

CENTRO UNIVERSITÁRIO DA FEI

CLAUDIO DE OLIVEIRA VILÃO JUNIOR

**UM SISTEMA DE VISÃO COMPUTACIONAL MONOCULAR PARA UM ROBÔ
MÓVEL HUMANOIDE**

São Bernardo do Campo
2015

CLAUDIO DE OLIVEIRA VILÃO JUNIOR

**UM SISTEMA DE VISÃO COMPUTACIONAL MONOCULAR PARA UM ROBÔ
MÓVEL HUMANOIDE**

Dissertação de Mestrado apresentada ao Centro Universitário da FEI para obtenção do título de Mestre em Engenharia Elétrica, orientado pelo Prof. Dr. Reinaldo Augusto da Costa Bianchi.

São Bernardo do Campo
2015

Vilão Júnior, Claudio de Oliveira

Um sistema de visão computacional monocular para um robô móvel
humanoide / Claudio de Oliveira Vilão Júnior. São Bernardo do Campo,
2015.

105 f. ill.

Dissertação - Centro Universitário da FEI.

Orientador: Prof. Dr. Reinaldo Augusto da Costa Bianchi.

1. Robô humanoide. 2. Visão Computacional. 3. Rastreamento de
objetos. I. Bianchi, Reinaldo Augusto da Costa, orient. II. Título.

CDU 658.52.011.56



CENTRO UNIVERSITÁRIO DA FEI

APRESENTAÇÃO DE DISSERTAÇÃO ATA DA BANCA EXAMINADORA

Mestrado

Programa de Pós-Graduação Stricto Sensu em Engenharia Elétrica

PGE-10

Aluno: Claudio de Oliveira Vilão Júnior

Matrícula: 113106-9

Título do Trabalho: Um sistema de visão computacional monocular para um robô móvel humanoide.

Área de Concentração: Inteligência Artificial Aplicada à Automação

Orientador: Prof. Dr. Reinaldo Augusto da Costa Bianchi

Data da realização da defesa: 01/10/2015

ORIGINAL ASSINADA

Avaliação da Banca Examinadora:

São Bernardo do Campo, / / .

MEMBROS DA BANCA EXAMINADORA

Prof. Dr. Reinaldo Augusto da Costa Bianchi Ass.: _____

Prof. Dr. Flávio Tonidandel Ass.: _____

Prof.ª Dr.ª Roseli Aparecida Francelin Romero Ass.: _____

A Banca Julgadora acima-assinada atribuiu ao aluno o seguinte resultado:

APROVADO

REPROVADO

VERSÃO FINAL DA DISSERTAÇÃO

**APROVO A VERSÃO FINAL DA DISSERTAÇÃO EM QUE
FORAM INCLUÍDAS AS RECOMENDAÇÕES DA BANCA
EXAMINADORA**

Aprovação do Coordenador do Programa de Pós-graduação

Prof. Dr. Carlos Eduardo Thomaz

A meus pais.

AGRADECIMENTOS

Agradeço pelo incentivo que me foi dado em momentos de dificuldade por todos os meus familiares, em especial meus pais Anna Maria Cortezi Oliveira Vilão e Claudio de Oliveira Vilão.

Preciso agradecer, com uma ênfase maior, o Prof. Dr. Reinaldo Augusto da Costa Bianchi por me ajudar e suportar em todos os momentos que passei durante este período, mas principalmente por acreditar no meu trabalho e em mim. Não posso deixar de mencionar, e talvez nem precise, mas, gostaria de agradecer pelas oportunidades que tem me dado, além do apoio, da motivação, das críticas construtivas e das aulas. Muito obrigado Professor.

Aos professores do Centro Universitário da FEI pelo incentivos e pelas aulas ministradas: Prof. Dr. Carlos Eduardo Thomaz, Prof. Dr. Flávio Tonidandel, Prof. Dr. Paulo Sérgio Silva Rodrigues, Prof. Dr. Paulo E. Santos.

Agradeço ao colega Ms. Isaac Jesus da Silva por sua contribuição me ajudando nas pesquisas científicas que vem sendo desenvolvidas com sugestões e críticas construtivas. Agradeço aos colegas Ms. Danilo Hernani Perico e Ms. Thiago Pedro Donadon Homem por terem me ajudado com sugestões e críticas, mas também transmitindo motivação e foco.

Agradeço à minha esposa Mônica Farias da Silva pelo suporte e paciência, além de auxílio e dicas do tratamento de imagens.

Agradeço a Deus.

“A good scientist is a person with original ideas.”
Freeman Dyson

RESUMO

Uma única imagem representa um conjunto de dados de tamanho considerável e tipicamente várias operações precisam ser feitas em cada pixel da referida imagem. Em uma estrutura de vídeo, a qual pode ser descrita como uma sucessão de várias imagens, esta tarefa torna-se ainda mais difícil, já que a taxa de quadros analisadas necessita ser mantida mesmo com a câmera em movimento. Este trabalho descreve um sistema de visão monocular para quatro robôs humanoides desenvolvidos para participar da liga humanoide na categoria KidSize, da RoboCup. O sistema de visão proposto permite que os robôs acompanhem uma bola e detectem companheiros e adversários, fornecendo informações como distâncias e orientações de todos esses objetos simultaneamente, de forma que, todos os processos possam ser executados em tempo real com diferentes resoluções de câmera. Devido a mudança constante de regras da competição, aumentando cada vez mais a complexidade do ambiente, o uso de técnicas de alto nível começam a parecer atraentes. Dessa forma o uso do Haar-Adaboost e do HOG-SVM para detecção de objetos pertencentes ao jogo, apresentaram resultados relevantes. Técnicas de baixo, médio e alto nível foram utilizadas em nossos robôs com poucas adversidades, com taxa de quadros por segundo condizentes com um robô de ação rápida e com a capacidade de generalização e identificação dos objetos demonstrados nas Curvas de Características de Operação do Receptor (ROC).

Palavras-chave: Robô humanoide, Visão computacional, Controle servovisual, Rastreamento de objetos, Reconhecimento de objetos

ABSTRACT

A single image represents a dataset of considerable size, and typically several operations has to be done in each pixel of that image. In a video frame, which can be assimilated to a succession of multiple images, this task becomes even more challenging, due to shaking camera, and since the frame rate per second needs to be maintained. This paper aims to describe a monocular vision system for four humanoid robots developed to participate in the humanoid league in the category Kid Size, RoboCup. The proposed vision system allows robots to be able to keep tracking of a ball and detect teammates and opponents, providing information such as distances and estimated orientations of all these objects simultaneously. It is possible for all threads to run in real time with different camera resolutions. Due to constantly changing competition rules which has continually increased the environment complexity, the use of high-level techniques gradually becomes more attractive. Thus the use of Haar-AdaBoost and HOG-SVM to detect objects belonging to the game, presented relevant results. All levels of techniques were used in our robots with few adversities, with frame rate per second consistent with a fast-acting robot and the ability to generalize and identify the soccer game belonging objects. All results are shown in the Receiver Operating Characteristics Curves (ROC).

Keywords: Humanoid robot, Computational vision, Visual servoing, Object tracking, Object recognition

LISTA DE ILUSTRAÇÕES

Ilustração 1 – Robôs desenvolvidos.	11
Ilustração 2 – Exemplo Filtro Gaussiana.	21
Ilustração 3 – Segmentação do Laranja usando como entrada uma imagem no modelo de cores HSV.	22
Ilustração 4 – Exemplo de erosão e dilatação para um objeto circular, com elemento estruturante circular.	23
Ilustração 5 – Divisão de regiões para aplicação do gradientes.	29
Ilustração 6 – Representação esquemática da classificação em duas dimensões do SVM.	30
Ilustração 7 – Um exemplo da maximização da margem.	31
Ilustração 8 – Algumas características simples usadas para descrever objetos.	33
Ilustração 9 – Características usadas para descrever objetos.	34
Ilustração 10 –Um exemplo de fronteira complexa.	36
Ilustração 11 –Um exemplo de vários classificadores mais simples ou mais fracos trabalhando em conjunto para determinação de fronteiras mais complexas.	37
Ilustração 12 –Planos que cortam o ser humano utilizados em robôs humanoides.	47
Ilustração 13 –A bola utilizada em 2014.	48
Ilustração 14 –A bola utilizada em 2015.	50
Ilustração 15 –A diversidade de robôs da liga humanoide.	52
Ilustração 16 –Divisão esquemática do quadro em 9 regiões.	56
Ilustração 17 –Visão Geral do Software a ser Implementado.	59
Ilustração 18 –O Robô Darwin Modificado.	61
Ilustração 19 –O robô Darwin seguindo uma bola parcialmente oculta pelo seu braço.	63
Ilustração 20 –Robô Milton seguindo a bola com a influência da luz do sol e sombras.	63
Ilustração 21 –B1 utilizando a condensação para identificar probabilisticamente e qualitativamente a posição da bola laranja no quadro.	64
Ilustração 22 –Rastreamento da bola usando características HAAR em um campo de futebol sobre carregado de objetos, pessoas e regiões não pertencentes ao jogo.	66
Ilustração 23 –Curva ROC para detecção da bola usando características HAAR.	66
Ilustração 24 –Função relacionando pixels com distância de uma bola de tamanho conhecido.	67

Ilustração 25 –Curvas ROC. A linha verde mostra o classificador clássico HOG-SVM, usado para reconhecimento de pessoas. A linha vermelha demonstra o mesmo classificador treinado com pessoas usado para reconhecer robôs e a linha amarela mostra o classificador treinado com 100 imagens de robôs positivas e 100 imagens negativas usado na classificação dos robôs.	69
Ilustração 26 –Detecção do robô Milton usando imagens de pessoas como conjunto de treinamento.	70
Ilustração 27 –Rastreamento do robô B2 usando características HAAR no campo de futebol com grama artificial.	71
Ilustração 28 –Identificação do Robô Bold Hearts (BOLD..., 2014b) da universidade de Hertfordshire usando HOG-SVM (Figura a) e HAAR-AdaBoost (Figura b).	72
Ilustração 29 –Curvas ROC. A linha verde mostra o algoritmo HOG-SVM clássico treinado com imagens de robôs usado para detecção de robôs. A linha vermelha demonstra o resultado do classificador HAAR-AdaBoost treinado com o mesmo propósito.	73
Ilustração 30 –Desempenho dos descritores HAAR-AdaBoost e HOG-SVM em termos de quadros por segundo após processamento.	74
Ilustração 31 –Vários robôs identificados usando o HOG-SVM e o HAAR-AdaBoost.	75
Ilustração 32 –Quadro de captura do robô B2 identificando vários robôs usando o HOG-SVM (Figura a) e o HAAR-AdaBoost (Figura b).	76
Ilustração 33 –Cálculo das Médias Móveis Simples.	89

LISTA DE TABELAS

Tabela 1 – Medidas Importantes do Campo, da Bola e dos Jogadores	14
Tabela 2 – Quadro de referência	44
Tabela 3 – Resultados dos algoritmos para reconhecimento da Bola	64

LISTA DE ALGORITMOS

Algoritmo 1	Algoritmo Adaboost.	35
Algoritmo 2	Algoritmo de rastreamento da bola laranja - Círculos de Hough.	49
Algoritmo 3	Algoritmo de Identificação da bola branca. (HAAR-Adaboost Offline)	51
Algoritmo 4	Algoritmo de Identificação da bola branca. (HAAR-Adaboost Online)	51
Algoritmo 5	Algoritmo para Identificação de Robôs. (HOG-SVM Offline)	53
Algoritmo 6	Algoritmo para Identificação de Robôs. (HOG-SVM Online)	54
Algoritmo 7	Algoritmo de Identificação de robôs. (HAAR-Adaboost Offline)	55
Algoritmo 8	Algoritmo de Identificação de robôs. (HAAR-Adaboost Online)	55
Algoritmo 9	Algoritmo para buscar e centralizar o objeto	57

SUMÁRIO

1	Introdução	10
1.1	Uma Breve Introdução à Visão Computacional de Robôs Móveis Humanoides	10
1.1.1	Tipos de Sistemas de Visão Computacional para Robôs Móveis	11
1.2	Motivação	12
1.3	Objetivos	12
1.4	Justificativas	12
1.5	Histórico	13
1.6	Domínio	14
1.7	Organização do Trabalho	15
2	Fundamentação Teórica: Visão Computacional	16
2.1	Processamento de imagens coloridas	17
2.2	Baixo Nível - Filtros	18
2.2.1	Transformações de intensidade	19
2.2.2	Histogramas	19
2.2.3	Filtros Espaciais	20
2.2.4	Segmentação	21
2.2.5	Morfologia Matemática	22
2.3	Transformada de Hough	24
2.4	Propagação da Densidade Condicional - Condensação	26
2.5	Descriptor Histograma de Gradientes Orientados (HOG) com Máquina de Vetores de Suporte (SVM)	27
2.5.1	SVM	29
2.5.1.1	<i>Um exemplo em duas dimensões</i>	30
2.6	Descriptor HAAR com Boosting	33
2.6.1	Classificador Boosting	34
2.6.2	HAAR-Adaboost	37
2.7	Controle Servo-Visual	39
2.8	Identificação de objetos	40
2.8.1	Reconhecimento da Bola	41
2.8.2	Identificação dos Robôs	43
2.9	Considerações finais	44

3	Sistema Desenvolvido	46
3.1	Bola	46
3.1.1	A Bola Laranja	47
3.1.2	A Bola Branca	50
3.2	Adversários e Companheiros de Equipe	52
3.3	HOG-SVM Online	54
3.4	Controle Servo-Visual	56
3.5	O sistema de visão	58
4	Experimentos Realizados e Resultados	60
4.1	O robô	60
4.2	Experimento - Encontrando uma bola laranja usando segmentação e transformada de Hough para círculos	61
4.3	Experimento - Encontrando uma bola branca usando características HAAR	64
4.4	Experimento - Determinação da distância da bola ao robô	65
4.5	Experimento - Detectando Robôs usando o Descritor HOG com o Classificador SVM	68
4.6	Experimento - Detectando Robôs usando o Descritor HAAR com o Classificador Boosting	70
4.7	Discussão - Comparando os dois descritores e seus classificadores	71
5	Conclusão	77
5.1	Contribuições	78
5.2	Trabalhos Futuros	78
	REFERÊNCIAS	80
	ANEXO A – Média Móvel	87
A.1	Séries Temporais	88
A.2	Métodos Simples de Previsão de Série Temporal	88

1 INTRODUÇÃO

Um robô é um sistema computacional com sinais de entrada provenientes dos sensores e um conjunto de atuadores que permitem ao agente interagir com o ambiente ao qual está inserido. O conceito de robôs jogadores de futebol foi introduzida em 1993 no Japão, e até hoje, é uma das mais importantes divisões da RoboCup. O objetivo oficial da organização é apresentar, até 2050, um time de robôs jogadores de futebol totalmente autônomos capazes de vencer uma partida, usando as regras oficiais da FIFA, contra a seleção humana vencedora da copa do mundo.

Quatro robôs autônomos (figura 1) foram desenvolvidos pelo time humanoide da FEI para atuar na liga KidSize da RoboCup, de maneira que cada um pudesse apresentar somente características humanas, ou seja, precisavam ter o formato humanoide. O dispositivo de aquisição de imagens precisa estar posicionada na cabeça, além de, necessitar de um processo de decisão e de visão individual livre de controles externos e, é claro, nenhum sensor, a não ser que represente um sentido humano.

1.1 UMA BREVE INTRODUÇÃO À VISÃO COMPUTACIONAL DE ROBÔS MÓVEIS HUMANOIDES

A robótica móvel foca em robôs que possuem a capacidade de se mover em um determinado ambiente. Os robôs móveis precisam interagir com o ambiente, muitas vezes complexo, e isso por si só não é trivial. As dificuldades decorrem de como esse agente artificial sente e manipula esse ambiente. Robôs humanoides foram criados devido ao desejo do ser humano pela criação de um robô que fosse capaz de realizar as atividades feitas pelo homem, e que também possuísse uma semelhança física, mas também, para sobressair as limitações que os robôs sobre rodas possuem.

A câmera, sendo um importante sensor do ambiente, proporciona imagens cheias de informação que o robô precisa processar e, a partir daí, tomar uma decisão. Os robôs humanoides atuais possuem uma câmera com é o caso do Darwin-OP (HA et al., 2011) ou duas câmeras que podem estar posicionadas de maneira estéreo como é o caso do robô do time Nubots (THE..., 2014) (EDROM..., 2014), ou duas câmeras posicionadas de maneira a uma voltada para o horizonte e outra voltada 45 graus na direção do chão, como é o caso do NAO (2015).

Figura 1 – Robôs desenvolvidos.



Fonte: Autor

1.1.1 Tipos de Sistemas de Visão Computacional para Robôs Móveis

Dois tipos de sistemas de visão computacional são normalmente utilizados o sistema de visão passiva e o sistema de visão ativa. O sistema de visão computacional passiva para robôs móveis se vale do conceito reativo de agir (TSOTSOS, 1992). Tsotsos (1992) ao se referir ao sistema de visão passiva menciona a pouca complexidade da busca pelo objeto de forma passiva reagindo ao seu movimento. Aloimonos, Weiss e Bandyopadhyay (1988) introduziram o primeiro formato geral para um sistema de visão ativa no final de 1980 a fim de melhorar a qualidade de percepção de rastreamento de objetos procurando-os e estimando sua posição futura. O sistema de visão ativa é particularmente importante para lidar com problemas como oclusões, campo de visão limitado e resolução da câmera limitada (DENZLER; ZOBEL; NIEMANN, 2003). Outras vantagens são: Reduzir o borrão causado por um objeto em movimento (RIVLIN; ROTSTEIN, 2000), reforçar a percepção de profundidade de um objeto concentrando-se duas câmeras no mesmo objeto, ou movimentar as câmeras (ALOIMONOS; WEISS; BANDYOPADHYAY, 1988) com o mesmo fim. Finalmente, câmeras autônomas são câmeras que podem direcionar a si mesmas no ambiente. Denzler, Zobel e Niemann (2003) usaram um conjunto de câmeras estéreis para acompanhar o movimento de um objeto.

1.2 MOTIVAÇÃO

A motivação do presente trabalho é desenvolver um sistema de visão computacional robusto e rápido para um robô humanoide, já que um dos maiores problemas da visão computacional é ter que lidar com a quantidade de informação, muitas vezes irrelevante, e algoritmos que requerem grande esforço de processamento. Ser capaz de identificar quais os pixels de uma imagem compõem um objeto é um problema não-trivial. Os trabalhos publicados para o domínio proposto se baseiam em operações que lidam diretamente com todos os pixels da imagem, ditas operações de baixo e médio nível, geralmente com baixo custo computacional. De modo que o desenvolvimento desse sistema complemente essas técnicas com o uso de técnicas de alto nível, ou seja com alguma inferência de contexto ao agrupamento de pixels.

1.3 OBJETIVOS

Este projeto visa desenvolver um sistema servo-visual para um robô humanoide atuante na competição RoboCup, Liga humanoide, categoria KidSize. Serão desenvolvidos sistemas de visão computacional que permitam ao agente:

- a) rastrear uma bola, de tamanho conhecido, fornecendo as coordenadas de sua posição;
- b) identificar agentes do mesmo time e do time rival;

1.4 JUSTIFICATIVAS

O desenvolvimento de um time de futebol de robôs representa uma aplicação do uso de agentes autônomos com comportamento inteligente, visando a execução de uma tarefa, geralmente complexa, em equipe. O estudo e desenvolvimento da inteligência artificial, fornece desafios e problemas onde diversas tecnologias e metodologias podem ser combinadas para a obtenção dos resultados desejados, especialmente em visão computacional.

1.5 HISTÓRICO

A RoboCup é uma iniciativa educacional de pesquisa, é também uma competição cujo objetivo é incentivar a pesquisa robótica fornecendo desafios nos quais uma grande quantidade de tecnologias possam ser integradas. A cada ano uma nova cidade é escolhida em um país diferente para sediar esse evento. Em 1992, o professor Alan Mackworth, em suas próprias palavras, considerou a possibilidade de robôs jogarem futebol:

“Suppose we want to build a robot to play soccer. Quite apart from all the difficult robotics and perception problems involved, we have substantial challenges in representation for planning and action”
 (MACKWORTH, 1993)

Na tradução livre do inglês: ‘Suponha que quiséssemos construir um robô para jogar futebol. Além de, todos os problemas robóticos e de percepção envolvidos, teremos desafios substanciais em representações do planejamento e das ações’. Após essa menção, os pesquisadores: Minoru Asada, Yasuo Kuniyoshi, e Hiroaki Kitano organizaram um workshop em Tóquio, fato que culminou, um ano mais tarde, em uma competição de robótica limitada apenas às fronteiras nipônicas chamada Robot J-League. Em 1997, essa competição tornou-se mundial e teve seu nome alterado para RoboCup.

O objetivo e meta da RoboCup, foi bem sintetizada na frase da própria RoboCup Federation:

“Em meados de século 21, uma equipe totalmente autônoma de robôs humanoides jogadores de futebol deve vencer um jogo contra o time de humanos campeão da última Copa do Mundo da FIFA, utilizando as regras da FIFA.” (SITE..., 2014).

Essa frase propõe que este objetivo será um dos grandes desafios compartilhados pelas comunidades de robótica e de Inteligência Artificial para os próximos 50 anos. A RoboCup, por possuir um intuito científico, possui além das diversas competições, o simpósio que acontece logo após as mesmas. Nele os participantes, podem publicar seus trabalhos científicos relevantes da área.

1.6 DOMÍNIO

O projeto tem como base a principal competição de futebol para robôs, a RoboCup, sendo a liga de atuação do agente a liga Humanoide na categoria KidSize. Todos os dados dessa seção estão de acordo com o livro de regras de 2014 da RoboCup. (ROBOCUP..., 2014)

O campo de jogo consiste de uma superfície plana e nivelada, coberta por um carpete verde liso na regras de 2014 (ROBOCUP..., 2014) e com grama artificial nas regras de 2015. Segmentos de linha de 10 centímetros de largura são usados para a marca de penalty e a posição inicial do início da partida. O campo é limitado pelas linhas maiores, chamadas laterais, e por linhas menores, definidas como sendo de fundo, onde se encontram os gols. Todas as áreas localizadas fora dessas quatro linhas são consideradas desconhecidas e indefinidas. Os gols são compostos por duas travessas, um travessão e uma rede. As condições de iluminação dependem do local da competição. Em volta do campo, um local é definido onde somente o juiz, seus assistentes e mais dois manipuladores dos robôs podem atuar. Nenhum participante pode usar, abaixo da cintura, cores iguais ou similares às definidas para o campo, bola ou gols.

Apenas os sensores equivalentes aos sentidos humanos são permitidos, mas, não são obrigatórios. Robôs humanoides devem ter o formato de seus corpos semelhante ao do ser humano, devem ter pernas e braços. Uma série de restrições são aplicadas às medidas dos agentes, uma delas, que tem mais relevância para a pesquisa, é a relação entre a altura do corpo do robô $H_{Robô}$ e a altura da cabeça $H_{Cabeça}$, de forma que, a altura da cabeça, incluindo o pescoço, deve satisfazer a seguinte expressão (ROBOCUP..., 2014):

$$0,05 * H_{Robô} \leq H_{Cabeça} \leq 0,25 * H_{Robô} \quad (1)$$

Onde a altura da cabeça é definida, como sendo, a distância do eixo da primeira junta do braço, no ombro, até o topo da cabeça.

Tabela 1 – Medidas Importantes do Campo, da Bola e dos Jogadores

Descrição	Comprimento (cm)
Linha Lateral (Comprimento do Campo)	900
Linha de Fundo (Largura do Campo)	600
Diâmetro do grande círculo	150
Diâmetro da Bola	10
Altura Máxima dos Agentes	90

Fonte: Livro de Regras RoboCup - (RoboCup Rules Book), 2014 (ROBOCUP..., 2014)

O campo de visão dos robôs é limitado a 180 graus, isso significa que o ângulo máximo entre quaisquer dois pontos na sobreposição do campo de visão de todas as câmeras montadas sobre o robô precisam, obrigatoriamente, ser inferiores a 180 graus. O mecanismo para girar (pan) a câmera é limitado a 270 graus, 135 graus para cada lado, contados da posição de olhar para frente. Já o mecanismo para erguer (tilt) a câmera é limitado a 90 graus medidos da linha horizontal, essas restrições simulam as mesmas limitações das juntas do pescoço humano. Seguindo essa linha de limitações humanas, fica definido que de qualquer posição do robô e de qualquer ângulo de câmera ou do robô, o mesmo não pode ser capaz de visualizar os dois gols ao mesmo tempo. O número de câmeras fica limitado a duas, porém, a visão monocular também é permitida, de qualquer forma deve(m) estar na cabeça do robô.

Robôs participantes das competições da liga humanoide devem ter corpos, predominantemente, pretos, cinzas, prateados ou brancos, com as limitações de serem não reflexivos e terem os pés pretos. Qualquer cor do campo, gol, bola ou time adversário deve ser evitada no robô. Braços, pernas e corpos devem ter aparência sólida e devem estar marcadas com a cor magenta (para agentes do time vermelho) e com a cor ciano (para agentes do time azul), ambas as marcas devem ter no mínimo 5 centímetros e estar visíveis nos dois lados das pernas e braços.

1.7 ORGANIZAÇÃO DO TRABALHO

As seções anteriores apresentaram uma breve introdução sobre alguns tipos de visão para robôs móveis, a motivação, os principais objetivos, as justificativas, um breve histórico da RoboCup, e finalmente, o domínio ao qual este trabalho esta inserido, compondo o capítulo introdutório. Os demais capítulos estão estruturados da seguinte maneira:

Conceitos básicos de aquisição, processamento e representação de imagens digitais são abordados no capítulo 2, apresentando, em forma de revisão, alguns dos mais conceituados algoritmos de processamento digital de imagens contextualizando-os com a visão computacional. A pesquisa prossegue nesse mesmo capítulo, pondo em perspectiva o modo como algumas equipes participantes da RoboCup solucionaram os mais variados problemas visuais do jogo de futebol. A descrição desse sistema de visão, com fluxogramas e testes a serem feitos, é apresentada na proposta de trabalho abarcada pelo capítulo 3. O capítulo 4 traz os testes já executados com resultados. Por fim, nas Considerações Finais, no capítulo 5, é feita uma breve síntese das atividades realizadas até o momento apresentando as contribuições e apontado para perspectivas futuras.

2 FUNDAMENTAÇÃO TEÓRICA: VISÃO COMPUTACIONAL

A percepção automática ou visão computacional feita por computadores, se vale do processamento digital de imagens para simular a visão humana, incluindo o aprendizado e a capacidade de fazer inferências, assim como, agir com base nas informações adquiridas.

Por sua vez, o processamento digital de imagens executa operações matemáticas em imagens digitais em domínios diferentes, é utilizado, de maneira geral, para melhorar imagens para percepção humana, e, como é abordado no presente trabalho, com enfoque computacional, a fim de armazená-las em um espaço com melhor alocação, transmití-las com maior velocidade e/ou representá-las com maior desempenho.

De maneira a simplificar o estudo do processamento digital de imagens, Gonzalez e Woods (2010), dividiram os processos digitais em três tipos de operações. São listadas as operações de baixo nível, de médio nível e de alto nível, como seguem:

- a) operações de baixo nível englobam operações primitivas, como pré-processamento de imagens, como melhorar contraste, brilho, assim como, aguçamento de imagens e redução de ruídos. Geralmente possuem como entrada a imagem original e como saída a imagem tratada;
- b) nas operações de médio nível, regiões ou objetos são identificados descrevendo-os individualmente, de modo, a classificá-los ou reconhecê-los, sem considerar em que contexto essas regiões sem enquadram. Como imagem de entrada para essas operações pode-se usar a imagem original, porém, é mais comum o uso de imagens previamente tratadas com operações de baixo nível. Após o processamento, essas operações retornam atributos dessa imagem, como, regiões, contornos e bordas;
- c) já as operações de alto nível pretendem vincular ou inferir a um conjunto de objetos reconhecidos um determinado contexto. As entradas desses tipos de operações variam, porém, de maneira geral, descritores são utilizados. Após encontradas todas as regiões e objetos é possível identificar em qual contexto estão inseridos.

Para entender melhor como o processamento digital de imagens funciona, convém iniciar pelo menor elemento em um dispositivo de exibição/aquisição, o pixel¹, ao qual é possível atribuir-se uma propriedade, comumente uma cor, ou como será visto, uma intensidade. De

¹pixel - Contração do inglês: "Picture Element"

uma forma mais simples, um pixel é o menor ponto que forma uma imagem digital, sendo que o conjunto de pixels forma a imagem inteira.

Uma imagem pode ser definida como sendo a representação visual de um objeto. Do ponto de vista matemático, uma imagem é considerada uma função bidimensional $f(x,y)$ onde x e y são coordenadas planas, e a amplitude de f em qualquer par de coordenadas (x,y) é chamada de intensidade ou nível de cinza da imagem no referido ponto. Quando (x,y) e a amplitude fazem parte de um conjunto de valores finitos ou discretos, a imagem é chamada de imagem digital (GONZALEZ; WOODS, 2010).

Em outras palavras, uma imagem digital pode ser representada através de uma matriz $n \times m$, onde, n é a resolução horizontal da captura e m é a resolução vertical da captura, cada elemento da matriz corresponde à intensidade ou nível de cinza $f(x,y)$ em um determinado ponto da imagem. Para imagens binárias (em preto e branco) os valores dos pixels podem assumir os valores 0 e 1. Para imagens em tons de cinza esses valores podem variar de 0 a 255 e, finalmente, para imagens coloridas, tem-se que o valor do pixel é representado por três valores variando de 0 a 255 cada um, no modelo de cores RGB.

2.1 PROCESSAMENTO DE IMAGENS COLORIDAS

Para entender como os descritores de cor são utilizados, é necessário definir previamente o conceito de cor e listar algumas de suas características. A cor pode ser definida como um fenômeno perceptual da luz quando incide e é refletida por um objeto (GONZALEZ; WOODS, 2010). Como se pode observar, o conceito de cor é diretamente relacionado ao conceito de luz (GONZALEZ; WOODS, 2010). Portanto, ao se referir ao conceito de cor deve-se mencionar, obrigatoriamente, a luz.

Dois conceitos preliminares são particularmente importantes para o entendimento do conceito de percepção de cor. São eles: a luminância e a crominância. A luminância contém a informação da quantidade das cores pretas e brancas presentes em uma imagem. O cérebro humano interpreta essa informação como a quantidade de cinza presente na cor (ou brilho). A crominância informa a respeito da tonalidade de uma cor. É a frequência dominante do raio de luz. A combinação destes dois conceitos, em diferentes proporções, permite ao cérebro perceber o espectro de cores visível em uma determinada imagem ou cena. Para representar as cores existem modelos ou sistemas de cores (GONZALEZ; WOODS, 2010), como por exemplo RGB

(“red, green, blue”), CMY (“cyan, magenta, yellow”), HSV (“matiz, saturação, intensidade”), YIQ (luminância, em-fase, quadratura) e LAB (luminosidade, verde-vermelho, azul-amarelo).

O sistema Red, Green, Blue é um sistema de representação de cor aditivo (COLOMBO; BIMBO; PALA, 1999) e se baseia na teoria dos três estímulos proposta por Sherman (1981). Segundo essa teoria, o olho humano percebe a cor através do estímulo de três pigmentos visuais presentes nos cones da retina, que possuem sensibilidades para alguns comprimentos de onda, como por exemplo, 630 nanômetros (vermelho – red), 530 nanômetros (verde – green) e 450 nanômetros (azul – blue). Uma representação desse sistema pode ser formulada através de um cubo com os eixos R, G e B. Podemos dizer que a origem representa a cor preta, os vértices das coordenadas (1, 1, 1) representam o branco, os vértices que estão sobre os eixos representam as cores primárias e os vértices restantes representam os complementos das primárias.

É importante observar que cada ponto no interior do cubo corresponde a uma cor, representada por uma tripla (R, G, B), com os valores de R, G e B variando de 0 a 1. Os tons de cinza são representados ao longo da diagonal principal do cubo, que se inicia no ponto de origem até o vértice que foi apontado como a cor branca. Portanto, é possível, afirmar que cada tom de cinza presente no cubo é formado por contribuições iguais das cores primárias que o compõem.

2.2 BAIXO NÍVEL - FILTROS

Os processos de visão computacional, muitas vezes, necessitam de uma etapa de pré-processamento envolvendo o processamento de imagens. As imagens de onde queremos extrair alguma informação em certos casos precisam ser convertidas para um determinado formato ou tamanho e precisam ainda ser filtradas para remover ruídos provenientes do processo de aquisição da imagem. (GONZALEZ; WOODS, 2010)

Nessa concepção, ruídos podem aparecer de diversas fontes, como por exemplo, o tipo de sensor utilizado, a iluminação do ambiente, as condições climáticas no momento da aquisição da imagem, a posição relativa entre o objeto de interesse e a câmera. Note que ruído não é apenas interferência no sinal de captura da imagem, mas também interferências que possam atrapalhar a interpretação ou o reconhecimento de objetos na imagem. Os filtros aparecem como ferramentas básicas, para remover ruídos de imagens.

No domínio espacial, plano que contém os pixels de uma imagem, a aplicação de filtros se limita à alteração direta dos pixels ou em sua vizinhança, de forma que, a uma janela é movida pixel a pixel na imagem de entrada para gerar uma imagem de saída.

2.2.1 Transformações de intensidade

As transformações de intensidade estão entre as técnicas mais simples de processamento digital de imagens, nessas transformações uma função matemática de transformação é aplicada no nível de intensidade dos pixels. Exemplos básicos incluem transformações logarítmicas, polinomiais de n-ésima potência e de negativo. Em uma transformação de negativo, a ser utilizada na identificação das linhas de campo, a intensidade dos pixels é revertida segundo a expressão (GONZALEZ; WOODS, 2010):

$$S = L - 1 - E \quad (2)$$

Onde S representa o valor do pixel depois do processamento, (L-1) é o limite máximo de nível de intensidade dos pixels da imagem original e E o valor do pixel original antes do processamento.

Na verdade, o objetivo dessa expressão é reverter a intensidade de todos os pixels da imagem, de forma a se obter algo semelhante a um negativo fotográfico, de modo que a imagem resultante se torne mais adequada para a aplicação do que a original.

2.2.2 Histogramas

Histograma é a representação de um conjunto de dados previamente tabulados e divididos em classes uniformes. Na estatística, um histograma é uma representação gráfica da distribuição de frequências de uma massa de medições (WAND, 1996). Quando o número de dados aumenta indefinidamente e o intervalo de classe tende a zero, a distribuição de frequência passa para uma distribuição de densidade de probabilidades (WAND, 1996). A construção de histogramas tem caráter preliminar em qualquer estudo e é um importante indicador da distribuição de dados. Pode indicar se uma distribuição aproxima-se de uma função normal, como pode indicar mistura de populações quando se apresentam bimodais. (WAND, 1996)

Em processamento de imagens é usado de forma geral para se obter estatísticas espaciais úteis da imagem (GONZALEZ; WOODS, 2010). Algumas aplicações do histograma de uma imagem digital incluem, mas não se limitam, a da probabilidade de ocorrência, de intensidades de pixels, de cores e de gradientes. Após sua normalização, a soma de todas as probabilidades de um determinado histograma é 1.

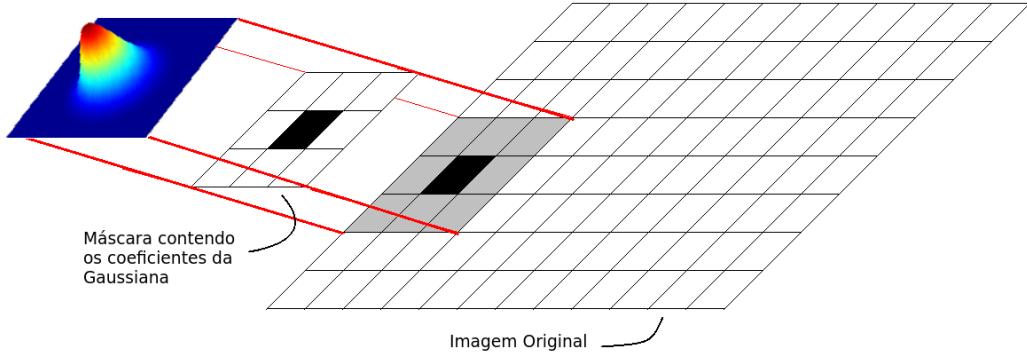
Umas das mais conceituadas formas de realçar uma imagem é equalizando seu histograma de cores ou de intensidades (GONZALEZ; WOODS, 2010). Essa equalização procede gerando uma função de transformação que busca produzir uma imagem de saída que tenha um histograma uniforme. O histograma de cores pode representar a distribuição global de cores completa da imagem ou de apenas uma determinada parte da mesma. E, como veremos nos próximos capítulos, o histograma de gradientes normalizado pode ser utilizado para extrair informações importantes sobre o formato de objetos presentes em uma determinada imagem identificando transições de borda e mostrando sua magnitude.

Histogramas de contraste e de brilho podem dizer se a imagem foi exposta corretamente, já os de luminância levam em consideração que o olho humano é mais sensível a certos tipos de cores que a outros. No histograma de luminância cada pixel é convertido de modo a representar a luminosidade. Isso é feito baseado em uma média com pesos diferentes para cada uma das cores em cada pixel (WAND, 1996). Esses pesos assumem que o verde representa 59% da luminosidade percebida, enquanto que o vermelho e o azul representam apenas 30% e 11%, respectivamente. Uma vez que cada pixel foi convertido em luminosidade, um histograma de luminância é produzido ao se contar quantos pixels estão em cada valor de luminância, ou seja, são produzidos da mesma forma que histogramas de cores apenas substituindo as cores por luminâncias.

2.2.3 Filtros Espaciais

Em processamento digital de imagens, o termo filtro refere-se ao fato de limitar, ou permitir, a passagem de certas características ou componentes de uma imagem original para uma tratada (GONZALEZ; WOODS, 2010). De forma geral isso é executado usando um pixel e/ou sua vizinhança de pixels. A definição do tamanho da vizinhança depende do tipo de filtro, do tipo de operação e do resultado que se deseja alcançar. Valores de vizinhança comuns são de 9, 25, 49 pixels, e formatos comuns são círculos e quadrados (GONZALEZ; WOODS, 2010). Uma ou mais operações de filtragem podem assumir o comportamento linear ou não linear. Entre os diversos tipos de filtros existentes destacam-se os filtros de suavização, esses são utilizados para redução de pequenos detalhes da imagem antes da extração de regiões maiores. Como exemplos de filtros lineares existem os filtros de média, também chamados de filtros passa-baixa, e como exemplos de filtros não-lineares, os filtros de estatística de ordem.

Figura 2 – Exemplo Filtro Gaussiana.



Fonte: Autor

Esse último filtro tem como principal representante o filtro da mediana, no qual o valor do pixel central é substituído pela mediana de sua vizinhança, de forma que se baseia na classificação dos pixels na área de imagem coberta por essa vizinhança (máscara) (GONZALEZ; WOODS, 2010). Essa classificação pode ser feita por qualquer operação que classifique os pixels, como uma Gaussiana, por exemplo. O pixel central da máscara Gaussiana, torna-se o pico da Gaussiana e os pixels vizinhos são alterados segundo a curvatura da Gaussiana, (ver figura 2), criando coeficientes da máscara e após aplicação causando um borramento, mas, também, reduzindo regiões menores. Esses filtros são bastante populares, devido a seus excelentes resultados na redução de ruídos aleatórios e impulsivos.

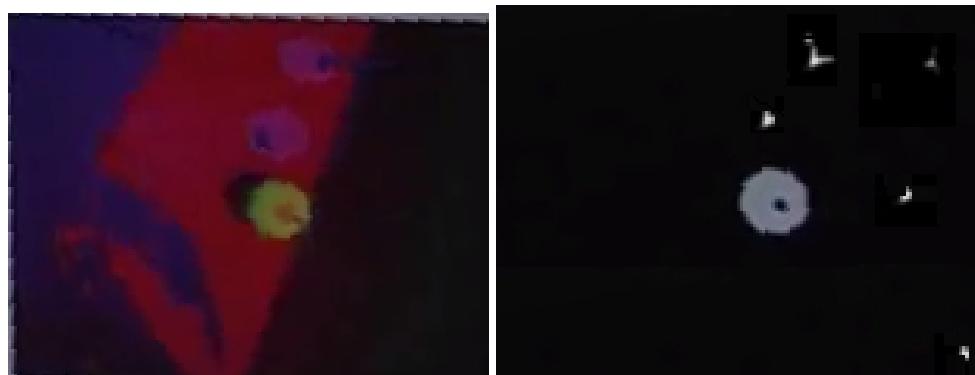
2.2.4 Segmentação

A segmentação, em visão computacional, é o processo de dividir ou segregar uma imagem digital em duas ou mais regiões diferentes, com o objetivo de simplificar uma imagem visando facilitar a sua análise ou interpretação (GONZALEZ; WOODS, 2010). Os retornos esperados, como resultados do processo de segmentação, são atributos dessa imagem, como, regiões e contornos. Convém elucidar a diferença entre borda e contorno, a borda se verifica como sendo uma transição de uma parte do contorno de um objeto. A maior dificuldade é interpretar quais bordas pertencem, efetivamente, ao contorno de uma região ou objeto.

Cada um dos pixels em uma mesma região é segmentada, levando-se em consideração alguma característica ou propriedade computacional. São propriedades comuns a cor, a intensidade, a textura ou, mesmo, a continuidade (GONZALEZ; WOODS, 2010).

Todas as regiões selecionadas devem possuir limiares de características que permitam sua discretização umas das outras. De acordo com Benallal e Meunier (2003) e Broggi et al. (2007) é possível efetuar a segmentação seguida de reconhecimento de formato em tempo real.

Figura 3 – Segmentação do Laranja usando como entrada uma imagem no modelo de cores HSV.



Fonte: Autor

A segmentação é especialmente importante para determinar a região de interesse. Assim, esse método será usado para encontrar uma região que possua uma característica tão importante quanto a cor, como é o caso da bola laranja do domínio Robocup Kid Size.

2.2.5 Morfologia Matemática

Morfologia é o estudo da forma, sendo o estudo morfológico o estudo das estruturas dessa forma. Em linguística, morfologia é o estudo da estrutura das palavras. Em biologia, morfologia está mais diretamente relacionada à forma de um organismo, por exemplo, a forma de uma folha pode ser usada para identificar uma planta ou a forma de uma colônia de bactérias pode ser usada para identificar sua variedade. Por outro lado, os matemáticos consideram a morfologia uma parte da teoria de conjuntos, e por consequência, denomina-se morfologia matemática como sendo um conjunto de métodos, inicialmente desenvolvidos por Haas, Matheron e Serra (1967), que tem em comum o objetivo de estudar a estrutura geométrica de um objeto ou parte dele.

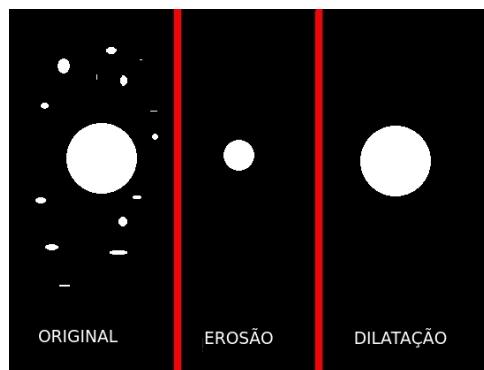
Segundo Dougherty (1992), é possível definir a morfologia digital como sendo um caminho para descrever ou analisar a forma de um objeto digital, levando-se em conta sua estrutura geométrica, baseado no fato de que uma imagem consiste em um conjunto de pixels, que podem ser reunidos em grupos, criando uma estrutura bidimensional (forma). No presente fica suben-

tendida a referência à morfologia digital. Certas operações matemáticas, quando aplicadas em conjuntos de pixels, podem ser usadas para ressaltar aspectos específicos das formas permitindo que sejam contadas ou reconhecidas.

A base da morfologia segundo Gonzalez e Woods (2010) consiste em extrair as informações relativas à geometria e à topologia de um conjunto desconhecido (no caso uma imagem) pela transformação através de outro conjunto bem definido, chamado elemento estruturante. Operações morfológicas estão divididas em binárias (aplicadas a imagens com pixels pretos ou brancos), e sobre imagens coloridas (aplicadas a imagens em tons de cinza ou, como o próprio nome diz, em imagens coloridas).

Após processamento, um objeto é considerado um conjunto matemático de pixels brancos, sendo, cada pixel, identificado pelos seus índices de linha e coluna. No caso de operações binárias, um pixel afetado por essa operação será substituído pelo seu valor oposto. Em operações executadas em imagens coloridas pode ocorrer apenas a modificação parcial do valor do pixel.

Figura 4 – Exemplo de erosão e dilatação para um objeto circular, com elemento estruturante circular.



Fonte: o Autor

As operações básicas da morfologia digital são a erosão, em que pixels que não atendem a um dado padrão são apagados da imagem, e a dilatação, em que uma pequena área relacionada a um pixel é alterada para um dado padrão. Todavia, dependendo do tipo de imagem que está sendo processada (preto e branco, tons de cinza ou colorida), a definição dessas operações muda, assim, cada tipo deve ser considerado separadamente.

Tanto a erosão como a dilatação aparecem como forma tratamento de ruídos provenientes do ambiente, eliminando regiões e áreas não pertencentes a região de interesse, reduzindo a área de atuação dos algoritmos subsequentes para detecção da bola laranja.

2.3 TRANSFORMADA DE HOUGH

Sendo uma ferramenta conceituada e de uso comum em visão computacional, a transformada de Hough é um processo matemático que detecta formas geométricas em imagens digitais. Essa técnica, desenvolvida por Hough (1959), encontra formas que podem ser parametrizáveis em imagens digitais como linhas, círculos e elipses, mesmo em imagens com pouca visibilidade da forma ou imagens fortemente ruidosas.

Em geral, a transformada é aplicada após a imagem sofrer um pré-processamento, comumente a detecção de bordas. A transformada de Hough encontra uma relação entre o espaço de imagem e o espaço de parâmetros. Cada transição (ou borda) de uma imagem é transformada por um mapeamento para determinar sua relação no espaço de parâmetros através de um vetor.

As posições do vetor são incrementadas e indicarão quais os parâmetros correspondentes à forma especificada, através do máximo local do acumulador. A detecção de retas foi sua primeira aplicação mas, a transformada de Hough foi modificada para possibilitar a localização de outras formas geométricas. A transformada de Hough utilizada, majoritariamente, atualmente pode ser atribuída a Duda e Hart (1972) e a Guil, Villalba e Zapata (1995).

Hough usou a equação definida por $y = a.x + b$ como representação paramétrica de uma linha, fato que conduziu a soluções infinitas para linhas paralelas ao eixo y, cuja solução será abordada posteriormente.

O algoritmo de Hough requer um acumulador de dimensão igual ao número de parâmetros desconhecidos. Por exemplo: achar segmentos de linhas usando a equação $y = ax + b$, requer achar dois parâmetros para cada segmento: a e b, ou seja, duas dimensões da matriz acumuladora.

Assim, usando uma matriz acumuladora A, o procedimento de Hough examina cada pixel e calcula os parâmetros da curva (equação) especificada que passa pelo pixel. O procedimento examina cada pixel e calcula os parâmetros da equação de reta que passa por esse pixel, incrementando a matriz acumuladora.

Caso uma imagem que não foi pré-processada com algoritmo de detecção de bordas, fato incomum na transformada de Hough, será examinado o pixel e sua vizinhança na imagem para determinar se há evidência de borda ou transição naquele pixel. Quando todos pixels tiverem sido processados, procura-se na matriz acumuladora os maiores valores. Esses valores indicam os parâmetros de prováveis linhas na imagem.

Ao procurar os máximos na matriz acumuladora, lança-se o artifício do uso de um limiar, visando encontrar uma quantidade mínima de pontos colineares. Qualquer valor da matriz acumuladora que não for superior ao limiar será ignorado. As detecções de outras formas, utilizando a transformada de Hough, usam o mesmo princípio, com a ressalva na quantidade de parâmetros da equação que será empregada, e em consequência na dimensão do acumulador.

Duda e Hart (1972) sugeriram coordenadas polares para representação de uma linha. Essa modificação manteve a quantidade de dimensões da matriz acumuladora e resolveu a limitação inicial de linhas paralelas ao eixo y. Usando esta parametrização, todo o ponto (x,y) na equação da reta satisfará a equação $r = x \cdot \cos(q) + y \cdot \sin(q)$. Onde os parâmetros r e q são respectivamente a distância e a orientação da linha normal à reta candidata.

Cada pixel do espaço da imagem produz uma função senoidal no espaço de Hough que alimenta um histograma. Com muitas senóides, ou seja, muitas possíveis linhas, o histograma gera diversos máximos que não correspondem efetivamente às linhas. Para resolver essa dificuldade, encontra-se a maior linha da imagem e remove-se sua contribuição do histograma. Repetindo esse procedimento até o fim das linhas encontradas, encontra-se o máximo global.

Como uma forma de inferir um formato circular à bola, é usada a transformada de Hough para círculos, na implementação sugerida por Davies (1988). Essas formas são parametrizados por x, y e r, onde x e y referem-se a posição do centro e r ao raio do círculo. Agora que a matriz acumuladora têm três dimensões, é necessário reduzir a complexidade computacional da transformada de Hough. Determinar o centro e em seguida, encontrar o raio, divide o problema em duas fases distintas amenizando as dificuldades computacionais dessa técnica.

Fica definido que a normal tangente de qualquer pixel pertencente a um círculo passa pelo centro desse círculo. Calcula-se a tangente como a linha que se ajusta melhor a todos pixels de uma vizinhança pequena (usando o método mínimos-quadrado), isso permite calcular a normal e registrá-la em um histograma. Finalmente, os máximos do histograma definem os locais de centros dos possíveis círculos.

Encontrar o raio esconde uma dificuldade prática, existem infinitos círculos que podem ter o mesmo centro. Para sobrepor essa dificuldade, alimenta-se um histograma unidimensional com a distância de cada pixel até o centro de um determinado círculo. Assim, os máximos dos histogramas correspondem aos raios dos círculos.

Há ainda, a transformada probabilística de Hough, que afirma que é suficiente computar a transformada de Hough só de uma proporção de pixels na imagem. Estes pixels são escolhidos de forma aleatória usando-se uma função de densidade de probabilidade uniforme. Essa técnica foi proposta por Kiryati, Eldar e Bruckstein (1991). A transformada de Hough vem para inferir

o formato circular à bola segmentada por cores e com ruídos reduzidos nas etapas de erosão e diliação.

2.4 PROPAGAÇÃO DA DENSIDADE CONDICIONAL - CONDENSAÇÃO

O algoritmo de condensação (Propagação densidade condicional) é um algorítimo probabilístico de visão computacional, cuja aplicação principal é detectar e rastrear objetos que se movem em um ambiente desordenado. Usa como base o filtro de partículas proposto por Thrun (2002) para fins de localização de robôs móveis.

O próprio algoritmo é descrito em detalhes por Isard e Blake (1998) e foi usado para rastrear determinados tipos de curvas. Um dos aspectos mais interessantes do algoritmo é que ele não é calculado em cada pixel da imagem. Em vez disso, os pixels são escolhidos ao acaso, e apenas um subconjunto dos pixels acaba sendo processado. A presença de desordem tende a produzir distribuições de probabilidades para o estado do objeto que são multi-modais e, portanto, poderiam ser mal modelados pelo filtro de Kalmman.

O uso de bordas e cores foi proposto por Ross (2006) onde o rastreamento de objetos usa a informação dessas características para calcular a verossimilhança das partículas na técnica de condensação.

Rastrear objetos envolve modelar sistemas não gaussianos e não lineares. Filtros de partículas são métodos sequenciais de Monte Carlo com base em representações de densidades de probabilidade, que são aplicados a qualquer modelo de estado.

O filtro de partículas pesa partículas com base em uma pontuação de probabilidade e, em seguida, propaga essas partículas de acordo com um modelo de movimento ou no caso da condensação uma cor ou uma característica. Assim, aproxima a distribuição a posteriori, filtrando ponderadamente um conjunto de partículas. Essa técnica assume o modelo de Markov para estimar o estado do sistema, de forma que o estado atual é condicionalmente independente dos estados passado e futuro. Assim as observações são dependentes apenas do estado atual.

Nessa técnica ocorre a criação das partículas, onde diversas partículas são distribuídas uniformemente pela imagem. Cada partícula se refere à uma possível parte do objeto a ser rastreado. Na condensação diferentemente do filtro de partículas a atualização das partículas não ocorre. A distância euclideana entre a características esperada do objeto e a presente na imagem capturada fornece o peso dessa partícula, ou seja, qual a probabilidade dessa partícula fazer parte do objeto. No último passo uma reamostragem das partículas onde partículas pouco

prováveis são descartadas, reiniciando o processo. Usando como característica a cor, o método de condensação pode apontar em qual região da imagem há uma concentração de pixels que pode representar probabilisticamente onde está a bola.

2.5 DESCRIPTOR HISTOGRAMA DE GRADIENTES ORIENTADOS (HOG) COM MÁQUINA DE VETORES DE SUPORTE (SVM)

No cálculo vetorial, o gradiente (ou vetor gradiente) é um vetor que indica o sentido e a direção na qual obtém-se o maior incremento possível no valor de uma grandeza. Por exemplo, a variação de intensidade de luz de uma lâmpada decai em forma de gradiente conforme nos afastamos dela. Em outras palavras, existe um operador linear que se aplicado à função de intensidade dessa lâmpada produz um vetor que sempre aponta para o ponto máximo dessa função.

Em visão computacional, o principal uso dessa técnica é a detecção de borda, quanto maior for a variância dos pixels maior será a magnitude do gradiente, enquanto, o ângulo do vetor mostra em qual direção a borda se encontra. O cálculo se inicia deslizando uma janela retangular de pixels através da imagem, e para cada janela são calculados a magnitude do gradiente e o ângulo do vetor gradiente.

Calcula-se a magnitude do gradiente varrendo a imagem pixel a pixel nas direções x e y. Subtraindo o valor da esquerda, do pixel a ser calculado, com o seu valor da direita e fazendo o mesmo com o pixel de cima e de baixo, a magnitude é dada:

$$\text{magnitude} = \sqrt{(p_{x+1} - p_{x-1})^2 + (p_{y+1} - p_{y-1})^2} \quad (3)$$

Já o ângulo do gradiente é dado por:

$$\hat{\text{ângulo}} = \arctan \frac{(p_{x+1} - p_{x-1})^2}{(p_{y+1} - p_{y-1})^2} \quad (4)$$

Onde o ponto p representa a intensidade dos pixels nas coordenadas x e y. Ao usar o conceito de gradiente, mesmo pixels diferentes podem ter o mesmo vetor de gradiente, isso se torna uma vantagem, já que torna o uso de gradientes mais robusto à variação de iluminação.

Com o uso do conceito de gradiente, é possível prosseguir para a ideia sugerida por Dalal e Triggs (2005), em que um histograma é usado em conjunto com os vetores de gradiente, para um detector de pessoas. O detector de pessoas proposto usa uma janela de detecção de

64 x 128 pixels e para calcular o descritor HOG uma janela de 8 x 8 pixels dentro da janela de detecção.

Assim, segundo Dalal e Triggs (2005), o próximo passo é criar um histograma para a imagem inteira. Inserindo todos os vetores calculados nessa imagem e colocando-os em um histograma de 9 bytes, que vai de 0 a 180 graus. Nesse histograma, Dalal e Triggs (2005) usaram gradientes sem sinal, de forma que, as orientações vão apenas de 0 a 180 graus ao invés de 0 a 360. Assim, 9 bytes distribuídos em 180 graus, significam que cada byte será responsável por uma variação de 20 graus nos vetores de gradientes. A imagem é representada por um histograma, ou, em outras palavras, um único vetor de características, ao invés de muitos vetores representando pedaços menores de muitos objetos.

A contribuição de cada vetor de gradiente para o histograma é a parte mais importante do algoritmo. Essa contribuição é dada pela magnitude do vetor. Dividindo a contribuição entre dois bytes mais próximos, é possível minimizar o problema de gradientes que estão na fronteira entre esses dois bytes. Basicamente, se um vetor estiver com ângulo 145, como bytes possíveis, o histograma terá os valores 130 e 150, assim 25 % da contribuição desse vetor irá para o byte de 130 graus e 75% para o byte de 150 graus.

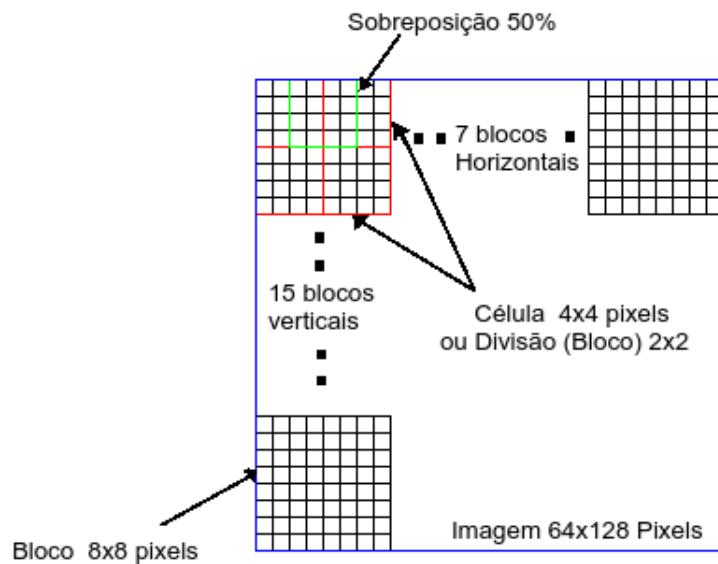
De outra forma, se um gradiente forte estiver bem na borda de um byte, uma pequena mudança no ângulo do gradiente (que poderia anular o gradiente no próximo byte) poderia causar um grande impacto no histograma.

Tendo como objetivo fazer o vetor de gradientes mais robusto a mudanças de iluminação, é efetuada uma normalização dividindo todos os vetores por suas respectivas magnitudes. Dividir o vetor por sua magnitude é equivalente a normalizá-lo com tamanho máximo igual a um. Normalizar um vetor afeta apenas sua magnitude e não sua orientação. Na verdade, ao invés de normalizar cada vetor, ou mesmo cada histograma, a normalização ocorre em blocos. Cada histograma é normalizado de acordo com todos os histogramas contidos em cada bloco.

Dalal e Triggs (2005) usaram um bloco com quatro células de tamanho dois por dois, de forma que há uma sobreposição de 50% dessas células. A normalização dos blocos é executada concatenando os histogramas das quatro células dentro do bloco em um vetor de trinta e seis componentes (Quatro histogramas e nove bytes por histograma) e dividindo esse vetor pela sua magnitude.

A estimativa do tamanho final do descritor para uma janela de detecção de 64x128 pixels é: A janela será dividida em 7 blocos de largura e 15 de altura, perfazendo um total de 105 blocos, ver figura 5. Cada bloco contém 4 células com 9 bytes num total de 36 valores por bloco. Então $7 \times 15 \times 4 \times 9 = 3.780$ valores são armazenados em um arquivo.

Figura 5 – Divisão de regiões para aplicação do gradientes.



Fonte: Autor “adaptado de” Dalal e Triggs, 2005

Segundo Dalal e Triggs (2005), gradientes são comumente utilizados como detectores de borda, essa técnica permite que todas as bordas detectadas de um objeto digital, como o ser humano ou robôs como é o caso dessa pesquisa, sejam armazenadas em um vetor. Com um histograma apresentando a distribuição de todos os ângulos das bordas detectadas é possível classificar o histograma de teste com um conjunto de treinamento, e portanto, dizer se o vetor de teste tem um histograma semelhante ao do ser humano.

2.5.1 SVM

Uma máquina de vetores de suporte (SVM - Do inglês: support vector machine), é um conceito criado por Vapnik e Chervonenkis (1995) que se refere a um conjunto de métodos de aprendizado supervisionado que analisa dados e reconhece padrões, usado para classificação e análise de regressão.

Um conjunto de hiperplanos é construído em um espaço de altas dimensões que pode ser usado para classificação. Intuitivamente uma boa separação é alcançada pelo hiperplano que tem a maior distância até o dado de treinamento de qualquer classe.

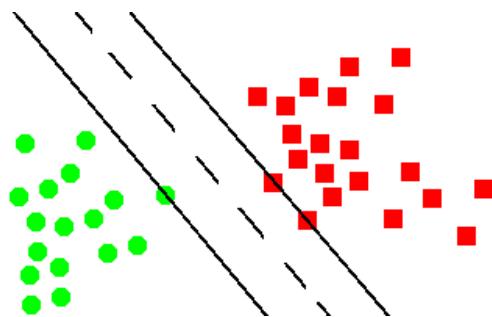
Na literatura, uma variável de previsão é chamada atributo e um atributo transformado que é usado para definir o hiperplano é chamado de característica. A tarefa de escolher a re-

presentação mais adequada é conhecida como seleção de recursos. Um conjunto de recursos que descreve um caso (ou seja, uma linha de valores de previsão) é chamado um vetor. Por isso o objetivo de modelar o SVM é encontrar o hiperplano ótimo que separa os conjuntos de vetores, de tal maneira que os dados de uma categoria alvo estejam de um lado do hiperplano e a outra categoria esteja do outro lado. Os vetores que se encontram próximos do hiperplano são chamados de vetores de suporte.

2.5.1.1 *Um exemplo em duas dimensões*

Para realizar uma classificação em duas dimensões, considerando que os dados possuem duas categorias, e assumindo que há duas variáveis de previsão com valores contínuos. Na figura 6 uma categoria de variável alvo é representado por retângulos, enquanto, a outra categoria é representada por círculos, também são demonstrados dois possíveis candidatos, já que existem um número infinito de linhas possíveis de separação, um exemplo de dois possíveis candidatos são mostrados nessa mesma figura.

Figura 6 – Representação esquemática da classificação em duas dimensões do SVM.

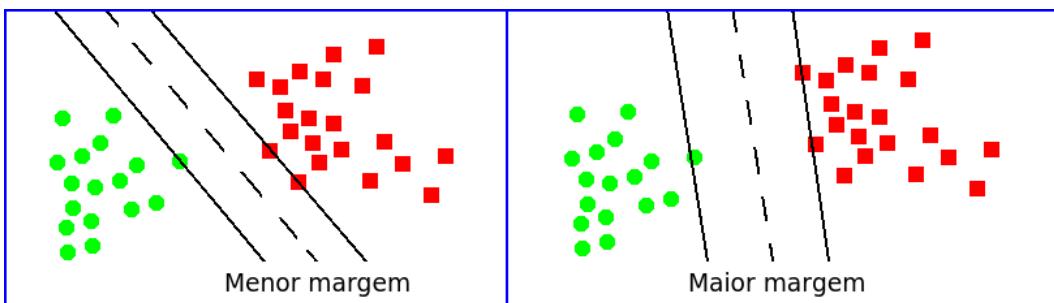


Fonte: Autor “Adaptado de” Vapnik e Chervonenkis, 1995

Continuando na figura 6, as linhas contínuas são os vetores de suporte e a tracejada representa um hiperplano separador de uma dimensão, ou seja, uma linha. As linhas contínuas paralelas à linha de separação marcam a distância entre essa linha e os vetores mais próximos da linha. À distância entre as linhas contínuas (Vetores de Suporte) é dado o nome de margem. Os vetores de características são representados na figura 7 em uma dimensão, como pontos. Os pontos que mais distanciam as classes servem de base para os vetores de suporte, e estes, por sua vez, abrigam a margem que deve ser maximizada.

Na análise do SVM, encontra-se a linha (ou o hiperplano) que está orientado de forma que a margem entre os vetores de suporte é maximizada. No exemplo da figura 6, os dados são

Figura 7 – Um exemplo da maximização da margem.



Fonte: Autor “Adaptado de” Vapnik e Chervonenkis, 1995

facilmente separados por uma linha, mas nem sempre esse é o caso. Para separar dados envoltos por regiões não lineares, lança-se o artifício do uso de uma função central (Kernel Function) como a RBF², que pode transformar os dados em um espaço com mais dimensões de modo que seja possível classificar esses dados.

Visando permitir alguma flexibilidade na classificação do SVM, um fator de erro chamado parâmetro de custo, (C)³ é inserido à margem, permitindo que alguns dados sejam aceitos como pertencentes à outra classe, de forma que seriam mal classificados. Reduzir o valor desse parâmetro exigirá uma classificação mais rigorosa, mas também reduzirá o poder de generalização do classificador. Aplicando o conceito de forma reversa, ou seja, aumentando o valor do parâmetro de custo, o efeito poderia ser de não haver classificação, portanto, a escolha desse parâmetro determina a eficácia desse classificador.

A precisão de um modelo SVM é dependente da seleção dos parâmetros do modelo. São propostos por Cristianini e Shawe-Taylor (2000) dois métodos para encontrar os valores dos parâmetros ideais: a pesquisa de grade e o padrão de busca.

A pesquisa grade testa valores de cada parâmetro em toda a gama de busca específica, utilizando passos geométricos. A busca padrão, também conhecida como busca bússola ou pesquisa em linha, começa no centro da faixa de busca e faz passos experimentais em cada sentido para cada parâmetro. Se o ajuste do modelo aumenta, o centro de pesquisa move-se para o novo ponto e o processo é repetido. Se nenhuma melhoria for encontrada, o tamanho do passo é reduzido e a pesquisa é iniciada novamente. O padrão de pesquisa é interrompido quando o tamanho do passo de pesquisa é reduzido à uma tolerância especificada.

Pesquisas de grade são computacionalmente mais caras porque o modelo deve ser avaliado em vários pontos dentro da grade para cada parâmetro. Por exemplo, se uma pesquisa

²RBF - Do inglês: "Radial Basis Function"

³C - Do inglês: "Cost Parameter"

de grade 10 é usada com intervalos de pesquisa e uma função de kernel RBF é utilizada com dois parâmetros (C) e gama, o modelo deve ser avaliado em $10 * 10 = 100$ pontos da grade. Uma análise Epsilon-SVR⁴ tem três parâmetros (C, Gama e P) para uma pesquisa de grade com 10 intervalos exigirá $10 * 10 * 10 = 1000$ avaliações no modelo. Se a validação cruzada⁵ é utilizada para a avaliação de cada modelo, o número de cálculos reais do SVM seria, ainda, multiplicado pelo número de dados da validação cruzada (tipicamente 4 a 10). Para grandes modelos esta abordagem pode ser computacionalmente inviável.

A busca padrão geralmente exige muito menos avaliações do modelo do que uma pesquisa de grades. Começando no centro geométrico do intervalo de busca, uma busca padrão faz passos de teste com valores passo positivo e negativo para cada parâmetro. Se for encontrado um passo que melhora o modelo, o centro da pesquisa é movido para esse ponto. Se nenhum passo melhora o modelo, o tamanho do passo é reduzido e o processo é repetido. A pesquisa termina quando o tamanho do passo é reduzido à uma tolerância especificada. A fraqueza de uma busca padrão é que ela pode encontrar um mínimo local em vez de ponto ótimo global para os parâmetros.

Os resultados da aplicação desta técnica são comparáveis aos obtidos por outros algoritmos de aprendizado, como as Redes Neurais Artificiais (RNAs) (HAYKIN; NETWORK, 2004) e em algumas tarefas têm se mostrado superiores, tal como na detecção de faces em imagens feitas por Osuna, Freund e Girosi (1997), e na categorização de textos, como proposto por Dumais et al. (1998). Segundo Lorena e Carvalho (2003), como vantagens do SVM, é possível citar:

- a) **boa capacidade de generalização:** mostrando eficiência na classificação de dados que não pertençam ao conjunto utilizado em seu treinamento;
- b) **robustez em grandes dimensões:** No caso das imagens, as SVMs são robustas diante de conjunto de dados de grandes dimensões (LORENA; CARVALHO, 2003);
- c) **convexidade da função objetivo:** a aplicação das SVMs implica em apenas um mínimo global. Há a necessidade da otimização de uma função quadrática. Isso se torna uma vantagem sobre as Redes Neurais Artificiais na qual há a presença de mínimos locais na função objetivo;

Na fase de treinamento, cada imagem, de um conjunto de imagens positivas e negativas, tem o seu histograma de gradientes calculado, gerando diversos vetores individuais. O classifi-

⁴Support Vector Regression (PEDREGOSA et al., 2011)

⁵Do inglês: "Cross Validation"

cador SVM cria um hiperplano que separa os conjuntos positivos dos negativos, maximizando a distância entre os vetores de treinamento.

Na fase de teste, cada imagem adquirida da câmera, já convertida em um histograma de gradientes orientados, se torna o vetor de teste do classificador SVM.

2.6 DESCRIPTOR HAAR COM BOOSTING

Na matemática, as ondaletas de HAAR⁶ são uma sequência de funções quadradas que juntas formam uma família ou base de ondaletas. A análise de Haar, foi proposta por Haar (1910) e é similar à análise de fourier que permite que uma função alvo durante um intervalo seja representada por uma função de base ortonormal. As ondaletas de HAAR não são diferenciáveis pois se tratam de funções contínuas, mesmo que isso seja uma desvantagem técnica, essa função pode determinar transições súbitas, especialmente em imagens identificando bordas.

Basicamente, quando todos os pixels de uma imagem têm que ser computados, o cálculo de características tende a ser custoso. Papageorgiou, Oren e Poggio (1998) propôs um conjunto de características alternativa baseada nas ondaletas de HAAR ao invés de usar as intensidades dos pixels. A ideia de usar ondaletas de HAAR foi melhorada por Viola e Jones (VIOLA; JONES, 2001), e originou as características de HAAR, que são características de imagens digitais usadas no reconhecimento de objetos. Uma das primeiras aplicações dessa técnica foi um reconhecimento facial em tempo real.

Para computar as características de HAAR é necessário considerar regiões retangulares adjacentes em um local específico da imagem, computar a soma das intensidades dos pixels e então calcular a diferença entre as somas de regiões diferentes. Essa diferença é então usada para categorizar subseções dessa imagem. A posição desses retângulos é definida com relação a uma janela limite de detecção do objeto alvo.

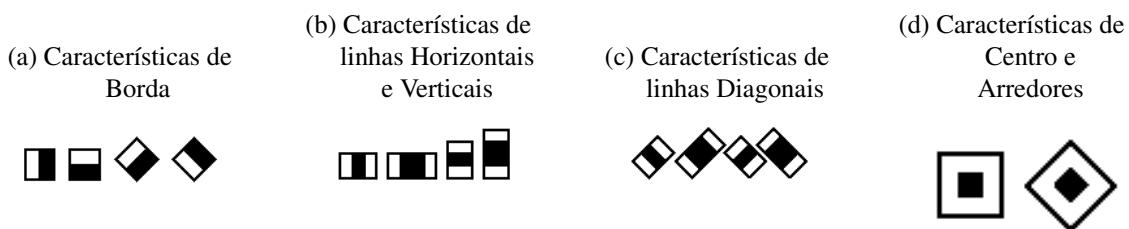
A figura 9 mostra algumas características comumente usadas.

2.6.1 Classificador Boosting

As técnicas de Boosting checam um número de características e descobrem qual é o melhor preditor para o objeto baseado nas amostras (AVIDAN, 2007). Depois de escolhido o

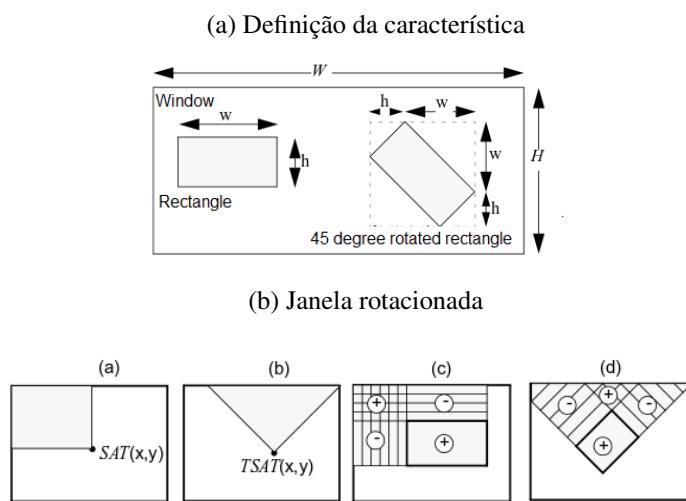
⁶Do inglês: "HAAR Wavelets"

Figura 8 – Algumas características simples usadas para descrever objetos.



Fonte: Autor “Adaptado de” Kuranov, Lienhart e Pisarevsky, 1998

Figura 9 – Características usadas para descrever objetos.



Fonte: Autor “Adaptado de” Kuranov, Lienhart e Pisarevsky, 1998

melhor classificador, ele continuará a encontrar outro e outro até que algum limiar seja atingido e todas as características combinadas fornecerão o resultado final.

O algoritmo Adaboost (Adaptive Boosting) é uma variante do algoritmo de Boosting, foi introduzido por (FREUND; SCHAPIRE, 1997), e é um algoritmo de classificação, cuja abordagem consiste em criar uma regra de predição altamente precisa, combinando, em forma de cascata, regras ou classificadores relativamente fracos e imprecisos. Uma cascata de classificadores é uma árvore de decisão onde, em cada estágio, um classificador é treinado para detectar objetos (VIOLA; JONES, 2001). Existem diversas variações do Adaboost, como o gentil (SCHAPIRE; SINGER, 1999a) (SCHAPIRE; SINGER, 1999b), o real(FRIEDMAN et al., 2000) e o discreto (FREUND; SCHAPIRE, 1997).⁷

⁷Do inglês: "Adaboost Gentle", "Adaboost Real" e "Adaboost Discrete"

O Adaboost chama um classificador fraco repetidamente usando iterações $t = 1, \dots, T$. Para cada chamada, a distribuição D é atualizada para indicar a importância do conjunto de dados de exemplo usado para a classificação. A cada iteração os pesos de cada exemplo mal categorizado são aumentados, de forma oposta os pesos dos exemplos corretamente classificados ou categorizados são reduzidos. (SCHAPIRE; SINGER, 1999a)

Além disso, o Adaboost é adaptativo (FREUND; SCHAPIRE, 1997) no sentido que classificações subsequentes anteriormente executadas são ajustadas em favor das instâncias adversamente classificadas. Usa um número de imagens de amostragem de treinamento para escolher uma quantidade de características. Nesse caso, para o reconhecimento de objetos, uma característica é tipicamente somente um retângulo de pixels que têm uma certa intensidade de cor média e um tamanho relativo.

O algoritmo 1 demonstra o algoritmo Adaboost. Nele, um conjunto de dados pertencentes à duas classes, é inserido no classificador com uma determinada distribuição $D_t(i)$. Para cada estágio t pertencente à T , onde T é o total de estágios de treinamento, um classificador simples é treinado e dele é retirado uma hipótese h_t , ou uma classificação fraca da qual é calculado um determinado erro ϵ_t . Após determinado o erro a distribuição é atualizada e normalizada para aquele determinado estágio. Esse passo é repetido tantas vezes até que todos os estágios sejam concluídos e a hipótese final do classificador H_x seja apresentada.

A ideia, mostrada na figura 10, é que o sistema de classificação divida o espaço de dados em porções menores, e teoricamente mais fáceis de aprender, fazendo com que a fronteira de decisão possa ser aproximada por meio de uma combinação apropriada das fronteiras de decisão de classificadores mais fracos.

Em estatística e aprendizagem de máquina, métodos de conjunção⁸ de algoritmos usam vários algoritmos de aprendizagem para obter um melhor desempenho preditivo do que poderia ser obtido a partir de qualquer um dos algoritmos de aprendizagem individualmente (POLIKAR, 2006). A figura 10 mostra um conjunto de dados e uma fronteira complexa a ser determinada, já na figura 11, é demonstrado graficamente como vários classificadores mais simples podem ser combinados para se ter uma classificação mais acurada.

Os métodos dos mínimos quadrados e do K vizinhos mais próximos (KNN) são, na literatura, os classificadores fracos mais comumente utilizados, são também os mais eficazes (AVIDAN, 2007) (PARAG; PORIKLI; ELGAMMAL, 2008). O KNN é um dos algoritmos de classificação clássicos e bem simples, usado para classificar objetos com base em exemplos de treinamento que estão mais próximos no espaço de características. Isso é possível determinando

⁸Do inglês: "Ensemble"

Algoritmo 1– Algoritmo Adaboost.

```

1 Dado  $x_1, y_1 \dots x_m, y_m$  onde  $x_i \in X, y_i \in Y = \{+1, -1\}$ 
2 Inicialize  $D_1(i) = \frac{1}{m}$ 
3 para  $t=0 \dots T$  faça
4   Treine um classificador fraco usando a distribuição D.
5   Extraia uma hipótese fraca  $h_t \rightarrow X \in \{+1, -1\}$  com erro
 $\epsilon_t = Pr_{i \sim D_t}[h_t(x_i) \neq y_i]$ 
6   Escolha  $\alpha_t = \frac{1}{2} \ln(\frac{1-\epsilon_t}{\epsilon_t})$ 
7   Atualize:
8     
$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} * \begin{cases} e^{\alpha_t}, & \text{Se } h_t(x_i) = y_i \\ e^{-\alpha_t}, & \text{Se } h_t(x_i) \neq y_i \end{cases}$$

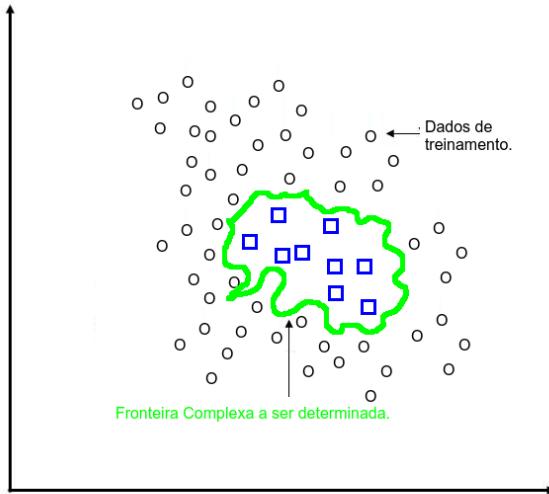
9     
$$Z_t = \sum_i D_t(i) e^{-\alpha_t y_i h_t(x_i)}$$

10    Onde  $Z_t$  é um fator de normalização escolhido de forma que  $D_{t+1}$  seja uma
        distribuição.
11    Hipótese final:  $H_x = sign(\sum_{t=0}^T \alpha_t h_t(x))$ 
12 fim

```

Fonte: Autor “Adaptado de” Freund e Schapire, 1997

Figura 10 – Um exemplo de fronteira complexa.



Fonte: Autor “adaptado de” Polikar, 2006

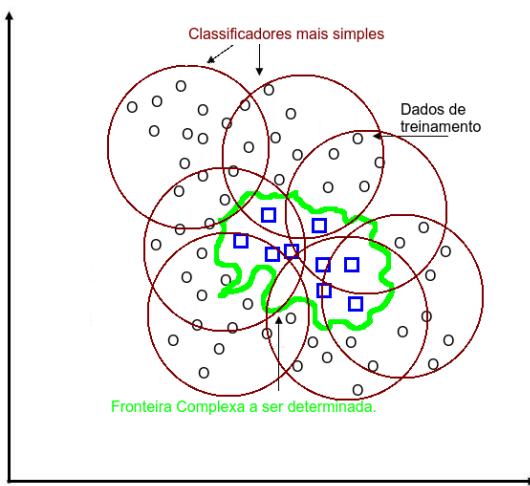
a classe do exemplo desconhecido a partir da lista de vizinhos mais próximos usando uma métrica de distâncias entre a classe de treinamento e de teste. A precisão da classificação utilizando o algoritmo KNN depende fortemente do modelo de dados (DUDA; HART, 2001). A escolha do valor de k , ou seja, a quantidade de vizinhos a ser considerada é importante, já que um valor de k muito pequeno torna a classificação sensível a ruídos e se for muito grande a vizinhança pode incluir elementos de outra classes. O método dos k vizinhos mais próximos será utilizado como classificador no método AdaBoost, pois, segundo Duda e Hart (2001) possui como vantagens:

- a) técnica simples e facilmente implementada.
- b) bastante flexível.

2.6.2 HAAR-Adaboost

Somando-se todos os valores de cores de uma imagem, de forma que cada o valor em cada posição será a soma de todos os pixels acima e a esquerda daquela posição e armazenando esses valores em um vetor bidimensional, é possível calcular rapidamente o valor médio da cor em qualquer retângulo da imagem. Isso é feito subtraindo o valor encontrado no topo à esquerda do valor encontrado no fundo a direita e dividindo pelo número de pixels do retângulo.

Figura 11 – Um exemplo de vários classificadores mais simples ou mais fracos trabalhando em conjunto para determinação de fronteiras mais complexas.



Fonte: Autor “adaptado de” Polikar, 2006

Segundo Ju, Zhang e Xue (2013), outra particularidade dessa técnica é o tamanho da amostra. O tamanho da amostra se refere ao tamanho padrão de entrada dos classificadores de uma amostra de uma determinada imagem. As amostras são trechos retirados da imagem que possuem um determinado tamanho padrão. A imagem de origem é rotacionada aleatoriamente em volta de todos os eixos. Então pixels que possuem a intensidade dentro de um intervalo específico são interpretados como transparentes. Um ruído branco é adicionado às intensidades do plano de frente. A imagem obtida é então colocada dentro de um plano de fundo arbitrário proveniente do arquivo de descrição de fundos, redimensionada para a largura e altura desejados e então armazenadas em um vetor.

Muitos pesquisadores têm observado que adicionar mais classificadores fracos pode reduzir o erro de treinamento, mas dificilmente reduz o erro de generalização (LI; ZHANG, 2004) (WU et al., 2008) (MITA et al., 2008). Segundo Paisitkriangkrai, Shen e Zhang (2010) o desempenho da detecção decai drasticamente no estágios mais avançados da cascata, além disso, o desempenho da detecção usando características simples e únicas não são suficientemente discriminantes para separar imagens de faces das imagens onde faces não estão presentes.

Mita et al. (2008) afirma que o uso da co-ocorrência de características em cada classificador fraco demonstrou um desempenho de classificação melhor do que se fossem utilizadas características simples. Assim, Paisitkriangkrai, Shen e Zhang (2010) propõe aplicar o conceito de unir características para criar a co-ocorrência de características.

A contagem de amostras positivas $N_{\text{AmostrasPositivas}}$ é um parâmetro muito importante, sem o qual o treinamento pode apresentar alguns erros ou pode nem mesmo prosseguir (MITA et al., 2008). Isso ocorre porque algumas amostras já utilizadas anteriormente por estágios já executados já podem ter sido classificadas isso pode manter o classificador em um mínimo local, a cada estágio a contagem de amostras positivas diminui até se tornar insuficiente. Para solucionar isso Can... (2012) propôs o seguinte:

Suponha que um subconjunto de amostras positivas de tamanho $N_{\text{AmostrasPositivas}}$ foi selecionado para treinar no estágio i , com i começando em zero, e $Min_{\text{TaxaAcerto}}$ é a constante de treinamento para cada estágio.

Após treinar o estágio atual, pelo menos $Min_{\text{TaxaAcerto}} * N_{\text{AmostrasPositivas}}$ amostras desse subconjunto devem passar por esse estágio, ou seja, o estágio atual da cascata com $i+1$ estágios tem que reconhecer a quantidade $Min_{\text{TaxaAcerto}}$ das amostras selecionadas como positiva. Algumas amostras já utilizadas anteriormente por estágios já executados já podem ter sido classificadas como plano de fundo, mas não mais que $((1 - Min_{\text{TaxaAcerto}}) * N_{\text{AmostrasPositivas}})$ em cada estágio. Veja equação 5 onde, $N_{\text{ImagensPositivas}}$ é a quantidade de imagens positivas total.

A equação 6 foi usada para determinar o parâmetro $N_{\text{AmostrasPositivas}}$.

$$Amostras = (N_{\text{Estágios}} - 1) * (1 - Min_{\text{TaxaAcerto}}) \quad (5)$$

$$N_{\text{ImagensPositivas}} = [N_{\text{AmostrasPositivas}} + Amostras * N_{\text{AmostrasPositivas}}] + N_{\text{ImagensNegativas}} \quad (6)$$

$$\therefore N_{\text{AmostrasPositivas}} = \frac{N_{\text{ImagensPositivas}} - N_{\text{ImagensNegativas}}}{1 + Amostras}$$

A contagem de amostras do vetor que pode ser reconhecida como plano de fundo imediatamente é a variável $N_{\text{ImagensNegativas}}$ na equação 6.

Kuranov, Lienhart e Pisarevsky (1998) afirmam que um tamanho de amostra de 20x20 pixels atingiu a maior taxa de acerto para detecção de faces. A cascata de classificadores foi implementada em formato de árvore, de forma que os ramos da árvore pudesse dividir em 2 ou mais nós. Kuranov, Lienhart e Pisarevsky (1998) afirmam também que para um tamanho de amostra de 18x18 e quatro nós de separação na fase de cascata tiveram um desempenho bom, enquanto para 20x20, dois nós a técnica teve um desempenho ligeiramente melhor, sendo a vantagem de desempenho entre usar 2, 3 ou 4 nós de separação nos classificadores fracos é menor do que a sua superioridade em relação a percorrer seus ramos. Sendo uma técnica usada

para determinação de objetos o Haar-Adaboost será utilizado para identificar robôs da mesma forma que o HOG-SVM, de forma que uma comparação possa apontar qual deles se aplica nos robôs desenvolvidos.

2.7 CONTROLE SERVO-VISUAL

O controle servo-visual é uma técnica que usa o retorno de dados extraídos de um sensor de visão para controlar o movimento do robô. Esse tipo de controle, para o domínio RoboCup, se restringe ao controle dos dois servo-mecanismos referentes à cabeça do robô, mais especificamente do pescoço.

Uma das primeiras abordagens referentes ao controle servo-visual foi publicado por Hutchinson et al. (1996). Trabalhos mais recentes foram publicados por Chaumette e Hutchinson (2006) e por Chaumette e Hutchinson (2007), aqui de uma forma mais abrangente no qual entende-se por controle servo-visual todas as ações feitas pelo robô que possuem como origem a visão do mesmo, inclusive movimentos das mãos, pés, rodas e articulações.

Técnicas de controle servo visual são classificados nos seguintes tipos:

- a) Baseada em Imagem (IBVS)
- b) Baseado Posição / Pose (PBVS)
- c) Abordagem Híbrida

A IBVS foi proposta por Sanderson e Weiss (1980). A lei de controle é baseada no erro entre as características atuais e desejadas no plano da imagem e não envolve qualquer estimativa da pose do alvo. Os recursos podem ser as coordenadas de recursos visuais, linhas ou momentos de regiões. A IBVS tem dificuldade com movimentos muito grandes, como rotações, o que veio a ser chamado retiro de câmera.⁹

A PBVS é uma técnica baseada em uma única câmera. Isto porque a posição do objeto de interesse é estimada em relação à câmera e, em seguida, é emitido um comando para o controlador de robô, que por sua vez, controla o robô. Neste caso, as características da imagem são extraídas mas, são, também, utilizadas para estimar a pose do objeto no espaço cartesiano.

Abordagens híbridas usam uma combinação dos controles previamente descritos. De forma geral, essas duas técnicas são consideradas passivas já que aguardam os dados do sensor de visão para tomar alguma decisão.

⁹Do inglês: "Camera Retreat"

Uma outra forma de encarar esse desafio é usar uma visão ativa, como proposta por Sharma (1998), que, de fato, procura ativamente o objeto a ser identificado e para tanto precisa tomar algumas decisões, mesmo que não haja dados relevantes vindos da câmera.

Um sistema de visão ativo é capaz de interagir com o seu meio ambiente (TSOTSOS, 1992), alterando o seu ponto de vista em vez de observá-lo passivamente e operando em sequências de imagens, em vez de em um único quadro (WALTHER et al., 2005). Além disso, a capacidade de acompanhar fisicamente um alvo reduz o borrão de movimento, aumentando a resolução de destino para tarefas de nível mais elevado, tais como classificação (WALTHER et al., 2005).

O controle servo-visual armazena as posições de trajetória do objeto e, por meio de regressão, estima as próximas posições do mesmo. Usando um algoritmo guloso encontra-se o melhor movimento antecipado da câmera em direção ao objeto. Para garantir um movimento plausível da câmera, é necessário considerar a cinematografia e a dinâmica desse movimento, otimizando a trajetória da câmera para que essa possa manter o objeto em seu campo de visão.

Uma das vertentes da visão ativa é a configuração escravo-mestre (HAJ et al., 2011, p.2), no qual uma câmera está fixa e outra está em movimento. Ambas precisam estar em um mesmo sistema de coordenadas para que as informações trocadas entre elas sejam facilmente interpretadas. Essa configuração poderá ser usada em um projeto futuro, no qual o goleiro exerça o papel de câmera estática de supervisão enquanto os outros jogadores seriam as câmeras ativas em movimento.

2.8 IDENTIFICAÇÃO DE OBJETOS

Atualmente a visão computacional aplicada à robótica melhora no mesmo ritmo que a capacidade computacional. A quantidade de informação que uma imagem pode ser redundante ou desnecessária. Dessa forma o sistema de percepção deve reduzir ou tratar essa informação para que os objetivos sejam alcançados.

Diversas abordagens têm sido propostas para o reconhecimento de objetos, sendo as mais recentes se valendo do uso de Redes Neurais Profundas (DNN) (SZEGEDY; TOSHEV; ERHAN, 2013) (ERHAN et al., 2014) (MCCANN; REESMAN, 2013). A maioria usa uma ou mais características, geralmente únicas a esse objeto, fazendo-o sobressair aos demais itens na imagem capturada. Um ponto em comum às muitas das abordagens, decorre do uso de câmeras estáticas, nesse sentido o sistema poderia interpretar os dados provenientes da(s) câmera(s)

de forma equivocada, já que a vibração da câmera causada pelo movimento do robô causa o borramento da região de interesse e arredores além de, provocar o movimento de pixels não pertencentes à essa região.

Para reconhecer um objeto é necessário distingui-lo do plano de fundo usando suas características únicas ou informação temporal, verificando quadro a quadro a região de interesse.

Com a finalidade de inferir algum formato à região de interesse, muitos algoritmos são propostos para a detecção de borda, incluindo o uso de gradientes. Reconhecer objetos parcialmente observáveis, em tempo real, torna-se uma tarefa particularmente difícil de solucionar, já que um objeto parcialmente ocluso tem seu formato alterado.

De forma análoga, a cor pode ser usada como uma forma de segmentar a região de interesse, que apesar de ser uma característica simples de se computar, é muito influenciada pela iluminação do ambiente. O time AUTMan, da Politécnica de Teerã (AUTMAN..., 2014), separa a imagem em três camadas, em cada qual segmenta uma cor referente à um objeto, sendo que cada uma dessas camadas serve de entrada para a próxima. Primeiramente as linhas de campo são segmentadas, em seguida o gol e por último, a bola. Dessa forma não é possível ver a bola se a segmentação do campo não estiver concluída.

Todos os times participantes da Robocup são unânimes quanto ao uso de uma tabela de pesquisa (Look-up Table - LUT), na qual diversos valores de cor são armazenados previamente para uso na segmentação. Alguns times usam uma GUI (Graphical User Interface), Interface gráfica para Usuário para calibrar esses valores durante a competição.

2.8.1 Reconhecimento da Bola

Devido a importância da bola em jogo de futebol, o algoritmo de identificação da bola precisa ser rápido, robusto à oclusão e às diversas condições de iluminação. Algumas características da bola, como seu formato circular, sua cor e seu tamanho, são largamente utilizadas por todos os times, não só para encontrar a bola mas, também, para calcular sua distância do robô.

O filtro de Kalman (PATEL; G., 2013) e o filtro de partículas proposto por Nummiaro, Koller-Meier e Gool (2002), também chamado de algoritmo de condensação (ISARD; BLAKE, 1998) para uso em imagens digitais, são amplamente usados em outros domínios como formas probabilísticas de inferir a posição de um objeto digital. São técnicas mais custosas em termos

de tempo de computação mas, com mais poder de processamento, seria possível usá-las como uma forma complementar de identificar a posição na qual a bola se encontra.

Calcular a distância da bola ao robô, quando esta encontra-se parcialmente oclusa, adiciona uma componente de dificuldade, já que o tamanho conhecido da bola é usado como forma de computar essa distância. Nesse sentido, o time NUBots, da universidade de Newcastle (BUDDEN et al., 2012), criou linhas de varredura horizontais e verticais onde todos os pixels são considerados de forma individual. Determinam a linha de horizonte verde e, usando uma LUT, geram as transições referentes à classe de cor da bola.

Esse mesmo time lidou com o problema da oclusão em 2012, adicionando uma etapa de verificação, onde quatro pontos de interesse, sem oclusão, podem ser usados em um método de atualização de centro, quando um determinado limite (threshold) de pixels consecutivos não pertencentes à bola são alcançados. Apesar de robusto contra ruído e alguma oclusão, esse método ainda precisa de duas verificações de tamanho e mais duas de cor.

Esse algoritmo trabalha em baixo nível, segmentando pixels de forma individual e, por esse motivo, pode-se dizer que está efetuando a detecção com pouquíssimos recursos computacionais, infelizmente, trabalhando nesse nível, várias verificações precisam ser executadas de maneira que os pixels possam ser agrupados em regiões com uma mínima confiabilidade. A transformada de Hough probabilística usada pelo time Bold Hearts Bold... (2014b), da universidade de Hertfordshire, Inglaterra, parece fornecer essa confiabilidade ao custo de mais processamento. Nenhum time parece ter usado a informação temporal da câmera. Na verdade, o processamento aparenta ter sido efetuado como se tirassem diversas fotos e cada foto tivesse sido processada individualmente, sem que a informação do quadro anterior passasse para o próximo quadro.

Segmentar uma região por movimento é feito, de maneira geral, comparando os pixels do quadro atual com os pixels do quadro anterior, de forma individual ou em forma de máscaras. O rastreamento por fluxo ótico (FLEET; WEISS, 2006) funciona muito bem para câmeras estáticas onde os pixels que representam o plano de fundo tem pouca ou nenhuma alteração são facilmente subtraídos, deixando apenas os pixels que podem significar um objeto em movimento. Para o robô proposto, a câmera se movimenta e oscila frequentemente fazendo com que essa técnica apresente regiões provenientes de falsos movimentos, visto que, se a câmera inteira se mover muitos pixels que deveriam ser considerados parte do plano de fundo se moverão também.

2.8.2 Identificação dos Robôs

Um importante item de um jogador de futebol é saber se seu adversário está a caminho de roubar-lhe a bola, ou saber a posição de seu companheiro de equipe para efetuar um passe. A identificação de robôs é especialmente difícil já que a variedade de robôs da liga é grande e a cada ano esses são modificados por suas equipes. Na verdade, na liga nenhum time demonstrou em seu trabalho de descrição de times qualquer menção ao reconhecimento de robôs.

O maior problema do algoritmo de reconhecimento de robôs é executá-lo em tempo real, devido ao número de dimensões ou características dos robôs e do fundo serem consideravelmente elevados, assim como, seus custos computacionais. Devido ao exposto, deve haver uma redução de características menos importantes. Na verdade o problema de reconhecimento dos robôs pode ser dividido em três partes:

- a) Redução de características com a extração dos descritores mais relevantes;
- b) Treinamento e classificação (É robô ou não é robô);
- c) Identificação do time pela sua respectiva cor. (Sorteada no começo da partida)

Um dos mais populares e bem sucedidos detectores de pessoas é o Histograma de Gradients Orientados¹⁰, usado juntamente com o classificador Máquina de Vetores de Suporte¹¹.

Estivill-Castro e Radev (2013) utilizaram essa abordagem com uma segmentação de cores para identificar a qual time cada robô pertence após um reconhecimento do torso do robô NAO (NAO, 2015)¹². Na standard platform league, todos os robôs obedecem ao padrão NAO, portanto se faz necessário identificar apenas um robô, e todos os outros terão o mesmo formato. Decorre do treinamento mais um problema dessa aplicação, a variedade de robôs presentes na categoria KidSize pode dificultar esse objetivo.

Há outros algoritmos que poderiam ser usados, como o Centrist (WU; REHG, 2011). Outros detectores usam o movimento para iniciar o rastreamento e, como já foi visto, a câmera na cabeça do robô vibra muito. Outros classificadores como o LBP (VIOLA; JONES, 2001), o PCA/LDA (MARTÍNEZ; KAK, 2001), entre outros, podem ser usados já que a classificação ocorre comparando vetores de descritores de treinamento com vetores de teste.

¹⁰HOG - Do inglês: "Histogram of Oriented Gradients"

¹¹SVM - Do inglês: "Support Vector Machine"

¹²TDP - Do inglês: "Team Description Papers"

Tabela 2 – Quadro de referência

Time	Sistema Cores	Segmentação	Bola	Gol	Linhas Campo	Robô
CIT Brains	RGB - HSV	LUT	Transição de pixels	Transição de pixels	Segmentação Branco com Verificação Linear	–
Bold Hearts	RGB - HSV	LUT	Transição de pixels	Transformada Hough Probabilística	Segmentação Branco e Regressão Linear	–
Baset	RGB - HSV	LUT	Transição de pixels Contornos	Transição de pixels Contornos	Segmentação Branco RANSAC	–
Eros	YUYV nativo	LUT	Transição de pixels	Transformada de Hough	Segmentação Branco	–
AUTMan	RGB - HSV	LUT	Transição de pixels	Transição de pixels	Segmentação Branco	–

Fonte: Artigos de Descrição de Equipes⁴, 2014 (AUTMAN..., 2014), (BOLD..., 2014b), (CIT..., 2014), (BASET..., 2014), (EROS..., 2014)

2.9 CONSIDERAÇÕES FINAIS

Vimos nesse capítulo que as equipes, inseridas no domínio RoboCup, trabalham em baixo nível e em médio nível (AUTMAN..., 2014), (BOLD..., 2014b), (CIT..., 2014), (BASET..., 2014), (EROS..., 2014). Essas técnicas são bem utilizadas e têm resultados pertinentes porém, com a mudança das regras, esses algoritmos vão precisar de cada vez mais estágios de verificação. Vimos também que diversas adversidades são inerentes do domínio, e que por esse motivo precisam de técnicas que possam identificar características, mas, que também, possam levar em conta algum contexto do objeto pertencente ao jogo. Dessa forma, ao lançar o artifício de utilizar técnicas de alto nível e considerando o que outras equipes têm feito, serão abordados no próximo capítulo dois aspectos do sistema de visão:

- a) a identificação e rastreamento da bola.
 - Algoritmo proposto conforme algoritimo 2
 - Descritor e Classificador Haar-AdaBoost
- b) a detecção de robôs.
 - Descritor e Classificador HOG-SVM

- Descritor e Classificador Haar-AdaBoost

Nos dois casos, as técnicas serão postas a prova com o mesmo conjunto de imagens pertinentes aos seus respectivos objetivos. De maneira que possam coexistir no sistema de visão do robô sem onerá-lo em demasia.

3 SISTEMA DESENVOLVIDO

No embasamento teórico, no capítulo 2, foi feita uma revisão bibliográfica de alguns algoritmos de visão computacional bem conceituados, além de, o que outras equipes têm feito com esses algoritmos no sentido de solucionar os problemas visuais do jogo de futebol. Utilizando o conhecimento desse capítulo, a proposta desse trabalho tem por objetivo criar um sistema de visão rápido e eficaz, abordando o uso dessas técnicas no domínio proposto. Visando facilitar o entendimento de suas diversas aplicações, as seções a seguir foram sub-divididas em: 3.1, que lida com as propostas escolhidas para o rastreamento da bola. A seção 3.2 apresenta a proposta para a detecção de robôs (companheiros ou adversários). A seção 3.4 trata do controle dos servo-mecanismos pan-tilt e como cada um dos algoritmos podem controlá-lo em um determinado instante de tempo. E, finalmente, a seção 3.5 propõe como todos esses algoritmos estariam interligados.

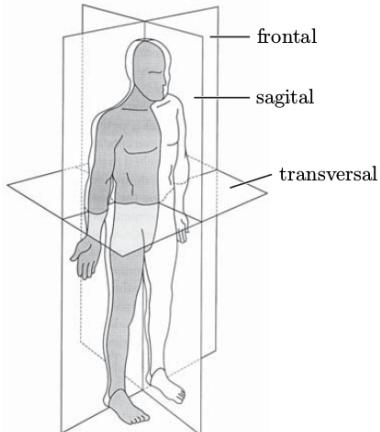
3.1 BOLA

A bola tem características próprias, que podem ser usadas para identificá-la, além de outras duas informações que são do campo, de fundo verde, e estar abaixo da linha do horizonte, entenda-se por linha do horizonte pela linha mediana da imagem de captura quando a câmera está posicionada paralelamente ao plano transversal do robô e paralelo ao piso. Ver figura 12 para referência sobre o plano transversal.

Na verdade, nenhum robô da liga é capaz de erguer a bola, isso quer dizer que, os robôs atualmente desenvolvidos ainda não chutam a bola de forma a erguê-la mais que alguns centímetros do chão, assim é razoável estimar que a bola estará sempre abaixo da linha do horizonte, tipicamente envolta por uma região verde com a matiz entre 190 e 270 graus. Isso irá evoluir gradativamente e significará uma quebra de paradigma para o rastreamento da bola, que não será abordado neste trabalho de pesquisa.

O rastreamento clássico, usando o fluxo ótico de pixels (FLUXO..., 2007), falha nessa aplicação devido à tremulação da câmera e também ao fato dela não estar fixa em um determinado ponto do campo. O quadro adquirido da captura da câmera é inserido no sistema de rastreamento da bola e é uma variável bloqueada via mutex. Esse bloqueio permite que diver-

Figura 12 – Planos que cortam o ser humano utilizados em robôs humanoides.



Fonte: Parte integrante do livro Westervelt et al., 2007.

sos processos paralelos acesssem a captura da câmera sem modificá-la, o que inviabilizaria o uso dessa imagem por vários processos de forma paralela.

A ideia aqui é começar fazendo um pré-processamento da imagem transformando o sistema de cores do RGB para o HSV, isso é especialmente importante, já que o HSV possui uma certa robustez à mudanças de iluminação.

Duas abordagens serão comparadas, uma usando segmentação e outra usando a técnica HAAR-AdaBoost para reconhecimento de objetos. Para a bola laranja a abordagem de segmentação e círculos de Hough com a imagem devidamente pré-processada e utilizando a técnica HAAR-AdaBoost para a detecção da bola branca. Dividindo a resolução da tela em nove regiões, é possível inferir de forma qualitativa se a bola está à direita, esquerda e, de forma quantitativa calculando valores de distância. Isso será melhor explicado na seção que trata do controle servo-visual Pan-tilt na seção 3.4.

3.1.1 A Bola Laranja

No caso da bola laranja, após segmentação usando uma LUT (Look-Up Table), duas operações morfológicas, Dilatação e Erosão, são aplicadas, ambas para retirar áreas e ruídos que não fazem parte da região de interesse. Ao mover-se pelo campo de visão do robô a bola provoca um borramento em sua imagem, para minimizar esse borramento aplica-se uma suavização usando uma Gaussiana. A transformada de Hough para círculos é aplicada e, com isso, temos a posição do centro da bola e seu raio expressos em pixels. Note que a bola usada em 2014, figura

13, é lisa em termos construtivos, chegando inclusive a refletir a luz branca, porém, em termos de cor possui uma heterogeneidade bem marcante.

Figura 13 – A bola utilizada em 2014.



Fonte: Autor.

A fim de incorporar a informação temporal, os centros dos círculos e raios encontrados serão armazenados na forma de média móvel (Ver referência de séries temporais no “ANEXO A”, trecho do trabalho de Faria et al. (2008)). Isso é importante para melhorar a estimativa da posição da bola. Essa média garante que a bola estará em uma posição muito próxima de sua última detecção. Como o tamanho da bola é conhecido, podemos achar uma relação entre o tamanho real em centímetros e o tamanho da bola em pixels, dada a resolução da tela.

Na medida que a bola se afasta, seu tamanho em pixels se reduz até o momento que o algoritmo acaba tratando-a como ruído, nesse momento a ideia é utilizar o algoritmo de condensação, a fim de, tentar inferir sua posição de forma qualitativa se está a direita ou a esquerda no campo de visão do robô. Infelizmente seu tamanho não será calculado até que o robô se aproxime novamente da bola. Perceba aqui que, para distinguir a bola do plano de fundo, a cor foi utilizada como característica antes do formato, isso foi feito, visando diminuir o espaço de ação da transformada de Hough para círculos, tornando-a mais rápido. Veja algoritmo 2 proposto.

No capítulo 2 fica claro que a transformada de Hough para círculos testa todas as possibilidades de círculo para todos os pixels encontrados usando três parâmetros, dessa forma, se esse algoritmo fosse aplicado à toda a imagem, provavelmente o algoritmo ficaria mais robusto porém, às custas de mais processamento.

Algoritmo 2– Algoritimo de rastreamento da bola laranja - Círculos de Hough.

```

1 Entrada: Abra Captura da Câmera (Dispositivo 0)
2 se Captura vazia então
3   | Erro! Dispositivo não está pronto. Aborte Algoritimo.
4 fim
5 enquanto Não for pressionada a tecla esc e Quadro não vazio faça
6   | Extraia um quadro RGB da captura, converta para uma imagem HSV.
7
8

$$H = \begin{cases} \theta, & SeB \leq G \\ 360 - \theta, & SeB > G \end{cases} \quad (7)$$

9

$$\theta = \cos^{-1} \left[ \frac{\frac{1}{2} * [(R - G) + (R - B)]}{\sqrt{[(R - G)^2 + (R - B)(G - B)]}} \right] \quad (8)$$

10

$$S = 1 - \frac{3}{(R + G + B)} * [min(R, G, B)] \quad (9)$$

11

$$I = \frac{1}{3} * (R + G + B) \quad (10)$$

12 Entrada: Tabela de pesquisa (LUT). ( $0 < H < 35$ ,  $0 < S < 180$ ,  $0 < V < 255$ )
13 para todo Pixel dentro da imagem HSV e dentro dos limites da LUT faça
14   | Altere o valor do Pixel de uma matriz binária na mesma posição
15 fim
16 Na matriz binária aplique Erosão seguida da Dilatação. ( 5x5 Pixels)
17 Aplique filtro espacial. (Gaussiana 9x9 Pixels)
18 Transformada de Hough para círculos:
19   - Atualize a média móvel do raio (MRB) e da posição XY dos círculos
        encontrados.
20
21 fim
22 retorna  $D_{bola}$ 

```

Fonte: Autor

3.1.2 A Bola Branca

A modificação do regulamento da competição fez com que todas as equipes tivessem que pensar em uma forma diferente de reconhecer a bola. A bola branca da figura 14, diferentemente da usada anteriormente, possui uma textura que pode ser usada como característica discrepante. Assim, a proposta é o descritor HAAR com o classificador AdaBoost, onde diversas imagens serão coletadas com a própria câmera do robô. A bola será posicionada em diferentes distâncias e posições no campo de futebol, essas imagens denominadas imagens positivas possuem a bola em qualquer posição da imagem. A bola se encontrará por completo na imagem, isso quer dizer que não haverá oclusão nas imagens de treinamento.

Figura 14 – A bola utilizada em 2015.



Fonte: Autor.

As imagens negativas são adquiridas rotacionando o robô em diversas posições do campo de futebol. Imagens de multidões também foram usadas. Isso é feito para que se possua uma visão 360 graus do campo e para que de qualquer posição do robô o mesmo possa identificar um plano de fundo negativo, mesmo com muitas pessoas em volta e aumenta a probabilidade de que a bola será encontrada nessas condições.

Tanto o algoritmo de treinamento 3, quanto o de identificação online 4, estão sendo propostos utilizando-se os parâmetros de seus respectivos artigos originais, mas alguns parâmetros serão postos a prova no capítulo 4.

Como forma de checar a qualidade do classificador uma curva ROC será plotada e como forma de verificar o desempenho será usada a taxa de quadros por segundo após processamento para a resolução de 1920x1080 pixels. Essa resolução foi escolhida devido ao seu potencial de

Algoritmo 3– Algoritmo de Identificação da bola branca. (HAAR-Adaboost Offline)

```

1 Entrada: Conjunto de imagens positivas - Contém robôs participantes
2 Entrada: Conjunto de imagens negativas - Não contém robôs
3 Entrada: Posição da bola contida em todas as imagens positivas
4 para todo Estágio da árvore e para todas as imagens faça
5   enquanto  $MinTaxaAcerto = 0,999$  não atingida faça
6     Extraia as características de HAAR para as imagens positivas.
7     Execute o Adaboost conforme algoritmo 1 usando como parâmetros:
8       - Tamanho da amostra = 20x20 Pixels;
9       - Tamanho mínimo do objeto = 30x30 Pixels;
10      - Tamanho máximo do objeto = Resolução vertical x  $\frac{ResoluçãoHorizontal}{2}$  Pixels
11      - Classificador 8 Vizinhos mais próximos.
12    fim
13    Armazene a hipótese final do estágio.
14 fim
15 Armazene a hipótese final em um arquivo .xml

```

Fonte: Autor “Adaptado de” Freund e Schapire, 1997

Algoritmo 4– Algoritmo de Identificação da bola branca. (HAAR-Adaboost Online)

```

1 Entrada: Árvore proveniente do Adaboost (Arquivo .xml)
2 Entrada: Aquisição da câmera
3 enquanto Quadro não vazio faça
4   Extraia o vetor de características (descritores) da imagem adquirida.
5   Percorra os nós (Estágios) da árvore presentes no arquivo xml encontrando a
melhor classificação para a imagem adquirida
6   se Houver bola presente na imagem então
7     Para cada bloco da imagem adquirida, encontre em qual bloco a bola se
encontra
8     Calcule seu raio em pixels
9     Armazene o centro do círculo e a raio em pixels
10    Calcule sua distância real até o robô em centímetros
11  fim
12 fim

```

Fonte: Autor “Adaptado de” Freund e Schapire, 1997

alcance de detecção da bola, já que existe uma correlação entre o tamanho da mesma em pixels e sua distância da câmera.

3.2 ADVERSÁRIOS E COMPANHEIROS DE EQUIPE

A ideia geral é identificar objetos conhecidos e parecidos com relação à alguma característica. Com os robôs não seria diferente, entretanto, existem poucas características semelhantes entre os robôs da liga. Na verdade, mesmo que muitas equipes ainda usem uma plataforma (Darwin-OP) (HA et al., 2011) com algumas modificações, os robôs, de maneira geral, têm formas e tamanhos variados, veja figura 15, e identificar esses robôs de forma confiável é complexo.

Os descritores HOG e HAAR e seus classificadores em sua forma clássica, respectivamente, SVM e AdaBoost serão testados para a identificação dos robôs em campo.

Figura 15 – A diversidade de robôs da liga humanoide.



Fonte: Autor.

Ambos os descritores são computados para todas as imagens de um conjunto de imagens positivas e negativas, entenda-se imagens positivas, como sendo imagens que possuam apenas um robô em qualquer pose e em diversos cenários. Esses vetores de descrição são inseridos em seus respectivos classificadores, sendo esse treinamento efetuado em um computador externo ao robô com as mesmas especificações. O classificador treinado é inserido no robô, que, durante o jogo, calcula o descriptor de um quadro adquirido e classifica o vetor de descrição no classificador treinado. A resposta do classificador informa se há robô no quadro e qual região tem a maior quantidade de descritores, apontando em qual região o robô foi identificado na imagem. Os algoritimos propostos [5] [6] [7] [8] são apresentados conforme revisão feita nos capítulos anteriores.

O mesmo conjunto de imagens positivas e negativas serão computadas tanto pelo HAAR com o AdaBoost como pelo HOG com o SVM, de forma que seja possível definir qual possui maior precisão, com maior taxa de quadros por segundo após processamento, para o robô proposto.

Algoritmo 5– Algoritmo para Identificação de Robôs. (HOG-SVM Offline)

- 1 **Entrada:** Conjunto de imagens positivas - Contém robôs participantes
- 2 **Entrada:** Conjunto de imagens negativas - Não contém robôs
- 3 Extraia, do diretório de imagens negativas, todas as imagens que possuam tamanho e extensão compatíveis (jpg,jpeg,png,bmp)
- 4 Calcule o histograma de gradientes para todas as imagens.
- 5 Armazene o vetor de características (descritores) de cada uma das imagens negativas.
- 6 Extraia, do diretório de imagens positivas, todas as imagens que possuam tamanho e extensão compatíveis (jpg,jpeg,png,bmp)
- 7 Calcule o histograma de gradientes para todas as imagens desse diretório.
- 8 Armazene o vetor de características (descritores) de cada uma das imagens positivas.
- 9 Armazene o vetor de características (descritores) de cada uma das imagens negativas.
- 10 **Entrada:** Vetores de características (descritores) das imagens positivas
- 11 **Entrada:** Vetores de características (descritores) das imagens negativas
- 12 Determine os vetores de suporte e o hiperplano que melhor separa as duas classes usando o SVM.
- 13 **Entrada:** Tabela com todas as alturas de todos os robôs de todos os times
- 14 **Entrada:** Tabela com a cor do time adversário

Fonte: Autor “Adaptado de” Dalal e Triggs, 2005.

3.3 HOG-SVM ONLINE

Algoritmo 6– Algoritmo para Identificação de Robôs. (HOG-SVM Online)

```

1 Entrada: Hiperplano proveniente do SVM
2 Entrada: Aquisição da câmera
3 enquanto Quadro não vazio faça
4   Calcule o histograma de gradientes da imagem adquirida da câmera
5   Extraia o vetor de caracterísiticas (descritores) da imagem adquirida.
6   Use o SVM para classificar o vetor de caracterísiticas da imagem adquirida.
7   se Houver robô presente na imagem então
8     Para cada bloco da imagem adquirida, encontre em qual bloco do histograma
      normalizado o robô se encontra
9     Calcule sua altura em pixels
10    Armazene o centro do bloco e a altura em pixels
11    Segmentação da cor do time.
12    Se a área na qual o robô for identificado conter o centroide referente à cor do
      time
13    Busque na tabela de alturas qual a altura em centímetros do robô que foi
      identificado
14    Calcule sua distância real até o robô em centímetros
15  fim
16 fim

```

Fonte: Autor “Adaptado de” Dalal e Triggs, 2005.

Com essas informações, a altura do robô em pixels e a informação de que existe pelo menos um robô na imagem, é necessário identificar a qual equipe o(s) robô(s) pertence(m). Para tal, usa-se a informação da cor da camisa via segmentação simples, ou seja, se a região, delimitada pela janela de detecção, onde foi identificado o robô coincidir com a região que possui uma área azul (ciano), esse é um robô pertencente ao time azul. As duas cores, azul ciano ou magenta, são sorteadas antes do início da partida. Com o(s) robô(s) identificado(s), é necessário calcular sua(s) distância(s) até o robô que está vendendo a ação. Por se tratar de uma visão monocular, calcular a distância dos robôs necessita da informação de sua altura real assim, uma tabela de pesquisa é alimentada com as informações de altura de todos os robôs da liga em centímetros e, finalmente, com a altura do robô em pixels calculado no passo anterior e a altura real, é possível estimar a distância entre esses dois robôs.

Algoritmo 7– Algoritmo de Identificação de robôs. (HAAR-Adaboost Offline)

```

1 Entrada: Conjunto de imagens positivas - Contém robôs participantes
2 Entrada: Conjunto de imagens negativas - Não contém robôs
3 Entrada: Posição dos robôs contidos nas imagens positivas
4 para todo Estágio da árvore e para todas as imagens faça
5   enquanto  $MinTaxaAcerto = 0,999$  não atingida faça
6     Extraia as características de HAAR para as imagens positivas.
7     Execute o Adaboost conforme algoritmo 1 usando como parâmetros:
8       - Tamanho da amostra = 20x20 Pixels;
9       - Tamanho mínimo do objeto = 30x30 Pixels;
10      - Tamanho máximo do objeto = Resolução vertical  $\times \frac{ResoluçãoHorizontal}{2}$  Pixels
11      - Classificador 8 Vizinhos mais próximos.
12    fim
13    Armazene a hipótese final do estágio.
14 fim
15 Armazene a hipótese final em um arquivo.

```

Fonte: Autor “Adaptado de” Freund e Schapire, 1997

Algoritmo 8– Algoritmo de Identificação de robôs. (HAAR-Adaboost Online)

```

1 Entrada: Árvore proveniente do Adaboost (Hipótese Final)
2 Entrada: Aquisição da câmera
3 enquanto Quadro não vazio faça
4   Extraia o vetor de características (descritores) da imagem adquirida.
5   Percorra os nós (Estágios) da árvore para a imagem adquirida
6   se Houver robô presente na imagem então
7     Na imagem encontre em qual bloco o robô se encontra
8     Calcule sua altura em pixels
9     Armazene o centro do bloco e a altura em pixels
10    Segmentação da cor do time.
11    Se a área na qual o robô for identificado conter a cor do time
12    Busque na tabela de alturas qual a altura do robô que foi identificado
13    Calcule sua distância real até o robô em centímetros
14  fim
15 fim

```

Fonte: Autor “Adaptado de” Freund e Schapire, 1997

3.4 CONTROLE SERVO-VISUAL

O controle servo-visual toma atitudes ativas e reativas, procurando os objetos quando não estão em seu campo de visão, de forma ativa, e seguindo-os, de forma reativa, quando estão à vista. A proposta do controle servo-visual foi feita dividindo o quadro capturado pela câmera em nove regiões distribuídas da seguinte forma: três na horizontal e cada uma dessas regiões é novamente dividida em três, dessa vez na vertical. As regiões não são divididas igualmente. A região central é menor do que as outras oito regiões pois, seria o equivalente do ser humano ao foco principal da visão, ou visão central, e as outras seriam equivalentes à visão periférica.

Independentemente da região na qual o objeto de interesse se encontra, o controle de pan-tilt deve mover a cabeça do robô até levá-lo à região central, ver algoritmo 9. Na figura 16 essa região se refere à região 5.

Figura 16 – Divisão esquemática do quadro em 9 regiões.

Região 1	Região 2	Região 3
Região 4	Região 5	Região 6
Região 7	Região 8	Região 9

Fonte: Autor

O tamanho da região 5 é fundamental para o bom funcionamento do sistema. Se for muito pequena, o robô terá problemas ao centralizar o objeto que sairá muito facilmente dessa região, se o contrário for verdade, a cabeça do robô se movimentará muito pouco e o objeto não será centralizado corretamente. A proposta aqui é que a região 5 tenha 5% da resolução da câmera nos dois eixos.

Quando o objeto a ser procurado não estiver no campo de visão, um algoritmo de varredura entra em ação, varrendo em três diferentes níveis, na vertical começando em 0 graus, passando por -22,5 graus e finalmente -45 graus, enquanto movimenta a cabeça na horizontal da esquerda para a direita e da direita para a esquerda, em cada um desses níveis.

Algoritmo 9– Algoritmo para buscar e centralizar o objeto

```

1 Divilde o quadro de captura em 3 regiões horizontais, sendo a região central com
  5% da resolução horizontal.
2 Divilde o quadro de captura em 3 regiões verticais, sendo a região central com 5%
  da resolução vertical.
3 enquanto Quadro não vazio faça
4   | se Dentro do quadro houver objeto então
5     |   Encontre em qual região o objeto se encontra usando sua posição.
6     |   se O objeto estiver na região 5 (central) então
7       |     | Mantenha o objeto nessa região parando o movimento
8     |   fim
9     |   se O objeto estiver nas regiões 1, 4 e 7 então
10    |     | Some a posição atual do servo com a quantidade de graus para o pan
11    |   fim
12    |   se O objeto estiver nas regiões 3, 6 e 9 então
13    |     | Subtraia a posição atual do servo com a quantidade de graus para o pan
14    |   fim
15    |   se O objeto estiver nas regiões 1, 2 e 3 então
16    |     | Some a posição atual do servo com a quantidade de graus para o tilt
17    |   fim
18    |   se O objeto estiver nas regiões 7, 8 e 9 então
19    |     | Subtraia a posição atual do servo com a quantidade de graus para o tilt
20    |   fim
21    |   Guarde posição dos servos
22   fim
23   senão
24     |   Chame algoritmo de busca varredura
25   fim
26 fim
27 retorna PosicaodoServo

```

Fonte: Autor.

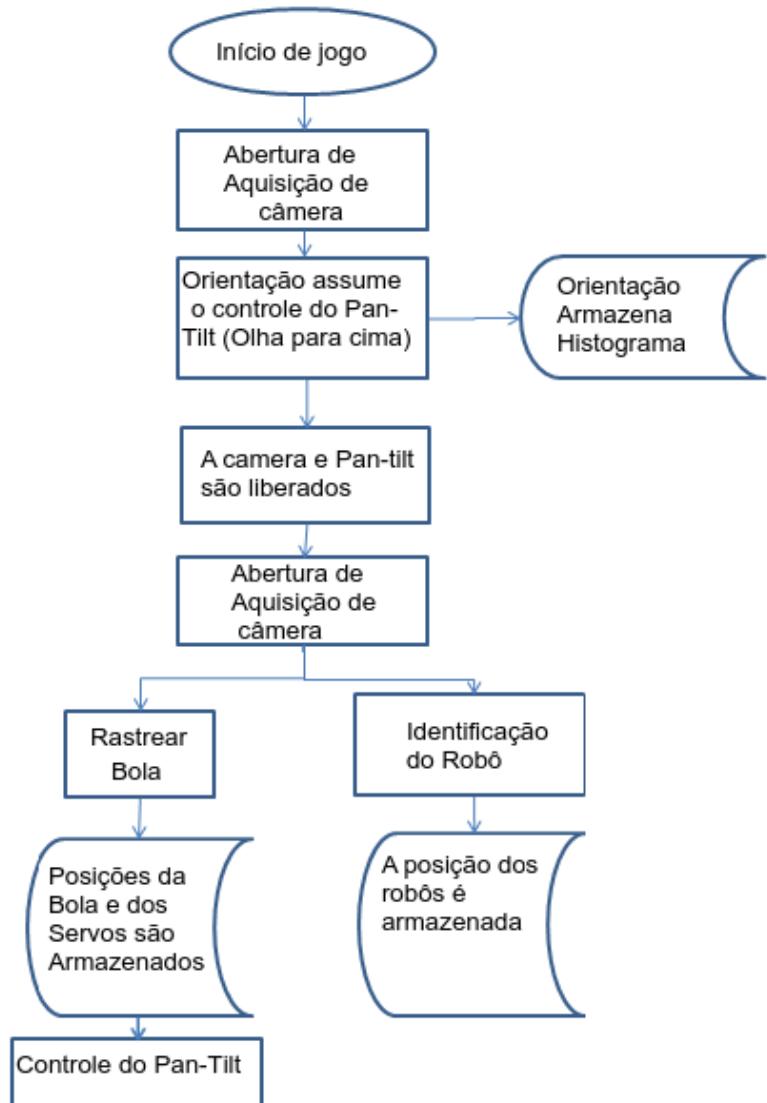
3.5 O SISTEMA DE VISÃO

A visão do robô, a ser desenvolvida, será um sistema com dois processos paralelos e dois sequenciais. Todos usarão um quadro disponível no momento de aquisição da câmera. Isso permitirá que todos os processos de visão adquiram informação do ambiente praticamente ao mesmo tempo.

Apenas um processo poderá ter controle sobre o Pan-Tilt em um determinado momento, isso, entretanto, não impedirá que os outros processos continuem a adquirir e processar as informações provenientes do ambiente, ver figura 17. Haverá um atraso entre processos devido à diferença de complexidade entre as tarefas a serem executadas, isso se torna óbvio já que algumas são computacionalmente mais custosas do que outras.

O que acontece é que a variável quadro é protegida por mutex, o bloqueio permite que apenas um dos processos utilize essa variável. Por exemplo, o algoritmo da bola é mais rápido que o de identificação de robôs, assim pode ser que o quadro disponível para reconhecimento de robôs seja diferente do que foi usado para rastreamento da bola. O mutex ou exclusão mútua, é um recurso técnico usado em programação paralela para evitar que dois processos tenham controle de um recurso que precisa ser compartilhado. O meio mais comum de se efetuar a exclusão mútua é através de um semáforo, que bloqueia o recurso para cada um dos processos liberando-o do processamento em seguida, de forma que no período de bloqueio apenas o processo definido pelo mutex pode controlar o recurso.

Figura 17 – Visão Geral do Software a ser Implementado.



Fonte: Autor

4 EXPERIMENTOS REALIZADOS E RESULTADOS

Há muitos aspectos na construção de um sistema de visão confiável, este deve ser compatível com um robô de ação rápida, sem sobrecarregar nenhum dos processos de visão. Esse capítulo começa na seção 4.1 fazendo uma breve introdução ao hardware do robô que foi utilizado para execução dos experimentos, o mesmo continua apresentando experimentos para identificação da bola laranja seção 4.2 e da bola branca 4.3, assim como o cálculo de suas distâncias ao robô na seção 4.4. Os experimentos da detecção de robôs se iniciam com a seção 4.5 onde o algoritmo HOG-SVM foi comparado com o detector de pedestres clássico com imagens de humanos e com poucas imagens de treinamento para robôs. Na seção 4.6 o detector de robôs foi implementado com o uso do HAAR-Adaboost. Na seção 4.7, os algoritmos HOG-SVM e HAAR-Adaboost são postos em perspectiva, ambos com os mesmos parâmetros e quantidade de imagens, de modo que algumas conclusões podem ser tomadas apresentando os resultados de cada experimento em sua respectiva seção.

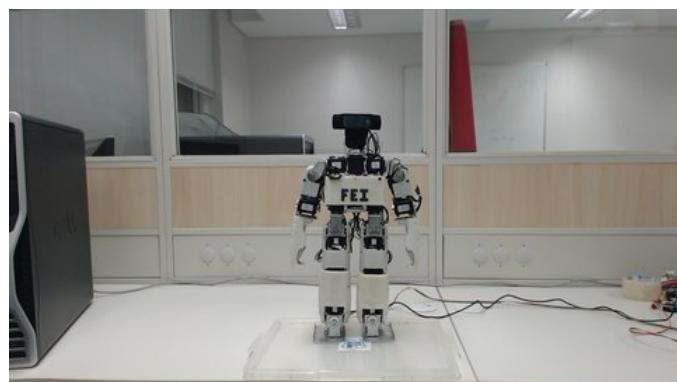
4.1 O ROBÔ

Quatro robôs humanoides autônomos foram desenvolvidos para atuar na liga KidSize da RoboCup. Para os experimentos do presente, o robô utilizado foi o B2, robô baseado no DARwIn-OP (HA et al., 2011), veja figura 18 desenvolvido no Centro Universitário da FEI. O robô DARwIn-OP tem sua plataforma aberta, que foi criada e desenvolvida pela fabricante coreana de robôs ROBOTIS em colaboração com a Virginia Tech, a Universidade de Purdue e a Universidade da Pennsylvania. O DARwIn-OP têm vinte graus de liberdade sendo cada um desses graus controlado por um servo-mecanismo DYNAMIXEL MX-28T.

Duas grandes modificações foram feitas: A primeira foi a extinção do microcontrolador ARM 32-bit Cortex-M3, responsável pelo controle de baixo nível dos servos mecanismos, isso implica que os pacotes enviados para os servos do pan-tilt precisam ser enviados da porta USB do computador através de um conversor RS485 para os servos, a segunda foi a instalação da câmera modelo c920, full hd e 78 graus de campo de visão, da Logitech.

O peito do robô teve que ser redesenhado e reimpresso em uma impressora 3D, de modo que, seu espaço foi aumentado para que fosse possível a substituição do mini-computador com

Figura 18 – O Robô Darwin Modificado.



Fonte: Autor

processador Intel Atom Z530 1.6GHz por um Intel NUC Core I5 com 2.5GHz, 4Gb de RAM e placa de vídeo HD Graphics 4000 com capacidades de trabalhar com resoluções de 4K.

O sistema operacional usado foi o linux Ubuntu 12.04. Usando a posição pan-tilt proveniente da biblioteca fornecida pela Dynamixel, fornecedora dos servo-mecanismos. Vários testes foram executados com os diversos objetos presentes no campo, sob diversas condições de luz. Os algoritmos foram desenvolvidos em C++, em conjunto com a biblioteca open source OpenCV... (2014).

4.2 EXPERIMENTO - ENCONTRANDO UMA BOLA LARANJA USANDO SEGMENTAÇÃO E TRANSFORMADA DE HOUGH PARA CÍRCULOS

Esse experimento foi implementado conforme o algoritmo 2 e foi testado em diversas situações de variação de luz e oclusão.

Os parâmetros do HSV podem assumir valores de 0 à 359 para a matiz, de 0 à 255 para a saturação e para a intensidade. Como parte dos requisitos de calibração da cor referente à bola, um algoritmo utilizando barras de rolagem (trackbars) foi implementado de forma que era possível alterar os parâmetros da segmentação apresentando-a em tempo real. Alterando esses três parâmetros foi possível visualizar quando a bola se tornava discrepante do fundo. Os valores encontrados variaram de 0 a 35 graus para a matiz, de 0 a 200 para a saturação e 0 a 180 para a intensidade dependendo da iluminação do local.

Dessa forma, para a LUT a matriz foi discretizada em 7 passos de 5 graus cada, a saturação em 10 passos e a intensidade em 10. Isso reduz o tempo de segmentação, pois o algoritmo tem que percorrer menos estados para efetuar a segmentação.

O pré-processamento é um dos fatores cruciais para se utilizar os círculos de Hough com alguma precisão. Ao testar diversos valores para o limiar do acumulador, este parece ser um parâmetro chave. Basicamente quanto maior for esse limiar menos círculos são detectados, mesmo assim esses círculos têm uma probabilidade maior de serem detectados corretamente (DAVIES, 1988). É possível fazer uma busca de parâmetros binária até que um certo critério seja atingido, no caso apenas um círculo. O valor definido, após algumas tentativas, ficou estabelecido em 50.

Para os círculos de Hough, o tamanho máximo do círculo a ser detectado foi limitado à 30 pixels de diâmetro, essa escolha visa reduzir os falsos positivos e ruídos, mas também possui o revés de reduzir o alcance de detecção da bola, que nesse caso ficou limitado a 3 metros. De forma análoga, o tamanho máximo de detecção ficou definido como sendo a largura da tela dividido por 2, visto que a bola, estando no chão, nunca terá um tamanho maior que esse.

Outra forma de reduzir os falsos positivos foi utilizar a informação temporal da câmera e armazenar na forma de média móvel (Ver referência de séries temporais no “ANEXO A”, trecho do trabalho de Faria et al. (2008)) a posição e raio atual da bola. As posições cartesianas da bola dos últimos 10 quadros são armazenados e têm sua média aritmética calculada em forma de fila, assim uma falsa detecção que não representa uma bola tem sua influência reduzida na posição real da bola.

Durante os experimentos, enquanto a bola esteve no campo de visão da câmera, esse algoritmo de rastreamento bola provou ser muito confiável mesmo quando um borramento é causado pelo movimento da bola. Este algoritmo foi amplamente testado para oclusão parcial (Veja figura 19) e robustez em diferentes condições de iluminação, incluindo a luz direta do sol e sombras (ver Figura 20).

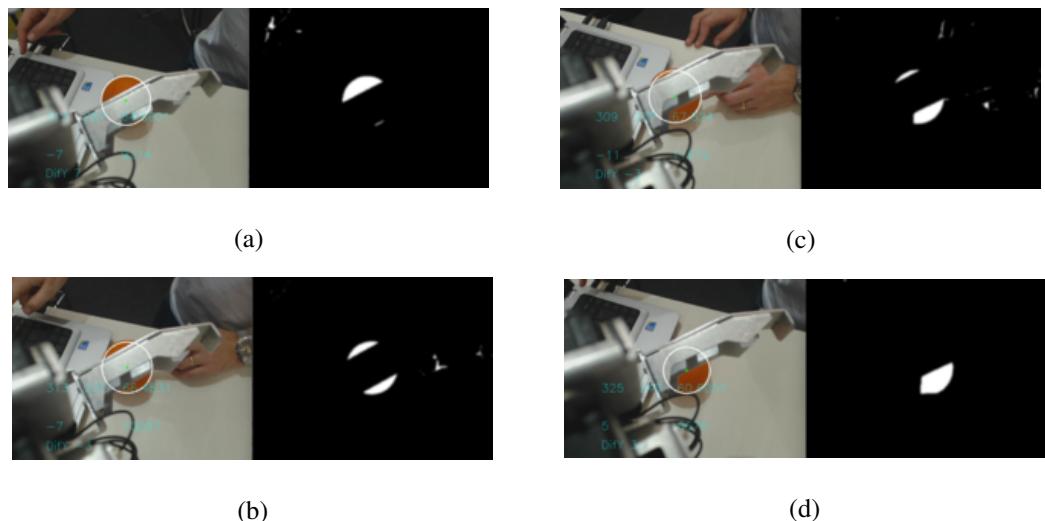
Na medida que o tamanho da bola, em pixels, é extremamente importante para calcular a distância verdadeira bola, a oclusão é um problema difícil que tem de ser resolvido. Os testes mostrados na figura 19 demonstram o algoritmo estimando o raio da bola, mesmo quando a maior parte da esfera está oculta. Se o formato for prejudicado por qualquer motivo a distância real da bola também o será.

Foi feito um teste com a bola, no qual o algoritmo proposto foi complementado com o algoritmo de condensação, ver figura 21, citado no capítulo de trabalhos correlatos. Nesse teste, enquanto a bola se encontrava à uma distância menor do que 2 metros, o algoritmo de

rastreamento executava os círculos de Hough, assim que a mesma se distanciava além desse limite, o algoritmo mudava para o de condensação onde 1.500 partículas foram utilizadas como uma forma de identificar probabilisticamente a posição da bola. Essa quantidade de partículas foi definida após contabilizar a área, em pixels, referente à bola quando essa se encontrava a 6 metros do robô, na resolução full-hd, onde possuía o raio de 22 pixels.

Apesar dos resultados animadores, onde foi possível estimar a posição da bola a mais de 6 metros, ver tabela 3, não foi possível adequar essas duas técnicas juntamente com o restante do sistema de visão. Esse algoritmo, de forma individual, manteve a taxa de quadros por segundo de 28, porém, ao incluir os outros algoritmos, o mesmo apresentou problemas de lentidão e acabou cessando de funcionar após alguns segundos, indicando que possa ter havido uma exigência muito grande do processador, fato comprovado pelo monitor de sistema do Ubuntu.

Figura 19 – O robô Darwin seguindo uma bola parcialmente oculta pelo seu braço.



Fonte: Autor

Figura 20 – Robô Milton seguindo a bola com a influência da luz do sol e sombras.



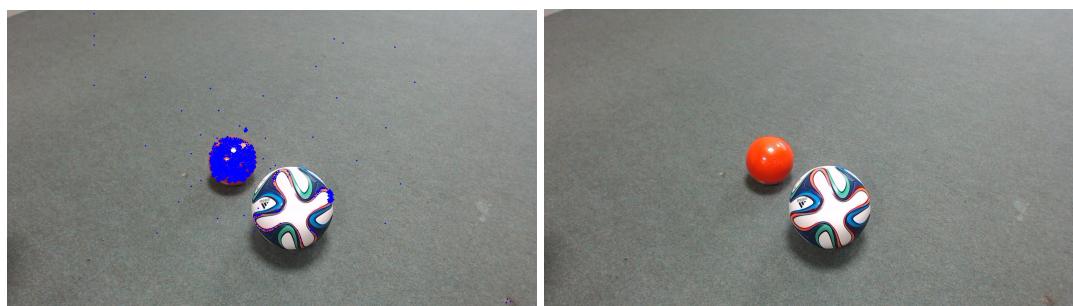
Fonte: Autor

Tabela 3 – Resultados dos algoritmos para reconhecimento da Bola

Descrição	Círculos de Hough	Condensação
Alcance da detecção (Centímetros)	300	600
Quadros por segundo	30	28

Fonte: Autor

Figura 21 – B1 utilizando a condensação para identificar probabilisticamente e qualitativamente a posição da bola laranja no quadro.



Fonte: Autor

4.3 EXPERIMENTO - ENCONTRANDO UMA BOLA BRANCA USANDO CARACTERÍSTICAS HAAR

A liga humanoide da RoboCup, vislumbrando a meta de 2050, vem fazendo diversas mudanças no campo e em seus objetos, incluindo a bola. A principal alteração efetuada foi a troca de sua cor predominantemente alaranjada, passando a ser no mínimo 80 por cento branca. A mudança das características da bola fez com fosse necessário uma mudança de paradigma para efetuar sua detecção. A segmentação por cores passou a apresentar diversos falsos positivos, basicamente devido ao brilho causado pelo reflexo branco da luz em diversos objetos que não representavam a bola. Assim, tendo como meta sanar esse revés, foi implementado um algoritmo utilizado as características de HAAR como descritor.

Um conjunto de imagens positivas e negativas foram coletadas na resolução full-hd (1920x1080 pixels). Aqui um conjunto de imagens negativas foi adquirida visando a diversidade contendo imagens de locais diversos como prédios, campos de futebol, multidões e ruas, sem que a bola estivesse envolvida.

As imagens positivas foram reunidas posicionando a bola em diferentes distâncias da câmera do robô e em diversas posições no campo de futebol. As imagens estão à disposição no endereço (<https://www.copy.com/id-568856/imagens>). O total de imagens positivas colhidas foi de 2135 imagens, já as negativas somaram 1192 imagens.

A janela mínima de detecção foi definida com 20x20 pixels, ou seja, o menor objeto (bola) a ser detectado terá essa dimensão. Essa janela foi escolhida visto que ela representa uma bola a 8 metros de distância. Essa distância é suficientemente pequena para que o robô estando em um lado do campo possa reconhecê-la do outro lado e, de forma oposta, não tão pequena que possam ser detectados falsos positivos. Durante o treinamento o número máximo de estágios foi determinado como sendo de 20 estágios, a taxa de acerto mínima foi configurada em 0.999, o parâmetro $N_{\text{AmostrasPositivas}}$ após cálculo utilizando as equações 5 e 6 ficou fixada em 920, finalmente, a janela de amostragem foi de 100x100 pixels. O classificador utilizado foi k vizinhos mais próximos (K-nearest neighbors) com valor de $k = 11$. Outros valores menores foram escolhas claramente ruins, já que geraram diversos falsos positivos. Bons resultados apareceram com o valor de $k=9$ mas, também geraram diversos falso positivos em diversas circunstâncias. A classificação ficou prejudicada com valores maiores valores de k.

Todos os objetos, foram selecionados manualmente usando uma caixa delimitadora retangular a partir das imagens positivas. O utilitário Criar amostras da biblioteca opencv (OPENCV..., 2014) foi usado para criar amostras de treinamento de cada imagem, enquanto alimentava um vetor.

Cada estágio do classificador foi dividido em dois nós, de modo que o resultado do classificador para cada nó foi armazenado em um arquivo xml.

No experimento apresentado na figura 22, duas bolas brancas idênticas estão posicionadas em posições diferentes (5 e 7 metros) em um campo de futebol de grama sintética sobre-carregado de objetos e pessoas se movendo.

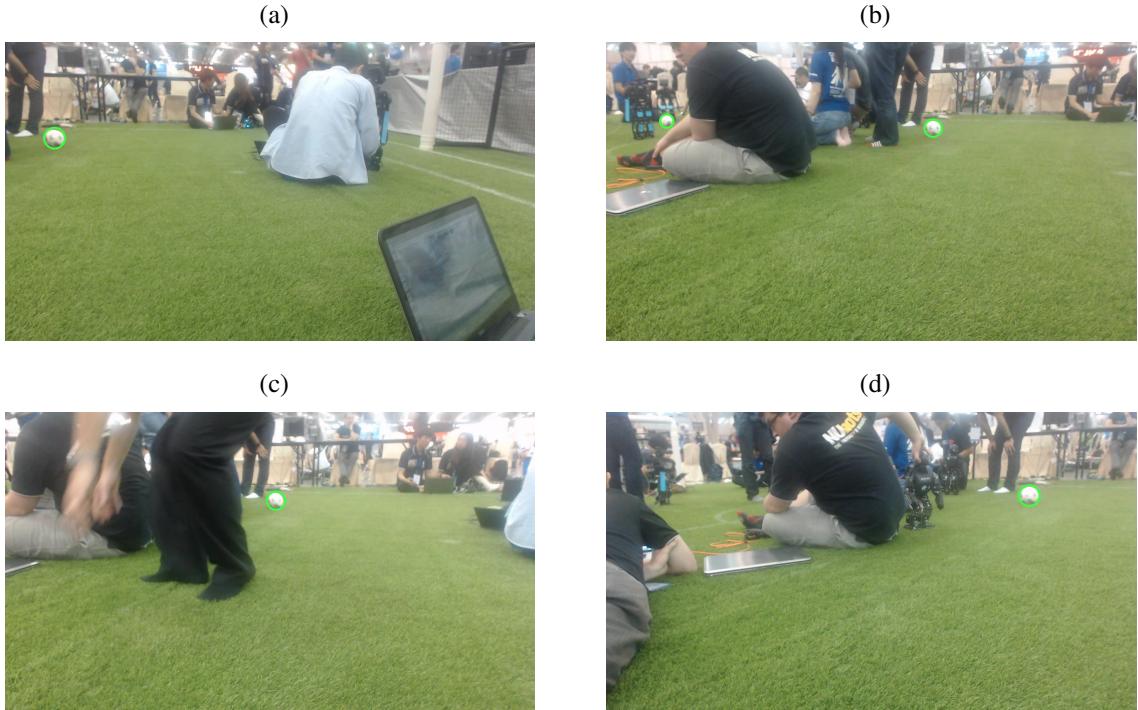
Na figura 23, a curva ROC demonstra os resultados da detecção.

4.4 EXPERIMENTO - DETERMINAÇÃO DA DISTÂNCIA DA BOLA AO ROBÔ

Os cálculos das distâncias da bola e do robô até outros robôs, foram determinados de forma experimental, tabelando a distância em centímetros e a área dos objetos em pixels, encontrando por meio de regressão a função que representasse essa relação, considerando, é claro, a resolução da captura.

O principal objetivo da média móvel simples é fornecer o valor médio do raio da bola e sua posição em pixels dentro de um determinado período, no caso, os últimos 10 quadros da captura de video. Médias Móveis eliminam grandes oscilações nos valores, filtrando somente a posição principal. Quanto maior o período utilizado na Média Móvel maior é o grau de

Figura 22 – Rastreamento da bola usando características HAAR em um campo de futebol sobrecarregado de objetos, pessoas e regiões não pertencentes ao jogo.



Fonte: Autor

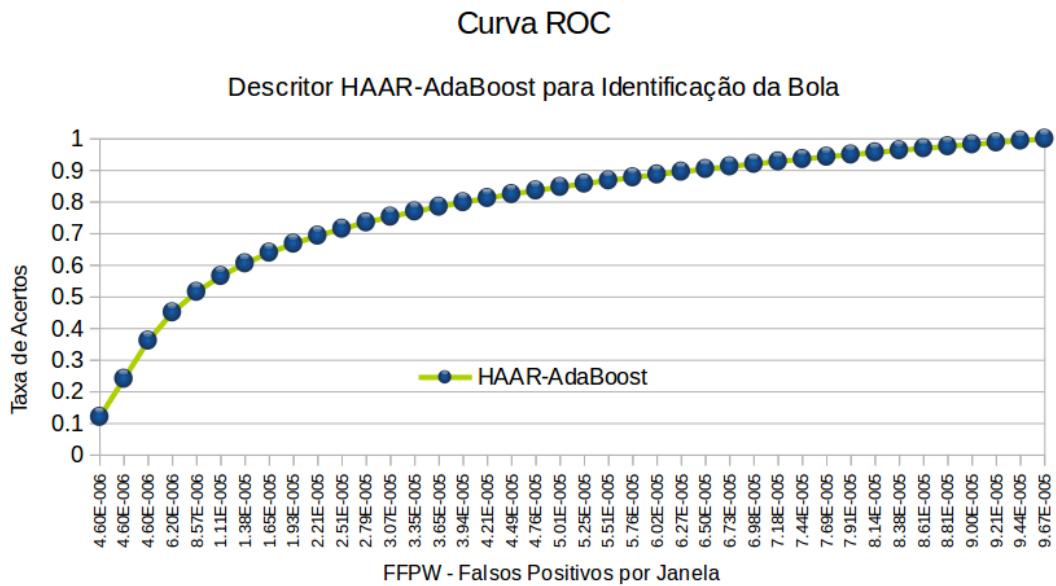
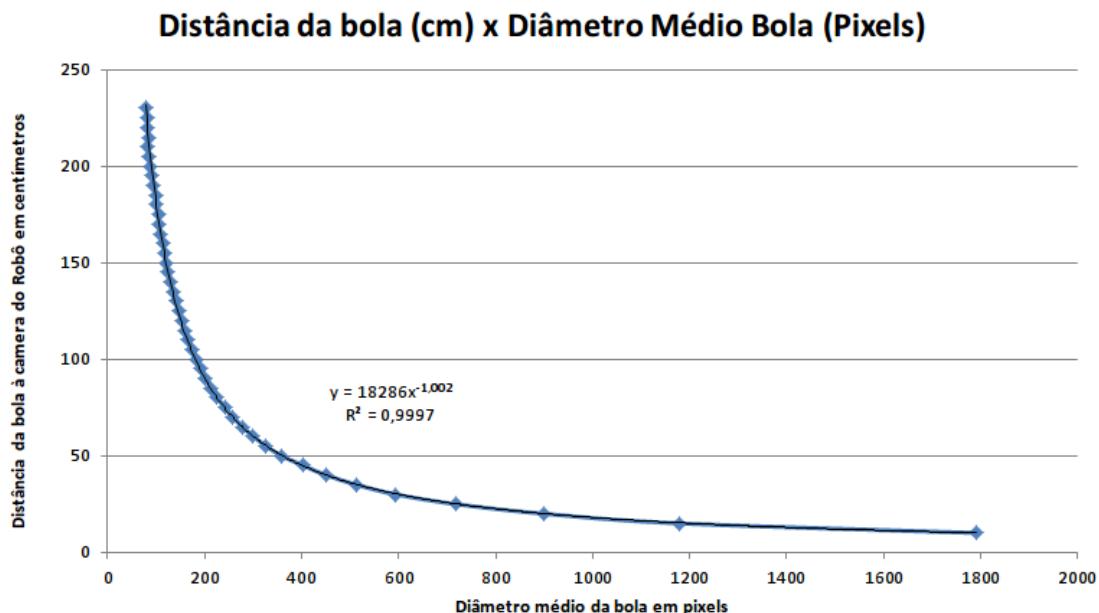


Figura 23 – Fonte: Autor

filtragem. Assim, para cada valor incluído no cálculo da média, o valor mais antigo é excluído. Na média móvel simples (SMA), cada dado utilizado no cálculo da média terá o mesmo peso.

$$MRB = \frac{[R(q) + R(q-1) + R(q-2) + \dots + R(q-9)]}{10} \quad (12)$$

Figura 24 – Função relacionando pixels com distância de uma bola de tamanho conhecido.



Fonte: Autor.

MRB é a média móvel do raio da bola e R é o raio da bola ambos em pixels, já q se refere ao índice do quadro atual da captura da câmera. Tabelando os raios médios da bola e sua distância real em centímetros, é possível por meio de regressão estimar uma função que representa os dados. O experimento foi realizado posicionando a bola no pé do robô e afastando-a do pé do robô com incrementos de 5 centímetros, a cada incremento foi coletado o diâmetro médio da bola em pixels, após coletar 30 pontos o gráfico na figura 24 foi plotado.

A função que relaciona o tamanho da bola em pixels com a distância da bola em centímetros foi extraída por meio de uma regressão não-linear exponencial, disponível no software Ms-Excel (MS..., 2015), para a resolução de 1920x1080 pixels, bola de tamanho fixo e conhecido de 13 centímetros, onde D_{Bola} é a distância da bola ao robô em centímetros, e MRB é a média móvel do raio da bola em pixels.

O coeficiente de determinação R^2 deve ser interpretado como a proporção de variação total da variável dependente do eixo Y (Distância da bola em centímetros) que é explicada pela variação da variável independente no eixo X (Tamanho da bola em pixels). Nesse caso pode-se concluir que 99,97 % das variações de Y são explicadas pela variação de X. Como o gráfico foi plotado em relação ao diâmetro da bola , porém, como ambos os algoritmos fornecem como saída o raio da bola, se faz necessário a conversão do diâmetro em raio dividindo-o por dois.

$$D_{bola} = \frac{18286}{MRB} \quad \therefore \quad D_{bola} = \frac{9143}{MRB} \quad (cm) \quad (13)$$

É claro que, haverá um erro de medição nessas fórmulas, já que foram calculadas sem oclusão e com o objeto inteiro no campo de visão da câmera, nesse caso, o erro será proporcional à oclusão.

4.5 EXPERIMENTO - DETECTANDO ROBÔS USANDO O DESCRIPTOR HOG COM O CLASSIFICADOR SVM

Sendo o algoritmo mais custoso computacionalmente, o reconhecimento de robôs roda com a resolução de 640x480 pixels e 15 quadros por segundo. O SVM foi treinado com um conjunto de imagens de pedestres humanos, os teste foram executados visando dois tipos de reconhecimento:

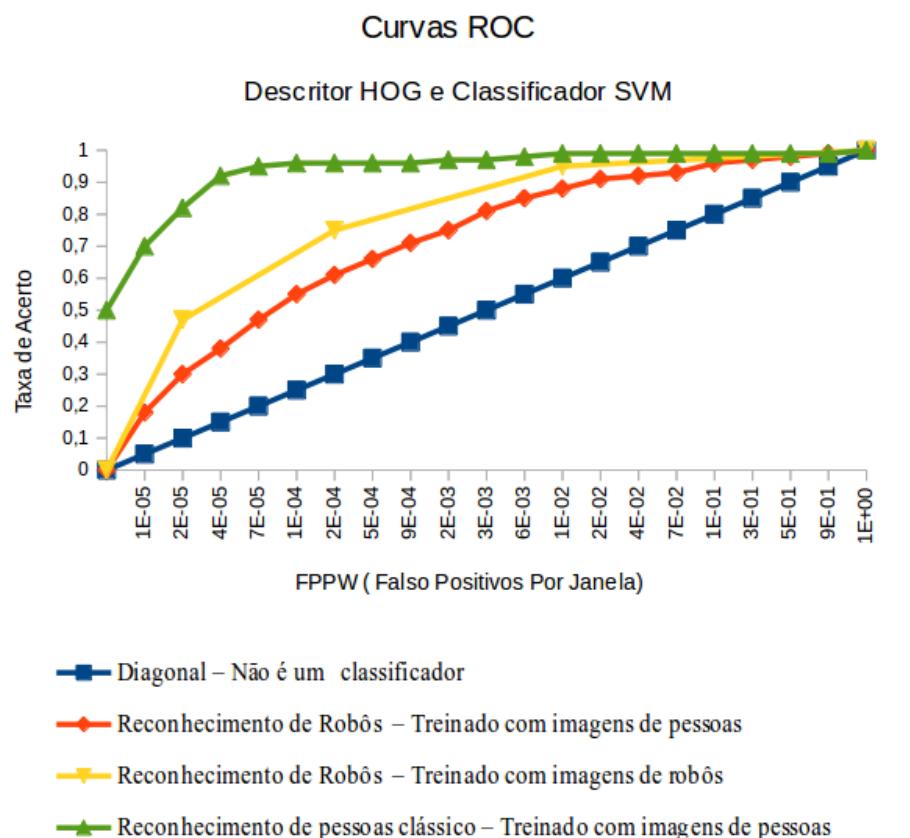
- a) SVM treinado com imagens humanas reconhecendo humanos- (Grupo de controle);
- b) SVM treinado com imagens humanas reconhecendo robôs;
- c) SVM treinado com imagens de robôs reconhecendo robôs.

Para o descriptor HOG um descriptor padrão de 64x128 pixels foi usado da mesma forma que o artigo original, uma janela de 16x16 pixels, a célula que desliza pela imagem tem 32x32 pixels e um limiar de acerto de 0,5. Para o classificador SVM, o parâmetro de custo C ficou definido em 10 e o parâmetro Γ ficou igual a 0.001, ambos escolhidos por uma busca em grade.

Como esperado, o classificador SVM, em conjunto com o descriptor HOG, treinado com imagens de pessoas para reconhecimento de robôs é muito menos preciso do que o reconhecimento clássico de imagens de pessoas reconhecendo pessoas, entretanto, mesmo nessas circunstâncias, parece atingir a sua meta. Também foi reunida uma, mesmo que pequena, quantidade de imagens normalizadas contendo robôs, numa tentativa de verificar o quanto bom esse algoritmo pode ser. Uma curva ROC foi plotada com as três situações, demonstrado na figura 25 e o resultado de um quadro na figura 26:

As curvas ROC sugerem que o classificador treinado com imagens de robôs, mesmo com poucas imagens, parece ser, assintoticamente, melhor que a primeira abordagem usando o conjunto de imagens de pedestres. Nossa expectativa é fazer um classificador de robôs tão bom quanto o que reconhece pedestres. Devido a diversidade de robôs e sendo os seres humanos o

Figura 25 – Curvas ROC. A linha verde mostra o classificador clássico HOG-SVM, usado para reconhecimento de pessoas. A linha vermelha demonstra o mesmo classificador treinado com pessoas usado para reconhecer robôs e a linha amarela mostra o classificador treinado com 100 imagens de robôs positivas e 100 imagens negativas usado na classificação dos robôs.

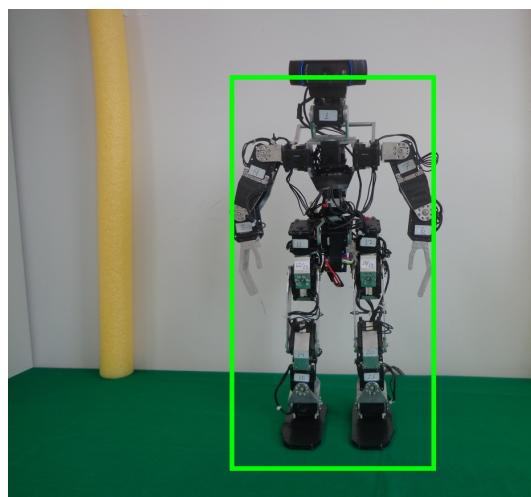


Fonte: Autor Vilao et al., 2014

nível máximo da forma humanoide, este classificador mostrou uma capacidade de generalização que poderia ser usada para reconhecer uma grande quantidade de robôs humanoides, dada uma quantidade suficiente de imagens de treinamento normalizadas.

É possível ver que a cabeça do robô não foi incluída no retângulo de detecção, talvez, pela falta de imagens normalizadas de boa qualidade ou, talvez, pelo fato da cabeça não estar em conformidade com a cabeça humana. Como a aquisição de imagens tão específicas com uma mínima qualidade é de certa forma difícil de ser obter, a abordagem "Leave-One-Out" foi utilizada, já que lida bem com poucas amostras de treinamento.

Figura 26 – Detecção do robô Milton usando imagens de pessoas como conjunto de treinamento.



Fonte: Autor Vilao et al., 2014

4.6 EXPERIMENTO - DETECTANDO ROBÔS USANDO O DESCRIPTOR HAAR COM O CLASSIFICADOR BOOSTING

No contexto atual da liga KidSize da RoboCup, outros robôs não têm a necessidade de serem detectados enquanto estiverem a mais de 3 metros de distância, devido a sua velocidade de andar ser bastante reduzida, dessa forma, foi inserida uma limitação com relação à distância do robô a ser detectada. Assim, a janela mínima de detecção foi definida com 150x150 pixels, ou seja, o menor objeto (robô) a ser detectado terá essa dimensão na imagem. O fato de limitar a janela de detecção mínima tem o principal objetivo de reduzir falsos positivos mas, também de acelerar o processo de detecção, já que menos janelas estarão contidas dentro da imagem.

Durante o treinamento o número máximo de estágios foi determinado como sendo de 20 estágios, a taxa de acerto mínima foi configurada em 0.999, o parâmetro $N_{\text{AmostrasPositivas}}$ após cálculo utilizando as equações 5 e 6 ficou fixada em 920 para 2135 imagens positivas e 1192 negativas, finalmente, a janela de amostragem foi de 150x150 pixels. O algoritmo foi implementado usando como classificador o k vizinhos mais próximos (K-nearest neighbors) com valor de $k = 9$. Valores como $k=3$, $k = 5$ e $k=7$ foram escolhas claramente ruins, já que geraram diversos falsos positivos. O valor de $k=11$ teve bons resultados mas, também ignoraram o objeto em diversas circunstâncias. Maiores valores de k não geraram classificação. Mesmo apresentando falsos positivos o Haar conseguiu identificar o robô em diversas situações. Na

figura 27 uma sequência de quadros do robô B2 sendo identificado, com o uso do Haar, em um campo de futebol com grama sintética.

Figura 27 – Rastreamento do robô B2 usando características HAAR no campo de futebol com grama artificial.



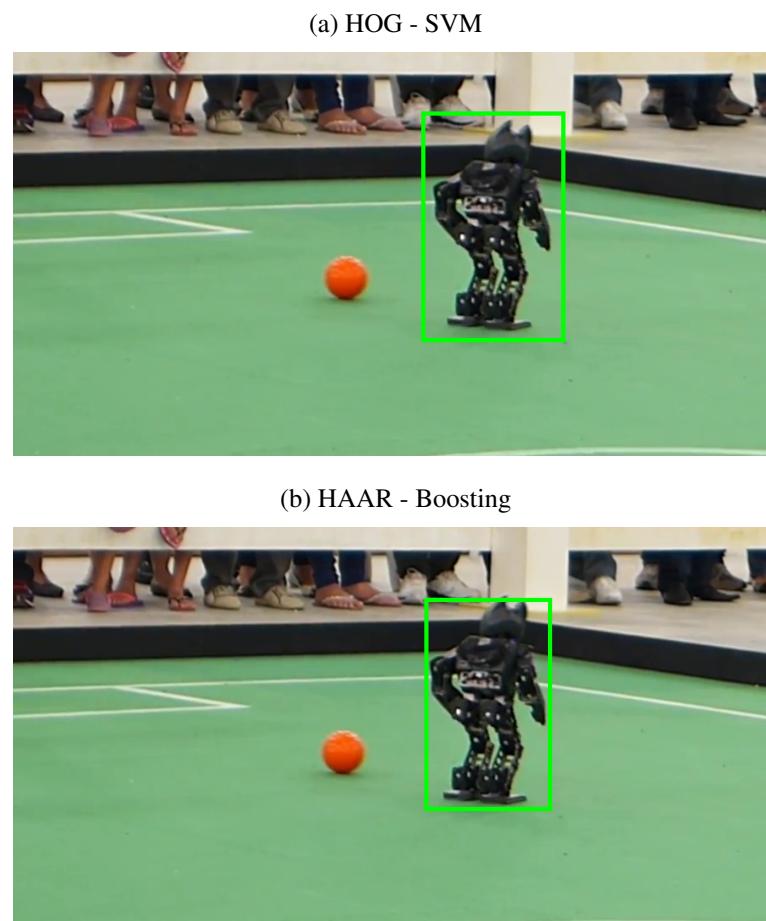
Fonte: Autor

4.7 DISCUSSÃO - COMPARANDO OS DOIS DESCRIPTORES E SEUS CLASSIFICADORES

Diferentemente do experimento apresentado na seção 4.5 nessa seção, as imagens utilizadas para treinamento das técnicas HOG-SVM e HAAR-Adaboost foram as mesmas, ou seja, 2135 imagens positivas e 1192 imagens negativas. Os parâmetros se mativeram os mesmos dos experimentos das seções 4.5 e 4.6. Duas curvas ROC foram plotadas e os resultados estão na

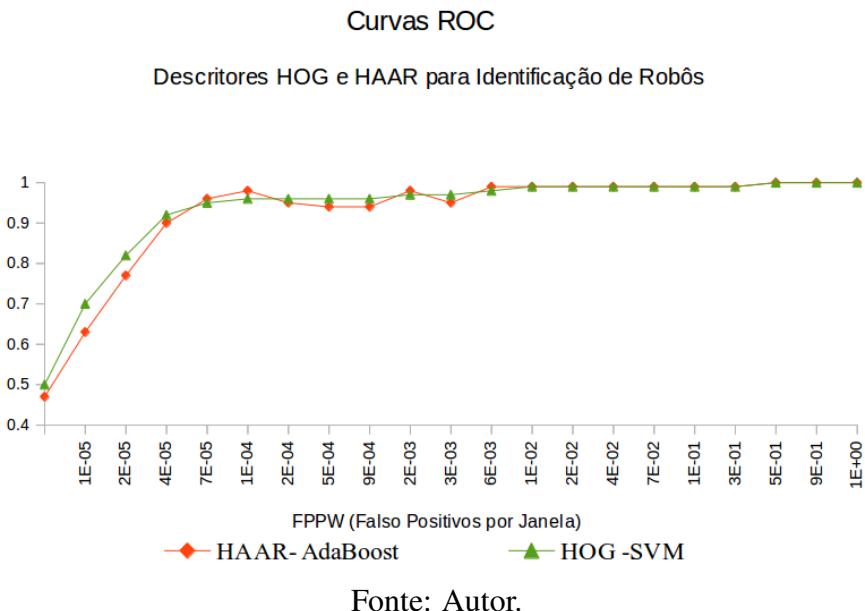
figura 29. A quantidade de quadros por segundo foi extraída usando um código simples que armazenava e contava os quadros dentro do período de 5 minutos, sendo portanto o resultado apresentado na figura 30 a média de quadros por segundo em 5 minutos. Esse período foi escolhido devido a limitação de tempo dos videos que foram usados para o teste. Na figura 28, os resultados dos dois métodos aplicados à mesma imagem do robô Bold Hearts. É possível verificar que não há grande diferença entre as duas detecções. O descriptor HAAR teve um desempenho melhor que o descriptor HOG em termos de velocidade, especialmente nas maiores resoluções. Entretanto quando o robô foi levado para o ambiente real, a iluminação se tornou um verdadeiro problema. Para ambos os algoritmos foi necessário incluir imagens negativas referentes ao ambiente. Na figura 29 uma curva ROC que compara as capacidades e limitações dos dois algoritmos.

Figura 28 – Identificação do Robô Bold Hearts (BOLD..., 2014b) da universidade de Hertfordshire usando HOG-SVM (Figura a) e HAAR-Adaboost (Figura b).



Fonte: Quadro extraído do vídeo de qualificação do time Bold Hearts (BOLD..., 2014b) disponível em Bold..., 2014a.

Figura 29 – Curvas ROC. A linha verde mostra o algoritmo HOG-SVM clássico treinado com imagens de robôs usado para detecção de robôs. A linha vermelha demonstra o resultado do classificador HAAR-AdaBoost treinado com o mesmo propósito.



Fonte: Autor.

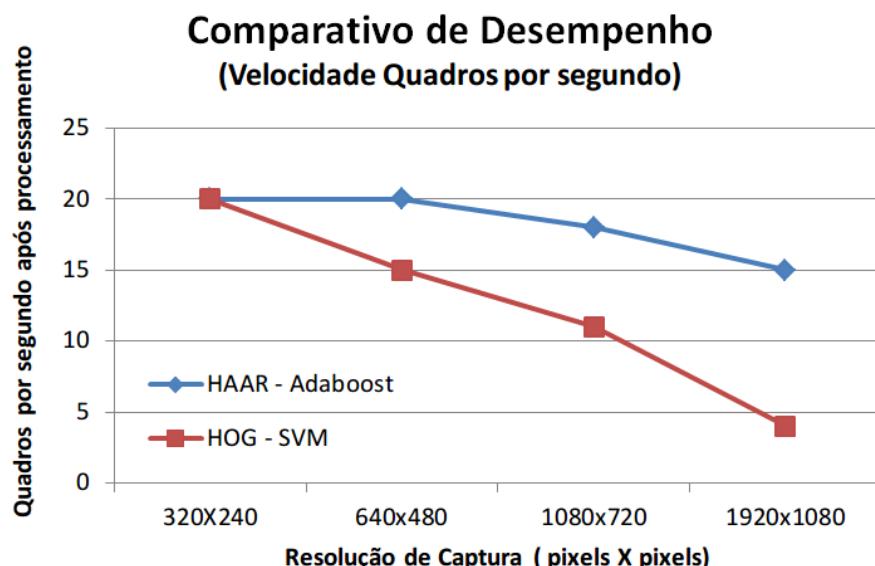
As duas curvas ROC apresentadas na figura 29 sugerem que os classificadores treinados parecem ser equivalentes. Devido à diversidade de robôs ambos os classificadores demonstraram capacidade de generalização que pode reconhecer diversos tipos de robôs dadas imagens de treinamento normalizadas suficientes.

Outro parâmetro de desempenho utilizado foi o de resposta dos algoritmos em quadros por segundo. É vital para um robô jogador de futebol que consiga detectar outros robôs em tempo real. A comparação desses parâmetros de desempenho estão na figura 30, esta informação é particularmente vital quando se trata de um robô que precisa agir rapidamente.

Os resultados mostram alguma vantagem para o HAAR-Adaboost em quadros por segundo, que pôde ser utilizado em full-hd 1920x1080 pixels mas, com alguns falso positivos, já que é uma técnica sensível à mudanças de iluminação. Já o HOG-SVM foi mais preciso em termos de detecção, porém com a limitação da taxa de velocidade e por consequência a resolução máxima de 640x480 pixels. Nas figuras 31 e 32 as mesmas imagens demonstrando as detecções para os dois algoritmos. Nessas figuras as detecções mostraram alguma vantagem para o HOG, que demonstrou mais estabilidade detectando menos falso positivos.

Aqui cabe uma ressalva, quando se trata de avaliar os descritores, os parâmetros escolhidos e o hardware foram verdadeiramente responsáveis pelo desempenho geral dos dois algoritmos.

Figura 30 – Desempenho dos descritores HAAR-AdaBoost e HOG-SVM em termos de quadros por segundo após processamento.

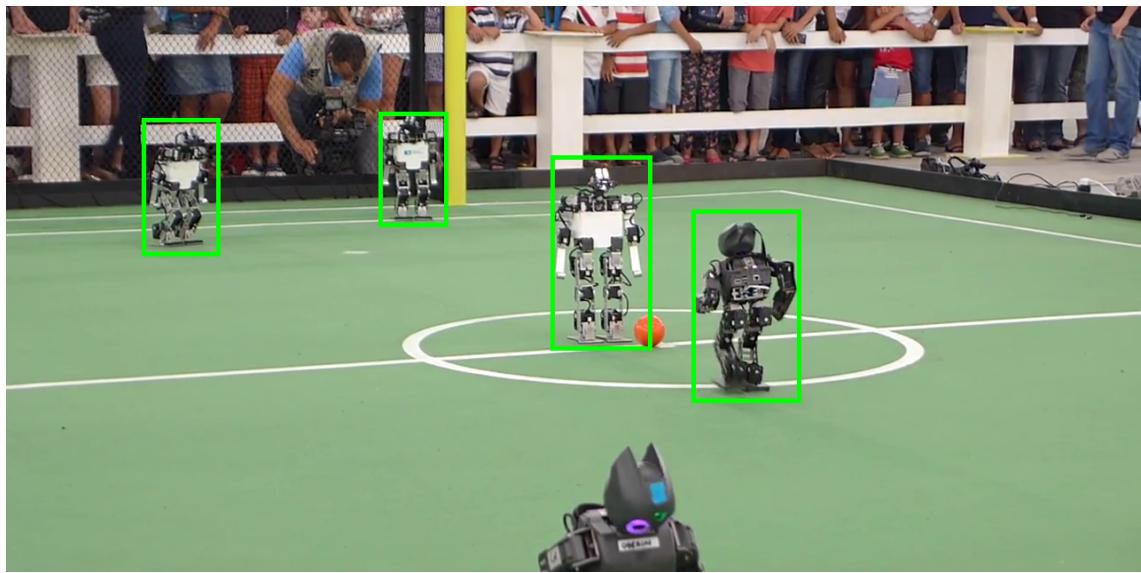


Fonte: Autor

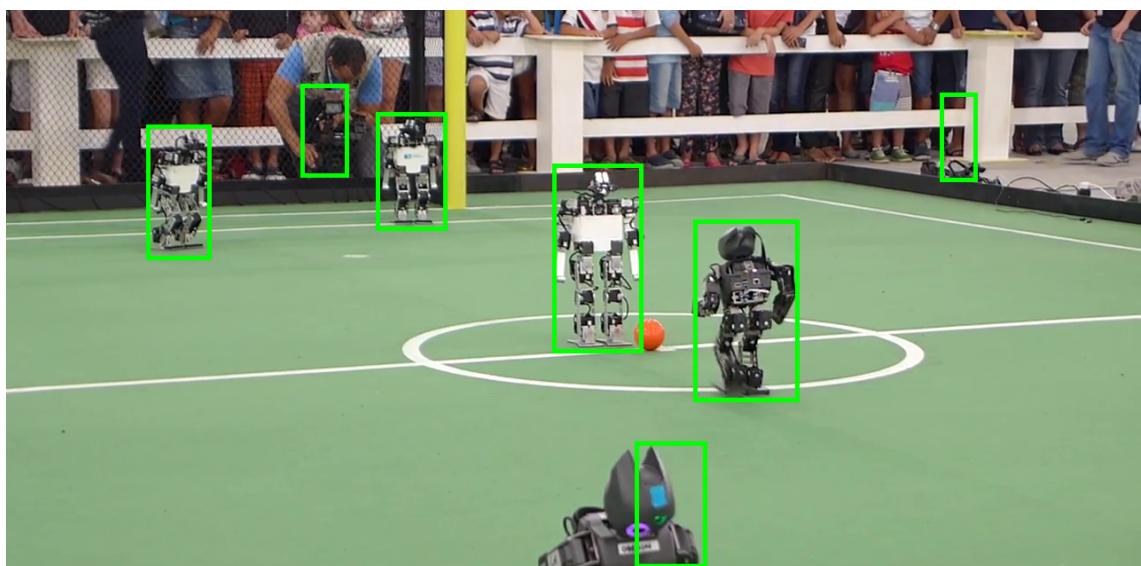
O tempo gasto no treinamento e na normalização e redimensionamento das imagens também foi uma questão complicada. Enquanto o HOG-SVM determinava todas as características por si só e levava no máximo 2 horas para efetuar o treinamento, o HAAR-Adaboost precisava que uma pessoa determinasse onde os objetos estavam em cada imagem positiva e seu treinamento levou de 5 a 12 horas para ser concluído. É válido lembrar que as imagens positivas do HOG-SVM precisaram ser redimensionadas o que levou um tempo considerável.

Figura 31 – Vários robôs identificados usando o HOG-SVM e o HAAR-AdaBoost.

(a) HOG - SVM



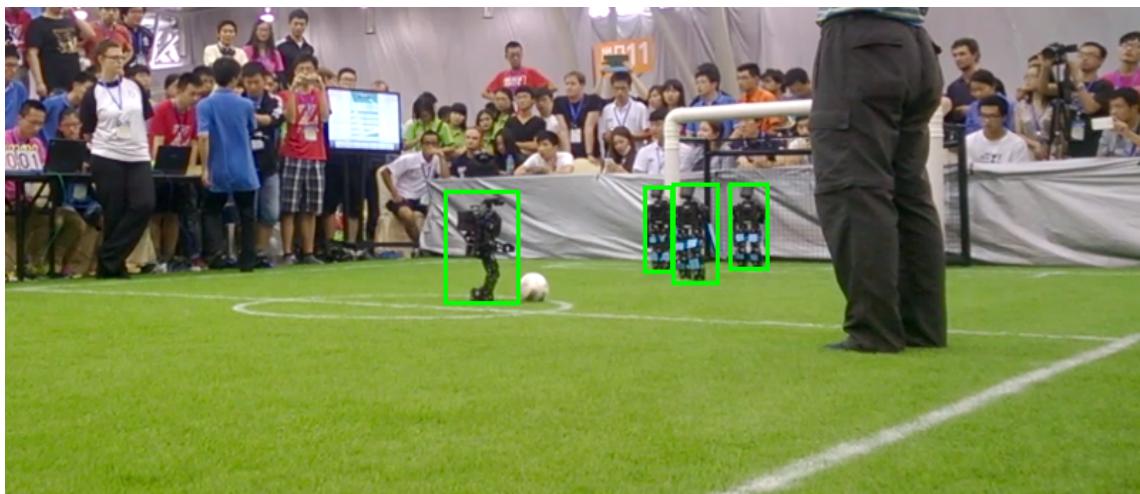
(b) HAAR - Boosting



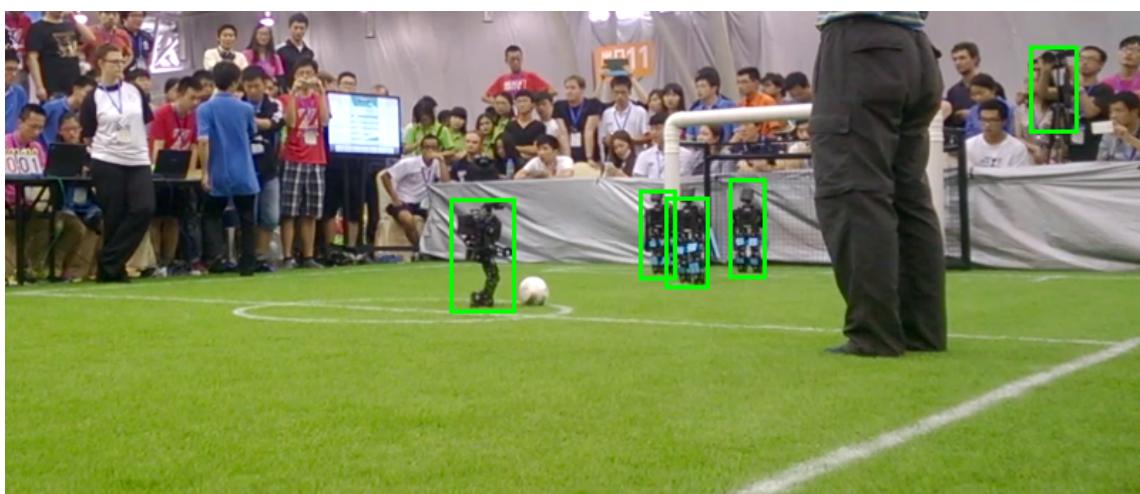
Fonte: Quadro extraído do vídeo de qualificação do time Bold Hearts (BOLD..., 2014b)
disponível em Bold..., 2014a.

Figura 32 – Quadro de captura do robô B2 identificando vários robôs usando o HOG-SVM (Figura a) e o HAAR-AdaBoost (Figura b).

(a) HOG - SVM



(b) HAAR - Boosting



Fonte: Autor.

5 CONCLUSÃO

Esse trabalho apresentou uma solução de visão computacional para o ambiente Kid-Size, robusto à mudanças de iluminação, usando algoritmos de visão básicos e bem conhecidos, entretanto, o ambiente real provou ser bastante desafiador.

Houveram diversas mudanças nas regras da RoboCup para a categoria entre os anos de 2014 e 2015, as duas principais mudanças são relativas às cores do gol(amarelo) e da bola (laranja), que passaram a ter suas cores alteradas para a cor branca.

Essas mudanças, apesar de drásticas para o domínio proposto, parecem levar ao encontro da meta da RoboCup. A abordagem apresentada permitiu que diversas técnicas pudessem ser utilizadas para identificar características como cores, texturas, bordas e contornos.

Dada a limitação da aplicação em tempo real, HAAR teve um melhor desempenho levando-se em consideração apenas a velocidade de computação. No entanto, possui alguns problemas com iluminação e, portanto, têm uma alta taxa de falsos alarmes especialmente em ambientes reais e complexos. Os resultados mostram algumas vantagens para a técnica de Haar em quadros por segundo, de forma que pode ser executado em Full HD, mas com alguns falsos positivos, uma vez que é uma técnica muito sensível às mudanças de iluminação e, embora o descritor HOG demonstrou mais precisão em termos de detecção, mostrou ter uma limitação da taxa de velocidade e resolução máxima de 640x480 pixels. Quando se trata de avaliar os descritores e seus classificadores a escolha dos parâmetros e o hardware utilizado tiveram fortes influências no desempenho total do sistema.

O tempo gasto em treinamento também foi uma questão complicada. Enquanto Hog determinava todas as características por si só e levava apenas alguns minutos para o treinamento, o HAAR precisava de uma pessoa para determinar onde os objetos se encontravam na imagens, e levando de 5 à 10 horas para ser concluído. As imagens de treinamento usadas para classificação usando o descritor HOG tiveram que ser reduzidas e normalizadas para o tamanho proposto no trabalho original de Dalal e Triggs (2005).

Na competição, no entanto, fazer o robô ignorar o ambiente de fundo provou ser uma tarefa muito desafiadora. Dessa forma, dado o hardware utilizado e os objetivos propostos, o descritor HAAR parece ser mais a técnica mais apropriada para o domínio proposto, principalmente por sua velocidade de processamento.

Construir uma base de imagens de robôs, a variedade de robôs atuantes na liga é muito grande e para se ter um classificador confiável, é necessário um conjunto de imagens de treinamento provenientes dos mais diversos tipos de robôs, assim como, suas respectivas dimensões.

O time qualificou-se e participou da RoboCup, liga KidSize, pela primeira vez em 2014, atingindo a segunda fase de grupos, terminando entre os 16 melhores colocados de 23 inscritos.

5.1 CONTRIBUIÇÕES

As seguintes contribuições são decorrentes desse, e são citadas:

- a) a construção do sistema de visão computacional do robô para a competição do time RoboFEI em diversos campeonatos de futebol (VILAO et al., 2014) (PERICO et al., 2014);
- b) quebra de um paradigma para o domínio, onde se acreditava que era necessário o uso de segmentação por cores, com o desenvolvimento de um sistema de visão computacional para identificação da bola que não é dependente de segmentação de cores;
- c) um sistema de detecção de robôs ainda pouco explorado nesse domínio (VILAO et al., 2015);
- d) a equipe foi campeã Latino-Americano de futebol de robôs (RESULTADOS..., 2014) e ficou entre os 16 melhores colocados na categoria KidSize da Robocup nos anos de 2014 em João Pessoa (RESULTADOS..., 2014) e 2015 na cidade de Hefei na China (RESULTADOS..., 2015).

5.2 TRABALHOS FUTUROS

Certos aspectos do desenvolvimento do sistema de visão computacional foram abordados no presente visando solucionar os mais diversos problemas inerentes desse tipo de sistema. Com o intuito de melhorar, expandir ou modificar o presente serão citados alguns itens que surgiram durante a elaboração do presente e que são descritos brevemente como possíveis trabalhos futuros.

- a) uma comparação entre o HOG e o HAAR foi proposta, porém outras técnicas podem ser utilizadas, como por exemplo, o CenTrist. Trata-se de uma técnica computacionalmente

eficiente para capturar e determinar contornos comparado com o HOG, mas ainda há a escassez de artigos que implementaram e testaram a precisão do descritor Centrist para identificar robôs;

- b) um estudo para identificar os níveis de processamento e robustez de redes neurais profundas (Deep Neural Networks - DNN) para identificação de robôs e outros objetos com o vídeo de uma câmera em movimento também pode ser vislumbrado;
- c) implementar o filtro de Kalman no rastreamento da bola pode ser uma alternativa para o filtro de partículas;
- d) limitar a área de rastreamento da bola, já que a bola precisa estar no campo, é possível considerar que a bola estará contida em uma região predominantemente verde;
- e) uma comparação entre o Haar e a transformada de Hough para círculos para a bola laranja;

REFERÊNCIAS

- ALOIMONOS, J.; WEISS, I.; BANDYOPADHYAY, A. Active vision. **International Journal of Computer Vision**, Kluwer Academic Publishers, v. 1, n. 4, p. 333–356, 1988. ISSN 0920-5691. Disponível em: <<http://dx.doi.org/10.1007/BF00133571>>.
- AUTMAN Humanoid KidSize Team Description Paper. 2014. Disponível em: <<https://www.robocuphumanoid.org/hl-2015/teams/>>. Acesso em: 01 Set. 2015.
- AVIDAN, S. Ensemble tracking. **Pattern Analysis and Machine Intelligence, IEEE Transactions on**, 2007.
- BASET Kid-Size 2014 Team Description Paper. 2014. Disponível em: <<https://www.robocuphumanoid.org/hl-2015/teams/>>. Acesso em: 01 Set. 2015.
- BENALLAL, M.; MEUNIER, J. Real-time color segmentation of road signs. In: **Electrical and Computer Engineering, 2003. IEEE CCECE 2003. Canadian Conference on**. [S.l.: s.n.], 2003. v. 3, p. 1823–1826 vol.3. ISSN 0840-7789.
- BOLD Hearts - RoboCup 2015 Qualification - Kid Size Humanoid League. 2014. Disponível em: <<https://www.youtube.com/watch?v=pzYHAp7b7sY>>. Acesso em: 03 Set. 2015.
- BOLD Hearts Team Description RoboCup 2014 Kid Size. 2014. Disponível em: <<https://www.robocuphumanoid.org/hl-2015/teams/>>. Acesso em: 01 Set. 2015.
- BROGGI, A. et al. Real time road signs recognition. In: **IEEE. Intelligent Vehicles Symposium, 2007 IEEE**. [S.l.], 2007. p. 981–986.
- BUDDEN, D. et al. A novel approach to ball detection for humanoid robot soccer. **AI 2012: Advances in Artificial Intelligence, Lecture Notes in Computer Science**, v. 7691, p. 827–838, 2012.
- CAN not get new positive sample. 2012. Disponível em: <<http://goo.gl/5qZl0j>>. Acesso em: 10 Abr. 2014.
- CHAUMETTE, F.; HUTCHINSON, S. Visual servo control. i. basic approaches. **Robotics & Automation Magazine, IEEE**, IEEE, v. 13, n. 4, p. 82–90, 2006.
- _____. Visual servo control, part ii: Advanced approaches. **IEEE Robotics and Automation Magazine**, v. 14, n. 1, p. 109–118, 2007.
- CIT Brains. 2014. Disponível em: <<https://www.robocuphumanoid.org/hl-2015/teams/>>. Acesso em: 01 Set. 2015.

COLOMBO, C.; BIMBO, A. D.; PALA, P. Semantics in visual information retrieval. **IEEE MultiMedia**, IEEE, n. 3, p. 38–53, 1999.

CRISTIANINI, N.; SHawe-Taylor, J. **An introduction to support vector machines and other kernel-based learning methods**. [S.l.]: Cambridge university press, 2000.

DALAL, N.; TRIGGS, B. Histograms of oriented gradients for human detection. In: **IEEE Computer Vision and Pattern Recognition. CVPR 2005. IEEE Computer Society Conference on**. [S.l.], 2005. v. 1, p. 886–893.

DAVIES, E. A modified hough scheme for general circle location. **Pattern Recognition Letters**, v. 7, n. 1, p. 37 – 43, 1988. ISSN 0167-8655. Disponível em: <<http://www.sciencedirect.com/science/article/pii/0167865588900426>>.

DENZLER, J.; ZOBEL, M.; NIEMANN, H. Information theoretic focal length selection for real-time active 3d object tracking. In: **IEEE. Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on**. [S.l.], 2003. p. 400–407.

DOUGHERTY, E. R. **An Introduction to Morphological Image Processing**. [S.l.]: SPIE Optical Engineering Press, 1992.

DUDA, R. O.; HART, P. E. Use of the hough transformation to detect lines and curves in pictures. **Communications of the ACM**, ACM, v. 15, n. 1, p. 11–15, 1972.

_____. **Pattern Classification**. 2. ed. [S.l.]: Wiley, 2001.

DUMAIS, S. et al. Using svms for text categorization. **IEEE Intelligent Systems**, Trends and Controversies—Support Vector Machines, v. 13, n. 4, p. 21–23, 1998.

EDROM Humanoid Kid Size 2014. 2014. Disponível em: <<https://www.robocuphumanoid.org/hl-2014/teams/>>. Acesso em: 03 Set. 2015.

ERHAN, D. et al. Scalable object detection using deep neural networks. In: **IEEE. Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on**. [S.l.], 2014. p. 2155–2162.

EROS Team Description for Humanoid Kid Size League of Robocup 2014. 2014. Disponível em: <<https://www.robocuphumanoid.org/hl-2015/teams/>>. Acesso em: 01 Set. 2015.

ESTIVILL-CASTRO, V.; RADEV, J. Humanoids learning who are teammates and who are opponents. **The 8th Workshop on Humanoid Soccer Robots 13th at IEEE-RAS International Conference on Humanoid Robots**, 2013.

FARIA, E. et al. Previsão de séries temporais utilizando métodos estatísticos. **Centro Brasileiro de Pesquisas Físicas—CBPF. Pontifícia Universidade Católica—PUC. Rio de Janeiro**, 2013.

Janeiro, 2008.

FLEET, D.; WEISS, Y. Optical flow estimation. In: **Handbook of mathematical models in computer vision**. [S.l.]: Springer, 2006. p. 237–257.

FLUXO Ótico. 2007. Disponível em: <http://www.verlab.dcc.ufmg.br/_media/cursos/visao/2007-1/alunos/alexandrewagner/optical_flow_article.pdf>. Acesso em: 12 Ago. 2014.

FREUND, Y.; SCHAPIRE, R. E. A decision-theoretic generalization of on-line learning and an application to boosting. **Journal of computer and system sciences**, Elsevier, v. 55, n. 1, p. 119–139, 1997.

FRIEDMAN, J. et al. Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). **The annals of statistics**, Institute of Mathematical Statistics, v. 28, n. 2, p. 337–407, 2000.

GONZALEZ, R. C.; WOODS, R. E. **Digital Image Processing**. 3. ed. [S.l.]: Prentice Hall, 2010.

GUIL, N.; VILLALBA, J.; ZAPATA, E. L. A fast hough transform for segment detection. **Image processing, IEEE transactions on**, IEEE, v. 4, n. 11, p. 1541–1548, 1995.

HA, I. et al. Development of open humanoid platform darwin-op. In: **IEEE. SICE Annual conference (SICE), 2011 proceedings of**. [S.l.], 2011. p. 2178–2181.

HAAR, A. Zur theorie der orthogonalen funktionensysteme. **Mathematische annalen**, Springer, v. 69, n. 3, p. 331–371, 1910.

HAAS, A.; MATHERON, G.; SERRA, J. Morphologie mathématique et granulométries en place. In: **Annales des mines**. [S.l.: s.n.], 1967. v. 11, n. 736-753, p. 7–3.

HAJ, M. A. et al. **Visual analysis of humans**: Looking at people. [S.l.]: Springer, 2011.

HAYKIN, S.; NETWORK, N. A comprehensive foundation. **Neural Networks**, v. 2, n. 2004, 2004.

HOUGH, P. V. Machine analysis of bubble chamber pictures. In: **International conference on high energy accelerators and instrumentation**. [S.l.: s.n.], 1959. v. 73.

HUTCHINSON, S. et al. A tutorial on visual servo control. **Robotics and automation, IEEE transactions on**, IEEE, v. 12, n. 5, p. 651–670, 1996.

ISARD, M.; BLAKE, A. Condensation—conditional density propagation for visual tracking. **International journal of computer vision**, Springer, v. 29, n. 1, p. 5–28, 1998.

- JU, Y.; ZHANG, H.; XUE, Y. Research of feature selection and comparision in adaboost based object detection system. **Journal of Computational Information Systems**, Binary Information Press, v. 9, p. 22, 2013. Doi = 10.12733/jcis7492.
- KIRYATI, N.; ELDAR, Y.; BRUCKSTEIN, A. M. A probabilistic hough transform. **Pattern recognition**, Elsevier, v. 24, n. 4, p. 303–316, 1991.
- KURANOV, A.; LIENHART, R.; PISAREVSKY, V. An empirical analysis of boosting algorithms for rapid objects with an extended set of haar-like features. **Intel Technical Report MRL-TR-July02-01**, 1998. [Http://www.lienhart.de/Publications/DAGM2003.pdf](http://www.lienhart.de/Publications/DAGM2003.pdf).
- LI, S. Z.; ZHANG, Z. Floatboost learning and statistical face detection. **Pattern Analysis and Machine Intelligence, IEEE Transactions on**, IEEE, v. 26, n. 9, p. 1112–1123, 2004.
- LORENA, A. C.; CARVALHO. Introdução as máquinas de vetores suporte. **Relatórios Técnicos do ICMC: Usp - São Carlos**, 2003.
- MACKWORTH, A. K. On seeing robots. In: BASU, A.; LI, X. (Ed.). **Computer Vision: Systems, Theory and Applications**. Singapore: World Scientific Press, 1993. p. 1–13. Reprinted in P. Thagard (ed.), Mind Readings, MIT Press, 1998.
- MAKRIDAKIS, S.; WHEELWRIGHT, S. C.; SONS, R. J. H. J. W. Forecasting: Methods and applications. **IIE Transactions**, v. 31, 1998.
- MARTÍNEZ, A. M.; KAK, A. C. Pca versus lda. **Pattern analysis and machine intelligence, IEEE transactions on**, IEEE, v. 23, n. 2, p. 228–233, 2001.
- MCCANN, S.; REESMAN, J. Object detection using convolutional neural networks. Stanford University, p. 331–371, 2013.
- MITA, T. et al. Discriminative feature co-occurrence selection for object detection. **Pattern analysis and machine intelligence, IEEE transactions on**, IEEE, v. 30, n. 7, p. 1257–1269, 2008.
- MORETTIN, P. A.; TOLOI, C. **Análise de séries temporais**. [S.l.]: Blucher, 2006.
- MS Excel 2013. 2015. Disponível em: <<https://www.office.com/>>. Acesso em: 03 Set. 2015.
- NAO. 2015. Disponível em: <<http://www.active8robots.com/wp-content/uploads/File-1400771269.pdf>>. Acesso em: 06/02/2015.
- NUMMIARO, K.; KOLLER-MEIER, E.; GOOL, L. V. Object tracking with an adaptive color-based particle filter. In: **Pattern recognition**. [S.l.]: Springer, 2002. p. 353–360.
- OPENCV - (Open Source Computer Vision Library). 2014. Disponível em: <[http:](http://)

//www.opencv.org>. Acesso em: 20 Mai. 2014.

OSUNA, E.; FREUND, R.; GIROSI, F. Training support vector machines: an application to face detection. In: IEEE. **Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on**. [S.l.], 1997. p. 130–136.

PAISITKRIANGKRAI, S.; SHEN, C.; ZHANG, J. Face detection with effective feature extraction. **Computer Vision – ACCV 2010 in 10th Asian Conference on Computer Vision, Queenstown, New Zealand**, v. 6494, 2010.

PAPAGEORGIOU, C. P.; OREN, M.; POGGIO, T. A general framework for object detection. In: IEEE. **Computer vision, 1998. sixth international conference on**. [S.l.], 1998. p. 555–562.

PARAG, T.; PORIKLI, F.; ELGAMMAL, A. Boosting adaptive linear weak classifiers for online learning and tracking. **Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on**, 2008.

PATEL, H. A.; G., T. D. Moving object tracking using kalman filter. **International Journal of Computer Science and Mobile Computing**, v. 02, p. 326–332, 2013.

PEDREGOSA, F. et al. Scikit-learn: Machine learning in Python. **Journal of Machine Learning Research**, v. 12, p. 2825–2830, 2011.

PERICO, D. H. et al. Hardware and software aspects of the design and assembly of a new humanoid robot for robocup soccer. In: IEEE. **Robotics: SBR-LARS Robotics Symposium and Robocontrol (SBR LARS Robocontrol), 2014 Joint Conference on**. [S.l.], 2014. p. 73–78.

POLIKAR, R. Ensemble based systems in decision making. **Circuits and Systems Magazine, IEEE**, v. 6, p. 21 – 45, 2006.

RESULTADOS da RoboCup. 2014. Disponível em: <<https://www.robocuphumanoid.org/hl-2014/results/>>. Acesso em: 03 Set. 2015.

RESULTADOS da RoboCup. 2015. Disponível em: <<https://www.robocuphumanoid.org/hl-2015/results/>>. Acesso em: 03 Set. 2015.

RESULTADOS do LARC/CBR. 2014. Disponível em: <<http://www.crobotica.org/>>. Acesso em: 25 Set. 2015.

RIVLIN, E.; ROTSTEIN, H. Control of a camera for active vision: Foveal vision, smooth tracking and saccade. **International Journal of Computer Vision**, Kluwer Academic Publishers, v. 39, n. 2, p. 81–96, 2000. Disponível em: <<http://dx.doi.org/10.1023/A%3A1008166825510>>.

ROBOCUP Official Website. 2014. Disponível em: <<http://www.robocup.org>>. Acesso em: 25 Mai. 2014.

ROSS, M. Statistical motion segmentation and object tracking without a-priori models. **Vision, Modeling, and Visualization 2006: Proceedings**, 2006.

SANDERSON, A.; WEISS, L. Image-based visual servo control using relational graph error signals. **Proc. ieee**, v. 1074, 1980.

SCHAPIRE, R. E.; SINGER, Y. Improved boosting algorithms using confidence-rated predictions. **Machine learning**, Springer, v. 37, n. 3, p. 297–336, 1999.

_____. A short introduction to boosting. **Journal-Japanese Society For Artificial Intelligence**, JAPANESE SOC ARTIFICIAL INTELL, v. 14, n. 771-780, p. 1612, 1999.

SHARMA, R. Role of active vision in optimizing visual feedback for robot control. In: **The confluence of vision and control**. [S.l.]: Springer, 1998. p. 24–40.

SHERMAN, P. D. **Colour vision in the nineteenth century: the Young-Helmholtz-Maxwell theory**. [S.l.]: Taylor & Francis, 1981.

SITE Oficial da RoboCup Brasil. 2014. Disponível em: <<http://www.robocup.org.br/objetivo.php>>. Acesso em: 01 Set. 2015.

SZEGEDY, C.; TOSHEV, A.; ERHAN, D. Deep neural networks for object detection. In: BURGES, C. et al. (Ed.). **Advances in Neural Information Processing Systems 26**. [s.n.], 2013. p. 2553–2561. Disponível em: <http://media.nips.cc/nipsbooks/nipspapers/paper_files/nips26/1210.pdf>.

THE NUBOTS Team Description Paper 2014. 2014. Disponível em: <<https://www.robocuphumanoid.org/hl-2015/teams/>>. Acesso em: 01 Set. 2015.

THRUN, S. Particle filters in robotics. In: **Proceedings of the 17th Annual Conference on Uncertainty in AI (UAI)**. [S.l.: s.n.], 2002.

TSOTSOS, J. On the relative complexity of active vs. passive visual search. **International Journal of Computer Vision**, Kluwer Academic Publishers, v. 7, n. 2, p. 127–141, 1992. ISSN 0920-5691. Disponível em: <<http://dx.doi.org/10.1007/BF00128132>>.

VAPNIK, V. N.; CHERVONENKIS, A. Y. Support-vector networks. **Machine Learning 20 (3): 273**, p. 273, 1995.

VILAO, C. et al. Evaluating the performance of two visual descriptors techniques for a humanoid robot. In: **LARS/SBR 2015**. Uberlândia - MG: [s.n.], 2015.

_____. A single camera vision system for humanoid robots acting at robocup kid-size soccer league. In: **LARS-SBR 2014**. USP São Carlos - SP: [s.n.], 2014.

VIOLA, P.; JONES, M. Rapid object detection using a boosted cascade of simple features. **Conference on Computer Vision and Pattern Recognition (CVPR)**, p. 511–518, 2001.

WALTHER, D. et al. Selective visual attention enables learning and recognition of multiple objects in cluttered scenes. **Computer Vision and Image Understanding**, v. 100, n. 1–2, p. 41 – 63, 2005. Special Issue on Attention and Performance in Computer Vision. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1077314205000433>>.

WAND, M. P. Data-based choice of histogram bin width. **The American Statistician**, v. 51, p. 59–64, 1996.

WESTERVELT, E. R. et al. **Feedback control of dynamic bipedal robot locomotion**. Boca Raton: CRC Press, 2007. (Control and automation). ISBN 978-1-4200-5372-2. Disponível em: <<http://opac.inria.fr/record=b1134075>>.

WU, J. et al. Fast asymmetric learning for cascade face detection. **Pattern Analysis and Machine Intelligence, IEEE Transactions on**, v. 30, 2008.

WU, J.; REHG, J. M. Centrist: A visual descriptor for scene categorization. **Pattern analysis and machine intelligence, IEEE transactions on**, IEEE, v. 33, n. 8, p. 1489–1501, 2011.

ANEXO A – MÉDIA MÓVEL

A.1 SÉRIES TEMPORAIS

O conceito de séries temporais está relacionado a um conjunto de observações de uma determinada variável feita em períodos sucessivos de tempo e ao longo de um determinado intervalo. Podemos citar como exemplos as cotações diárias da taxa do dólar, as vendas mensais de um determinado produto, a taxa de desemprego de um país, e etc. Segundo Morettin e Toloi (2006) os objetivos de se analisar uma série temporal são os seguintes:

- a) Descrição: propriedades da série como, por exemplo, o padrão de tendência, a existência de alterações estruturais, etc.
- b) Explicação: construir modelos que permitam explicar o comportamento da série no período observado.
- c) Controle de Processos: por exemplo, controle estatístico de qualidade.
- d) Previsão: prever valores futuros com base em valores passados.

Existem inúmeros métodos para se fazer previsão de séries temporais, desde os mais complexos que envolvem diferentes parâmetros aos métodos mais simples e de fácil entendimento. Para Makridakis, Wheelwright e Sons (1998) o fato de se utilizar métodos estatísticos mais complexos não significa necessariamente uma melhora nos resultados da previsão. Métodos simples podem apresentar resultados satisfatórios sobre certas condições, além de permitir uma total compreensão de suas limitações facilitando assim a interpretação dos resultados. Sendo assim deve-se primeiro avaliar os benefícios de se utilizar um método simples ou um mais complexo antes de se iniciar a previsão em uma determinada aplicação.

A.2 MÉTODOS SIMPLES DE PREVISÃO DE SÉRIE TEMPORAL

Muitos estudos envolvendo séries temporais têm como objetivo fazer previsões. Existem alguns métodos de previsão simples e que efetuam a previsão de valores futuros da série temporal através das observações de valores passados da série em interesse. Segundo Morettin e Toloi (2006) o propósito destes métodos é identificar um padrão básico presente nos dados históricos da série e através deste padrão prever valores futuros. Estes métodos simples têm uma grande popularidade, pois são simples de programar, e o custo computacional é muito pequeno

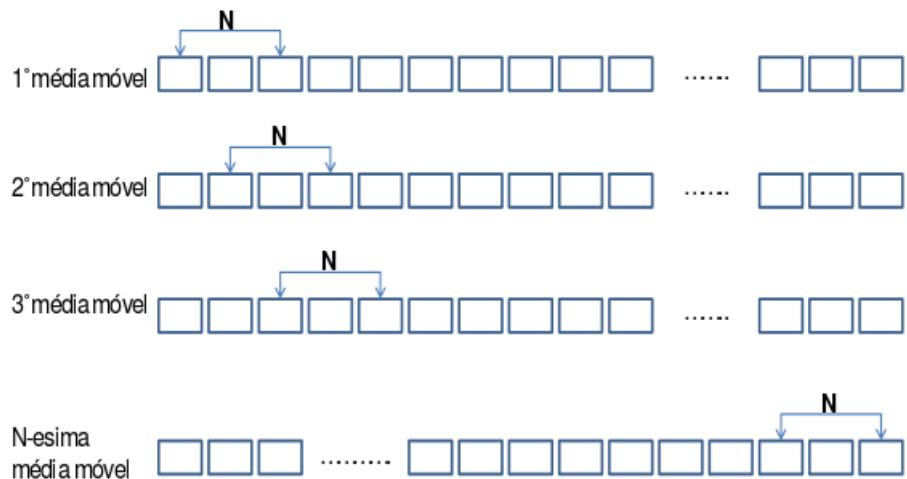


Figura 33 – Cálculo das Médias Móveis Simples.

além de obter uma razoável previsão. Dentre estes métodos podemos citar a média móvel, o alisamento exponencial simples e linear e também o alisamento exponencial sazonal e linear de Winter.

Os modelos de média móvel utilizam como previsão para um determinado período no futuro a média das observações passadas. As médias móveis podem ser simples, centradas ou ponderadas. Para os modelos de média móvel simples que são os modelos utilizados neste trabalho podemos definir sua equação como:

$$x_t = \frac{[x_{t-1} + x_{t-2} + x_{t-3} + \dots + x_{t-n}]}{n} \quad (14)$$

Na equação acima n representa o número de observações incluídas na média Xt, ou seja, n (também pode ser chamado de janela de observações) pode ser considerado como um parâmetro a ser ajustado. Neste caso quanto maior for a janela de observações maior o efeito de alisamento na previsão. Sendo assim, se a série em estudo apresentar muita aleatoriedade ou pequenas mudanças em seus padrões um número maior de observações podem ser utilizadas no cálculo da média móvel, ou seja, podemos dizer que a média móvel neste caso fica mais imune a ruídos e movimentos curtos. Já para o caso de séries que apresentam pouca flutuação aleatória nos dados ou mudança significativa, um número menor deve ser usado para o tamanho da janela de observações, pois caso contrário a série prevista poderá reagir de maneira muito lenta às estas mudanças significativas. Para Morettin e Toloi (2006) o termo média móvel é utilizado porque à medida que a próxima observação está disponível, a média das observações é recalculada, incluindo esta observação no conjunto de observações e desprezando a observação mais antiga, conforme podemos visualizar na figura 33:

Na figura acima o valor N corresponde ao número de observações (tamanho da janela igual a três) a ser utilizado para o cálculo da média móvel simples. Podemos perceber claramente que para cada nova janela de três dias a ser formada, a observação mais antiga é desprezada, e uma nova observação é inserida no conjunto N para o próximo cálculo.