de imagem, o Algoritmo 5 é proposto.

Algoritmo 5 Rastreamento da bola

```
Servo.x, Servo.y = posições panorâmica e vertical do servo
Ci.x, Ci.y = coordenadas (x,y) do centro da imagem
Co.x, Co.y = coordenada (x,y) do centro do objeto
varFolga = variável de folga para o posicionamento
if Co.x > Ci.x + varFolga then
    decrementar Servo.x
else
    incrementar Servo.x
end if
if Co.y > Ci.y + varFolga then
    decrementar posição Servo.y
else
    incrementar posição Servo.y
end if
```

5.4 EXPERIMENTOS E TESTES

Com o objetivo de testar o sistema de visão artificial proposto, dois experimentos foram realizados:

- No primeiro experimento, imagens foram classificadas em função da iluminação da cena com o objetivo de avaliar a robustez do classificador frente a variações de luminosidade do ambiente;
- No segundo experimento, os objetos são localizados em um suporte de calibração e os valores relativos às coordenadas polares reais (θ, d) e estimadas $(\hat{\theta}, \hat{d})$ (pelo processo de fusão de sensores) são determinados e avaliados.

No primeiro experimento, um luxímetro é utilizado para medição da luminosidade da cena (robôs e bola). Os objetos são localizados no cone de visão do sistema e correspondem a amostras semelhantes às da Fig. 5.7(a). Nenhum tipo de controle de iluminação é utilizado e a variação da intensidade de luz é obtida através do aumento ou diminuição da quantidade das fontes de luz difusas no ambiente.

No segundo experimento, um suporte de calibração foi utilizado, conforme ilustrado na Fig. 5.24. O objetivo do suporte é avaliar o erro entre o valor real e o valor obtido

pela rotina de determinação do ângulo a partir da imagem.

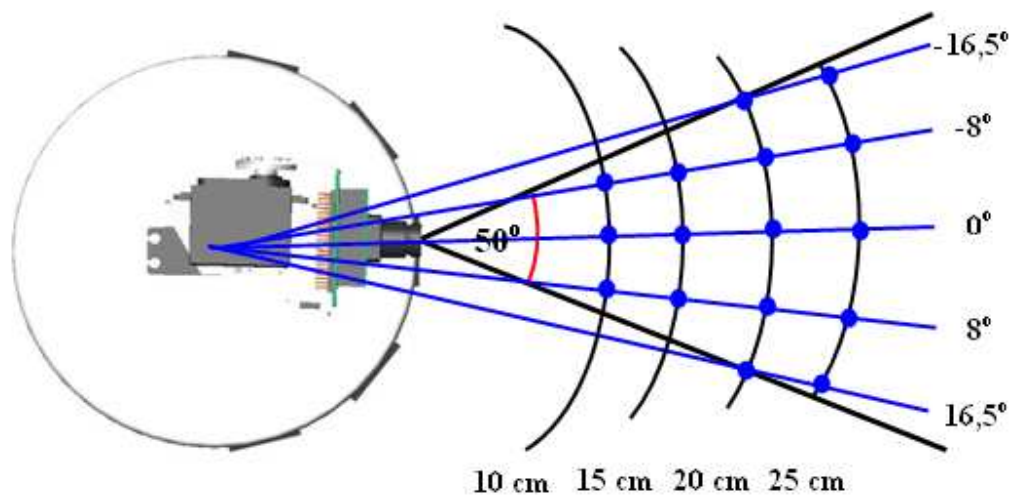


Figura 5.24. Suporte de calibração utilizado para avaliação da precisão do processador de fusão de sensores

Dessa forma, foram traçadas curvas de nível correspondentes às distâncias entre 10 e 55 cm. O objetivo de tais curvas de nível é avaliar de forma sistemática o comportamento do módulo responsável pela determinação dos ângulos a partir da imagem. Analisando a resposta do sistema para cada distância nos diferentes raios a partir do centro do anel inferior do robô ($-16,5^\circ$, -8° , 0° , 8° , $16,5^\circ$), bem como para quatro diferentes ângulos da cabeça de visão (20° , 35° , 50° , 65°), foi possível avaliar seu erro em diferentes situações.

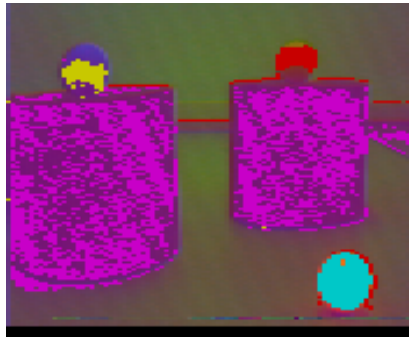
5.4.1 Resultados

A Tabela 5.10 apresenta a taxa de acerto de classificação em função da luminosidade. Para amostras de imagens de objetos diferentes das utilizadas para treinamento, este coeficiente indica a precisão de pixels classificados corretamente pelo classificador.

A faixa de luminosidade testada se encontra entre 570 and 980 luxs e está de acordo com as regras da Robocup (ROBOCUP, 2005) que define uma iluminação com variação entre 700 e 1000 luxs, para cada torneio. A Fig. 5.25 ilustra o resultado da classificação dos objetos de uma cena.

Tabela 5.10. Resultados em diferentes faixas de luz

Iluminação (lux)	Precisão
570	87.8%
660	84.1%
780	84.9%
800	86.4%
920	87.7%
980	90.0%

**Figura 5.25.** Resultado da classificação. Objetos classificados pela MVS. Cada cor representa um objeto classificado

Para avaliar a determinação dos ângulos pelo sistema, a Tabela 5.11 foi construída com os dados obtidos (ângulos em graus).

A partir da Tabela 5.11, os erros absolutos foram calculados entre os ângulos reais e as médias dos ângulos medidos em cada curva de nível. A Tabela 5.12 apresenta os resultados obtidos.

A partir da Tabela 5.12, pode-se observar que o somatório dos erros é diferente de zero. O valor ideal para os somatórios dos erros é próximo de zero. Como isto não ocorre, é possível se obter um valor de correção que multiplicado por cada valor de erro absoluto torne a soma próxima de zero. Os valores para correção dos erros absolutos (em graus) são apresentados na Tabela 5.13.

A Fig. 5.26 apresenta o gráfico correspondentes aos pontos da Tabela 5.13. A

Tabela 5.11. Dados levantados para 4 ângulos de cabeça de visão

Cabeça de visão = 20 graus								
Ângulo Real	Em 25cm	Em 30cm	Em 35cm	Em 40cm	Em 45cm	Em 50cm	Em 55cm	Média
-16,5	-16	-15	-15	-15	-15	-15	-15	-15,1
-8	-9	-10	-9	-9	-9	-9	-9	-9,1
0	-1	-1	-1	-1	-1	-1	-1	-1
8	5	4	4	5	5	5	5	4,3
16,5	14	14	14	14	14	14	14	14
Cabeça de visão = 35 graus								
Ângulo Real	Em 25cm	Em 30cm	Em 35cm	Em 40cm	Em 45cm	Em 50cm	Em 55cm	Média
-16,5	-15	-15	-16	-16	-16	-16	-16	-15,9
-8	-10	-10	-10	-10	-10	-10	-10	-10
0	-1	-1	-1	-1	-1	-1	-1	-1
8	5	4	4	6	6	6	6	5,6
16,5	14	14	14	14	14	14	14	14
Cabeça de visão = 50 graus								
Ângulo Real	Em 10cm	Em 15cm	Em 20cm					Média
-16,5	-18	-18	-17					-17,7
-8	-11	-10	-11					-10,7
0	-2	-2	-2					-2
8	4	5	7					5,3
16,5	13	12	12					12,3
Cabeça de visão = 65 graus								
Ângulo Real	Em 10cm	Em 15cm						Média
-16,5	-17	-17						-17
-8	-8	-9						-8,5
0	-1	-2						-1,5
8	4	4						4
16,5	13	12						12,5

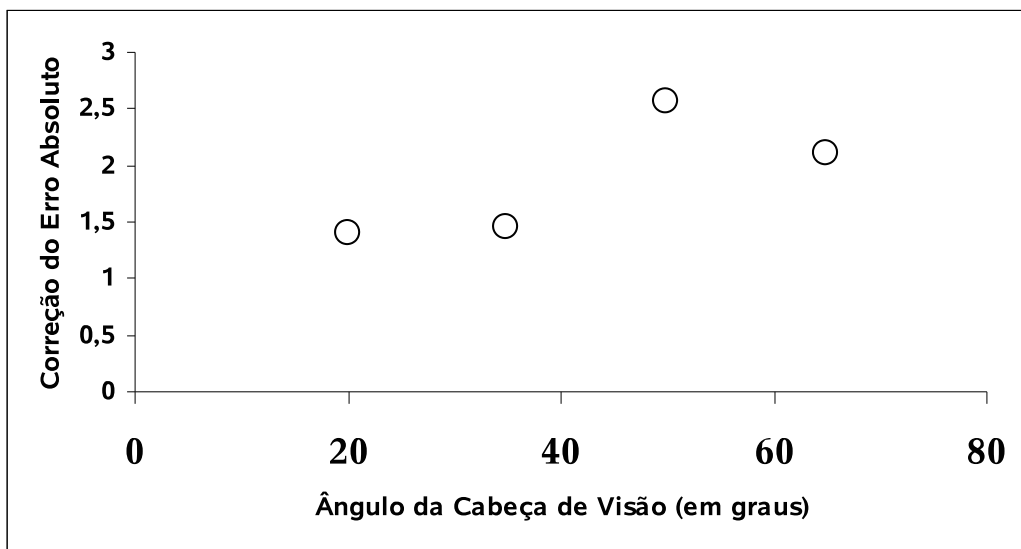
Tabela 5.12. Dados levantados para 4 ângulos de cabeça de visão

Cabeça de visão = 20 graus		
Ângulo Real	Média dos Ângulos estimados	Erro Absoluto
-16,5	-15,1	-1,4
-8	-9,1	1,1
0	-1	1,0
8	4,3	3,7
16,5	14	2,5
	Soma	7,0
Cabeça de visão = 35 graus		
Ângulo Real	Média dos Ângulos estimados	Erro Absoluto
-16,5	-15,9	-0,6
-8	-10	2,0
0	-1	1,0
8	5,6	2,4
16,5	14	2,5
	Soma	7,3
Cabeça de visão = 50 graus		
Ângulo Real	Média dos Ângulos estimados	Erro Absoluto
-16,5	-17,7	1,2
-8	-10,7	2,7
0	-2	2,0
8	5,3	2,7
16,5	12,3	4,2
	Soma	12,7
Cabeça de visão = 65 graus		
Ângulo Real	Média dos Ângulos estimados	Erro Absoluto
-16,5	-17	0,5
-8	-8,5	0,5
0	-1,5	1,5
8	4	4,0
16,5	12,5	4,0
	Soma	10,5

Tabela 5.13. Correção do erro absoluto em cada ângulo da cabeça de visão

Ângulo da cabeça de visão	Correção do erro
20	1,4
35	1,45
50	2,56
65	2,1
Média	1,88

média dos valores de correção é de 1,88 graus e deve ser acrescida aos ângulos obtidos pelo sistema de visão, a fim de corrigir seus valores.

**Figura 5.26.** Gráfico do ângulo da cabeça de visão x correção do erro absoluto

Pode-se observar que os valores para correção dos erros absolutos são todos positivos, o que indica um erro sistemático para os ângulos positivos, ou seja, o sistema possui um erro menor quando mede os ângulos do seu lado esquerdo, já que a diferença do valor de correção e o erro deve se aproximar de zero (ver Fig. 5.24). Esse fato pode ser justificado por algumas razões:

- **Distorção radial da lente da câmera** - maior distorção radial da lente da câmera para o lado direito
- **Papel de calibração** - Erro na construção do suporte de calibração

- **Iluminação** - Áreas de sombra podem modificar a determinação da centróide do objeto e por conseguinte alterar o ângulo obtido pelo sistema
- **Rotina de treinamento** - como a rotina de treinamento aplica um método determinístico/estatístico (MVS), existe uma margem de imprecisão na generalização

Apesar do erro sistemático, o valor para a correção de erro no sistema é pequeno (1,88 graus), o que demonstra a alta precisão no módulo de determinação de ângulos a partir da imagem

CAPÍTULO 6

CONCLUSÕES E PERSPECTIVAS

Um sistema de visão artificial aplicado em robótica móvel e baseado em fusão de sensores é apresentado nesta dissertação. O Capítulo 2 contextualiza a aplicação proposta nesta dissertação. Os Capítulos 3 e 4 representam o núcleo da dissertação: visão artificial e fusão de sensores. No Capítulo 5, aspectos da implementação são apresentados e os resultados gerais analisados.

A técnica de MVS é empregada tanto para o reconhecimento de padrões na imagem quanto na extração de atributos de decisão dos objetos e sua aplicação se justifica pelo baixo custo computacional e alta capacidade de generalização.

Para a fusão de sensores, a lógica difusa é utilizada. A aplicação desta técnica reside no fato de facilitar a tomada de decisão em regiões de ponderação, permitindo, assim, o tratamento de incerteza no processo de fusão.

O sistema proposto tem como objetivos principais o reconhecimento e localização dos objetos no campo de futebol. Apesar desses objetivos serem aplicados no ambiente de experimentação de futebol de robôs, seus princípios e arquitetura podem ser utilizados em outras áreas, por exemplo: na indústria, para inspeção e localização de objetos coloridos; na área de interfaces de máquina, como reconhecedor de gestos humanos; na área automotiva, para pilotagem automática de veículos em sistemas com direção automática; na indústria de alimentos, para inspeção de tamanho de frutas; não sendo esta uma lista exaustiva.

Os dados levantados no experimento para avaliação do módulo de localização demonstraram uma necessidade de correção de 1,88 graus no valores de ângulo fornecidos pelo sistema, justificado por um (ou mais) dos quatro fatores: distorção radial da lente da câmera, erro no papel de calibração, áreas de sombra provenientes da iluminação

difusa e imprecisão na rotina de treinamento da MVS de regressão. Apesar da necessidade de ajuste dos ângulos obtidos, os resultados evidenciam um bom desempenho do sistema para a localização dos objetos no campo.

6.1 PERSPECTIVAS

- Aplicação do sistema proposto em um robô físico real.
- Reconhecimento dos marcos estáticos: linhas e paredes.
- Aplicação de etapas que tornem o reconhecimento dos objetos mais robusto frente a variações maiores de luminosidade.
- Avaliação e possível alteração da disposição dos sensores no robô físico, com o mecanismo de chute.
- Diminuição no tempo de aquisição de imagens. Uma vez que este é o ponto crítico do sistema (tempo elevado), o teste com outros sensores de visão, bem como a alteração do hardware de aquisição da imagem da câmera, através do desenvolvimento de circuitos digitais com maior capacidade de transmissão, tornam-se alternativas para a melhoria da taxa de aquisição da imagem.

6.2 PUBLICAÇÕES

OLIVEIRA, L.; COSTA, A. *Visão Computacional: Uma Abordagem para Robótica Móvel baseada em Fusão de Sensores*. In: XXIV Congresso Brasileiro da Sociedade Brasileira de Computação, 2004, Salvador. I Encontro de Robótica Inteligente. 2004, v. 1, p. 1957-1964.

OLIVEIRA, L. et al. *An Artificial Vision Prototype Based on Sensor Fusion For Mobile Robot's Obstacle Localization*. In: 50. Internationales Wissenschaftliches Kolloquium. Ilmenau, Alemanha, 2005.

OLIVEIRA, L.; COSTA, A.; SCHNITMAN, L. *Uma Arquitetura de Visão Artificial: Aplicação em Robótica Móvel Baseada em Fusão de Sensores*. In: VII Simpósio Brasileiro de Automação Inteligente/2005 Latin American IEEE Robotics Symposium. Maranhão, Brasil, 2005.

OLIVEIRA, L.; COSTA, A.; SCHNITMAN, L. *An Architecture of Sensor Fusion for Spatial Location of Objects in Mobile Robotics*. In: 12Th Portuguese Conference on Artificial Intelligence (IROBOT 2005). Coimbra, Portugal, 2005. Springer, Lecture Notes in Artificial Intelligence.

REFERÊNCIAS

- ANISIMOVICH, K. et al. Using combination of structural, feature and raster classifiers for recognition of handprinted characters. In: *International Conference on Document Analysis and Recognition*. USA: IEEE, 1997. v. 1, p. 881–885.
- AXEBOT. *Axebot Project*. 2005. [Http://www.axebot.ufba.br](http://www.axebot.ufba.br).
- BALA, J. et al. Hibrid learning using genetic algorithms and decision trees for patten recognition. In: *IJCAI*. [S.l.: s.n.], 1995. v. 1.
- BASTOS, T.; SARCINELLI, M.; FREITAS, R. A multi-sensorial integration scheme to help mobile robot navigation through obstacle recognition. In: *International Conference on Emergin Technology and Factory Automation*. USA: IEEE, 1999. v. 1, p. 549–558.
- BONCI, A.; LEO, T.; LONGHI, S. Ultrasonic and video data fusion for mobile robot navigation. In: *Proceedings of the 10th Mediterranean Conference on Control and Automation*. [S.l.: s.n.], 2002.
- BRUDER, S.; FARONQ, M.; BAYOUMI, M. Robotic heterogenous multi-sensor fusion with spatial and temporal alignment. In: *Proceedings of the 30th IEEE Conference on Decision and Controle*. [S.l.]: IEEE, 1991. p. 506–511.
- CHANG, C.-C.; LIN, C.-J. *LIBSVM: A Library for Support Vector Machines*. 2001. Biblioteca disponível em: <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- CMUCAM2. *The Cmucam2*. 2005. Disponível em <http://www-2.cs.cmu.edu/cmucam/cmucam2/>. Acessado em 03 abril 2005.
- COMANICIU, D.; MEER, P. Mean shift: A robust approach toward feature space analysis. In: IEEE (Ed.). *IEEE Transactions Pattern Analysis and Machine Intelligence*. [S.l.: s.n.], 2002. v. 24, n. 5, p. 603–619.
- COSTA, A.; BITTENCOURT, G. From a concurrent architecture to a concurrent autonomous agents architecture. In: *IJCAI'99, Third International Workshop in RoboCup*. [S.l.: s.n.], 1999. p. 85–90. Springer, Lecture Notes in Artificial Inteligence.
- CRISTIANINI, N.; SHAWE-TAYLOR, J. *An Introduction to Support Vector Machine an Other Kernel-based Learning Algorithms*. [S.l.: s.n.], 2003.

- DAHYOT, R.; CHARBONNIER, P.; HEITZ, F. Robust visual recognition of color images. In: *Computer Vision and Pattern Recognition*. [S.l.]: IEEE, 2000. v. 1, p. 1685–1690.
- DASARATHY, B. Sensor fusion potential exploitation – innovative and illustrative applications. In: *Artigo convidado em Proceedings of the IEEE Special Issue on Sensor Fusion*. [S.l.]: IEEE, 1997. v. 85, n. 1, p. 24–38.
- DORNAIKA, F.; CHUNG, R. Cooperative stereo-motion: Matching and reconstruction. In: *Computer Vision and Image Understanding*. [S.l.: s.n.], 2000. v. 79, p. 408–427.
- DROLET, L.; MICHAUD, F.; COTE, J. Adaptable sensor fusion using multiple kalman filters. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. Japão: IEEE/RSJ, 2000. v. 2, p. 1434–1439.
- DUDA, R.; HART, P.; STORK, D. *Pattern Classification*. [S.l.]: Book News Inc., 2000.
- ENRIQUEZ, J.; ROBLES, L. Local estimation fusion for tracking objects under occlusion. In: *Fifth Mexican International Conference in Computer Science*. [S.l.]: IEEE, 2004. p. 256–261.
- FACELI, K.; SARCINELLI, M.; FREITAS, R. Combining intelligent techniques for sensor fusion. In: *International Conference on Neural Information Processing*. EUA: IEEE, 2002. v. 4, p. 1998–2002.
- FANG, J.; QIU, G. Face detection based on multiple regression and recognition support vector machines. In: *British Machine Vision Conference*. Inglaterra: [s.n.], 2003.
- FERAUD, R. et al. A fast and accurate face detector based on neural networks. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*. [S.l.]: IEEE, 2001. p. 42–53.
- FINLAYSON, G.; SCHIELE, B.; CROWLEY, J. Comprehensive colour normalization. In: *ECCV'98 Fifth European Conference on Computer Visio*. [S.l.: s.n.], 1998.
- FORSYTH, D.; PONCE, J. *Computer Vision: A Modern Approach*. [S.l.: s.n.], 2003.
- FRANCA, J. *Desenvolvimento de Algoritmos de Visão Estereoscópica para Aplicações em Robótica Móvel*. Tese (Doutorado) — Universidade Federal de Santa Catarina, 2003.
- GP2D02. *Manual de Dados de GP2D02*. [S.l.], 2005. Disponível em <http://www.acroname.com/robotics/parts/R19-IR02.html>. Acessado em 04 abril 2005.

- GROTTA, S. Anatomy of a digital camera: Image sensors. *Revista Eletrônica Extreme Tech*. Disponível em <http://www.extremetech.com/article2/0,3973,1157572,00.asp>. Acesso em 12 de março de 2005. 2003.
- GU, J. et al. Sensor fusion in mobile robot: Some perspectives. In: *Congress on Intelligent Control and Automation*. EUA: IEEE, 2002. v. 2, p. 1194–1199.
- HALL, D.; LLINAS, J. *Handbook of multisensor data fusion*. [S.l.: s.n.], 2001.
- HAYKIN, S. *Redes Neurais: Princípios e Práticas*. 2. ed. [S.l.: s.n.], 2001.
- HORN, B. *Robot Vision*. [S.l.]: The MIT Press, 1986.
- HOSEINNEZHAD, R.; ASHARIF, M. Sensor fusion for ultrasonic and laser arrays in mobile robotics: A comparative study of fuzzy, dempster and bayesian approaches. In: *IEEE SENSORS 2002 Conference*. EUA: IEEE, 2002.
- HSU, C.-W.; LIN, C.-J. A comparison methods for multi-class support vector machine. In: IEEE (Ed.). *IEEE Transactions on Neural Networks*. [S.l.: s.n.], 2002. v. 13, n. 2, p. 415–425.
- HUNDELSHAUSEN, F.; BEHNKE, S.; ROJAS, R. An omnidirectional vision system that finds and tracks color edges and blobs. In: *Proceedings of Fifth RoboCup International Symposium*. [S.l.: s.n.], 2001. p. 374–379.
- INTEL. *Intel PXA255 Processor: Design Guide*. março 2003. Disponível em <ftp://download.intel.com/design/pca/applicationsprocessors/manuals/278694-001.pdf>. Acessado em 30 março 2005.
- JAIN, A.; DUIN, R.; MAO, J. Statistical pattern recognition: A review. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*. [S.l.]: IEEE, 2000.
- JANG, J.; SUN, C.; MIZUTANI, E. *Neuro-fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*. [S.l.]: Prentice Hall, 1997.
- JOLION, J. Computer vision methodologies. In: *Computer Vision and Image Understanding*. [S.l.]: Ideal Library, 1994. p. 53–71.
- KATO, M. S. N. et al. A handwritten character recognition system using directional element feature and asymmetric mahalanobis distance. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*. [S.l.]: IEEE, 1999. p. 258–262.
- KATO, T.; NINOMIYA, Y.; MASAKI, F. An obstacle detection method by fusion of radar and motion stereo. In: *IEEE Transactions on Intelligent Transportation Systems*. EUA: IEEE, 2002. v. 3, p. 182–188.
- KECMAN, V. *Learning and Soft Computing*. [S.l.: s.n.], 2001.

- KITADAI, A.; NAKAGAWA, M. A learning algorithm for structured character pattern representation used in on-line recognition of handwritten japanese characters. In: *Eight International Workshop on Frontiers in Handwriting*. [S.l.]: IEEE, 2002. p. 163–168.
- KORTENKAMP, D.; BONASSO, R.; MURPHY, R. *Artificial Intelligence and Mobile Robots*. Menlo Park: [s.n.], 1998.
- KROSE, B. et al. Environment modeling and localization from an omnidirectional vision system. In: *IEEE Robotics and Automation Magazine*. EUA: IEEE, 2004. v. 11, n. 4, p. 45–52.
- KULKARNI, A. *Computer Vision and Fuzzy-Neural Systems*. [S.l.: s.n.], 2001.
- KURITA, T.; TAKAHASHI, T.; IKEDA, Y. A neural network classifier for occluded images. In: *International Conference on Pattern Recognition*. [S.l.]: IEEE, 2002. v. 3, p. 30045–30048.
- LAMPINEN, J.; OJA, E. Distortion tolerant pattern recognition based on self-organizing feature extraction. In: *IEEE Transactions on Neural Networks*. [S.l.]: IEEE, 1995.
- LIN, C. et al. A new approach to automatic reconstruction of a 3-d world using active stereo vision. In: *Computer Vision and Image Understanding*. [S.l.]: Academic Press, 2002. p. 117–143.
- LINGRAND, D. An exhaustive study of particular cases leading to robust and accurate motion estimation. In: *Computer Vision and Image Understanding*. [S.l.]: Academic Press, 2002. p. 159–188.
- LIU, C.; JAEGER, S. Online recognition of chinese characters: The state-of-the-art. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*. [S.l.]: IEEE, 2004. p. 198–213.
- LOY, G. et al. An adaptive fusion architecture for target tracking. In: *Fifth IEEE International Conference on Automatic Face and Gesture Recognition*. [S.l.]: IEEE, 2002. p. 261–266.
- MACHADO, A. *Metodologias para Reconhecimento de Padrões em Visão Computacional*. Dissertação (Mestrado) — UFMG, Belo Horizonte, 1994.
- MAKI, A.; NORDLUND, P.; EKLUNDH, J.-O. Attentional scene segmentation: Integrating depth and motion. In: *Computer Vision and Image Understanding*. [S.l.: s.n.], 2000. v. 78, p. 351–373.
- MAMDANI, E.; ASSILIAN, S. An experiment in linguistic synthesis with a fuzzy logic controller. In: *International Journal of Man-Machine Studies*. [S.l.: s.n.], 1975. v. 7, n. 1, p. 1–13.

- MARR, D. Vision. In: _____. [S.l.]: Freeman e Co, 1982. cap. Understanding Complex Information Processing Systems, p. 19–24.
- MECATEAM. *Mecateam*. 2005. [Http://www.axebot.ufba.br](http://www.axebot.ufba.br). Acessado em 06 de março de 2005.
- MINSKY, M. A framework to represent knowledge. In: *The Psychology of Computer Vision*. [S.l.]: McGraw-Hill, 1975. p. 211–277.
- MITZIAS, B. M. D. A neural multiclassifier system for object recognition in robotic vision applications. In: *Measurement*. [S.l.]: Elsevier, 2004. p. 315–330.
- MOGHADDAM, B.; JEBARA, T.; PENTLAND, A. Bayesian face recognition. In: *Pattern Recognition*. [S.l.: s.n.], 2000. v. 33, n. 11, p. 1771–1782.
- NASHMAN, M. et al. *A Unique Sensor Fusion System for Coordinate Measuring Machine Tasks*. [S.l.], abril 1998.
- PAL, N.; MANDAL, G.; KUMAR, E. Comments on a fuzzy neural network and its application to pattern recognition. In: *IEEE Transactions on Fuzzy Systems*. [S.l.]: IEEE, 1999. v. 7, n. 4.
- POLIUSP. 2001. [Http://www.pmr.poli.usp.br/lpa/research/omnidirecional.asp](http://www.pmr.poli.usp.br/lpa/research/omnidirecional.asp). Acessado em 03 abril 2005.
- QUINLAN, S. C. M.; MIDDLETON, R. Techniques for improving vision and locomotion on the sony aibo robot. In: *Australasian Conference on Robotics and Automation (ACRA)*. [S.l.: s.n.], 2003.
- REALI, A.; BIANCHI, R. A distributed control architecture for a purposive computer vision system. In: *IEEE International Joint Symposia on Intelligence and Systems (IJSIS '96)*. [S.l.]: IEEE, 1996.
- REALI, A. et al. Integrando visão e comportamento: uma aplicação de reconstrução propositiva. In: *XI Congresso Brasileiro de Automática (CBA96)*. [S.l.]: SBA, 1996. p. 573–578.
- RIVLIN, E.; ALOIMONOS, Y.; ROSENFELD, A. *Purposive Recognition: a framework*. [S.l.], 1991.
- ROBOCUP. *Laws of the F180 League 2004 – Release v3.00a*. 2004. Disponível em <http://www.itee.uq.edu.au/wyeth/F180%20Rules/index.htm>. Acessado em 30 março 2005.
- ROBOCUP. *Sítio da Robocup*. 2005. Disponível em <http://www.robocup.org>. Acessado em 30 março 2005.
- ROWE, A. *CMUcam2 Vision Sensor*. [S.l.], 2003.

- SHAMAIE, A. Bayesian fusion of hidden markov models for understanding bimanual movements. In: IEEE (Ed.). *IEEE International Conference on Automatic Face and Gesture Recognition*. [S.l.: s.n.], 2004. p. 602–607.
- SHAPIRO, L.; STOCKMAN, G. *Computer Vision*. [S.l.]: Prentice Hall, 2000.
- SINGHAL, A.; BROWN, C. Dynamic bayes net approach to multimodal sensor fusion. In: *Proceedings of the SPIE conference on Sensor Fusion and Decentralized Control*. [S.l.: s.n.], 1997.
- TAKAGI, T.; SUGENO, M. Fuzzy identification of systems and its applications to modelling and control. In: IEEE (Ed.). *IEEE Transactions on Systems, Man and Cybernetics*. [S.l.: s.n.], 1985. v. 28, p. 15–33.
- TARABANIS, A.; ALLEN, P.; TSAI, R. A survey of sensor planning in computer vision. In: IEEE (Ed.). *Transactions on Robotics and Automation*. [S.l.: s.n.], 1995. v. 11, n. 1.
- TSOUKALAS, L.; UHRIG, R. *Fuzzy and Neural Approaches in Engineering*. [S.l.]: John Wiley and Sons, Inc, 1997.
- TSUKAMOTO, Y. An approach to fuzzy reasoning method. In: *Advances in Fuzzy Set Theory and Applications*. [S.l.: s.n.], 1979. p. 137–149.
- VAPNIK, V. *The Nature of Statistical Learning Theory*. [S.l.]: Springer Verlag, 1995.
- WALLRAVEN, C.; CAPUTO, B.; GRAF, A. Recognition with local features: the kernel recipe. In: *IEEE International Ninth Conference on Computer Vision*. [S.l.: s.n.], 2003. v. 1, p. 257–264.
- WEBB, A. *Statistical Pattern Recognition*. Malvern, UK: QinetiQ Ltd., 2002.
- WECHESSER, P.; DILMAN, R. Navigating a mobile service-robot in a natural environment using sensor-fusion techniques. In: *International Conference on Intelligent Robots and System*. EUA: IEEE, 1997. v. 3, p. 1423–1428.
- WU, H.; SIEGEL, M.; ABLAY, S. Sensor fusion using dempster-shafer theory ii: static weighting and kalman filter-like dynamic weighting. In: IEEE (Ed.). *Proceedings of the 20th IEEE Instrumentation and Measurement Technology Conference*. [S.l.: s.n.], 2003.
- WU, H. et al. Sensor fusion using dempster-shafer theory. In: IEEE (Ed.). *Proceedings of the 19th IEEE Instrumentation and Measurement Technology Conference*. [S.l.: s.n.], 2002.
- ZADEH, L. Fuzzy sets. In: *Information Sciences*. [S.l.: s.n.], 1965. p. 338–353.
- ZADEH, L. The concept of a linguistic variable and its application to approximate reasoning – part i. In: *Information Sciences*. [S.l.: s.n.], 1975. p. 199–249.

REFERÊNCIAS

ZHANG, Z. Determining the epipolar geometry and its uncertainty: A review. In: *INRIA*. [S.l.: s.n.], 1996.